

ҚАЗАҚСТАН РЕСПУБЛИКАСЫНЫҢ БІЛІМ ЖӘНЕ ҒЫЛЫМ МИНИСТРЛІГІ
Ш.ЕСЕНОВ АТЫНДАҒЫ КАСПИЙ МЕМЛЕКЕТТІК ТЕХНОЛОГИЯЛАР
ЖӘНЕ ИНЖИНИРИНГ УНИВЕРСИТЕТІ

Э.А.Абдыкеримова

**ИНФОРМАТИКАНЫҢ ТЕОРИЯЛЫҚ НЕГІЗДЕРІ
ПӘНІНЕН ДӘРІСТЕР ЖИНАҒЫ**

Ақтау - 2012

УДК 004(075-8)

ББК 32.81 я73

А 14

Пікір жазғандар:

1. п.ғ.д., профессор Ғ.И.Имашев
2. т.ғ.д., профессор С.М.Ахметов
3. п.ғ.к., доцент Р.С.Шуақбаева

Абдыкеримова Э.А.

А 14

Информатиканың теориялық негіздері пәнінен дәрістер жинағы: Оқу құралы/Э.А. Абдыкеримова – Ақтау, 2012. – 161 бет.

ISBN 978-601-7349-81-3

Оқу құралында «Информатиканың теориялық негіздері» пәні бойынша дәрістер курсы ұсынылған. Дәрістер курсы 5В011100 - «Информатика» мамандығына арналған Қазақстан Республикасының мемлекеттік білім беру стандартының негізінде құрастырылды. Дәрістер курсы 5В011100 - «Информатика» мамандығының күндізгі және сыртқы оқу бөлімінің студенттері үшін арналған.

УДК 004(075-8)
ББК 32.81 я73

Ш.Есенов атындағы Каспий мемлекеттік технологиялар және инжиниринг университетінің оқу-әдістемелік Кеңесінің шешімімен баспаға ұсынылған.

ISBN 978-601-7349-81-3

© Ш.Есенов атындағы КМТЖИУ, 2012

АЛҒЫ СӨЗ

Теориялық информатика – математикалық ғылым. Ол математиканың бірнеше бөлімдеріне негізделген: автоматтар мен алгоритмдер теориясы, математикалық логика, формалды тілдердің және грамматикалардың теориясы, ақпарат теориясы және т.б. Теориялық информатика ақпаратты сақтау және өңдеу кезінде пайда болатын негізгі сұрақтарға дәл талдау әдістері арқылы жауап беруге тырысады, мысалы, кейбір ақпараттық жүйеде жинақталған ақпарат саны неге тең, сақтау немесе іздеу үшін ақпаратты тиімді ұйыдастыру, сонымен қатар ақпаратты түрлендіру алгоритмдердің бар болу және қасиеттер туралы. Ай сайын деректерді сақтауға арналған ең үлкен көлемі бар жаңа құрылғылар пайда болуда, бұл ақпарат теориясы мен кодтау теориясының дамуымен түсіндіріледі. Қолданбалы есептерді шешу үшін жақсы программалар бар, бірақ сауатты түрде қолданбалы есепті қою үшін, оны компьютерге түсінікті түрге келтіру үшін ақпараттық және математикалық модельдеу негіздерін және т.б. білу керек. Информатиканың тек осы бөлімдерін меңгере отырып, өзіңізді осы ғылымның маманы деп есептей аласыз.

Информатиканың теориялық негіздері курсы информатиканың фундаментальды ұғымдары: ақпараттар теориясының негізі, сандық автоматтар теориясы, алгоритмдер теориясы, алгоритмдер тиімділігінің анализі, ақпараттық модельдеу және информатиканың семантикалық негізі туралы түсінікті қалыптастыру және машықтандыру; логикалық айнымалылар және арифметикалық амалдардың моделі және ЭЕМ-нің элементтік базасын ықшамдау; салыстыру операцияларын және арифметикалық операцияларды танып-білу; берілген алфавиттер мен сандарды кодтау, алгоритмді іздеу және таңдау, алгоритмдердің тиімділігі мен күрделілігіне анализ жасау, сұрыптауды түсіндіру; көпіршік сұрыптауды түсіндіріп, анализ жасау және пирамидалық сұрыптаудың тиімділігі мен күрделілігіне анализ жасау мәселелерін қамтиды.

Дәріс №1. Информатика пәні және мақсаттары. Теориялық информатика

Дәріс жоспары:

- 1.1 Информатиканың пайда болуы және дамуы
- 1.2 Есептеу техникасының даму тарихы
- 1.3 Информатика ғылымының негізгі мәселелері, зерттеу объектісі
- 1.4 Информатиканың ғылымдар жүйесіндегі орны
- 1.5 Информатиканың әлеуметтік аспектілері
- 1.6 Информатика халық шаруашылығының саласы ретінде, фундаментальды ғылым және қолданбалы пән
- 1.7 Информатиканың мақсаты және міндеттері

1.1 Информатиканың пайда болуы және дамуы

Информатика термині 60 жылдары Францияда электрондық есептеу машинасы көмегімен ақпаратты автоматты түрде өңдеумен айналысатын саланы атау негізінде пайда болды. Informatique (информатика) француз термині information (ақпарат) және automatique (автоматика) екі сөздің бірігуінен пайда болған және "ақпараттық автоматика немесе ақпаратты автоматты өңдеу" дегенді білдіреді. Ағылшын тілді елдерде оның синонимі computer science (компьютерлік техника туралы ғылым) сәйкес келеді.

Информатика ақпаратты сақтау мен түрлендіруге бағдарланған арнайы құрылымдар - компьютерлер шыққанда және оларды адамның өндірістік және қоғамдық өмірінде кеңінен қолдана бастаған кезде, яғни өткен ғасырдың ортасында дами бастады.

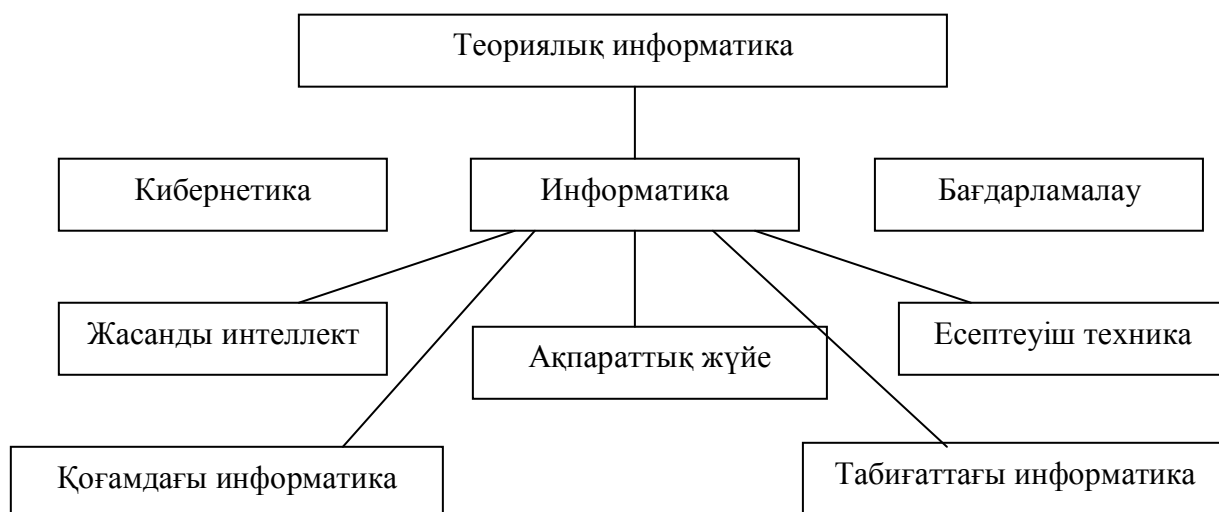
Информатиканың адам қызметінің жеке саласы болып бөлінуі ең алдымен есептеу техникасының дамуына байланысты. Бұған негізгі еңбек сіңірген микропроцессорлық техника болды. Оның 70-жылдардың ортасында пайда болуы екінші электрондық революция болып табылды. Осы уақыттан бастап есептеу машинасының элементтік базасы интегралдық схема және микропроцессор болды, ал компьютерді құру және қолдануға байланысты сала тез қарқынмен дамуда. «Информатика» термині жаңа мағынаға ие болды және ол тек компьютерлік техниканың жетістіктерін бейнелеу үшін ғана емес, сонымен қатар ақпараттарды жіберу және өңдеу процестерімен де байланысты.

Қазіргі кезде информатика - ақпаратты алудың, сақтаудың, түрлендірудің, таратудың, пайдаланудың барлық аспектілерін зерттейтін, адамның практикалық іс-әрекетінің ақпараттық технологияларды пайдалануымен байланысты күшті қарқынмен дамып келе жатқан және үнемі толықтырылып, кеңейіп отыратын ғылым.

Информатика іс-әрекеттің негізгі сегіз бағыттарын қамтитын ғылыми-техникалық пәндердің кешені (1-сурет).

Есептеуіш техника – есептеуіш жүйелерді құрудың жалпы қағидалары талданатын бөлім. Мұнда мәселе техникалық бөлшектер немесе электрондық схемалар жайында емес, есептеуіш (компьютерлік) жүйелердің архитектурасы тұрғысынан алғандағы олардың принциптік шешімі туралы болады. Мұндай классикалық шешім мысалы ретінде – алғашқы компьютерлердің неймандық

архитектурасын, үлкен кезеңге жататын ЭЕМ-дердің шиналық архитектурасын айтуға болады.



1-сурет. Информатиканың бағыттары

Бағдарламалау – программалық қамту жүйелерін талдауға байланысты іс-әрекет. Мұнда қазіргі программалаудың негізгі бөлімдерін атап өтуге болады: жүйелік программалық қамтуды құру және қолданбалы программалық қамтуды құру. Жүйелік программалар арасында – жаңа программалау тілдерін талдау, интерфейстік жүйе құру. Жалпы мақсаттағы қолданбалы программалар арасында ең танымалы – текстілерді өңдеу жүйелері, электронды кестелер және берілгендер базасын басқару жүйелері.

Ақпараттық жүйелер - әр түрлі күрделі жүйелердегі ақпарат ағынын талдау, оларды тиімділеу, құрылымдау, сақтау принциптері мен ақпарат іздеуге байланысты сұрақтарды шешуге бағытталған информатика бөлімі. Ақпараттық-анықтамалық жүйелер, ақпараттық-іздеу жүйелері, қазіргі заманғы аса үлкен ақпаратты сақтау мен іздеу жүйелері (Internet жүйесімен қоса), соңғы уақытта көпшілік назарын аудартуда. Солардың ішінде ең үлкен жүйе қатарына – WWW гипертекстік іздеу жүйесі, ал төменгі деңгейде – 09 нөмірі арқылы қолданылатын қарапайым анықтамалық жүйені жатқызуға болады.

Жасанды интеллект – психология, физиология, лингвистика т.б. ғылымдар тоғысында тұрған күрделі мәселелерді шешуге мүмкіндік беретін информатика саласы. Компьютерді адам сияқты ойлануға үйрете аламыз ба? Бірақ біз адамның өзі де қалай ойланатынын жете білмегендіктен, жасанды интеллект бойынша жасалған тұжырымдар жарты ғасырлық уақыт өтсе де, көптеген қиын мәселелерді әлі де болса шеше алмауда. Бұл облысқа тиесілі талдаулардың негізгі бағыттары - пікірлерді моделдеу, компьютерлік лингвистика, машиналық аударма, эксперттік жүйелерді құру, бейнелерді тану және т.б. Адам мен компьютер арасындағы интеллектуалды интерфейстік жүйе құру сияқты маңызды қолданбалы мәселе де осы жасанды интеллект саласындағы жұмыстардың жетістігіне байланысты.

Кибернетика, көбінесе табиғаттың әр түрлі объектілерін басқарудың жалпы ұстанымдарымен айналысады. Яғни, кибернетика қолданбалы

информатика ретінде жеке объектілерді басқарудан бастап (станоктар, кәсіпорындық құрылымдар, автомобильдер, т.б.) өте күрделі бүтін өндіріс салаларын, банк жүйелерін, байланыс жүйелерін, әуелі адамдар қоғамдастығын басқару жүйелеріне дейін автоматтық немесе күрделігі әр түрлі дәрежедегі автоматтандырылған басқару жүйесін жасау және қолдану саласында қарастырылуы мүмкін.

Кибернетикаға қаланған Н.Винердің негізгі концепциясы адам қызметінің әр түрлі саласындағы күрделі динамикалық жүйелерді басқару теориясын құруға байланысты. Кибернетика компьютердің бар, жоқтығына тәуелсіз болады.

Кибернетика - әр түрлі: техникалық, биологиялық, әлеуметтік және т.б. жүйелердегі басқарудың жалпы ұстанымдары туралы ғылым.

1.2 Есептеу техникасының даму тарихы

Барлығы машинаны санап үйрету немесе көп қатарлы бүтін сандарды қосу идеясынан бастады.

1500 жылы, Леонардо да Винчи – он үш разрядты есептеу құрылғысының жобасын жасады.

1623 жылы, Уильям Шикард алғашқы «есептеуіш машина» жасап шығарды, ол 7-таңбалы сандармен қарапайым арифметикалық амалдар (қосу, азайту) орындай алатын көлемді аппарат болатын.

1644 жылы, Блез Паскаль 5-таңбалы сандармен арифметикалық амалдар орындай алатын алғашқы есептеуіш машина жасап шығарды.

1667 жылы, Сэмюэль Морландтың жасаған есептеуіші қаржылық операцияларды атқаруға арналған болатын.

1674 жылы, Вильгельм Годфрид фон Лейбниц - механикалық есептеу машинасын жасап шығарды, ол тек қана қосу және азайту амалдарымен қатар көбейту амалын да орындай алатын еді.

1770 жылы, Евно Якобсонның механикалық есептеу машинасы Ресейде жасалып шықты.

1774 жылы, Филипп-Малтус Хан көпшілікке арналған алғашқы «есептеу машинасын» - механикалық калькуляторды жасап шығарды.

1820 жылы, Томас де Кольмард «арифмометрі» - алғашқы калькулятор жасады. Бұл 90 жыл көлемінде (аздаған өзгертулер арқылы) пайдаланылған, көптеп шығарылып кеңінен сатылуға түскен бірінші механикалық есептеу құрылғысы болды.

1822 жылы, атақты ағылшын математигі және инженері Чарльз Беббидж құрған және 30 жылдай құрастырылып, жетілдірілген машина алғашқыда «айырымдық», ал одан кейін жобаны көптеп жетілдіргеннен кейін «аналитикалық» деп аталды. «Аналитикалық» машинаға есептеу техникасы үшін негіз болған принциптер қойылды:

1. Амалдарды автоматты орындау.

Үлкен көлемді есептеулерді орындау үшін тек жеке арифметикалық амалдардың тез орындалуы ғана емес, сонымен қатар амалдар арасында

адамның араласуын қажет ететін «бос орындар» болмауы керек. Мысалы, қазіргі калькулятордың көпшілігі амалдарды тез орындағанымен бұл шарттарды қанағаттандыра алмайды. Амалдар бірінен кейін бірі тоқтамай орындалуы керек.

2. «Жүре отырып» енгізілген программамен жұмыс.

Амалдарды автоматты орындау үшін программа орындайтын құрылғыға амалдарды орындау жылдамдығындай жылдамдықпен енгізілуі керек. Бэббидж программаны алдын-ала жазып және оны машинаға енгізу үшін перфокартаны қолдануды ұсынды. Ол кезде перфокарта тоқыма станоктарын басқару үшін қолданылатын болған.

3. Деректерді сақтау үшін арнайы құрылғы – жадының қажеттілігі (Бэббидж оны «қойма» деп атады).

1829 жылы, Уильям Берт механикалық баспа құрылғысына – қазіргі принтерлердің алғашқы нұсқасына (прототипіне) патент алды.

1834 жылы, Чарльз Бэббидждің атақты «аналитикалық машинасы» - перфокарталарға жазылған программаларды орындайтын, алғашқы программаланатын қарапайым компьютер жобасы жасалды.

1871 жылы, Бэббидж компьютердің аналитикалық құрылғысы прототипін және баспа құрылғысы – принтерді жасап шығарды.

1886 жылы, Дорр Фелт мәліметтерді пернелер арқылы енгізуге арналған алғашқы құрылғыны жасап шығарды.

1890 жылы, АҚШ-та халықты санау ісі жүргізіліп, онда Герман Холлерит жасап шығарған «есептеу машинасы» тұңғыш рет қолданылды.

1896 жылы, есептеу машинасын шығаратын алғашқы фирмалардың бірі – Forms Tabulating Company компаниясы құрылды.

1899 жылы, металл сымдарға, кейіннен магниттік таспаларға мәлімет жаза алатын магниттік тәсілмен ақпарат сақтау принципі ашылды.

1919 жылы, Эдвард Хеберн «Энигма» атты мәліметті шифрлайтын машина ойлап шығарды. Кейіннен ол неміс әскери басшылығымен екінші дүниежүзілік соғыста пайдаланылды.

1935 жылы, International Business Machines (IBM) корпорациясы көпшілікке арналған IBM-601 есептеуіштерін шығара бастады.

Конрад Цузе Берлинде алғашқы компьютерлердің бірі болып саналатын Z1 машинасын жасап шығарды.

1936 жылы, Айов университетінің қызметкері Джон Атанасо компьютердің техникалық моделін жасады. Ол К.Берримен бірлесе отырып теңдеулерді шешуге арналған ABC деп аталатын алғашқы компьютерді ойлап тапты, бірақ бұл жоба аяқталмай қалды.

1937 жылы, математик Алан Тюринг кейіннен «Тюринг машинасы» деп аталып кеткен компьютердің алғашқы «математикалық моделін» жасады.

1941 жылы, Германияда Конрад Цузе Z3 машинасын – пайдаланылған кино таспасынан жасалған перфолентадан мәлімет енгізетін алғашқы компьютер жасап шығарды. Компьютерде 2000-нан аса механикалық реле пайдаланылды.

1943 жылы, немістің «Энигма» атты машинасының шифрланған мәліметін шешетін алғашқы электрондық компьютерді Colossus (Ұлыбритания) жасап шығарды. М.А.Ньюмен мен Т.Х.Флауерс ойлап шығарған бұл «Колосс» машинасы 1500-ден аса электрондық шамдардан тұратын еді.

1944 жылы, Говард Эйкен қазіргі компьютерлердің атасы болып саналатын «ASCC Mark I» машинасын шығарды. «Марктың» салмағы 7 тоннадан асып, ол 750000 бөліктен тұрды. Бұл машина артиллерия кестелерін есептеуге арналған әскери мақсатта қолданылды, ол программаларды перфолентадан оқи отырып, 24-разрядты сандарды өңдей алатын еді. Ол машина арқасында неміс армиясының радиохабарларында мәлімет беруге пайдаланылған жасырын кодтардың шешуі табылды.

1945 жылы, Джон фон Ньоманн әлемде бірінші болып сырттан енгізілетін программаны пайдаланатын компьютер құрылғыларының теориялық моделін жасады.

1945 жылы, Джон Мочли мен Преспер Эккрет электрондық шамдардан тұратын сол замандағы ең үлкен әрі қуатты ENIAC компьютерін жасап шығарды. Компьютердің салмағы 70 тоннадан асып, ол 18 мыңдай электрондық шамдардан тұратын. Компьютердің жұмыс істеу жиілігі 100 КГц-тен (секундына бірнеше жүз операция) аспайтын.

1949 жылы, Морис Уилкис есептеу программасын өз жадында сақтай алатын алғашқы электрондық цифрлық компьютер – EDSAC машинасын жасап шығарды. Осы EDSAC ақпарат шығару үшін катод түтікше негізінде (монитор прототипі) жасалған дисплейді пайдаланатын әлемдегі тұңғыш компьютер болды.

1950 жылы, Йосуито Накамато иілгіш магниттік дискілерге мәлімет жазу технологиясын ойлап тапты. Осы жылы, алғашқы Кеңес өкіметінің МЭСМ (Малая Электронная Счетная Машина) атты компьютерін Украина Кеңестік Социалистік Республикасының Ғылым академиясындағы Электротехника институтында С.А.Лебедев жасап шығарды.

1953 жылы, көпшілікке арналған IBM 650 компьютері жасалды.

1955 жылдан Массачусет технологиялық институтында транзисторлардан тұратын алғашқы тәжірибелік компьютер TX-0 жасалды.

1954 жылы, Texas Instruments компаниясы кремний транзисторларын өндірісте шығара бастады.

1955 жылы, алғашқы транзисторлық компьютер TRIDAC іске қосылды.

1956 жылы, IBM корпорациясы алғашқы көлемді ақпарат жинақтауыш құралды – винчестер прототипі болып табылатын КАМАС 305 қатты дискісін жасады. Бұл монстр бір қорапта орналасқан әрқайсысының диаметрі бір метр шамасында болатын 50 магниттік дискіні біріктірді, олардың жалпы көлемі 5 Мбайт шамасында болды.

1957 жылы, алғашқы арнайы бизнес–компьютер NCR 304 жасалды.

1958 жылы, Джек Килби мен Роберт Нойс бір-бірінен тәуелсіз түрде логикалық элементтер тізбегін кремний кристалы бетіне орналастыруға мүмкіндік беретін технологияны ойлап табады. Осы технология арқылы қазіргі микропроцессор прототипі – интегралдық микросхема пайда болды.

1959 жылы, С.Лебедевтің басшылығымен Үлкен Электрондық Есептеу Машинасы (БЭСМ) жасалып шықты. Есептеу жылдамдығы секундына 10000 операция болды.

1960 жылы, IBM фирмасы IBM 7030 қуатты есептеу жүйесін жасап шығарды.

1961 жылы, Манчестер университетінің қызметкерлері Т.Килбурн басшылығымен тұңғыш рет виртуальды жады концепциясын жүзеге асырып, «Atlas» есептеу машинасын жасап шығарды.

1962 жылы, Teletype компаниясы бірінші рет пернетақта мен монитор құрылғысын жасап шығарды.

1964 жылы, IBM фирмасы әлемде тұңғыш рет IBM 2250 компьютеріне арналған монитор шығарды. Оның ақ-қара түсті (монохромды) дисплейінің мөлшері 12*12 дюйм болды, экраны 1024*1024 нүктелерден құралып, экрандағы көрініс 40Гц жиілікпен жаңартылып тұрды.

1964 жылы, Сеймур Крэй алғашқы рет CDC 6600 суперкомпьютерін шығарды.

1965 жылы, Digital Equipment компаниясы алғашқы PDP-8 мини компьютерін шығарды.

1967 жылы, IBM фирмасы флоппи-дискінің алғашқы прототипін көрсетті. Осы жылы, «Бір кристалдағы компьютер» концепциясы жасалды.

1968 жылы, Уэйн Пикетт «винчестер» концепциясын жасап, қатты магниттік дискідегі прототипті мәлімет жинақтауыш құрылғы жасауға болатынын көрсетті.

1979 жылы, Zilog фирмасы өзінің 16-разрядты микропроцессорын шығарды. Оған жалғаса Intel фирмасы өзінің жаңа Intel 8088 процессорын шығара бастады. Компьютерде алғашқы бейнелік ойындар көрсетіліп, енді соларды көрсетуге арналған қосымша құрылғылар шығарыла бастады.

1980 жылы, Atari компьютері осы жылдың ең кең таралған компьютеріне айналды.

Seagate Technologies фирмасы дербес компьютерлерге арналған диаметрі 5,25 дюйм, сыйымдылығы 5 Мбайт, «Винчестер» деп аталған қатты дискіні алғаш шығарды.

1981 жылы, IBM PC (IBM 5150) компьютері шығарылды, ол жұмыс істеу жиілігі 4,77 МГц, жедел жады көлемі 64 Кбайт, тұрақты жады көлемі 16-64 Кбайт болатын Intel 8088 процессоры негізінде жасалып шыққан болатын. Компьютер дисплеймен және сыйымдылығы 160 Кбайт флоппи-дискі жетегімен жабдықталған болатын.

Intel фирмасы тұңғыш рет жылжымалы нүктелі сандармен күрделі есеп-қисап жұмыстарын жүргізуге арналған арнайы процессор – алғашқы сопроцессорды әлемге жария етті.

1981 жылы, Apple фирмасы Apple III компьютерін шығарды. Osborn компаниясы өзінің алғашқы портативті компьютерін шығарды.

Көпшілікке арналған Seagate қатты дискісі шығарылды, оның сыйымдылығы 5 Мбайт.

1982 жылы, Commodore фирмасы өзінің 64 Кбайт RAM жадымен, 20 Кбайт тұрақты жадымен жабдықталған атақты Commodore 64 компьютерін шығарды. Осы компьютермен үзеңгілес Sinclair фирмасының Sinclair ZX компьютері де осы жылдың кең таралған тұрмыстық компьютерлері қатарына жатқызылды.

1982 жылы барлығы 20-дан аса фирмалар өз компьютерлерін сатуға шығарды, олардың ішінде – Toshiba, Sharp, Matsushita, Sanyo, т.с.с. бар. Сатуға IBM PC компьютерлерінің алғашқылары да шыға бастады. IBM корпорациясы өзінің 130000 транзистордан тұратын алғашқы 16-разрядты 80286 процессорын шығарды. Жұмыс жиілігі – 6-12 МГц. Жылдамдығы – секундына 1,5 млн операция шамасында.

1983 жылы, Commodore фирмасы түрлі түсті экранды (5 түсті) алғашқы портативті компьютер шығарды. Компьютер салмағы - 10 кг.

1983 жылы, IBM корпорациясы 8088 процессорлы (алғашқы моделінің жұмыс істеу жиілігі - 4,77МГц), 10-20 Мбайт қатты дискісі, 360 Кбайтты дискеті және 128-256 Кбайт жедел жады бар PC XT компьютерін жасап шығарды. Компьютерге Microsoft фирмасы жасаған MS-DOS 2.0 операциялық жүйесінің жаңа нұсқасы орнатылды.

1984 жылы, IBM корпорациясы жұмыс жиілігі 6-8 МГц 80286 процессоры негізінде 256 Кбайт жедел жады, 1,2 Мбайт иілгіш дискетке арналған екі дискжетектері бар, 20-30 Мбайт қатты дискімен жабдықталған компьютер жасап шығарды.

1988 жылы, Compaq фирмасы бұдан кейінгі барлық DOS жүйелеріне стандарт түрінде кірген - 640 Кбайт жедел жады бар алғашқы компьютер шығарды.

1988 жылы, Intel фирмасы 386 процессорының 80386SX класындағы «қысқартылған» нұсқасын (сопроцессор қосылмаған) шығарды. Оның жұмыс жиілігі - 16-33 МГц, жылдамдығы секундына 2-3 млн операция болды.

1989 жылы, Intel фирмасы жиілігі 20 МГц және есептеу жылдамдығы секундына 20 млн операция болатын 486DX процессорлар тобының (1,25 миллион транзисторлары бар, кейінгі моделдерінде - 1,6 млн) алғашқысын шығарды.

1990 жылы, Intel фирмасы жаңадан 32-разрядты 80486SX процессорын шығарды. Жылдамдығы - секундына 27 млн операция. MSDOS 4.01 және Windows 3.0 жүйелері шықты.

1990 жылы, IBM корпорациясы экраны түрлі-түсті екпінді кристалды матрица (ACLCD) негізінде жасалған алғашқы ноутбук – Thinkpad 700C шығарды.

1991 жылы, AMD фирмасы тактілік жиілігі 40 МГц болатын жетілдірілген Intel - 386DX процессорының жаңа түрін, ал Intel фирмасы – жиілігі 20 МГц болатын 486 SX процессорын (900 000 транзистор шамасында) ұсынды.

1992 жылы, Intel фирмасы жүйелік шинасы жиілігі «екі еселенген» (1,25 млн транзистор) 486DX2/40 процессорын шығарды. Жылдамдығы секундына 41 млн операция.

1993 жылы, Intel компаниясы шиналардың жаңа стандартын шығарып, қосымша карталарды қосуға арналған PCI слотын орната бастады. Осы жылы процессорлардың жаңа буыны болып саналатын 32-разрядты Pentium (3,1 млн транзисторы бар) компьютерлеріне арналған алғашқы Intel процессоры жасалды. Оның жұмыс жиілігі - 6 МГц, жылдамдығы секундына 110 млн операциядан жоғары

1994 жылы, Intel фирмасы 486DX4-75 (1,6 млн транзисторы бар) процессорын шығарды. AMD фирмасы 486 процессорлардың соңғы нұсқасы ретінде AMD486DX - 120 жылдамдығы, секундына 90 млн операция процессорларын көпшілікке ұсынды. Intel қуатты жұмыс станцияларына арналған Pentium Pro процессорын шығарды. Оның кэш-жады көлемі 256 Кбайт шамасында, жұмыс жиілігі – 150 МГц-тан жоғары, құрамында 5,5 млн транзисторлары болды.

1996 жылы, Intel фирмасы мультимедиялық құрылғылармен жұмыс істейтін, жаңа мүмкіндіктерді сүйемелдейтін Pentium MMX процессорын (жиілігі - 166 МГц-тен жоғары, 4,4 млн транзисторы бар) ұсынды. «Үлкен» тұрмыстық компьютерлерге арналған сұйық кристалды мониторлар өндірісі іске қосылды.

1997 жылы, Intel фирмасының жаңа процессоры - Intel Pentium II (7,5 млн транзисторы бар) шықты. Жиілігі – 233 МГц-тен жоғары, есептеу жылдамдығы - секундына 250 млн операция.

1.3 Информатика ғылымының негізгі мәселелері, зерттеу объектісі

Адамның өзінің іс-әрекетінде ақпарат жиымдарын пайдалануы жылдан жылға көбейіп келеді. Мысалы, егер 1940 жылдан 1950 жылға дейін ақпарат көлемі, шамамен 10 жылда екі есе көбейсе, қазіргі кезде осындай екі еселену 1-2 жылда болады. Ақпаратпен жұмыс істегенде, оны сақтаудың, жіберудің, іздеудің, өңдеудің қолайлы және тиімді түрлерімен байланысты көптеген мәселелерді шешуге тура келеді. Бұдан басқа, ақпараттың құрылымын анықтаумен байланысты мәселелер де кездеседі. Сондай-ақ ақпараттың жалпы қасиеттерін зерттеу керек. Мұның бәрімен информатика деп аталатын жаңа ғылым шұғылданады.

Информатика - ақпарат алу, жіберу, өңдау, сақтау, ұсыну процестерін зерттеумен, қоғам өмірінің барлық саласында ақпараттық техника мен технология жасау, ендіру және тиімді пайдалану мәселелерін шешумен шұғылданатын ғылыми-техникалық іс-әрекет саласы, ғылыми-техникалық прогрестің басты бағыттарының бірі.

Информатика – есептеу техникасы құралдарының көмегімен мәліметтерді қабылдау, құру, сақтау, түрлендіру, өңдеу және жеткізу тәсілдерін жүйеге келтіретін жаңа техникалық ғылыми пән.

Информатика:

- Ақпарат жинау, сақтау, жіберу, өңдеу және беру әдістері мен құралдарын зерттеумен байланысты – техникалық;

- Ақпараттың мәнін сипаттау тәсілін анықтайтын, оның сипаттау тілдерін зерттейтін – семантикалық;

- Ақпаратты кодтау әдістерін сипаттайтын - прагматикалық;

- Ғылыми-ақпараттық іс-әрекеттің кейбір түрлерін, оның ішінде индекстеуді, автоматты реферлеуді, машиналық аударманы формальдау және автоматтандыру мәселелерін шешумен байланысты синтактикалық мәселелер топтарын зерттейді.

Информатиканың негізгі мәселесі – есептеу техникасының аппараттық және программалық құралдармен жұмыс істеу тәсілдерін жүйелеу болып табылады. Бүгінгі таңда информатиканың негізгі мәселелерінің құрамына келесі бағыттарды жатқызуға болады:

- Есептеу жүйелерінің архитектурасы (мәліметтерді автоматты түрде өңдеуге арналған жүйелерді құрудың тәсілдері мен әдістері);

- Есептеу жүйелерінің интерфейсі (аппараттық және программалық жасақтаманы басқару тәсілдері мен әдістері);

- Бағдарламалау (компьютерлік бағдарламаларды даярлау әдістері);

- Мәліметтерді түрлендіру;

- Ақпаратты қорғау;

- Автоматтандыру;

- Стандарттау (әр түрлі есептеу техникасы құралдарына сәйкес аппараттық және программалық құралдар арасындағы, мәліметтердің берілу пішімі арасындағы үйлесімділікті қамтамасыз ету).

1.4 Информатиканың ғылымдар жүйесіндегі орны

Информатиканың дәстүрлі ғылымдар жүйесіндегі (техникалық, жаратылыстану, гуманитарлық және т.б.) орнын қарастырайық. Бұл информатиканың жалпы білім беру курсының басқа да оқу пәндерінің арасында алатын орнын анықтауға мүмкіндік береді.

А.П.Ершов бойынша, информатика – “фундаментальды жаратылыстану ғылымы”. Академик Б.Н.Наумов информатиканы “ақпараттың жалпы қасиеттерін, оны өңдеудің (жинақтау, сақтау, түрлендіру, орын ауыстыру, беру) процестерін, әдістері мен құралдарын зерттейтін жаратылыстану ғылымы” ретінде анықтайды.

Фундаменталды және жаратылыстану ғылымдары арасындағы айырмашылыққа тоқталайық. Негізгі ұғымдары жалпы ғылымдық сипатта болатын, басқа да ғылымдар мен іс-әрекет түрлерінде қолданылатын ғылымдар фундаменталды ғылымдарға жатады. Мысалы, математика және философия ғылымдарының осы топқа жататынына ешкім дау айтпайды. Осы қатарға информатиканы да қоюға болады, өйткені “ақпарат”, “ақпаратты өңдеу процесі” т.б. ұғымдар жалпы ғылымдық сипатқа ие.

Жаратылыстану ғылымдары – физика, химия, биология және т.б. - әлемнің біздің санамызға тәуелсіз объективті болмысымен байланысты. Олардың қатарына информатиканы жатқызу – табиғи, биологиялық және қоғамдық ғылымдар жүйесіндегі ақпаратты өңдеу заңдарының бірлігін көрсетеді.

Алайда, көптеген ғалымдар информатиканың басқа да ғылымдар тобымен - техникалық және гуманитарлық (немесе қоғамдық) ғылымдармен байланысты екенін атап өтуде.

Ақпаратты өңдеудің машиналық жүйелерін құру және қызмет етуі информатикаға техникалық сипат береді. Айталық, академик А.А.Дородницын информатиканың құрамын “үш бөлінбейтін және байланысқан бөліктер: техникалық, программалық, алгоритмдік құралдар” түрінде анықтайды. “Информатика және есептеуіш техника негіздері” пәнінің бастапқы атауы қазір “Информатика” болып өзгертілді. Ол техникалық, программалық және алгоритмдік құралдарды оқытуға байланысты бөлімдерді қамтиды. Информатика ғылымына гуманитарлық (қоғамдық) ғылымдар сипаты да тән. Бұл информатика ғылымының әлеуметтік ортаның дамуына қосқан үлесімен түсіндіріледі. Сонымен, информатика – ғылыми білімнің кешендік, пәнаралық саласы болып табылады.

1.5 Информатиканың әлеуметтік аспектілері

“Әлеуметтік аспектілер” терминін көптеген ғылымдарға, соның ішінде фундаменталды ғылымдарға қолдану үйлесімсіз екені белгілі. Айталық, “Математиканың әлеуметтік аспектілері” деген сөйлемнің еш мағынасы жоқ. Алайда информатика – тек қана ғылым емес, жоғарыда айтылғандай, “...өндірістік, коммерциялық, әкімшілік және әлеуметтік негізде бірлесе әрекет ету кешені”.

Қоғамды ақпараттандыру – ақпараттық технологияларды қоғамның барлық өмірі мен іс-әрекет ету салаларына ендіру процесі. Көптеген әлеуметтанушылар мен саясаттанушылар әлемнің ақпараттық қоғам сатысында тұрғанын айтуда. В.А.Извозчиков мынадай анықтама береді: “Ақпараттанған” (“компьютерленген”) қоғам ретінде біз сол қоғам мүшелерінің барлық өмірі мен іс-әрекетіне компьютер, телематика, кітапханалардың бағалы қорына қол жеткізе алатын, есептеулерді орасан шапшаңдықпен жүргізетін, кез келген ақпаратты өңдей алатын, нақты және болжам оқиғаларды, процестер мен құбылыстарды моделдей алатын, өндірісті басқара алатын, оқытуды автоматтандыратын информатиканың басқа да құралдарын интеллектуалды ой еңбегінің құралы ретінде енгізуді түсінеміз”. “Телематика” - ақпаратты ара қашықтықтан (дәстүрлі телефон, телеграфтан басқа) өңдеу құралдары.

Соңғы жарты ғасырда ақпараттандыру – адамдардың тікелей материалдық өндірістен ақпараттық салаға қарай бет бұруының ең басты себебі болып отыр. XX ғасырдың ортасында дамыған елдердегі өндірістік жұмысшылар мен шаруалар бүкіл халықтың 2/3-ден астам бөлігін құраса, қазіргі кезде бұл көрсеткіш 1/3-ден кем. Уақыт өткен сайын “ақ жағалылар” деген атқа ие болған, яғни тікелей материалдық құндылықтар жасамайтын, керісінше ақпаратты өңдеумен айналысатын адамдар саны көбеюде: оларға мұғалімдер, банк қызметкерлері, программистер және т.б. қызметкерлер тобы жатады.

Ақпараттандыру өндіріс пен ауыл шаруашылығының дәстүрлі салаларының келбетін де өзгертті. Өндірістік роботтар, станоктар жай құралға

айналды. Ауыл шаруашылығы өндірісіндегі жаңа технологиялар тек еңбек өнімділігін арттырып қана қоймай, ондағы еңбекті жеңілдетеді, білімді адамдарды шоғырландырады.

Осылайша, компьютерлендіру мен ақпараттық технологиялар әлемге тек қана жақсылық алып келетіндей көрінгенмен, мұның “көлеңке” тұстары да бар. Бұл бір жағынан, көптеген адамдардың материалдық және интеллектуалдық деңгейін, білімді адамдар санын көбейтсе, екінші жағынан, жоғары әлеуметтік тартыс тудырады. Айталық, өндірісте өнеркәсіптік роботтардың пайда болуы көптеген адамдарды жұмыссыз қалдыруы мүмкін. Соған байланысты мамандық номенклатурасы да өзгереді. Көптеген адамдар не мамандығын, не жұмысын ауыстыруына тура келеді.

Информатиканың жедел дамуы демократиялық қоғам үшін қауіпті келесі процестің пайда болуына әкелді, яғни әрбір азамат жайлы барлық мәліметтер әр түрлі деректер банкісінде (мемлекеттік және мемлекеттік емес) жинақталуы. Бұл деректерде адамның жұмысы, денсаулығы, мүлкі т.б. туралы құнды деректер жинақталатындықтан және ол басқа тұлғаларға (мемлекет немесе қылмыскер) белгілі болуы мүмкін жағдайда бұл құбылыс өте қауіпті.

Міне, осындай теріс әлеуметтік жағдайларды үкімет пен қоғам үнемі естен шығармай, басқа да механизмдерді қарастыруы қажет.

1.6 Информатика халық шаруашылығының саласы ретінде, фундаментальды ғылым және қолданбалы пән

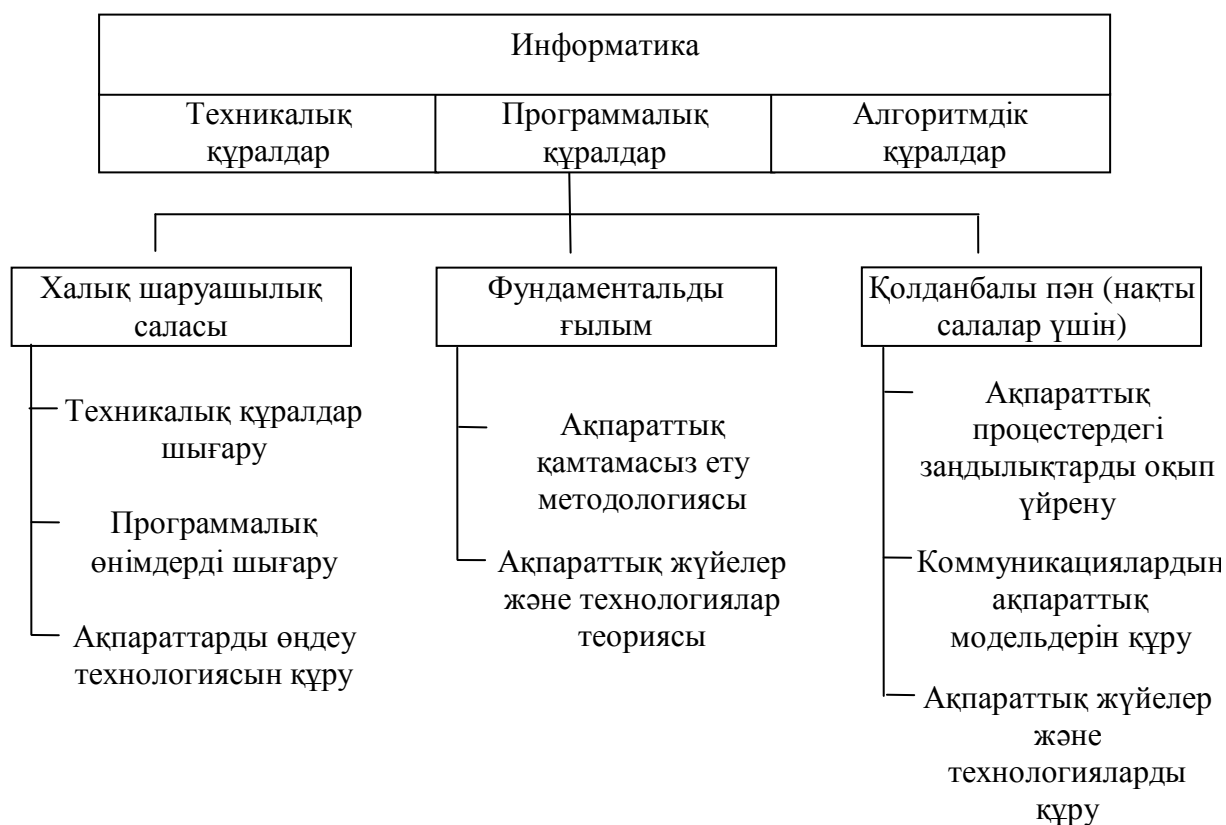
Информатика кең мағынада ғылымның, техниканың және өндірістің әр түрлі саласының ең бастысы компьютерлер көмегімен және адамзат қызметінің барлық саласындағы телекоммуникациялық құралдар байланысының көмегімен бірігуі болып табылады.

Информатиканы тар мағынада – үш өзара байланысты бөліктерден – техникалық құралдар (hardware), программалық құралдар (software), алгоритмдік құралдардан (brainware) тұрады деп қарастыруға болады. Жалпы информатиканы, сондай-ақ оның әрбір бөлігін әдетте әр түрлі тұрғыдан (2-сурет) халық шаруашылық саласы, фундаментальды ғылым, қолданбалы пән деп қарастырады.

Информатика халық шаруашылығының саласы ретінде шаруашылықтың әр түрлі формасындағы бірыңғай кәсіпорындар жиынынан тұрады, онда компьютерлік техника шығарумен, программалық өнімдер және ақпараттарды өңдеудің қазіргі заманғы технологияларын құрумен айналысады.

1.7 Информатиканың мақсаты және міндеттері

Информатикадағы фундаментальды зерттеулердің мақсаты – кез-келген ақпараттық жүйелер жөніндегі жалпы білімдерді талдап қорыту, оларды құру және қызметінің жалпы заңдылықтарын айқындау.



2-сурет. Информатиканың сала, ғылым, қолданбалы пән ретіндегі құрылымы

Информатика қолданбалы пән ретінде:

- ақпараттық процестердегі заңдылықтарды зерттеу (жинақтау, өңдеу, тарату);
- адам қызметінің әр түрлі саласындағы қатынастардың ақпараттық моделін құру;
- нақты салалардағы ақпараттар жүйесі мен технологияларды құру және олардың өмір сүру кезеңдеріне қатысты кепілдемелер өндіру: жүйелерді жобалау және құру кезеңдері үшін, оларды өндіру, қызмет ету және т.б.

Информатиканың басты қызметі ақпараттарды түрлендіру құралдары мен әдістерін құру және оларды ақпараттарды өңдеудің технологиялық процестерін ұйымдастыру болып табылады.

Информатиканың негізгі міндеттері мыналар:

- табиғаттағы кез-келген ақпараттық үрдістерді зерттеу;
- ақпараттық (информатикалық) техниканы жасау;
- ақпараттық үрдістерді зерттеудің нәтижесінде алынған қорытындылардың негізінде ақпаратты өңдеудің жаңаша технологиясын жасау;
- компьютерлік техника мен технологияны жасау және оларды қоғамдық өмірдің барлық салаларында енгізу мен тиімді пайдаланудың ғылыми және инженерлік проблемаларын шешу.

Дәріс №2. Ақпарат және оның қасиеттері

Дәріс жоспары:

2.1 Ақпарат философиялық категория ретінде

2.2 Ақпаратты түрлендіру

2.3 Хабар – ақпаратты тасымалдаушы

2.4 Хабарды сигналдар көмегімен жіберу.

2.5 Хабарды дискретизациялау

2.6 Ақпарат мөлшерінің бірліктері

2.7 Кодтау теориясы. Абстрактілі алфавит. Кодтау. Декодтау

2.1 Ақпарат философиялық категория ретінде

Ақпарат (лат. *informatio* – түсіндіру, мазмұнын айту, хабардар болу) – ғылымның жалпы ұғымдарының бірі, ол қандай да бір мәлімет, қандай да бір деректер, білімнің және т.б. жиынтығы.

Жүйелік-кибернетикалық тәсілде ақпарат кез-келген кибернетикалық жүйенің үш фундаментальды аспектілерінің контекстінде қарастырылады:

- *ақпараттық*, жүйеде сыртқы дүние мен ішкі ортаның бейнелену процестерінің белгілі бір жиынын сәйкес сигналдарды жинау, жинақтау және өңдеуге байланысты;

- *басқару*, жүйенің қызмет ету, алынған ақпарат әсерімен оның жүру бағытын және мақсатқа жету дәрежесін ескеретін процестерді;

- *ұйымдастырылатын*, жүйедегі құрылғыны және жүйенің жетілдіру дәрежесін сипаттайтын.

Ақпарат ұғымының өзі кем дегенде үш объектінің – ақпарат көзі, ақпаратты қолданушы және жіберуші ортаның болуын ұйғарады.

Ақпарат таза түрінде берілмейді, қабылданбайды және сақталынбайды. Оны тасушы хабар болып табылады. *Хабар* – ақпарат көзі арқылы бекітілген және шартты физикалық символдардың тізбектерінің көмегімен өрнектелген кодталған оқиғаның эквиваленті. Хабарды жіберу құралдары ретінде – ақпараттардың байланыс каналы болып табылады. Байланыс каналдары арқылы хабар осы каналға ғана тән сигнал түрінде беріледі.

Сигнал - байланыс каналдары арқылы таралатын және белгілі бір оқиға туралы, бақылау немесе тексеру объектісінің қалпын, басқару каналдары нұсқаулары және т.б. белгі, физикалық процесс немесе құбылыс. Сонымен, электр сигналдары электрондық және электрлік тізбектерде, акустикалық газ, сұйық және қатты денелер ортасында таралады. Ортамен берілетін және қолданушы қабылдайтын ақпаратты алып жүретін сигнал қолданушы үшін осы ақпараттың түсуінен өзгеше белгілі бір мағынасы бар. Бұл арнайы келісім арқылы жүзеге асырылады, соған сәйкес сигнал интерпретацияланады, яғни алынатын сигналдан мағынасы қолданушыға түсініктісі алынады. Яғни, қабылданған сигналды қарапайым физикалық тіркеу, егер бұл сигналдан қолданушыға белгілі ережелер көмегімен мағынасы бар сигнал алынбаса ақпарат көзінен ақпарат алды дегенді білдірмейді. Мұнда атап өтетін нәрсе, сигналдың өзі алып келетін оқиға немесе құбылыс туралы ақпаратпен тікелей

физикалық байланыста болмауы да мүмкін. Бұл кезде ақпарат объектілердің және құбылыстардың қасиеті ретінде көптеген жағдайларды тудырады, олар бейнелеу көмегімен бір объектіден екіншісіне беріледі және оның құрылымында сақталынып қалады.

Адам миы сезім органдарының көмегімен ақпараттардың кең көлемін алады. Ақпарат негізі ойлау материалы болып табылады және барлық ойлау қызметінің негізі болып табылады. Ақпараттар теориясының негізі – мазмұны әр түрлі болатын әр түрлі хабарлар ортақ тілге келтірілуі мүмкін, ал осы хабарлар әкелетін ақпараттар сандық өлшенуі мүмкін. Осындай сандық өлшем арқылы берілетін хабарларды олардың түріне тәуелсіз бағалауға болады. Бұл жалпы ақпараттар теориясының негізін құрады. Энергия ұғымын енгізу арқылы табиғаттың барлық құбылысын бір ортақ көзқараспен қарастырған сияқты бірдей өлшемді ақпараттар санындағы ақпарат ұғымы әр түрлі процестерді оқып үйренуді көздейді. Берілетін ақпарат саны және қабылдаушыға ақпарат әсері оны жеткізгенге жұмсалған энергия санымен анықталмайды. Мысалы, автоматталған робототехникалық кешендерде қазіргі заманғы машиналардың күрделі детальдары мен тораптарын жасау және оларды жинақтау энергетикалық мағынада қуаты аз басқару сигналдарының көмегімен жүзеге асырылады. Сигналдарды технологиялық тізбектерге ендірілген микропроцессорлар станоктар мен роботтардың орындаушы органдарына жіберіледі.

Философияда отыз жылдан астам уақыттан бері ақпараттың бір-біріне қарама-қайшы, екі әр түрлі концепциясы – атрибуттік және функционалдық өмір сүруде.

Атрибуттік концепция ақпаратты барлық материалдық объектінің қасиеттері деп қарастырады.

Функционалдық концепция, керісінше, ақпаратты өздігінен ұйымдастырылған жүйелердің қызметімен байланыстырады.

Бұл концепциялардың әрбіреуі ақпараттың белгілі бір аспектіні бейнелейді және сондықтан оларды бірлікте қарастыруға болады. Мұнда атрибуттік концепция ақпараттың материалдық объектінің атрибуты ретінде оны қолдану процесіне тәуелсіздігіне екіпін түсірді, осылайша ақпараттың статикалық аспектісін бейнелейді. Кибернетикалық жүйенің қызметі ақпараттың динамикалық аспектісін көрсетеді, ол ақпаратты ақпараттық процестердің динамикасы арқылы анықтайды. Сондықтан объектілер процестер тудырады, өйткені осы объектілер ақпараттан тұрады, ол белгілі бір шарттарда бейненің жеткізуші бөлігі ретінде жүзеге асады. Субъект объектіден ақпарат алады және оны таным немесе басқару контурына қосады. Яғни, ақпараттың мазмұны, мағынасы айқындалады, бағалы болады, оның семантикалық және прагматикалық аспектілері ашылады. Бұл жағдайда ақпараттың салыстырмалы және абстрактілі сипаттамасы туралы айтуға болады.

Ақпараттың *салыстырмалы* сипаттамасы – қандай да бір объект тек басқа, қатаң анықталған және сол объектімен өзара әрекеттесетін объектіге қатысты ақпарат көзі болып табылатындығынан көрінеді. Ол берілген нақты жағдайларда осы ақпараттарды ала алады және өз мақсатында қолданады.

Ақпараттың *абсолютті* сипаттамасы – ақпарат сияқты қасиетке ие болатын материалдық объектілердің жоқ екендігінен көрінеді. Бұл ашық объектілерге (өзара әрекеттесуші жүйелер) қатысты да, сондай-ақ жабық объектілерге қатысты орындалады.

Ақпарат материя мен энергия сияқты фундаментальды категориялармен қатар тұрып, кең ұғымға айналды және одан да терең және кең айқындала түсуде. Зерттеу саласына байланысты ақпараттың көптеген анықтамасы бар:

- сыртқы дүниеге бейімдеу процесінде алынған мазмұнды белгілеу (Н.Винер);

- энтропияны жоққа шығару (Бриллюэн);

- коммуникация және байланыс процесінде анықталмағандық жойылады (К. Шеннон);

- әр түрлі хабарлар беру (Эшби);

- құрылымның күрделілігінің өлшемі (Моль);

- таңдау ықтималдығы (Яглом).

Осы анықтамалардың әрбіреуі кеңістіктегі және уақыт бойынша материя мен энергияның таралуының біртекті емес өлшемі ретінде және әлемдегі барлық процестердің өзгеру өлшемі ретіндегі ақпарат ұғымының белгілі бір жақтарын ашады.

Ақпарат ғылыми зерттеу объект ретінде *техникалық, семантикалық және прагматикалық* аспектілерге бөлінеді.

Техникалық аспектіде дәлдік, сенімділік, хабарларды беру жылдамдығы, сигналдарды беру каналдарының техникалық құралдары мен әдістерін құру, оларды бөгеуілдерден қорғау мәселелері қарастырылады.

Семантикалық аспектіде зерттеулер кодталған сигналдар көмегімен берілген хабардың мағынасын дәл беру мәселелерін шешуге бағытталған.

Ақпараттарды зерттеудің прагматикалық аспекті – алынған хабар қолданушы үшін оның кейінгі іс-әрекетіне ықпалдылығы тұрғысынан қаншалықты бағалы екендігінде.

Ақпарат философиялық категория ретінде болмыстың жалпы формаларын, олардың байланысын және өзара шарттылығын ғана емес, сондай-ақ табиғатта, қоғамда, танымда төменнен жоғарыға қарай даму факторы болып табылады. Осыған байланысты ақпараттың үш формасы бар: *биологиялық* ақпарат түрі организмдер және олардың арасындағы, *машиналық* ақпарат машиналар арасындағы және олардың ішіндегі және *әлеуметтік* ақпарат адам қоғамындағы формалар.

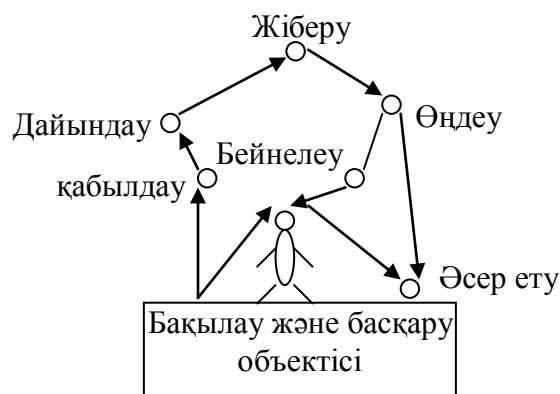
2.2 Ақпаратты түрлендіру

Жүйеде ақпараттарға қатынау барысында жеке кезеңдерді атап өтуге болады.

Ақпараттың материалды жеткізушісі сигнал болғандықтан, бұл сигналдардың қатынасу және түрлендіру кезеңдері болып табылады (3-сурет).

Ақпаратты қабылдау кезеңінде белгілі объект туралы ақпаратты алады және анализ жасалады, соның нәтижесінде объект бейнесі құрылады, оны тану және бағалау жүргізіледі. Мұнда бізді қазіргі уақытта қызықтыратын ақпаратты

кедергі болатын ақпараттан бөліп алу керек, ол көп жағдайда бірқатар қиыншылықтарға байланысты. Қабылдаудың қарапайым түрі екі қарама-қарсы күйді: бар болу («иә») және жоқ болу («жоқ») айыру.



3-сурет. Сигналдардың қатынасу және түрлендіру кезеңдері

Ақпаратты даярлау кезеңінде нормальдау, аналогтық-сандық түрлендіру, шифрлау сияқты амалдар жүргізіледі. Кей жағдайда бұл кезең қабылдау кезеңіндегі көмекші кезең ретінде қарастырылады. Қабылдау және дайындау нәтижесінде жіберу немесе өңдеуге ыңғайлы түрдегі сигнал болады.

Жіберу және сақтау кезеңінде ақпарат бір орыннан екінші орынға немесе бір уақыттан екіншісіне беріледі. Бұл кезеңде пайда болатын теориялық есептер бір-біріне жақын болғандықтан, ақпаратты сақтау кезеңі жеке кезең болып бөлінбейді. Мұнда ақпаратты жіберу кең мағынаға ие болады. Қашыққа ақпаратты жіберу үшін физикалық табиғаты әр түрлі каналдар қолданылады, олардың ішінде кең таралғаны электрлік және электромагниттік. Соңғы он жылдықта алдыңғы қатарлы оптикалық канал кең таралды. Ақпаратты сақтау үшін негізінен жартылай өткізгішті және магниттік тасымалдаушылар қолданылады. Шуылға ұшырайтын каналдың шығысында сигналды алу екінші ретті түсіну сипатында болады.

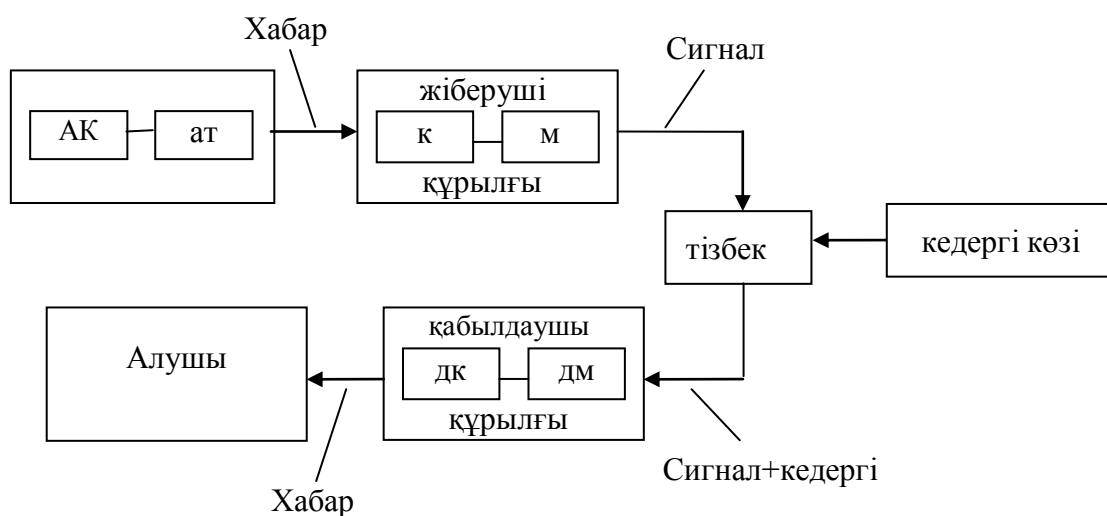
Ақпаратты өңдеу кезеңінде ақпаратты түрлендіру (басқа кезеңдердегі сияқты) ақпараттық техника құралдары немесе адамның көмегімен жүзеге асырылады. Егер өңдеу процесін формализацияласақ, оны техникалық құралдар орындайды. Қазіргі кездегі күрделі жүйелерде бұл функция ЭЕМ-ге және микропроцессорге жүктеледі. Егер өңдеу процесі формализацияланбаса және шығармашылық тәсілді қажет етсе, ақпаратты өңдеуді адам орындайды. Басқару жүйелеріндегі өңдеудің негізгі мақсаты басқару әсерлерін таңдау есептерін шешу болып табылады (шешім қабылдау кезеңі).

Ақпаратты бейнелеу кезеңі адамның қатысуына байланысты кезеңдердің алдында болу керек. Бейнелеу кезеңінің мақсаты – адамға қажетті ақпаратты оның сезім органдарына әсер етуге қабілетті құрылғы көмегімен көрсету.

Әсер ету кезеңінде ақпарат жүйеде қажетті өзгерістерді жүзеге асыру үшін қолданылады.

2.3 Хабар – ақпаратты тасымалдаушы

Ақпаратты жіберудің бір каналды жүйесінің құрылымдық сұлбасы 4-суретте келтірілген.



4-сурет. Ақпаратты жіберудің бір каналды жүйесінің құрылымдық сұлбасы

Ақпарат жүйеге хабар түрінде түседі. *Хабар* деп ақпараттардан тұратын алғашқы сигналдар немесе белгілер жиынтығын түсінеміз. Хабар көзі жалпы жағдайда (зерттелетін немесе бақыланатын объектінің) ақпарат көзі - *АК* және (датчик, оператор-адам және т.б.) алғашқы түрлендіруші - *АТ* жиынтығын құрайды. Хабар дискретті және үздіксіз болып бөлінеді.

Дискретті хабар ақпарат көзінің жеке элементтерін - белгілерді біртіндеп шығару нәтижесінде қалыптасады. Әр түрлі белгілердің жиыны *хабар көзінің алфавиті*, ал белгілер саны – *алфавит көлемі* деп аталады. Дискретті хабардың кең тараған түрі – деректер болып табылады.

Үздіксіз хабарлар элементтерге бөлінбейді. Олар уақыт функцияларымен сипатталады. Оның мысалы ретінде сөйлеу, телевизиялық бейнелер.

Байланыс каналдармен хабарларды жіберу үшін оған белгілі бір сигналды сәйкестендіру керек. Ақпараттық жүйелерде *сигналдар* деп хабарды бейнелейтін (тасымалдайтын) физикалық процесті түсінеміз. Хабарды берілген байланыс каналы бойынша жіберуге ыңғайлы сигналға түрлендіруді *кең мағынадағы кодтау* деп атайды. Қабылданған сигнал бойынша хабарды қалпына келтіру операциясын *декодтау* деп атайды.

Берілген белгілерді белгілер саны аз басқа алфавитке – символдарға түрлендіруді *тар мағынада кодтау* деп атайды. Осы амалды орындайтын құрылғыны *Кодер (К)* деп атайды. Символдар алфавиты белгілер алфавитінен аз болғандықтан, әрбір белгіге қандай да бір символдың тізбегі сәйкес келеді, оны *кодтық комбинация* деп атайды.

Символдарға берілген алфавит символдарын сәйкестендіру үшін «декодтау» термині қолданылады. Оны жүзеге асыратын құрылғыны *декодер (ДК)* деп атайды.

Жіберуші құрылғы үздіксіз хабарлар немесе белгілерді нақты байланыс тізбегі бойынша жүруге ыңғайлы сигналдарға түрлендіреді. Мұнда таңдалған тасымалдаушының бір немесе бірнеше параметрі жіберілетін ақпаратқа сәйкес өзгереді. Мұндай процесс *модуляция* деп аталады. Оны *модулятор (М)* жүзеге асырады. Керісінше сигналдарды символға түрлендіруді *демодулятор (ДМ)* жүзеге асырады.

Байланыс тізбегі деп – жіберуші құрылғыдан қабылдаушыға сигналдардың түсуін қамтамасыз ететін физикалық орта (ауа, металл, магниттік таспа және т.б.). Байланыс тізбегінің шығуында сигналдар кедергілердің әсер етуі, бұрмалануы, сөнуі салдарынан берілгендерден өзгешелеу болуы мүмкін. *Кедергілер* деп – берілген сигналдардың алынған сигналдардан кездейсоқ ауытқуларын тудыратын кез-келген кедергі жасайтын ауытқуларды айтамыз. Олар сыртқы кедергілер (атмосфералық кедергілер, өнеркәсіп кедергілері) және ішкі кедергілер көзі ретінде байланыс аппаратурасы болып табылады.

Сигналдар және кедергілерден *қабылдаушы құрылғы* сигнал шығарады және декодер арқылы хабарды қалпына келтіреді, ол жіберілгеннен өзгеше болуы мүмкін.

Қабылданған хабар байланыс жүйесінің шығысында алушыға келіп түседі.

Хабарды жіберуге арналған құралдар жиынтығын *байланыс каналы* деп атайды.

Бір пунктте шоғырланған ақпараттар көзінің тобынан басқа пунктте орналасқан алушылар тобына ақпаратты жіберу үшін көбінесе бір байланыс тізбегін қолданған жөн, бұл тізбекте қажетті каналдар санын ұйымдастыру керек. Мұндай жүйелер *көпканалды* деп атайды.

2.4 Хабарды сигналдар көмегімен жіберу

Біздің ұғымымыздағы «сигнал» ақпараттық жүйеде хабарды жіберу үшін құрылған сигнал деп түсінеміз. Сигналдың материалдық негізі ақпарат (хабар) тасымалдаушы деп аталатын физикалық объект немесе процесс. Тасымалдаушы модуляция процесінде сигнал болады. Берілетін хабарға сәйкес уақытта өзгертін тасымалдаушы параметрін *информативті* деп атайды.

Ақпарат тасымалдаушы ретінде табиғаты әр түрлі тербелістер қолданылады. Техникалық ақпараттық жүйелерде кең таралғаны электр кернеуі немесе тогы түріндегі тасымалдаушылар. Тербелістер детерминделген және кездейсоқ болып бөлінеді.

Детерминделген деп кез-келген уақыт аралығында дәл анықталған тербелістерді айтамыз.

Кездейсоқ тербелістердегі кейбір параметрлердің мәндерін алдын-ала айту мүмкін емес. Олар бізді қызықтыратын ақпараттарды тасымалдайтын сигнал немесе бізді қызықтыратын сигналдарды бақылауға кедергі жасайтын кедергілер түрінде қарастырылады.

Байланыс каналдарының, сигналдарының және кедергілердің жалпы қасиеттерін оқып үйренгенде біз оның нақты физикалық табиғатын, мазмұнын және қызметін оның модельдерімен ауыстырамыз. Модель есепті шешудегі негізгі факторларды бейнелейтін объектіні, процесті немесе құбылысты сипаттайтын тәсіл.

Ақпараттық жүйелердің жұмыс істеуінің тиімділігін арттыру есептері – ақпарат көзі және байланыс каналымен сипатталатын негізгі параметрлерінің сандық қатынастарын анықтаумен байланысты. Сондықтан зерттеу кезінде математикалық модельді қолданады. Математикалық модель – бізді қызықтыратын көрсеткіштерді анықтау тәсілдеріне байланысты әр түрлі әдістермен жүзеге асырылады.

2.5 Хабарды дискретизациялау

Ақпараттық параметрлердің құрылымына байланысты сигналдар дискретті, үздіксіз және дискретті-үздіксіз болып бөлінеді.

Дискретті хабарлар – уақыт аралығында біртіндеп ақпарат көзі құратын шектеулі элементтер жиынынан тұрады. Элементтер (символдар) жиынтығы ақпарат көзінің алфавитін құрайды.

Үздіксіз хабарлар уақыт аралығында өзгертін қандай да бір физикалық шамамен беріледі. Белгілі бір санаулы уақыт аралығында хабарлардың шектелген жиынын алу *дискретизациялау* (уақыт бойынша), *кванттау* (деңгей бойынша) арқылы жүзеге асырылады.

Көп жағдайда қандай да бір физикалық процестің жүруі туралы ақпарат уақыт бойынша үздіксіз өзгертін сигналдар сәйкес датчиктер арқылы өндіріледі. Аналогтық сигналдардың цифрлы сигналға өтудегі артықшылығы – ақпаратты жеткізу, сақтау немесе өндеуде. Түрлендіруді арнайы құрылғы – үздіксіз сигналдарды түрлендіргіш жүзеге асырады және уақыт бойынша дискретизациялау және деңгей бойынша кванттау арқылы орындалады.

2.6 Ақпарат мөлшерінің бірліктері

«Ақпарат мөлшері» ұғымын анықтау өте қиын. Бұл мәселені шешу екі негізгі тұрғыда қарастырылды. XX ғасырдың 40-жылдары кибернетиканың негізін салушы Клод Шеннон бұл мәселені ықтималдық тұрғыдан қарастырды.

Ықтималдық тұрғы

Мысал ретінде N жағы бар дұрыс ойын сүйегін лақтырумен байланысты тәжірибені қарастырайық (көбінесе N=6). Тәжірибе нәтижесі төмендегідей болуы мүмкін: яғни 1, 2, ..., N белгілерінің бірі бар жақпен түсу.

Анықталмағандықты өлшейтін H – энтропия сандық шамасын енгіземіз. N және H қандай да бір функциялық байланыста: $H=f(N)$ (1.1)

Ал f - өспелі N=1..6 үшін анықталған функция.

Сүйекті лақтыру процедурасы нақтырақ:

1) сүйекті лақтыруға дайындық;

тәжірибе нәтижесі белгісіз, яғни анықталмағандық - H1 шамасын енгіземіз.

2) сүйек лақтырылды;

тәжірибе нәтижесі туралы ақпарат алынды, оның мөлшерін I деп белгілейміз

3) тәжірибе іске асырылғаннан кейінгі анықталмағандық - H2.

Осы тәжірибені жүзеге асыру кезіндегі ақпарат мөлшері ретінде тәжірибеге “дейін” және “кейінгі” белгілердің айырымын аламыз:

$$I = H_1 - H_2 \quad (1.2)$$

Егер нақты нәтиже алынса, яғни $H_2 = 0$ болса, онда $I = H_1$, яғни бастапқы энтропияға тең.

Басқаша айтқанда, тәжірибедегі анықталмағандық - сол тәжірибенің нәтижесі туралы ақпаратпен беттеседі.

Енді (1.1)-дегі f функциясының түрін анықтау қажет. Егер N жақ пен лақтыру саны $-M$ десек, онда жалпы нәтиже саны: $X = N^M$ (1.3)

Мысалы, 6 жақты сүйекті екі рет лақтырғанда $X = 6^2 = 36$, яғни әрбір $X = (x_1, x_2)$ жұбы болып табылады: x_1 – бірінші лақтыру, x_2 – екінші лақтыру.

(1.3)-ке логарифм т.б. әрекеттер қолданып,

$$H = \log_2 N \quad (1.4) \quad (\text{Хартли})$$

және $H = -\sum P_i \log_2(1/P_i)$ (1.5) (Шеннон) формулаларын аламыз.

Мысалдар:

1) Орыс тілінде жазылған сөздегі әр символдың пайда болуына байланысты ақпарат мөлшері: орыс алфавитінде 33 әріп және ашық орын белгісін есептегенде: (1.4) бойынша, $H = \log_2 34 \approx 5,09$ бит.

2) Латын, француз, неміс т.б. тілдерде 26 әріп және ашық орын белгісін есептегенде: $H = \log_2 27 \approx 4,76$ бит.

3) 0, 1 алфавитінде $H = \log_2 2 = 1$ бит.

Көлемдік тұрғы

Екілік санау жүйесінде 0, 1 таңбалары бит (binary digit – екілік цифр) деп аталады. Компьютер құрастырушылар осы екілік жүйені қолданады, өйткені техникалық құрылғыларда екі әр түрлі қарама-қарсы күйді (қалыпты) оңай жүзеге асыруға болады: мысалы, қандай да бір физикалық элементтің екі бағытта магниттелуі немесе прибордың электр тогын өткізу - өткізбеуі, конденсатордың зарядталған – зарядталмағаны т.б.

Компьютерде бит – ең аз мүмкін ақпарат бірлігі. Компьютер жадысында немесе сыртқы жадыда екілік таңбамен жазылған ақпарат көлемі осы жазуға қажет екілік символдардың санымен саналады. Үқтималдық тұрғымен салыстырғанда мұнда биттердің бүтін емес саны болмайды.

Ақпараттың көлемі деп – хабардың ұзындығын, яғни хабарды жазу үшін пайдаланылған символдар санын айтады. Есептеуіш техникада кез келген хабардың жазылуы екілік алфавитте жасалатыны мәлім. Осылай қарастырғанда ақпараттың ең кіші өлшем бірлігі екілік разрядқа немесе 1 битке тең болады.

Одан қолайырақ өлшем бірлігі - сегіз екілік разрядтың жиынтығы – 1 байт, ол символдарды кодтауда жиі қолданылады.

Бір байт 8 битке тең.

1 байт = 8 бит;

1Кбайт (килобайт) = 1024 байт \approx 1000 байт;

1Мбайт(мегабайт) = 1024Кбайт \approx 1(миллион байт)

1Гбайт(гигабайт) = 1024Мбайт \approx 1(миллиард байт).

1 Тбайт (терабайт) = 1024 Гбайт.

Көлемі 100 Мегабайт жадқа төмендегілерді орналастыруға болады:

- 50000 беттік мәтінді:
- 150 түсті слайдты:
- 1,5 сағаттық сөздік аудио жазбаны:
- 10 минуттық стерео музыкалық үзіндіні:
- 15 секундтық фильмді:
- Windows98, 2000, Word 97 бағдарламаларын.

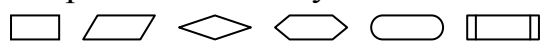
2.7 Кодтау теориясы. Абстрактілі алфавит. Кодтау. Декодтау

Абстрактілі алфавит

Ақпарат хабарлама түрінде беріледі. Дискретті ақпарат кейбір таңбалардың ақырлы сан жиынтығы арқылы жазылады. Таңбалар - әріптер деп аталады. Бұл жерде әріптер – мәлімет алмасу үшін кейбір ұйғарым бойынша тағайындалған кез-келген таңба. Реттелген таңбалар жиыны – алфавит деп аталады.

Алфавит түрлері:

- Орыс әріптерінің алфавиті А...Я;
- IBM дербес компьютерінің клавиатуралық символдарының алфавиті;
- Дұрыс алты жақты ойын сүйегінің таңбалар алфавиті;
- Морзе алфавиті;
- Араб цифрларының алфавиті: 0...9;
- Он алтылық символдар алфавиті: 0...9 А...F;
- Екілік цифр алфавиті: 0, 1;
- “Нүкте”, “тире” екілік алфавиті: .-;
- Римдік санау жүйесі алфавиті: I, V, X, L, C, D, M;
- Алгоритмді бейнелеудің блок-схема тілінің алфавиті:



Кодтау және кері кодтау

Ақпарат берілген кезде байланыс каналында бір алфавиттің символдарынан құралған хабарлама екінші алфавит символдарынан құралған хабарламаға түрленуі мүмкін. Осындай түрлендіру кезіндегі алфавит әріптерінің бірімәнді сәйкестігін сипаттайтын ереже – *код* деп аталады. Ал түрлендіру процесін *кодтау* деп атайды. Мұндай түрлендіру ақпарат көзінен байланыс каналына хабарлама түскен кезде (*кодтау*) және қабылдаушы хабарламаны қабылдаған кезде (*кері кодтау*) жүзеге асуы мүмкін.

Ақпаратты ЭЕМ-де өрнектеу үшін көбінесе *екілік кодтау* қолданылады. Мұнда ақпарат 0, 1 әріптерінен тұратын екілік алфавит құралдары арқылы өрнектеледі. Екілік алфавиттің құрамына кіретін символдар саны өте аз (екі) болғандықтан, әріптердің көп санынан тұратын алфавитті екілік кодтау кезінде, әрбір әріпке бірнеше екілік таңба немесе екілік сөз сәйкестендіріледі. Осындай тізбектер кодтық тіркестер деп аталады. Кодталатын алфавиттің барлық әріптерінің екілік жазылуына сәйкес кодтық тіркестердің толық жиынтығы – код деп аталады.

Тең өлшемді және тең өлшемді емес кодтар бар. Тең өлшемді екілік кодтардың кодтық тіркестері екілік таңбалардың бірдей санынан, ал тең өлшемді емес кодтар әр түрлі санынан тұрады.

Тең өлшемді емес екілік кодқа Морзе алфавиті мысал бола алады. Мұнда әрбір әріп пен цифр үшін қысқа және ұзын сигналдардың екілік тізбегі анықталған. Морзе әліппесінде мысалы, Е әрпіне бір қысқа сигнал (нүкте), ал Ш әрпіне – төрт ұзын сигнал (төрт тире) сәйкестендірілген. Тең өлшемді емес кодтау хабарламаны беру жылдамдығын арттыруға жағдай жасайды, өйткені берілетін текстідегі көп кездесетін символдарға (мысалы, Е әрпі) неғұрлым қысқа тіркес тағайындалады.

Есептеу техникасында, негізінен, кодтық тіркестері сегіз екілік таңбалар тізбегінен құралатын тең өлшемді кодтар қолданылады. Мысалы, ЭЕМ-де ақпарат енгізу мен шығаруда қолданылатын ақпарат алмасу коды – КОИ-8; ақпараттың ішкі құрылымын көрсетуге қолданылатын ақпарат алмасудың екілік коды – ДКОИ.

Трисиме коды (латын алфавитінің таңбаларына 3 таңба: 1, 2, 3 тіркесі сәйкестендіріледі). Трисиме коды – тең өлшемді код (яғни барлық кодтық тіркестер екілік таңбалардың бірдей санынан тұрады). Тең өлшемді емес код – Морзе алфавиті.

Кодтық тіркестерді құрайтын символдар саны – код ұзындығы деп аталады. Екілік кодқа қатысты код ұзындығы терминімен қатар, код разрядтылығы термині қолданылады.

Байттық кодтаудың халықаралық жүйесі

N екілік таңба топтарынан тұратын тең өлшемді код арқылы қанша әр түрлі кодтық тіркестер құруға болатынын $K=2^N$ формуласы арқылы табуға болатыны белгілі.

Бірнеше алфавитті кодтау үшін қажет N-нің ең кіші мәні – 8, яғни $K=2^8=256$. Осы екілік символдардың 256 тіркесі арқылы кез келген алфавитті кодтауға болады. Сегіз екілік цифр байтты құрайды, сондықтан жүйе – байттық кодтау жүйесі деп аталады. Көбінесе мына екі жүйе қолданылады:

- EBCDIC (Extended Binary Coded Decimal Interchange Code)
- ASCII (American Standard Code for Information Interchange)

Біріншісі үлкен машиналарда, ал екіншісі мини, микроЭЕМ, дербес компьютерлерде қолданылады. Бұл жүйелер 1963 жылдары құрылған.

Қазіргі кезде ASCII кестесінің жетілдірілген балама кодировкасы қолданылады. Мұнда дербес компьютер алфавитінің таңбаларына он алтылық сандар төмендегі ереже бойынша сәйкес қойылған: біріншісі – баған номері, екіншісі – жол номері:

‘А’ - коды $41_{16}=4*16^1+1*16^0=64+1=65_{10}$

‘и’ - коды $A8_{16}=10*16^1+8*16^0=160+8=168_{10}$

Бүтін оң сандардың тура коды қосымша кодпен бірдей:

1) Сан екілік жүйеге аударылады

2) Санның қай типке жататынына сәйкес, оның жазылуына сол жақтан қажет 0-дер тіркеледі .

Мыс: $37_{10}=100101_2$

- егер integer деп хабарланса, онда тура коды: 2 байт, яғни $2 \cdot 8$ бит = 16 бит
- егер longint деп хабарланса, онда тура коды: 4 байт, яғни $4 \cdot 8$ бит = 32 бит
- қысқа жазу үшін он алтылық жүйе қолданылады.

Бүтін теріс санның қосымша коды:

1. сан тобының тура кодын жазу
2. инверттеу (бірді –нөлмен, нөлді – бірмен алмастыру)
3. инвертті кодқа 1-ді қосу

Дәріс №3. Ақпараттық жүйелер және технологиялар

Дәріс жоспары:

- 3.1 Ақпараттық жүйе ұғымы
- 3.2 Ақпараттық жүйелердің даму кезеңдері
- 3.3 Ақпараттық жүйедегі үдерістер
- 3.4 Ақпараттық жүйенің құрылымы
- 3.5 Ақпараттық жүйелердің жіктелуі
- 3.6 Ақпараттық жүйелерді енгізуден не күтуге болады?
- 3.7 Ақпараттық технология
- 3.8 Ақпараттық технология дамуының кезеңдері

3.1 Ақпараттық жүйе ұғымы

Жүйе дегеніміз бір мезгілде әрі біртұтас, әрі әр текті элементтердің жиынтығы, алға қойған мақсаттарға қол жеткізуге орай біріктірілген ұғым ретінде қарастырылатын кез келген объект болып табылады. Жүйелер өзара құрамы жағынан да, басты мақсаттары жағынан да едәуір ерекшеленеді.

Әр түрлі элементтерден тұратын және әр түрлі мақсаттарды жүзеге асыруға бағытталған бірнеше жүйелерді келтіреміз.

Жүйе	Жүйе элементтері	Жүйенің басты мақсаты
Фирма	Адамдар, қондырғылар, материалдар, ғимараттар және т.б	Тауарлар өндірісі
Компьютер	Электронды және электр-механикалық элементтер, байланыс желілері және т.б	Мәліметтерді өңдеу
Телекоммуникациялық жүйе	Компьютерлер, модемдер, кабельдер, желілік бағдарламалық қамтамасыз ету және т.б	Ақпараттық бағдарлама
Ақпараттық жүйе	Компьютерлер, компьютерлік желілер, адамдар, ақпараттық және бағдармалық қамтамасыз ету	Кәсіби ақпаратты өндіру

Ақпараттық жүйе - алға қойылған мақсатқа жетуге орай ақпаратты сақтау, өңдеу және беру үшін пайдаланылатын құралдардың, әдістер мен жұмыскерлердің өзара байланысты жиынтығы.

Арнайы бағдарламалармен жарактандырылған компьютерлер ақпараттық жүйелердің техникалық базасы және құралы болып табылады. Ақпараттық жүйе компьютерлермен және телекоммуникациялармен өзара әрекет ететін қызметкерсіз ақылға қонбайды.

3.2 Ақпараттық жүйелердің даму кезеңдері

Алғашқы ақпараттық жүйелер 50-ші жылдары пайда болды. Бұл жылдары ол шоттар мен еңбекақы есептерін өңдеу үшін арналған еді, оған электромеханикалық бухгалтерлік есептеу машиналары қолданылды. Бұл қағаз құжаттарын дайындауға кететін шығындар мен уақытты үнемдеуге тиісінше көмектесті.

60-шы жылдары ақпарат жүйелеріне деген көзқарас жаңаша сипат алды. Олардан алынған ақпарат көптеген параметрлер бойынша мерзімді есептемеде қолданыла бастады. Бұл үшін ұйымдарға бұрынғыдай тек шоттарды өңдеу мен еңбекақыны есептеп қана қоймайтын, сонымен бірге көптеген функцияларды атқаруға қабілетті, ауқымды мақсаттағы компьютерлік қондырғы талап етілді.

70-ші жылдары және 80-ші жылдардың бас кезінде ақпараттық жүйе қабылданған шешімдер үдерісін қолдайтын әрі жеделдететін басқару бақылауының құралы ретінде кеңінен қолданыла бастады.

80-ші жылдардың аяғына қарай ақпараттық жүйені пайдаланудың тұжырымдамасы қайтадан өзгеріске түсті. Олар енді ақпараттың стратегиялық қайнар көзіне айналады және кез келген бейінді (профильді) ұйымдардың барлық деңгейлерінде қолданылады. Осы кезеңнің ақпараттық жүйелері уақытында керекті ақпаратты бере отырып, ұйымға өзінің қызметінде табысқа жетуді ұйымдастыруға, жаңа тауарлар мен қызмет көрсетулерді жүзеге асыруға, сатудың жаңа рыноктарын табуға, өзіне лайықты серіктес табуына, өнімді төменгі бағада шығаруды ұйымдастыруға және т.б. көмектесті

3.3 Ақпараттық жүйедегі үдерістер

Кез-келген мақсаттағы ақпараттық жүйе жұмысын қамтамасыз ету үдерістерін шартты түрде мынадай блоктардан тұратын сұлба түрінде (5-сурет) ұғынуға болады:

- сыртқы немесе ішкі қайнар көздерден ақпаратты енгізу;
- түсетін ақпаратты өңдеу және оны оңтайлы түрде беру;
- тұтынушыларға ұсыну немесе басқа жүйеге беру үшін ақпаратты шығару;
- кері байланыс – бұл түсетін ақпаратты түзету үшін аталған ұйым адамдарымен қайта өңделген ақпарат.

Ақпараттық жүйе төмендегідей қасиеттерімен айқындалады:

- кез-келген ақпараттық жүйе талдауға жатуы мүмкін, ол жүйелерді құрудың жалпы ұстанымдары негізінде құрылған және басқарылады;
- ақпараттық жүйе серпінді және қарқынды дамуда;
- ақпараттық жүйені құру кезінде жүйелілік тәсілді қолдану қажет;

- ақпараттық жүйенің шығатын өнімі ақпарат болып табылады, оның негізінде шешімдер қабылданады;
- ақпараттық жүйені ақпаратты өңдеудің адам-компьютерлік жүйесі ретінде қабылдаған жөн.



5-сурет. Ақпараттық жүйедегі үдерістер

Қазіргі кезде ақпарат жүйесі туралы оны компьютерлік техниканың көмегімен жүзеге асырылатын жүйе ретінде санау туралы пікір қалыптасты. Алайда жалпы жағдайда ақпараттық жүйені компьютерлік емес нұсқада да түсінуге болады.

Компьютерлік ақпарат жүйесінің мүмкіндігін айқындау кезінде қабылданған шешімдерді қолдау үшін мыналарды ескеру керек:

- шешілетін басқару міндеттерінің құрылымдылығы;
- шешім қабылдануға тиісті фирманың басқару иерархиясының деңгейі;
- бизнестің сол немесе басқа жұмыс істейтін саласындағы шешілетін міндеттің оған жатуы;
- пайдаланылған ақпараттық технологияның түрі.

Компьютерлік ақпараттық жүйедегі жұмыс технологиясы компьютерлік емес сала мамандары түсіну үшін қолайлы болуы тиіс және кәсіби қызметтің және оны басқарудың үдерістерін бақылау үшін табысты пайдаланылуы мүмкін.

3.4 Ақпараттық жүйенің құрылымы

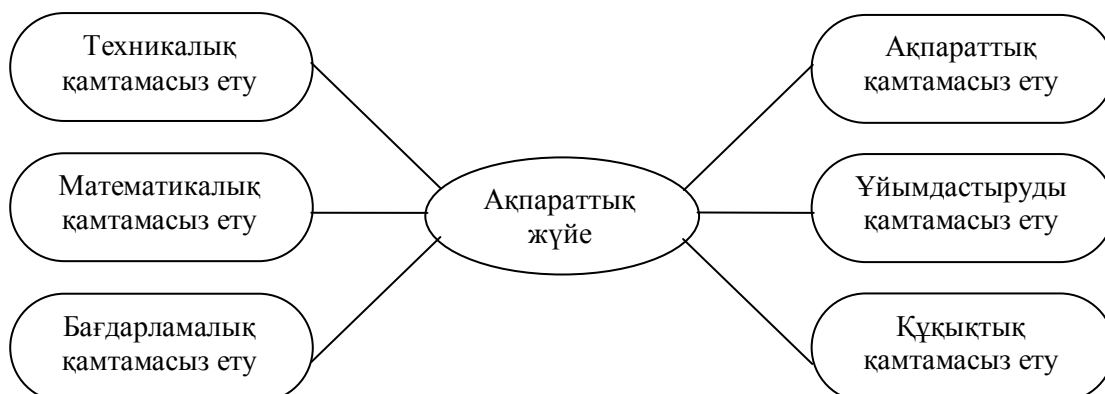
Ақпараттық жүйенің құрылымы *қосалқы жүйе* деп аталатын, оның жекелеген бөліктерінің жиынтығын құрайды.

Қосалқы жүйе – қандай да бір белгісі бойынша бөлініп алынған жүйенің бөлігі.

Ақпараттық жүйенің жалпы құрылымын қолданылу аясына тәуелсіз қосалқы жүйелердің жиынтығы деп қарастыруға болады. Кез-келген ақпараттық жүйенің құрылымы қамтамасыз етуші қосалқы жүйелердің жиынтығы түрінде көрсетілуі мүмкін (6-сурет).

Қамтамасыз етуші қосалқы жүйелер ақпараттық, техникалық, математикалық, бағдарламалық, ұйымдастырушы және құқықтық болып бөлінеді.

Ақпараттық қамтамасыз ету деп - ақпаратты кодтау және жіктеудің, бірыңғай құжаттаманың бірыңғай жүйелерінің, ұйымда айналып жүретін ақпараттық ағымдар сұлбаларын, сондай-ақ мәліметтер қорын құрудың әдіснамасының жиынын айтамыз.



6-сурет. Қосалқы жүйелерді қамтамасыз етуші жиынтық ретіндегі ақпараттық жүйенің құрылымы

Техникалық қамтамасыз ету – ақпараттық жүйелердің жұмыс істеуі үшін қажетті техникалық құралдардың, сондай-ақ осы құралдар мен процестерге сәйкес құжаттамалар кешені.

Математикалық және бағдарламалық қамтамасыз ету - ақпараттық жүйелердің мақсаттары мен мәселелерін іске асыру үшін қажетті алгоритмдер мен бағдарламалардың, математикалық әдістер, модельдер жиынтығын, сондай-ақ техникалық құралдар кешенінің дұрыс қызмет етуін айтамыз.

3.5 Ақпараттық жүйелердің жіктелуі

Ақпараттық жүйелер ақпараттарды жинақтау және оны әр түрлі мақсатқа тиімді пайдалану үшін қызмет етеді. Олардың қолданылуына байланысты:

- 1) ақпараттық іздестіру жүйелері
- 2) мәліметтерді өңдеу жүйелері болып жіктеледі.

Ақпараттық іздестіру жүйелері сақталған мәліметтерден белгілі бір іздестіру шарттарын қанағаттандыратын ішкі жиындарын алуға бағытталған. Мысалы анықтамалық қызмет қала тұрғындары туралы мәлімет алу үшін.

Мәліметтерді өңдеу жүйелері ақпараттарды жаңартуға алып келеді, ақпараттарды шығару мүлде болмауы мүмкін. Мысалы, қаланың жинақ банкілерінің жұмысы.

3.6 Ақпараттық жүйелерді енгізуден не күтуге болады?

Ақпараттық жүйелерді енгізу:

- математикалық әдістер мен интеллектуалды жүйелерді енгізу есебінен басқару міндеттерін шешудің неғұрлым ұтымды нұсқаларын алуға және т.б.;
- жұмысты автоматтандыру есебінен қызметкерлерді ескі көзқарастан арылтуға;

- қағазда жазылатындарды магниттік дискілерге немесе таспаға ауыстыруға, бұл компьютерде ақпаратты өңдеуді неғұрлым ұтымды ұйымдастыруға және қағаздарға жазылатын құжаттардың көлемін азайтуға жағдай жасайды;

- фирмадағы құжат айналымының жүйесі мен ақпарат ағындарының құрылымын жетілдіруге;

- өнімдерді өндіруге және қызмет көрсетулерге шығынды азайтуға;

- тұтынушыларға жоғары дәрежеде қызмет ұсынуға;

- жаңа нарықтың мүмкіндіктерді іздеп табуға;

- сатып алушылар мен жеткізушілерге әр түрлі жеңілдіктер мен қызмет көрсетулерді ұсыну есебінен оларды фирмаға тартуға жәрдемдеседі.

3.7 Ақпараттық технология

Ақпараттық технология ескішілік әдістен өндірістік әдіске дейін өтуді қамтамасыз етеді және адам әрекетінің түрлі аймағында ақпаратпен жұмыс істеу құралдарын тұрақты және тиімді қолдануды қамтамасыз ету.

Ақпараттық технология деп басқару есептерін шешу үшін қажетті әдістер және дамыған бағдарламалық жасау, есептеу техникасының құралдарының жиынтығын, сондай-ақ солардың көмегімен ақпараттарды қолданушыға жеткізу әдістерін айтамыз.

Өнеркәсіптегі ақпараттық технологиялардың мақсаты есептеу техникасының қазіргі заманғы құралдарының негізінде өндірісті автоматтандыру арқылы тиімділігін арттыру.

Басқаруды ұйымдастырудағы ақпараттық технологиялардың мақсаты - шешім қабылдау.

Ақпараттық технологияның үш деңгейін қарастыруға болады:

Бірінші деңгей – теориялық; негізгі тапсырма – параметрлі және критериялды сәйкес ақпараттық үрдістердің өзара әрекеттесуші кешенді моделін құру;

Екінші деңгей – зерттеулік; негізгі тапсырма – тиімді нақты ақпараттық технологияны автоматты жасауға мүмкіндік беретін әдіс жасау;

Үшінші деңгей – қолданбалы, яғни, екі бөлімді: құрал-сайманды және пәндік аймақты мақсатқа сай бөлу.

Құрал-сайманды бөлім (құрылғылар, станоктар, құрал жабдықтар) ақпараттық технологияны жүргізу жолын және құралын анықтайды, оны әдістемелік, ақпараттық, математикалық, алгоритмдік, техникалық және программалық деп бөлуге болады.

Пәндік бөлім нақты пәндік аймақтық құрамына байланысты және ол арнайы дайындалған ақпараттық технологияда көрініс табады, мысалы, ұйымдастырылған басқару, технологиялық үрдіспен басқару, автоматтандырылған жобалау, оқыту және т.б.

Нақты ақпараттық технология – базалық технологиялық операцияларды, арнайы технологияларды және жүргізу құралдарын компиляциялау және синтездеу нәтижесінде анықталады.

Технологиялық процесс – ақпарат жағдайының өзгеруі бойынша әрекеттерге (физикалық, механикалық және т.б.) ие, ақпараттық процесс бөлімі.

Ақпараттық технология кез-келген ақпараттық технологиялар үшін әркелкі базалық, мінездемелік бөлінулерді талап ететін ақпараттық процестердің жүргізілуіне негізделеді.

Базалық технологиялық процесс стандартты модельдер мен құрал-саймандық құралдарды қолдануға негізделген және ол ақпараттық технологияның құрамдық бөлігі негізінде қолданылуы мүмкін. Оның қатарына жататындар: шығару операциясы, тасымалдау, сақтау, өңдеу және ақпаратты ұсыну.

Олардың ішінде негізгі технологиялық процестерге жататыны:

- ақпаратты шығару;
- ақпаратты тасымалдау;
- ақпаратты өңдеу;
- ақпаратты сақтау;
- ақпаратты ұсыну және қолдану.

Ақпаратты *шығару процесі* пәндік аймақтың нақты ұсынылуынан формальді және мәліметтер түрінде, яғни осы ұсынуларды бейнелейтін сипатталуына өтуімен байланысты.

Тасымалдау процесінде ақпаратпен қашықтықта жылдам алмасу үшін және оған түрлендірудің әр түрлі әдістерін қолдану арқылы жылдам қол жеткізуді ұйымдастыру үшін ақпаратты тасымалдаулар жүзеге асырылады.

Ақпаратты *өңдеу процесі* дегеніміз – кейбір алгоритмдерді орындау арқылы бір ақпараттық объектіні басқа ақпараттық объектіден алу; бұл процесс ақпаратпен орындалатын негізгі операциялардың бірі болып табылады және ақпарат көлемі мен көптүрлілігін көбейтудің негізгі құралы болып табылады.

Сақтау процесі – ақпаратты көкейкестілікпен, бүтіндікпен, қауіпсіздікпен, қол жеткізумен қамтамасыз ете отырып, мәліметті жинақтау және ұзақ уақытқа сақтау қажеттілігімен байланысты.

Ақпаратты *ұсыну және қолдану процесі* қолданушының ақпаратқа қол жеткізуі үшін ыңғайлы форма тапсырмасын шешуге бағытталған.

3.8 Ақпараттық технология дамуының кезеңдері

Негізгі ақпараттық технология негізгі технологиялық операциялар негізінде тұрғызылады, бірақ бұдан басқа арнайы модельдер мен құрал-саймандар қатарын да құрамына алады. Технологияның бұл түрі тапсырманың нақты класын шешуге арналған және нақты технологияда жеке компонент түрінде қолданылады. Оның ішінде мыналарды бөлуге болады:

- Мультимедия-технологиялар;
- Геоақпаратты технологиялар;
- Ақпаратты қорғау технологиялар;
- CASE-технологиялар;
- Телекоммуникациялық технологиялар;
- Жасанды интеллект технологиялары.

Нақты пәндік аймақ құрамы арнайыландырылған ақпараттық технологиядан көрініс табады, мысалы, ұйымдастырылған басқару, технологиялық процестерді басқару, автоматтандырылған басқару, оқыту және т.б.

CASE-технология (Computer Aided Software Engineering – программалық қамтаманы компьютерлік автоматты жобалау) ақпараттық технологияны автоматты жобалауды жүзеге асырушы өз алдына «технологиялық жабдық» болып табылады.

Бірінші дәуір (1900-1955 ж.ж.) перфокарта технологиясымен байланысты, яғни мәліметтерді жазу екілік құрылым түрінде ұсынылған. 1915-1960 ж.ж. IBM компаниясының дамуы мәліметтерді картаға жазудың, іріктеудің және кесте құрудың электромеханикалық құрылғы жасаумен байланысты. Құрылғының ыңғайсыздығы, кең көлемді перфокартаны сақтау қажеттілігі жаңа технологияның, яғни, электромеханикалық компьютердің пайда болуына әкелді.

Екінші дәуір (1955-1980 ж.ж., жазбаны өңдеудің программаланған құрылғысы) әрқайсысы ондаған мың перфокарта ақпаратын сақтай алатын, магнитті лента технологиясының пайда болуымен байланысты. Ақпаратты өңдеу үшін сақталған программасы бар электронды компьютерлер жасалды, ол жүздеген жазбаларды бір секунд ішінде өңдей алатын болды. Бұл жаңа технологияның басты жетістігі, бағдарламалық қамтама болды, яғни соның көмегімен компьютерді программалау және қолдану жеңіл болды.

Осы заманның бағдарламалық қамтамасы жазбаларды файл негізінде өңдеудің моделін қалады. Мұндай бағдарламалар бірнеше кіріс файлдарын ретімен оқып, шығыста жаңа файлдарды өңдейді. Осы жазбаға бағытталған тізбектелген тапсырманы анықтауды оңайлату үшін COBOL және басқа да бірнеше бағдарламалау тәсілдері жасалды. Операциялық жүйе осы жазбаларды сақтау үшін, тапсырманы орындаудағы басқару тәсілін және жұмыс ағынын басқару үшін тапсырманы жобалаушы файл жағдайын қамтамасыз етті.

Үшінші дәуір (1965-1980 ж.ж., жылдам мәліметтер қоры) интернативті режимде ақпаратқа жылдам қол жеткізумен байланысты, яғни бұл жылдам транзакциялы мәліметтер қоры жүйесін қолдануға негізделген.

Интернативті компьютерлік терминал компьютеріне қосылу үшін техникалық құралдар телеграф аппаратынан қарапайым алфавитті-цифрлы дисплейге дейінгі жолдан өтті және де соңында әмбебап компьютер технологиясына негізделген қазіргі заманғы интеллектуалды терминалға жетті.

Жедел мәліметтер қоры мәліметтік дискілерде немесе барабандарда сақталды, яғни кез-келген мәлімет элементіне 1 секунд ішінде қол жеткізу қамтамасыз етілді. Мәліметті басқарудың бұл құрылғысы және программалық қамтамасы бағдарламаға бірнеше жазбаларды оқуға, өзгертуге және жедел қолданушыға жаңа мәндер қайтаруға мүмкіндігі бар. Алғаш жүйе мәліметті іздеудің қарапайым жолын қолдаған: не жазба нөмірі бойынша тікелей іздеу, не кілт бойынша ассоциативті іздеу.

Жазбаны қарапайым индексті-тізбектеп ұйымдастыру, жиынға бағытталған анағұрлым мықты модельді дамытты. Мәліметтер моделі

иерархиялық және желілік модельден реляциялық модельге дейін даму жолынан өтті.

Бұл ескі мәліметтер қорында мәліметтер схемасының үш түрі қолданылды:

- логикалық, мәліметтер қоры жазбасының геобальды логикалық жазбасын және жазбалар арасындағы байланысты анықтайды;

- физикалық, мәліметтер қоры жазбасының құрылғы жадысында және файлда физикалық орналасуын сипаттайды, сонымен қатар логикалық байланысты қолдау үшін қажетті индекстерді сипаттайды.

- әрбір қосымшаға ұсынылатын подсхема, бұл тек бағдарлама қолданатын логикалық схеманың бөлігін ғана көрсетеді.

Логикалық және физикалық схема және подсхема механизмі мәліметтердің тәуелсіздігін қамтамасыз етті және қазірдің өзінде сол дәуірде жазылған көптеген бағдарламалар, логикалық және физикалық схемалар өзгергенімен, әлі де бәрі басталған сол подсхемаларды қолдану арқылы жұмыс істеп келеді.

1980 жылдары жазба жиынына ұйымдастырылған мәліметтердің желісі моделі өте танымал болды. Бірақ, навигациялы бағдарламалық интерфейс төмен деңгейлі болды, ол ақпараттық технологияның одан әрі өмір сүруіне секірмелі қызмет көрсетті.

Төртінші дәуір (реляциялық мәліметтер қоры: «клиент-сервер» архитектурасы, 1980-1995 ж.ж.) төмен деңгейлі интерфейс үшін альтернатива болып табылады. Реляциялық модель идеясы маңыз бен байланыстың біркелкі ұсынылуынан тұрады.

Мәліметтің реляциялық моделі мәліметтерді анықтау үшін, мәліметтер бойынша навигация үшін және мәліметтерді манипуляциялау үшін бірыңғай тілді игереді. Бұл бағытта жұмыс істеу SQL атты тілдің пайда болуына әкелді, бұл тіл стандарт ретінде қабылданды.

Қазір барлық мәліметтер қоры жүйесі SQL интерфейсін қамтамасыз етеді. Бұдан басқа, барлық жүйелер стандарттан тыс жеке кеңейтілулерді қолдайды.

Өнімділігі мен қолдану қарапайымдылығынан басқа реляциялық модель кейбір күтпеген жетістіктерге ие. Ол «клиент-сервер» архитектурасында, параллельді өңдеуде және графикалық қолданушы интерфейсін қолдануда өте қажетті болып табылады. «Клиент-сервер» қосымшасы екі бөлімге бөлінеді. Клиенттік бөлім енгізуді қолдануға және қолданушы үшін мәліметтерін ұсынуға жауап береді. Сервер мәліметтер қорының сақталуына, мәліметтер қорына клиенттік сұраныстың өңделуіне, клиентке жалпы жауапты қайтаруға жауап береді. Реляциялық интерфейс әсіресе «клиент-сервер» архитектурасында қолдану үшін өте ыңғайлы және бұл жоғары деңгейлі сұраныстар мен жауаптардың ауысуына әкеледі. Жоғары деңгейлі SQL интерфейсін клиент пен сервер арасындағы коммуникацияны минимизациялайды. Қазір көптеген клиент-серверлік құрал Open Database Connectivity (ODBC) хаттамасы негізінде тұрғызылады, бұл клиент үшін серверді жоғары деңгейлі сұраныстың стандартты механизмімен қамтамасыз етеді. «Клиент-сервер» архитектурасы одан әрі дамуда.

Бесінші дәуір (мультимедиялық мәліметтер қоры, 1995 ж.). Интернет клиенті мен сервері апплеттерді қолдану арқылы тұрғызылады, бұл сол немесе

басқа типтегі мәліметтерді сақтайды, өңдейді және көрсетеді. Қолданушылар бұл апплеттерді браузерге немесе серверге қояды. Жалпылама үлестірілген апплеттер дыбысты, графиканы, видеоны, электронды кестелерді, графтарды басқарады. Осы апплетті қолданған әрбір мәліметтер типі үшін класстар библиотекасы бар. Стол үстілік компьютерлер және Web-браузерлер мәліметтердің көп бөлігінің үлестірілген қайнар көзі және қабылдаушы болып табылады.

Дәріс №4. Автомат ақпараттық жүйенің негізгі элементі ретінде. Абстрактылы автоматтар

Дәріс жоспары:

4.1 ЭЕМ - бағдарламалық басқарылатын цифрлы автомат

4.2 Тьюринг машинасы

4.3 Пост машинасы

4.1 ЭЕМ - бағдарламалық басқарылатын цифрлы автомат

Абстрактылы (яғни тек адам қиялында ғана болатын) Пост және Тьюринг машиналары бағдарламалардың қасиеттері туралы әр түрлі тұжырымдарды дәлелдеу үшін ойлап шығарылды. Бұл бір-біріне тәуелсіз екі есептеу машиналарының моделін (практикада бір уақытта) 1937 жылы Алан Тьюринг ұсынды. Бұл машиналар толық детерминделген әмбебап орындаушылар болып табылады. Олар алғашқы деректерді енгізіп, бағдарлама орындалғаннан кейін нәтижені оқуға мүмкіндік береді. Тьюринг машинасына қарағанда Пост машинасы қарапайым болғанымен кең таралмаған.

4.2 Тьюринг машинасы

Бұл елестегі машина - яғни “қағаз бетіндегі” машина немесе машинаның математикалық моделі.

Тьюринг машинасы - таза абстракция және ешқашан жасалмаған. Оның пайдасы түрлі есептер шешімінің алгоритмі бар немесе жоқ екендігін дәлелдеуге болады. Машина белгілі бір алгоритмді орындайтын болғандықтан, бұл машинаға алгоритмнің қасиеттерінен талаптар қойылады. Біріншіден, машина толықтай детерминенделген (есептеулер нақты және жалпы түсінікті) болуы қажет және тапсырылған ережелер жүйесі негізінде әрекет етуі керек. Екіншіден, “бастапқы мәліметтерді” енгізуге мүмкіндік беруі қажет. Үшіншіден, берілген машинаның жұмыс жасау ережелерінің жүйесі және шешілетін есептердің класы машина жұмысы нәтижесін оқи алатындай болып келістірілуі керек.

Тьюринг тезисі кез-келген алгоритмді Тьюринг машинасына салып шешуге болатынға негізделген.

Тьюринг машинасының жұмыс істеу принципі, сипаттамасы

Бір ленталы Тьюринг машинасын кибернетикалық құрылғы ретінде қарастырады және ол келесі элементтерден тұрады:

- ұяшықтарға бөлінген шексіз лента;
- лента ұяшықтарында болатын символдарды оқи алатын басқарушы головка;
- Тьюринг машинасының жағдайын көрсететін ішкі алфавит символы бар жадының ұяшығы.
- лента бойымен головканың қозғалысын қамтамасыз ететін басқарудың механикалық құрылғысы
- тек оқуға болатын Тьюринг машинасының бұйрықтарынан тұратын функционалды схема - жады аймағы.

Әдетте, Тьюринг машинасы схемалық түрде мынадай түрде көрсетіледі:



Лентаны магниттік жол немесе баспаның қағаз лентасы деп қарастырайық, ол бірнеше ұяшыққа бөлінген. Жұмыс барысында машина бар ұяшықтарға жаңа ұяшықтарды қоса алады, сондықтан лента екі жағынан да шексіз деп айтуға болады. Лентаның әрбір ұяшығы көп жағдайдың бірінде болуы мүмкін. Бұл жағдайларды біз a_0, a_1, \dots, a_m символдарымен немесе басқа символдармен белгілейміз. Осы символдар лента ұяшықтарында жазылған. Мұндай символдардың жиынтығы машинаның сыртқы алфавиті деп, ал лентаның өзі машинаның сыртқы жадысы деп аталады. Егер ұяшық бос болса, ол жерде λ шарты символы орналасқан деп есептейміз. Машинаның жұмысы кезінде лентаның ұяшықтары өздерінің жағдайын өзгертуі де, өзгертпеуі де мүмкін. Жаңа қосылатын барлық ұяшықтар бос болады. Сонымен, егер қандай да бір уақытта лентаның r ұяшығы болса және машинаның сыртқы алфавиті мынадай символдардан тұрса a_0, a_1, \dots, a_m , онда лентаның жағдайы былай жазылады:

$a_{i0}, a_{i1}, \dots, a_{im}$. a_{i1} -солдан бастап орналасқан бірінші ұяшықтың жағдайы, a_{i2} -екіншісінің жағдайы, т.с.с.

Басқару головкасы. Бұл лентаның бетімен қозғалатын және белгілі бір уақытта лентаның белгілі бір ұяшығында тұрады. Кейде керісінше басқару головкасы қозғалмайды, оған сәйкес лента қозғалады деп есептеледі. Ондай кезде головкаға лентаның ұяшығының біреуі әлде келесісі кіреді деп есептеледі. Егер лентаның ұяшығы головкада болса, машина бұл ұяшықты оқып немесе қабылдап жатыр деп есептеледі.

Машинаның ішкі жадысы - әрбір қараған уақытта белгілі бір жағдайда болатын құрылғы. Ішкі жадының жағдайының саны машинамен есептеліп отырады. Ішкі жадының жағдайын мынадай символдармен көрсетеміз S_0, S_1, \dots, S_n , олар машинаның сыртқы алфавитіне кірмейді. Ішкі жадының жағдайының символдарының жиынтығы машинаның ішкі алфавиті деп аталады.

Ішкі жадының жағдайын көбінесе машинаның ішкі жағдайлары деп атайды. Бұл жағдайлардың біреуі бастапқы деп аталады, бұл жағдайдан машина өз жұмысын бастайды, ол S_0 жағдайы болсын. Тағы да бір арнайы жағдай - аяқтаушы соңғы жағдай. Соңғы жағдайды көрсететін символ стоп-символ деп аталады. Стоп-символ Ω өрнектеледі. Егер қандай да бір уақытта машинаның ішкі жадысы Ω келсе, ол жұмысын тоқтатқан болып есептеледі. Машинаның S_i қандай да бір ішкі жағдайында ешқандай өзгеріс болмауы мүмкін, олай болса, машина шексіз жасайды деп есептеледі.

Машина ерекше механизммен жабдықталған деп есептеледі. Ол қабылданатын ұяшық пен ішкі жады жағдайына қарай ішкі жады жағдайын, ұяшық орнын өзгерте алады деп есептеледі.

Машинаның (Тьюринг) ағымдағы жағдайы немесе оның конфигурациясы деп ағымдағы ұяшық a_j және ішкі жады S_1 жағдайларының жиынтығын айтамыз.

Тьюринг машинасының командасы, яғни машина жұмысы бір жағдайдан белгілі бір уақытта механикалық құрылғының бір такті жұмысынан соң жаңа бір жағдайға өтуіне негізделген, содан соң тағы бір такті жұмысынан жаңа бір жағдайға өтуі т.с.с. Егер машина S_i ішкі жағдайында a_i символды лентаның ұяшығын қабылдап, ішкі жады жағдайын келесіге S_b аударып сонымен бірге қабылданған ұяшықтың мазмұнын a_r символына аударып, ал басқару головкасы (H) орнында тұрып, бір ұяшыққа оңға қарай (R), солға қарай (L) қозғалса, онда машина мынадай команда орындап жатыр деп айтады:

$$S_i a_i \rightarrow a_s S_b H$$

$$S_i a_i \rightarrow a_s S_b R$$

$$S_i a_i \rightarrow a_s S_b L$$

Машина орындай алатын барлық командалардың жиынтығы бағдарлама деп аталады. Машина жұмысы шарт бойынша толықтай оның ішкі жадысының S және қабылданатын ұяшықтың a_j жағдайымен анықталатын болғандықтан, барлық $S_i a_j$ ($i=1, \dots, n; j=0, 1, \dots, m$) үшін машина бағдарламасы тек қана бір $a_i a_j$ сөзінен басталатын командадан тұруы қажет. Сонымен $\{a_0, a_1, \dots, a_n\}$ символды және $\{S_0, S_1, \dots, S_m\}$ жағдайдағы машина бағдарламасы максимум $(n+1)(m+1)$ командаларынан тұрады. Бағдарламада мынадай жолдардың өңделуі мүмкін емес, бірақ формалды түрде олардың болуы қате болып саналмайды.

Тьюринг машинасының қарапайым түрінде кейбір алгоритмдерді құру қиындық туғызады. Мысалы, аралық мәліметтерді бір жерде сақтау немесе бірнеше элемент топтарының символын салыстыру және т.б. Кей жағдайда қосымша лентаның болуы немесе сөздерді бірнеше лентаға орналастыру дұрыс шешім қабылдауға көмектеседі.

Көпленталы машина әрбір лентаға арналған алфавитке ие. Машинадағы ленталар бір-бірінен тәуелсіз қозғалады. Алайда машина лентасының жағдайы бірдей, ол жағдай басқару механизмі болып табылады. Бір ленталы машинаны қарастырғанда лента қозғалмайды, ал головка берілген бағытта қозғалады деп есептелді. Бірақ көп ленталы машинаны қарастырғанда бұл жағдай ыңғайлы емес. Себебі ленталар бір-бірінен тәуелсіз, ал бір басқару механизмінде түрлі

бағыттағы қозғалысты көрсету қиын. Сонымен басқару механизмі қозғалмайды, ал ленталар оңға, солға бір-бірінен тәуелсіз қозғалады деп есептеледі.

Көп ленталы машинаның бағдарламасы - шешімге қарай келесі жазба тәртібі қолданылады: $S_i \{a,b,c\} \rightarrow \{a',b',c'\} \{R,L,H\} S_j$

Тьюринг белгілі бір U есептеу машинасының құрылысының мүмкіндігін көрсетті, U машинасы универсалды деп аталу себебі онда түрлі есептеулерді жасауға болады.

Тьюринг машинасының ерекшелігі сәйкес келетін кодтау жолымен кез-келген есептеуді берілген Тьюринг машинасында орындай алуында. Кодтау жеңіл болуы керек.

Тьюринг машинасының бағдарламасының кодталуы

Тьюрингтің универсалды машинасы ТУМ лентасында берілген Тьюринг машинасының ТМ код номері жазылады. ТУМ осы код номерін оқып, оның лентасында бағдарламасы жазылған машинаның барлық жұмысын атқаруы қажет. Осыған сәйкес мұндай машиналарға белгілі бір әдіспен жазылған кіріспе сөз қажет. Мүмкін белгілердің саны көп болғандықтан, барлық символдар басқа белгілердің ретімен кодталады. Егер A машинасы m символға ие A_i және n ішкі жағдайға S_j ие болса, кодты былай көрсету керек:

$A_i = 1 \dots 1$ ($A_1=1, A_2=11, A_3=111$ және т.б.)

$S_j = 2 \dots 2$ ($S_1=2, S_2=22, S_3=222$ және т.б.)

$R = 3$

$L = 33$

$H = 333$

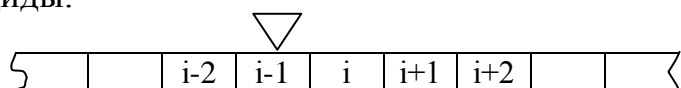
Мұндай жағдайда машина жұмысының бағдарламасын белгілі бір санмен жазуға болады. Жазбаның екі нұсқасы бар:

1. Команданың бөлу белгісі болмайды. Мұндай жағдайда командаларды мынадай форматта жазу қажет $A_{old} S_{old} A_{new} R S_{new}$ сонда бірінен соң бірі орналасқан екі командалар элементар анализатормен бөлінеді.

2. Команда бөлгіші бар. Мысалы X саны оны 4 саны кодтаймыз.

4.3 Пост машинасы

Абстрактылы Пост машинасы шексіз лента түрінде болады, ол жеке ұяшықтарға бөлінген, оған белгіні енгізеді немесе бастиек көмегімен белгіні жазады немесе оқиды.



Абстрактылы Пост машинасы

Лента немесе бастиек командаға байланысты бір қадам солға немесе оңға жылжиды. Лента бастиек қарама-қарсы ұяшыққа орналасатындай тоқтайды. Абстрактылы автоматтың құрамына төмендегі әрекеттердің біреуі кіреді:

Әрбір команданың өзінің i нөмірі болады. Стрелка жылжу бағытын көрсетеді. Команда соңындағы екінші j саны жөнелту (жіберу) деп аталады.

Басқаруды беру командасында екі жөнелту болады. Сондықтан абстрактылы автомат екі қасиетке ие:

Команда	лентаның қалып-күйі	
	командаға дейін	командадан кейін
Бастиекті оңға жылжыту $\rightarrow m$		
Бастиекті солға жылжыту $\leftarrow m$		
Белгіні жазу $M m$		
Белгіні өшіру $C m$		
Басқаруды беру $\begin{cases} m1 \\ m2 \end{cases}$		
Токта тоқта n		

- 1) бірінші орында нөмір 1 команда, екінші орында 2 нөмірі және т.с.с
- 2) кез-келген командадан жөнелту бағдарлама командасы алынады.

Лентаны солға немесе оңға жылжытқаннан кейін бастиек ұяшықтың қалып күйін оқиды (бос немесе белгі жазылған). Бос секциялар немесе белгіленген секциялар туралы ақпарат лентаның қалып күйін немесе автоматтың қалып күйін құрады. Автоматтың бағдарламасы деп командалардың бос емес шектелген тізімін айтамыз.

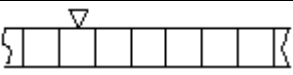
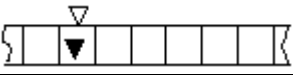
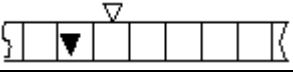


Абстрактылы автоматтың жұмыс істеуі үшін бағдарлама және бастапқы күйін беру керек, яғни бастиектің орны мен лента ұяшықтарының күйін беру керек. Әрбір команда бір қадамда орындалады, одан кейін жөнелтуде көрсетілген нөмірлі команданың орындалуы басталады. Егер команда екі жөнелтуден тұрса, егер бүркеншік бос ұяшықта тұрса, онда жоғарғы жөнелту орындалады. Егер бүркеншік белгісі бар ұяшықта тұрса, онда төменгі жөнелту орындалады. Басқаруды беру командасының орындалуы автоматтың күйін өзгертпейді (белгілердің бірде біреуі жойылмайды, қойылмайды және лента қозғалыссыз қалады). Автоматты іске қосқанда төмендегі жағдайлардың біреуі болуы мүмкін:

- 1) автомат орындалмайтын командаға жетті (белгіні бос емес ұяшыққа жазу, бос ұяшықтағы белгіні өшіру); бұл жағдайда орындалу аяқталады, автомат тоқтайды, нәтижесіз тоқтату болады.

- 2) автомат тоқта командасына жетті, бағдарлама орындалды деп есептеледі, нәтижелі тоқтату болады.

- 3) автомат нәтижелі тоқтатуға да, нәтижесіз тоқтатуға да жетпейді, шексіз жұмыс істеледі.

Пост машинасының типтік бағдарламасын орындау кезіндегі автомат жұмысын қарастырамыз. Бастиектің бастапқы күйі берілген және бос лентаға екі белгі жазу керек.

Бастапқы күйі	
$M 2$	
$\rightarrow 3$	
$M 3$	
Тоқта 3	

Пост машинасында қолданылатын сандар позициялық емес.

Дәріс №5. Сандық автоматтарда ақпаратты өрнектеу. Санау жүйелері

Дәріс жоспары:

5.1 Ақпаратты өрнектеу үшін санау жүйесін таңдау

5.2 Позициялық емес санау жүйесі

5.3 Позициялық санау жүйесі

5.4 Ондық санау жүйесі

5.5 Екілік санау жүйесі

5.6 Сегіздік санау жүйесі

5.7 Он алтылық санау жүйесі

5.1 Ақпаратты өрнектеу үшін санау жүйесін таңдау

Ақпарат ЭЕМ-ге қатысты сыртқы дүниеде үздіксіз немесе дискретті түрде өрнектеледі. ЭЕМ ішінде ақпарат әрқашанда белгілі бір санау жүйелеріндегі сандар түрінде өрнектеледі. Сандық автоматтардағы санау жүйелерін таңдау туралы сұрақ - жеке құрылғылардың алгоритмдерінің қызметін жобалаудың, сондай-ақ осы автоматтың техникалық сипаттамаларын есептеудің маңызды сұрақтарының бірі болып табылады.

Санау жүйелері - сандарды цифрлық таңбалар немесе символдармен жазудың тәсілдері мен ережелерінің жиынтығы. Бізге кеңінен таныс санау жүйесі ондық санау жүйесі болып табылады, онда сандарды жазу үшін 0, 1, ..., 9 цифрлар қолданылады. Сандарды цифрлы таңбалармен жазудың көптеген түрі бар. Практикалық қолдануға арналған кез келген санау жүйесі төмендегілерді қамтамасыз етуі тиіс:

- шамалардың қарастырылып отырған аралығында кез келген санды өрнектеу мүмкіндігі;

- өрнектеудің жалғыздылығы (символдардың әрбір тіркесіне тек қана бір шама сәйкес қойылуы қажет);

- сандарға амал қолданудың қарапайымдығы.

Сандарды өрнектеу жүйелерінің барлығы позициялық және позициялық емес болып бөлінеді.

Сандарды жазудың ең қарапайым тәсілі төмендегі өрнекпен сипатталады

$$A_D = D_1 + D_2 + \dots + D_k = \sum_{i=1}^{i=k} D_i,$$

мұндағы A_D - D санау жүйесінде A санын жазу; D_i - жүйенің $D = \{D_1, D_2, \dots, D_k\}$ негізін құрайтын символдары. Санаудың позициялық емес жүйесі осы ұстаным бойынша құрылған.

5.2 Позициялық емес санау жүйесі

Позициялық емес санау жүйесінде әрбір цифрдың мәні оның алатын орнына байланысты емес. Мұндай жүйелерді құру принциптері күрделі емес. Оларды құру үшін негізінен қосу және азайту амалдарын қолданады. Мысалы, бір символы (таяқшалар) бар жүйе көптеген халықтарда кездескен. Бұл жүйеде қандай да бір санды бейнелеу үшін сол санға сәйкес таяқшаларды жазу керек. Бұл жүйе тиімді емес, өйткені санның жазылуы ұзын болады. Позициялық емес санау жүйесінің басқа мысалы ретінде рим жүйесі болып табылады, ол жүйеде келесі символдардың жиынтығы қолданылады: I, X, V, L, C, D, M және т.с.с. Бұл жүйеде цифрдың сандағы орнына байланысты мәнінің тәуелсіздігінің ережелерінен ауытқулар болуы мүмкін. LX және XL сандарында X символы екі түрлі мән қабылдайды: +10 - бірінші жағдайда және - 10 - екінші жағдайда.

Жалпы жағдайда санау жүйелерін келесі ұстаным бойынша құруға болады:

$$A_B = \alpha_1 B_1 + \alpha_2 B_2 + \dots + \alpha_n B_n, \quad (5.1)$$

мұндағы A_B - негізі B_i болатын санау жүйесінде A санының жазылуы; α_i —негізі B_i болатын санау жүйесінің цифры (символы); B_i - жүйенің базасы, немесе негізі.

Егер $B_i = q^i$ деп есептесек, онда (5.1)-ді ескере отырып алатынымыз

$$B_i = q_i B_{i-1}. \quad (5.2)$$

Позициялық санау жүйесі - (5.2) теңдігін қанағаттандыратын жүйе.

5.3 Позициялық санау жүйесі

Позициялық санау жүйесінде цифрдың мәні оның сандағы орнымен анықталады: бірдей таңба әр түрлі мән қабылдайды. Мысалы, 222 санында оң жақтағы бірінші цифр екі бірлікті, оған көрші - екі ондықты, ал сол жақтағысы - екі жүздікті білдіреді.

Кез келген позициялық санау жүйесі негізімен сипатталады. Позициялы санау жүйесінің q негізі (базисі) - берілген жүйеде санды бейнелеуде

қолданылатын таңбалар немесе символдардың саны. Позициялы жүйенің шексіз көп жиыны болуы мүмкін, себебі, негізі ретінде кез келген санды алып жаңа жүйе құруға болады. Мысалы, келесі таңбалардың (цифрлардың): 0, 1, ..., 9, A, B, C, D, E, F (A, ..., F-тардың орнына кез келген басқа символдарды жазуға болады, мысалы, $\bar{1}, \bar{2}, \dots, \bar{5}$).

Позициялық санау жүйесі үшін келесі теңдікті жазуға болады:

$$A_q = \sum_{i=-m}^{i=n} a_i q^i \quad (5.3)$$

немесе $A_q = a_n q^n + \dots + a_1 q^1 + a_0 q^0 + a_{-1} q^{-1} + \dots + a_{-m} q^{-m}$, мұндағы A_q - негізі q болатын санау жүйесінде жазылған кез-келген сан; $n + 1, m$ - бүтін және бөлшек разрядтардың саны.

Санның позициялы санау жүйесінде p_i разрядының салмағы, төмендегі қатынаспен өрнектеледі:

$$p_i = q^i / q^0 = q^i, \quad (5.4)$$

мұндағы i - оннан солға есептегендегі разряд нөмірі.

Егер разрядтың салмағы $p_i = 10^k$ болса, онда келесі жоғарғы разрядтың салмағы $p_{i+1} = 10^{k+1}$ көрші кіші разряд салмағы - $p_{i-1} = 10^{k-1}$. Разрядтардың арасындағы мұндай өзара байланыс олардың арасындағы ақпараттарды беру мүмкіндігіне алып келеді.

5.4 Ондық санау жүйесі

Бұл жүйенің түп төркінінде он негізі жатыр. Бұл жүйеде санды жазу үшін он цифр қолданылады - 0, 1, 2, 3, 4, 5, 6, 7, 8, 9.

Ондық жүйе позициялық болып табылады, өйткені ондық санды жазуда цифрдың мәні оның позициясына немесе санда орналасқан орнына байланысты.

Санның цифрына бөлінетін позицияны *разряд* деп атайды.

Мысалы, 425 жазуы 4 жүздіктен, 2 ондықтан және 5 бірліктен тұратын сан екенін білдіреді. 5 цифры - бірліктер разрядында, 2 - ондықтар разрядында, 4 - жүздіктер разрядында тұрады.

Егер осы цифрларды басқа ретте жазатын болсақ, мысалы, 524, онда сан 5 жүздіктен, 2 ондықтан және 4 бірліктен тұрады.

Бұл кезде 5 үлкендеу салмақты болады және санның *үлкен цифры* деп аталады, ал 4 цифры кішкене салмақты болады да, осы санның *кішкентай цифры* деп аталады. Егер 524 санын қосынды түрінде жазатын болсақ:

$$5 \cdot 10^2 + 2 \cdot 10^1 + 4 \cdot 10^0$$

Бұл жазбадағы 10-саны санау жүйесін негіздеуші. Санның әрбір цифры үшін 10 негіздеуші цифрдың орнына байланысты дәрежеленеді және осы цифрға көбейтіледі. Бірліктер үшін негіздеуші дәреже - нөлге, ондықтар үшін – бірге, жүздіктер үшін екіге тең және т.с.с.

Егер ондық сан бөлшек болса, онда ол да қосынды түрінде оңай жазылады. Әрбір цифрдың бөлшек бөлігі үшін негіздеуші дәреже теріс және -1-ге тең - бұл бөлшек бөліктің үлкен цифры үшін, ал бөлшек бөліктің келесі цифры үшін -2-ге тең және т.с.с.

Мысалы, 384,9506 ондық саны мынадай қосындымен белгіленеді:

$$384,9506=3*10^2+8*10^1+4*10^0+9*10^{-1}+5*10^{-2}+0*10^{-3}+6*10^{-4}$$

$$856,25=8*10^2+5*10^1+6*10^0+2*10^{-1}+5*10^{-2}$$

$$12937,1=1*10^4+2*10^3+9*10^2+3*10^1+7*10^0+1*10^{-1}$$

Осылайша, ондық санның кез келген цифрының салмағы - оның белгілі бір бүтін дәрежесі, ал дәреженің мәнін сәйкес цифрдың позициясы бекітеді.

5.5 Екілік санау жүйесі

Компьютерде әдетте ондық емес позициялық екілік санау жүйесі, яғни 2 негіздеуші бар санау жүйесі қолданылады.

Екілік жүйеде кез келген сан екі 0 және 1 цифрларының көмегімен жазылады және екілік сан деп аталады.

Тек қана 0 және 1 цифрларынан тұратын екілік саннан ондық санды ажырату үшін екілік санды жазуда екілік санау жүйесінің индексіне белгі қосылады, мысалы, $110101,111_2$.

Екілік санның әрбір разрядын (цифрын) бит деп атайды.

Екілік жүйенің маңызды құндылығы - цифрды ұсыну ыңғайлылығы (мысалы, 1 цифрына электр кернеуінің бар болуы, ал 2 цифрына электр кернеуінің жоқ болуы сәйкес келуі мүмкін) және компьютер аппаратурасының қарапайымдылығы, жарым-жартылай екілік сандармен орындалатын арифметикалық және логикалық операцияларды жүзеге асыруға арналған арифметикалы-логикалық құрылғы.

Ондық сандар сияқты, кез келген екілік санды екілік санға кіретін цифрлар салмағының айырмашылығын анық бейнелейтін қосынды түрінде жазуға болады. Мысалы, $1010101,101$ екілік сан үшін қосынды мына түрде болады:

$$1*2^6+0*2^5+1*2^4+0*2^3+1*2^2+0*2^1+1*2^0+1*2^{-1}+0*2^{-2}+1*2^{-3}$$

Бұл қосынды ондық сан үшін жазылған қосындының ережесі бойынша жазылады. Берілген мысалда екілік сан жеті санды бүтін және үш санды бөлшек бөліктерінен тұрады. Сондықтан бүтін бөліктің үлкен цифры, яғни бірі $2^{7-1}=2^6$ -ға көбейтіледі, бүтін бөліктің нөлге тең келесі саны, 2^5 -ке көбейтіледі және т.с.с., кішкентайға, үшіншіге, дейін екінің дәрежесі кемуі бойынша цифрдың бөлшек бөлігі 2^{-3} -ке көбейтіледі. Осы қосындыда ондық жүйенің ережесі бойынша арифметикалық операцияларды орындай отырып, 85,625 санын аламыз. Осылайша, $1010101,101$ екілік саны 85,625 ондық санына сәйкес келеді, немесе $1010101,101_2=85,625_{10}$.

Екілік жүйенің елеулі кемшілігі - мұнда санды жазу үшін 0 және 1 цифрлары көп қажет болады. Бұл адамның екілік санды қабылдауын қиындатады. Мысалы, 156 ондық санының екілік жүйедегі түрі мынадай: 10011100 . Сондықтан екілік жүйе әдетте компьютердің "ішкі қажеттілігі" үшін

қолданылады, ал адамның компьютермен жұмыс істеуі үшін үлкен негіздеуішті санау жүйесі таңдалады.

Үштік жүйеде сандарды жазу үшін 0, 1, 2 цифрларын алады. Мысалы,
 $2122_3 = 2 * 3^3 + 1 * 3^2 + 2 * 3^1 + 2 * 3^0$.

5.6 Сегіздік санау жүйесі

Сегіздік санау жүйесі, яғни 8 негіздеуші санау жүйесі, сегіз цифрдың көмегімен санды көрсетеді: 0, 1, 2, 3, 4, 5, 6, 7. Мысалы, $357_8 = 3 * 8^2 + 5 * 8^1 + 7 * 8^0$, мұнда 357 санының индексі “8” санау жүйесін білдіреді. Жазылған қосындыда ондық жүйенің ережесі бойынша арифметикалық әрекеттерді орындай отырып, $357_8 = 239_{10}$ аламыз, яғни 357 сегіздік саны 239 ондық санға сәйкес келеді.

$$124,537_8 = 1 * 8^2 + 2 * 8^1 + 4 * 8^0 + 5 * 8^{-1} + 3 * 8^{-2} + 7 * 8^{-3}.$$

5.7 Он алтылық санау жүйесі

Екілік санау жүйесін компьютерден тыс жерде қолдану өте қолайсыз. Мысалы, $895128_{10} = 11011010100010011000_2$.

Екілік санды жазуды қысқарту үшін 16 негіздеуші бар санау жүйесі қолданылады. Бұл жүйені он алтылық деп атайды.

Он алтылық позициялық санау жүйесінде санды жазу үшін ондық санау жүйесінің цифрлары 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 және жетпейтін алты цифрды белгілеу үшін ондық сандарының мәні 10, 11, 12, 13, 14 және 15 болатын сәйкес латын алфавитінің алғашқы үлкен әріптері: А, В, С, D, E, F қолданылады. Осылайша ондық жүйенің барлық цифрлары және сонымен қатар латынның алты әріптері он алтылық жүйенің “цифрлары” болып табылады.

Он алтылық санау жүйесінде ондық жүйеде $15+1=16$ дегенге сәйкес F санынан кейін F+1 саны келеді.

Сондықтан он алтылық сандарда, мысалы, 3E5A1 түрі болуы мүмкін. Осы санды негіздеуіші 16 қосындысы түрінде есептеп жазсақ, мынаны аламыз:

$$3E5A1_{16} = 3 * 16^4 + E * 16^3 + 5 * 16^2 + A * 16^1 + 1 * 16^0$$

Ондық жүйенің ережесі бойынша арифметикалық операцияларды орындай және $A=10, E=14$ ескере отырып, $3E5A1_{16} = 255393_{10}$ аламыз.

2-кесте. Ондық цифрлардың әр түрлі санау жүйесіндегі эквиваленті

Ондық цифр	Басқа санау жүйелеріндегі эквиваленттері				
	q=2	q=3	q=5	q=8	q=16
0	0000	000	00	00	0
1	0001	001	01	01	1
2	0010	002	02	02	2
3	0011	010	03	03	3
4	0100	011	04	04	4
5	0101	012	10	05	5
6	0110	020	11	06	6
7	0111	021	12	07	7

8	1000	022	13	10	8
9	1001	030	14	11	9
10	1010	031	20	12	A
11	1011	032	21	13	B
12	1100	040	22	14	C
13	1101	041	23	15	D
14	1110	042	24	16	E
15	1111	050	30	17	F

Бүтін ондық сандарды екілік санау жүйесіне ауыстыру

Ереже: Бүтін ондық санды екілік санау жүйесіне ауыстыру үшін осы санды 2-ге бөлу қажет. Алынған бөліндіні 2-ден кіші болғанша бөлінеді қайтадан 2-ге бөле береді және т.с.с.нәтижеде бір қатарға соңғы бөлінеді және соңғысынан бастап барлық қалдықтарды жазу керек.

Мысалы 891 санын ондық жүйеден екілік санау жүйесіне аудару.

Шешімі:

891:2=445,	↑	1	
445:2=222,		1	
222:2=111,		0	
111:2=55,		1	
55:2=27,		1	
27:2=13,		1	
13:2=6,		1	
6:2=3,		0	
3:2=1,		1	
1:2=0,		1	(екілік санның үлкен цифры жазылады)

Соңынан бастап барлық қалдықтарды жазамыз.

$$891_{10} = 1101111011_2$$

Ондық бөлшектерді екілік санау жүйесіне ауыстыру

Ондық бөлшек сандарды екілік санау жүйесіне ауыстыру үшін оны 2-ге көбейтіп, бүтін бөлікті іздеу керек. (үтірден кейін төрт таңбаға дейін)

Мысалы: $0,322_{10}$		$8,83_{10}$	
$0.322 * 2 = 0.644$	↓	0	↑
$0.644 * 2 = 1.288$		1	8:2=4 қалдық 0
$0.288 * 2 = 0.576$	↓	0	4:2=2 қалдық 0
$0.576 * 2 = 1.152$		1	2:2=1 қалдық 0
			1:2=0 қалдық 1
Жауабы:			
$0,322_{10} = 0.0101_2$		$0.83 * 2 = 1.66$	бүтін бөлік 1 ↓
		$0.66 * 2 = 1.32$	бүтін бөлік 1 ↓
		$0.32 * 2 = 0.64$	бүтін бөлік 0 ↓
		$0.64 * 2 = 1.28$	бүтін бөлік 1 ↓
		Жауабы: $8,83 = 1000,1101$	

Ондық сандарды сегіздік санау жүйесіне ауыстыру

Ондық жүйеден сандарды сегіздік санау жүйесіне ауыстыру үшін екілік жүйесі сияқты сандарды тек 8 санына бөлеміз. Егер алынған бөлінді 7-ден көп болса, онда оны да, қалдықты сақтап 8-ге бөлуге болады. Мысалы: Ондық жүйедегі 891 санын сегіздік санау жүйесіне келтірейік.

Шешімі:	қалдық
891:8=111	↑ 3
111:8=13	7
13:8=1	5
1:8=0	1
891 ₁₀ =1573 ₈	(қалдық сегіздік санның үлкен цифры жазылады)

Ондық сандарды он алтылық санау жүйесіне ауыстыру

Ондық санды он алтылық санау жүйесіне ауыстыру үшін 16-ға бөлу керек.

Шешімі:	қалдық
891:16=55	11
55:16=3	7
3:16=0	3
891 ₁₀ =37B ₁₆	

Сандарды екілік жүйеден сегіздік санау жүйесіне ауыстыру

Кез келген цифрды сегіздік сан түрінде жазу үшін үш екілік цифрлар қажет. Сондықтан түрленетін екілік санды оңнан солға қарай екі цифрлар тобына үштен бөледі. Екілік жүйедегі бөлшек санды сегіздік санау жүйесіне аудару үшін санның бүтін бөлігін оңнан сола қарай бағытта, ал бөлшек бөлігін солдан оңға қарай бағытта үш екілік саннан бөліп жазып, кестені пайдаланып, сәйкесінше сегіздік санды жазамыз. Сол жақтан және оң жақтан жетпеген цифр орындарын нөлмен толықтырамыз.

Мысалы: 1101111011 екілік саны екілік цифрлар бойынша үштен топқа бөлінгенде, 1 101 111 011 сияқты бөледі. Кестеде көрсетілген цифр түрінде қарастырамыз. 1573₈;

$$1011101,10011 \text{ санын сегіздік жүйеге ауыстырайық,}$$

$$1\ 011\ 101,100\ 11 \rightarrow 001\ 011\ 101,100\ 110 \rightarrow 135,46_8;$$

Екілік санау жүйесі	000	001	010	011	100	101	110	111
Сегіздік санау жүйесі	0	1	2	3	4	5	6	7

Сандарды екілік жүйеден он алтылық санау жүйесіне ауыстыру

Екілік жүйеден он алтылық санау жүйесіне түрлендіргенде, екілік сан төрт екілік сан бойынша бөлінеді, өйткені он алтылық санның кез келген цифрын жазу үшін төрт екілік цифр қажет.

Мысалы:

1101111011 екілік саны төрт екілік цифр бойынша топқа бөлгеннен кейін, 11 0111 1011 сияқты жазуға болады. 37B₁₆;

Екілік жүйедегі бөлшек санды он алтылық санау жүйесіне аудару үшін санның бүтін бөлігін оңнан сола қарай бағытта, ал бөлшек бөлігін солдан оңға қарай бағытта төрт екілік саннан бөліп жазып, кестені пайдаланып, сәйкесінше он алтылық санды жазамыз. Сол жақтан және оң жақтан жетпеген цифр орындарын нөлмен толықтырамыз.

Мысалы: 101111,100011 санын он алтылық жүйеге ауыстырайық,
 $10\ 1111,1000\ 11 \rightarrow 0010\ 1111,1000\ 1100 \rightarrow 2F,8C_{16}$;

Екілік санау жүйесі	0000	0001	0010	0011	0100	0101	0110	0111
Он алтылық санау жүйесі	0	1	2	3	4	5	6	7

Екілік санау жүйесі	1000	1001	1010	1011	1100	1101	1110	1111
Он алтылық санау жүйесі	8	9	A	B	C	D	E	F

Дәріс №6. Екілік қосындылауыштарда арифметикалық амалдарды орындау

Дәріс жоспары:

6.1 Екілік арифметика ережелері

6.2 Екілік қосындылауышта нүктесі бекітілген түрде көрсетілетін сандарды қосу

6.2.1 Тура кодтың екілік қосындылауышы

6.2.2 Қосымша кодтың екілік қосындылауышы

6.2.3 Кері кодтың екілік қосындылауышы

6.1 Екілік арифметика ережелері

ЭЕМ-ге түсетін ондық жүйедегі сандарды процессор олардың екілік эквивалентіне түрлендіреді. ЭЕМ бағдарламаға байланысты ақпараттарды өңдегеннен кейін процессор екілік түрде алынған нәтижелерді ондық санау жүйесіне қайта түрлендіреді. Ондық сандардың екілік эквивалентіне амалдар қолданғанда процессор екілік арифметика ережелерін басшылыққа алады. Себебі екілік санау жүйесіндегі арифметикалық амалдардың орындалуы қарапайым:

Қосу	Азайту	Көбейту
$0+0=0$	$0-0=0$	$0*0=0$
$0+1=1$	$1-0=1$	$0*1=0$
$1+0=1$	$1-1=0$	$1*0=1$
$1+1=10$ бірліктерді көрші (үлкен) разрядқа тасымалдау жүреді	$0-1=1$ бірлікті көрші (үлкен) разрядтан аламыз	$1*1=1$

1-мысал: Екі екілік $101+11$ сандарын қосуды (ондық жүйеде бұл: $5+3=8$) орындайық.

Жетпеген нөлдерді қосып, қосу амалын бағанда орындаған жөн

$$\begin{array}{r} 101 \\ +011 \\ \hline \end{array}$$

Қосу процесін кезеңмен қарастырайық.

1. Алдымен қосу кіші разрядта орындалады: $1+1=10$. Қосындының кіші разрядына 0 жазылады да бірлік алдыңғы үлкен разрядқа тасымалданады.

2. Келесі сол жақ разрядтың цифрлары мен тасымалдың бірлігі қосылады: $0+1+1=10$. Қосындының бұл разрядына 0 жазылады да, бірлік тағы да келесі разрядқа тасымалданады.

3. Енді үшінші сол жақ разрядтың цифрлары мен тасымалдың бірлігі қосылады: $0+1+1=10$. Бұл разрядта 1 жазылады, ал бірлік келесі үлкен разрядқа тасымалданады.

Нәтижеде

$$\begin{array}{r} 101 \\ + 011 \\ \hline 1000 \end{array} \quad 1000_2 = 8_{10}$$

2-мысал: $1010-101$ екілік санның айырмасын табу. Кіші разрядтан бастап азайтуды бағанада орындаймыз:

$$\begin{array}{r} 1010 \\ - 101 \\ \hline \end{array} \quad \text{- азайту процесін кезеңімен қарастырайық:}$$

Кіші разряд үшін $0-1$ бар. Сондықтан үлкен разрядтан бірлікті аламыз және $10-1=1$ –ді табамыз.

Келесі разрядта $0-0=0$ болады.

Сол жақтағы разрядта тағы да $0-1$ болады. Үлкен разрядтан 1-ді аламыз және $10-1=1$ – ді табамыз.

Келесі разрядта 0 қалады.

$$\begin{array}{r} 1010 \\ - 101 \\ \hline 101 \end{array} \quad \text{алынады.}$$

3-мысал: $101*110$ екілік санының көбейтіндісін табу.

$$\begin{array}{r} 101 \\ \times 110 \\ \hline 000 \\ +101 \\ \hline 101 \\ \hline 11110 \end{array} \quad \begin{array}{l} \text{Тексеру: } 101_2 = 1*2^2 + 0*2^1 + 1*2^0 = 5 \\ 110_2 = 1*2^2 + 1*2^1 + 0*2^0 = 6 \end{array}$$

$$11110_2 = 1*2^4 + 1*2^3 + 1*2^2 + 1*2^1 + 0*2^0 = 16 + 8 + 4 + 2 + 0 = 30_{10}$$

яғни $5*6=30$

Көбейту кестесін кезеңмен қарастырайық:

1. Кіші разрядқа көбейте отырып, кесте бойынша 000 аламыз.

2. Келесі разрядқа көбейткенде, бір разряд солға жылжыған 101-ді аламыз.

3. Үлкен разрядқа көбейткенде де, тағы бір разряд солға жылжыған 101 -ді аламыз.

4. Енді екілік сандарды қосу кестесін есепке ала отырып, қосамыз да, 11110_2 нәтижені аламыз.

Кез келген ЭЕМ-нің арифметикалық–логикалық құрылғысының негізі ретінде қосындылауыш немесе азайтатын құрылғы алынады. Қосындылауыш – екілік екі цифрларды қосу амалын орындайтын цифрлық электронды құрылғы. Бұл құрылғыны екілік қосындылауыш деп атайды.

Арифметикалық әрекеттерді орындағанда әрқашан екі немесе одан да көп сан қатысады. Арифметикалық амал нәтижесінде жаңа сан пайда болады:

$$C=A \vee B \quad (1)$$

мұндағы - арифметикалық әрекет таңбасы (қосу, азайту, көбейту, бөлу).

Операнд-цифрасы автоматта орындағанда арифметикалық амалдарға қатысатын сан. Цифрлы автомат сандардың автоматтық бейнелеріне ғана амал орындағандықтан санның автоматтық бейнесі операнд болып табылады. Яғни, машиналық амалдар үшін (1) өрнекті төмендегідей жазған дұрыс:

$$[C]=[A] \vee [B] \quad (2)$$

мұндағы квадрат жақшада операндтардың автоматтық бейнесі белгіленген.

6.2 Екілік қосындылауышта нүктесі бекітілген түрде көрсетілетін сандарды қосу

Тура кодтың екілік қосындылауышы - үлкен цифрлық таңбалық разрядтардың арасында разрядтық тасымалдау тізбегі болмайтын қосындылауыш. Тура кодтың екілік қосындылауышы таңбалары бірдей сандарды қосады. Яғни мұндай қосындылауыш алгебралық қосу амалын орындай алады.

$$[A]_T = SgA \ a_1, a_2, \dots, a_n$$

$$[B]_T = SgB \ b_1, b_2, \dots, b_n \text{ операндтары берілсін}$$

Мұндағы SgA, SgB - A және B үшін разрядтық таңбалар

a_i, b_i – кескіндердің цифрлық разрядтары

Егер $SgA = SgB$ болса, онда сандардың қосындысының таңбасы қосылғыштың таңбасына ие болады, ал нәтиженің цифрлық бөлігі операндтардың цифрлық бөліктерін қосқанға тең.

Мысал 1. $A=0,1011$; $B=0,0100$ сандарын тура кодтық қосындылауышында қосу керек.

Шешуі:

$$[A]_T = 0,1011 \quad SgA=0$$

+

$$[B]_T = 0,0100 \quad SgB=0$$

$$[C]_T = 0,1111 \quad SgC=0$$

Жауабы: $[C]_T = 0,1111$

Мысал 2. $A=-0,0101$, $B=-0,1001$ сандарын тура кодтық қосындылауышында қосу керек.

Шешуі:

$$\begin{array}{r} [A]T=-0,0101 \quad SgA=0,0101 \\ + \\ \underline{[B]T=-0,1001 \quad SgB=0,1001} \\ SgC=0,1110 \end{array}$$

Жауабы: $[C]T=0,1110$

Тура кодтың екілік қосындылауышында сандарды қосқанда, операндтардың қосындысының абсолют мәні бірден асып кететін жағдай болуы мүмкін. Онда автоматтың разрядтық торының толып кетуі мүмкін. Толып кету белгісі - қосындылауыштың цифрлық бөлігінің үлкен разрядынан тасымалдау бірлігінің бар болуы. Бұл жағдайда $\phi=1$ толып кету сигналы пайда болуы керек. Сол сигнал бойынша машина автоматты түрде тоқтайды және толып кетуді болдырмайтындей масштабтық коэффициентті түзетеді.

Қосымша кодтың қосындылауышы – қосымша кодтағы сандардың кескіндеріне амалдар қолданатын қосындылауыш. Қосымша кодтың екілік қосындылауышының ерекшелігі цифрлық бөліктің үлкен разрядынан таңбалық разрядқа разрядтық тасымалдау тізбегінің бар болуы. Сандардың қосымша кодтарының қосындысы нәтиженің қосымша кодына тең.

Қосымша кодтың екілік қосындылауышында сандарды қосудың ережелерін анықтайық.

Теорема. Сандардың қосымша кодтарының қосындысы нәтиженің қосымша кодына тең.

Дәлелдеу. Сандар жылжымалы нүкте түрінде берілген деп есептейік. Мүмкін болатын жағдайларды қарастырамыз.

1) $A > 0, B > 0, A + B < 1$

$$[A]_q + [B]_q = A + B = [A + B] \text{ – нәтиже оң}$$

2) $A < 0, B > 0, |A| > B$

$$[A]_q = A + q \quad [B]_q = B \text{ болғандықтан}$$

$$[A]_q + [B]_q = A + B + q \text{ – нәтиже теріс}$$

3) $A < 0, B > 0 \quad |A| < B$

$$[A]_q = A + q \quad [B]_q = B + q$$

Сонда $[A]_q + [B]_q = A + B + q$. Бұл қосындының мәні q -ден үлкен болғандықтан, таңбалық разрядтан 1-ді тасымалдайды, бұл қосындыдан q бірлікті алғанмен теңбе тең, яғни нәтижеге тең.

$$[A]_q + [B]_q = A + B$$

4) $A < 0, B > 0 \quad |A + B| < 1$

$$[A]_q = A + q \quad [B]_q = B + q$$

$$\text{Сонда } [A]_q + [B]_q = A + B + q + q = [A + B]_q \text{ – нәтиже теріс}$$

Сонымен, теорема разрядтық тордың толып кетуі болмайтын барлық жағдай үшін орындалады. Бұл сандардың автоматтық кескіндерін екілік арифметика ережелері бойынша қосуға мүмкіндік береді.

Мысал 3. Қосымша код қосындылауышын қолданып $A=0,1010, B=0,0100$ сандарының қосындысын табу керек.

Шешуі:

Бұл сандардың машиналық кескіндері қосылады:

$$[A]_k=0,1010$$

+

$$[B]_k=0,0100$$

$$[C]_k=0,1110$$

Жауабы : $[C]_k=0,1110$

Мысал 4. Қосымша код қосындылауышында $A=-0,1011$; $B=0,0100$ сандарының қосындысын табыңдар.

Шешуі:

$$[A]_k=1,0101$$

+

$$[B]_k=0,0100$$

$$[C]_k=1,1001$$

Жауабы : $[C]_k=-0,0111$

Мысал 5. Қосымша код қосындылауышында $A=0,1011$; $B=-0,0100$ сандарының қосындысын табыңдар.

Шешуі:

$$[A]_k=0,1011$$

+

$$[B]_k=1,1100$$

$$[C]_k=-(1)0,0111$$

Жауабы : $[C]_k=-0,1001$

Кері кодтың екілік қосындылауышы – кері кодтағы сандардың кескіндеріне амалдар қолданатын қосындылауыш.

Теорема. Сандардың кері кодтарының қосындысы нәтиженің кері кодына тең.

Дәлелдеу. Негізгі жағдайларды қарастырамыз.

1) $A>0, B>0$ $A+B<1$ сонда

$$[A]_k+[B]_k= A+B=[A+B]_k$$

2) $A<0, B>0$ $|A|>B$ Мұнда $[A]_k= q-q(-n)+A$

Сонда $[A]_k+[B]_k= q-q(-n)+A+B$

$[A]_k+[B]_k$ оң болғандықтан, бұл өрнектің оң жағы q -ден үлкен болады. Ол таңбалық разрядтан 1-ді тасымалдауға алып келеді. Кері кодтың екілік қосындылауышында таңбалық разрядтан кіші разрядқа тасымалдау тізбегі бар болады (таңбалық разрядтан тасымалдау шамасы $q-q(-n)$ тең). $[A]_k+[B]_k=[A+B]_k$ нәтиже оң.

3) $A<0, B<0$ $A+B<1$

$$[A]_k= q-q(-n)+A;$$

$$[B]_k= q-q(-n)+B \text{ болғандықтан}$$

$[A]_k+[B]_k= q-q(-n)+ q-q(-n)+A+B$ мұнда таңбалық разрядтан бірді тасымалдау пайда болады. Ол қосындыдан $q-q(-n)$ шаманы алып тастағанға теңбе-тең. Яғни $[A]_k+[B]_k= [A+B]_k$

4) $|A|=B, A < 0, B > 0$ сонда $[A]_k + [B]_k = q - q(-n) + A + B = q - q(-n)$ нәтиже қосынды 0-ге тең екенін көрсетеді. Сонымен, кері кодтың қосындылауышында сандардың кескіні екілік арифметика ережесі бойынша қосылатыны көрсетіледі.

Мысал 6. Кері кодтың қосындылауышында сандардың $A=0,0101, B=0,0111$ қосындысын табыңдар.

Шешуі:

$$\begin{array}{r} [A]_k = 0,0101 \\ + \\ [B]_k = 0,0111 \\ \hline [C]_k = 0,0111 \end{array}$$

Жауабы : $[C]_k = 0,0111$

Мысал 7. Кері кодтың қосындылауышында сандардың $A=-0,0101, B=0,0111$ қосындысын табыңдар.

Шешуі:

$$\begin{array}{r} [A]_k = 1,1010 \\ + \\ [B]_k = 0,0111 \\ + (1)0,001 \\ \hline 1 \\ \hline [C]_k = 0,0010 \end{array}$$

Жауабы: $[C]_k = 0,0010$

Дәріс №7. Логикалық алгебраның негізгі ұғымдары

Дәріс жоспары:

7.1 Логикалық (бульдік) айнымалы

7.2 Логикалық амалдар

7.2.1 Логикалық көбейту

7.2.2 Логикалық қосу

7.2.3 Логикалық терістеу

7.1 Логикалық (бульдік) айнымалы

ЭЕМ қатысуымен шешілетін есептердің ішінде әдетте логикалық деп аталатын есептер де аз емес.

Логика дегеніміз – адамның ойлау нысандары мен заңдары туралы, нақты айтқанда, дәлелді пікір заңдары туралы ғылым.

Ғылыми пән ретінде логиканың бірнеше нұсқалары дараланады: формалды логика, математикалық логика, ықтималды логика, диалектикалық логика және т.б.

Адамдар әр түрлі кесте құрғанда, бір-біріне қайшы келетін куәлар жауаптарының дұрысын анықтағанда және басқа көптеген жағдайларда логиканың көмегіне жүгінеді.

Формалды логика сөйлеу тілімен білдіретін біздің кәдімгі мазмұнды пікірімізді талдаумен байланысты.

Математикалық логика формалды логиканың бөлігі болып табылады және оның дәлме-дәл анықталған объектілері мен пікірлері бар, олардың ақиқаттығын немесе жалғандығын бір мәнді шешуге болатын ойларды ғана зерттейді.

Математикалық логиканың саласы пікірлер алгебрасы ретінде (оны басқаша логика алгебрасы деп атайды, ол алғаш рет XIX ғасырдың ортасында ағылшын математигі Джордж Бульдің еңбектерінде пайда болды. Бұл - дәстүрлі логикалық есептерді алгебралық әдістермен шешуге талаптанудың нәтижесі) информатикада жақсы меңгерілген. Логика алгебрасының математикалық аппараты компьютердің аппараттық құралдарының жұмысын сипаттауға өте қолайлы, өйткені компьютердегі негізгі екілік санау жүйесі болып табылады, онда екі цифр: 0 мен 1 қолданылады, ал логикалық айнымалылардың мәндері де екі: «0» және «1». Бұл компьютерді конструкциялағанда, оның логикалық функциялары, схемаларының жұмысы айтарлықтай жеңілденеді және қарапайым логикалық элементтердің саны азаяды. Компьютердің негізгі тораптары ондаған мың осындай логикалық элементтерден тұрады.

Қазіргі кезде пікірлер алгебрасының негізгі операциялары еңбейтін бірде-бір программалау тілі жоқ. Логикалық есептерде тек сандар ғана емес, күтпеген, тым шиеленісті пікірлер де бастапқы деректер болып табылады. Адамдардың ақпарат алмасқандағы қатынас түрлерінің бірі – бұл сұрақтар мен жауаптарды кезектестіру. Әрбір сұрақ бізді қоршаған зат әлемі туралы мағлұматтарды білу қажеттігін білдіреді. Бұл білімді біз пайымдау түрінде айтамыз. Пайымдау, әдетте тікелей бақыланатын деректерді көрсете алады. «Күн жарқырап тұр», «Бұл тіктөртбұрыш – квадрат» және т.б. Алайда пайымдауларда ойдан шығарылған объектілер немесе әлі болып үлгермеген оқиғалар туралы тұжырымдар да айтылуы мүмкін: «су перісі бұтақта отыр», «Бүгін жаңбыр жауады» және т.с.

Пікір дегеніміз - жалған немесе ақиқат болуы мүмкін қандай да бір пайымдау. Мысалы, «қар – ақ», « $2*2=4$ » деген ақиқат, ал «Тау тегіс», « $2*2=5$ » деген – жалған пікірлер. Әдетте, біз бақылайтын деректер ақиқат ретінде қабылданады. Жалған пайымдаулар, көбінесе, пайымдаулар мен ұйғарымдардағы қателерден немес сондай болса екен деген тілегімізді шындық ретінде көрсетуге тырысудан пайда болады.

Пікірлер жалпы және жеке болып бөлінеді. Жеке пікір нақты фактілерді көрсетеді, мысалы, « $3+3<7$ », «Бүгін күн шуақты болды».

Жалпы пікірлер объектілер немесе құбылыстар тобының қасиеттерін сипаттайды, мысалы, «Егер жаңбыр жауып өткен болса, онда көше ылғалды болады», «Кез келген квадрат параллелограм болып табылады» және т.с.с. Жалпы пікір объектілердің қандай да бір бөлігі үшін ақиқат, ал басқа объектілер үшін жалған болуы мүмкін. Мысалы, «Иттер мысықтарды жақсы көрмейді» пікірі көпшілігі үшін рас, бірақ барлығы үшін емес. « $x*y>0$ » пікірі $x=1$ және $y=1$ үшін ақиқат және сонымен қатар y кез келген болғанда, $x=0$ үшін жалған. Егер пікір айтылған ой объектілерінің кез келгені үшін рас болса, онда жалпы

пікір тепе-тең ақиқат деп аталады. Мысалы, «Иттің төрт аяғы бар» пікірі кез келген ит үшін рас. Тепе-тең ақиқат пікірлер заттардың заңды байланыстарын көрсеткенде ерекше пайдалы. Мысалы, « $a+v=v+a$ » пайымдауы кез келген нақты сандар үшін орынды және ол «Қосылғыштардың орындарын ауыстырғаннан қосынды өзгермейді» деген арифметикалық заңды көрсетеді. Күрделі жағдайларда сұрақтардың жауабы ЖӘНЕ, НЕМЕСЕ, ЕМЕС логикалық жалғаулықтарын пайдаланып, құрамды пікірлер арқылы беріледі. Мысалы, «Бұл оқушы ақылды және зерек» пікірі екі қарапайым «Бұл оқушы ақылды» және «Бұл оқушы зерек» деген пікірлерден тұратын құрамды пікір болып табылады. Логикалық жалғаулықтардың көмегімен басқа пікірлерден құрастырылған пікірлерді құрамды деп атайды. Құрамды емес пікірлерді қарапайым немесе элементар деп атайды.

Құрамды пікірлердегі ЖӘНЕ жалғаулығы әрқашан бір мезгілдік ақиқатты болжайтын пікірлерді құрайды.

Құрамды пікірдегі НЕМЕСЕ жалғаулығы екі жақты рөл атқаруы мүмкін. Мысалы, «біз бүгін саябаққа демалуға барамыз немесе бақшада жұмыс істейміз» НЕМЕСЕ жалғаулығын «не» бөлушісімен ауыстыруға болады, «біз бүгін не саябаққа демалуға барамыз, не бақшада жұмыс істейміз», өйткені бір мезгілде саябақта демалу мен бақшада жұмыс істеу мүмкін емес.

Ал «Жаңбыр күндіз немесе кешке жауады» деген сөйлемде үш түрлі жағдай болуы мүмкін: «Жаңбыр күндіз жауады», не «Жаңбыр кешке жауады», не «Жаңбыр күндіз де, кешке де жауады». Бірінші мысалда НЕМЕСЕ жалғаулығы бөлуші рөл, екіншісінде біріктіруші рөл атқарады. Барлық компьютерлік бағдарламаларда және математикалық пайымдауларда НЕМЕСЕ жалғаулығы тек біріктіруші рөлде түсініледі. Мысалы « $x=0$ немесе $y=0$ » пайымдауындағы НЕМЕСЕ жалғаулығы не « $y=0$ » не « $x=0$ », не « $x=0$ » және « $y=0$ » дегенді білдіреді.

Математикада НЕМЕСЕ жалғаулығы бар құрамды пікірді құрайтындардың кемінде біреуі ақиқат болса, ол ақиқат деп есептеледі, ал егер оны құрайтындардың бәрі жалған болса, ол жалған деп есептеледі.

ЕМЕС жалғаулығы теріске шығаруды тұжырымдау үшін қолданылады. Мысалы, « $x=0$ » пайымдауының теріске шығаруы « $x=0$ екені дұрыс емес», бұл $x \neq 0$ пайымдауына тең. «Бұл адам сұлу» пайымдауының теріске шығаруы – «Бұл адам сұлу емес» пайымдауына тең.

7.2 Логикалық амалдар

Логикалық жалғаулықтар математикада күрделі айтылымдарды сипаттайтын логикалық операциялар болып табылады. Логикалық айтылымдармен жұмыс істеу үшін оларға ат қояды. «Айдар жазда теңізге барады» айтылымы А арқылы белгіленсін, ал В арқылы – «Айдар жазда тауға барады» айтылымы белгіленсін. Сонда «Айдар жазда теңізге де, тауға да барады» құрамды айтылымын А және В түрінде қысқаша жазуға болады. Мұндағы «және» - логикалық жалғаулық, А,В – логикалық айнымалылар, олар

тек екі мәнде болады: «ақиқат» немесе «жалған», сәйкесінше олар «0» не «1» арқылы белгіленеді.

Әрбір логикалық жалғаулық логикалық айтылымдармен орындалатын операция ретінде қарастырылады және олардың өз аты мен белгіленуі болады. Математикалық логикада ЖӘНЕ, НЕМЕСЕ, ЕМЕС логикалық операциялары ақиқаттық мәндер кестесімен анықталады.

Ақиқаттық кесте – бұл логикалық операцияның кестелік түрде ұсынылуы, онда кірістік айтылымдардың ақиқаттық мәндерінің барлық мүмкін терулері осы терулердің әрқайсысына арналған операцияның шығыстық нәтижесінің ақиқаттық мәнімен бірге аталған.

Негізгі логикалық операцияларды қарастырайық:

«ЖӘНЕ» конъюнкция (логикалық көбейту) A және B

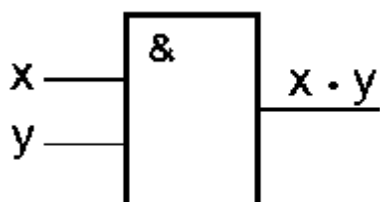
«НЕМЕСЕ» дизъюнкция (логикалық қосу) A немесе B

«ЕМЕС» терістеу A емес

7.2.1 Логикалық көбейту

ЖӘНЕ жалғаулығының көмегімен қарапайым екі A мен B айтылымдарының бір құрамдасқа бірігуі логикалық көбейту немесе конъюнкция, ал операцияның нәтижесі – логикалық көбейтінді деп аталады. ЖӘНЕ операциясы нүктемен белгіленеді.

ЖӘНЕ (конъюнкция) логикалық операциясының ақиқаттық кестесі:



«ЖӘНЕ» кестесі

x	y	$x \cdot y$
0	0	0
0	1	0
1	0	0
1	1	1

Мұндағы A мен B – иә немесе жоқ мәндерін қабылдай алатын екі айтылым. Мысалы, қарапайым айтылымдар берілсін: A - «Астана – Қазақстанның астанасы», B – «Астанада 500 мыңнан астам адам тұрады».

Онда бұл айтылымдардың логикалық көбейтіндісі немесе конъюнкциясы құрамды пайымдау болады: «Астана – Қазақстанның астанасы және Астанада 500 мыңнан астам адам тұрады» айтылымдары ақиқаттық мәніне байланысты *иә* немес *жоқ* мәндерін қабылдайды. « A » және « B » айтылымы да *иә* мәнін қабылдағанда ғана « A » және « B » құрамды айтылымы да *иә* мәнін қабылдайтыны кестеде көрсетілген.

Ақиқаттық кестесінен:

- Пікірдің екеуі де ақиқат болғанда, A және B конъюнкциясы *ақиқат*.

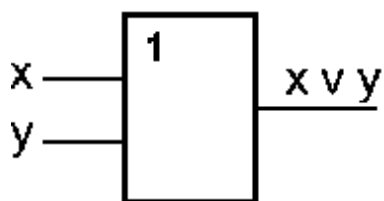
- A немесе B пікірлерінің бірі немесе екеуі де жалған болса, онда A және B конъюнкциясы *жалған* болады.

7.2.2 Логикалық қосу

Біріктіруші мағынада қолданылатын НЕМЕСЕ жалғаулығының көмегімен қарапайым А және В айтылымдарының бір құрамдасқа бірігуі *логикалық қосу* немесе *дизъюнкция*, ал операцияның нәтижесі – логикалық қосынды деп аталады.

НЕМЕСЕ логикалық операциясы \vee белгісімен (кейде $+$ белгісімен) белгіленеді.

НЕМЕСЕ логикалық операциясының ақиқаттық кестесі төмендегідей болады:



«НЕМЕСЕ» кестесі

x	y	$x \vee y$
0	0	0
0	1	1
1	0	1
1	1	1

Ақиқаттық кестесінен:

- А немесе В пікірлерінің тек біреуі ғана ақиқат болғанда, А және В дизъюнкциясы *ақиқат* болады.

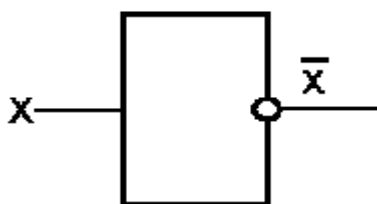
- Пікірдің екеуі де жалған болғанда, А және В дизъюнкциясы *жалған*.

7.2.3 Логикалық терістеу

Қарапайым А айтылымына ЕМЕС жалғауын қосу логикалық терістеу операциясы деп аталады, операцияны орындау нәтижесінде жаңа айтылым пайда болады.

ЕМЕС операциясы айтылымының үстіне сызықша салумен белгіленеді.

ЕМЕС терістеу амалының ақиқаттық кестесі:



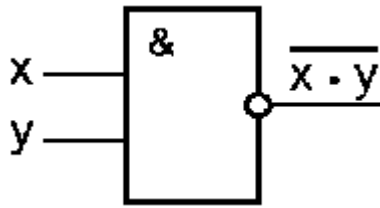
«ЕМЕС» кестесі

x	\bar{x}
0	1
1	0

Ақиқаттық кестесінен:

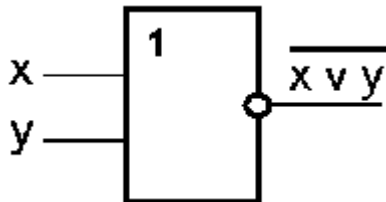
- Бастапқы пікір жалған болғанда, теріске шығару *ақиқат*.

- Бастапқы пікір ақиқат болғанда, теріске шығару *жалған*.



«ЖӘНЕ-ЕМЕС» кестесі

x	y	$\overline{x \cdot y}$
0	0	1
0	1	1
1	0	1
1	1	0



«НЕМЕСЕ-ЕМЕС» кестесі

x	y	$\overline{x \vee y}$
0	0	1
0	1	0
1	0	0
1	1	0

Дәріс №8. Алгоритм ұғымы. Алгоритмдер теориясының негізгі ұғымдары

Дәріс жоспары:

- 8.1 Алгоритм ұғымы
- 8.2 Алгоритмнің қасиеттері
- 8.3 Алгоритмдеу. Алгоритмді бейнелеу тәсілдері мен құрылымдық негіздері
- 8.4 Тьюринг және Пост машинасы көмегімен «алгоритм» ұғымын анықтау
- 8.5 Марковтың нормальды алгоритмдері

8.1 Алгоритм ұғымы

Электронды есептегіш машиналар – алуан түрлі есептерді шешуге арналған ақпараттарды өңдеудің жоғары өнімді құралы. Алайда ЭЕМ – бар болғаны адам құрастырған жарлықтарды тез және дәл орындайтын автомат. Мұндай жарлықтарды дайындау, яғни есептерді шешудің барлық жолын біле тұра жобалау – есептегіш машинаны пайдаланумен байланысты әрекеттің ажырамас бөлігі.

Жетілдірілген есептегіш машиналар есептеу бағдарламаларын зердесіне сақтайды, сонда адамға тек бағдарламаны құрастырып және оны машина зердесіне орналастыру ғана қалады, ал барлық қалғанын – тікелей есепті шешуді – машина автоматты түрде, яғни адамның араласуынсыз, жүзеге асырады. Бұл жағдайда машина бағдарламалы есептегіш техникаға айналады да, онда есепті шешу процесі тікелей бағдарламаны құрастыруға байланысты болады. Демек, есептегіш машинаға арналған бағдарлама негізін берілген есепті шешу алгоритмі, яғни нәтижеге жеткізуші әрекеттер тізбегінен тұратын дәл жарлықтар құрайды. Алгоритм бағдарламаға қарағанда өте жалпы ұғым, ал

бағдарлама берілген есептегіш машинаға түсінікті тілде жазылған алгоритм. Бұдан бағдарламалы басқарылатын техника көмегімен есептерді шешу процесінің негізгі бөлігі - осы есепті шешу алгоритмдерін дайындау екендігі көрінеді. Есепті шешу алгоритмі айқындалғанда, ол еш қиындықсыз бағдарламалау тілінде беріледі. Басқа сөзбен айтқанда, бағдарламалы басқарылатын есептегіш машинада есепті шығарудағы негізгі процесс – алгоритмдеу, яғни алгоритмдік жарлықтарды белгілі бір ережеге сәйкес құрастырып жазу.

Алгоритм сөзі арифметикалық амалдарды орындау ережелерін баян кылған IX ғасырдың ұлы математигі Ал-Хорезмий атының латынша аталуынан (algorithmi) келіп шыққан.

Алгоритмдер – математика және информатиканың шекарасындағы математикалық логикаға жанасатын – алгоритмдер теориясы деп аталатын ғылыми пәннің жүйесі зерттеулерінің объектісі болып табылады. «Алгоритм» терминінің бірнеше анықтамасы бар. Мысалы, академик А.Н. Колмогоровтың анықтамасы бойынша алгоритм дегеніміз қойылған есепті қандай да бір қадамнан кейін шешімге әкелетін нақты қатаң ережелер бойынша орындалатын есептеу жүйесі.

Мектеп оқулықтары да алгоритмге толы деп айтуға болады: арифметикалық амалдардың орындалу ережелері, физика, химия, биология заңдарының математикалық өрнектері, алуан түрлі теңдеулер жүйесін шешу ережелері және т. б. Әр түрлі әдебиетте алгоритмге әр түрлі анықтама берілгенімен, олардың түпкі мағыналары бір негізге саяды. Мәселен,

1. Изделініп отырған нәтижені алу мақсатында мәліметтермен атқарылатын әрекеттердің орындалу ретін көрсететін жарлықты - алгоритм дейміз.

2. Алгоритм дегеніміз - берілген мәндерді пайдаланып қажетті нәтижеге жетуді жүзеге асыратын әрекеттердің орындалу ережесі.

3. Алгоритм дегеніміз – алғашқы мәліметтерден ізделінді нәтижеге жеткізетін әрекеттерді көрсетуші.

4. Алгоритм – алғашқы және аралық мәліметтерді есеп шешімінің нәтижесінде түрлендіру процесін анықтайтын жарлықтардың бірқатар шекті тізбегі.

5. Алгоритм – алға қойылған мақсатқа жету барысында, біртіндеп қандай әрекеттер жасау керек екендігін, түсінікті және дәл көрсететін нұсқаулар мен жарлықтар тізбегі.

6. Алгоритм - берілген есептің шығару жолын реттелген амалдар тізбегі түріне келтіру. Жоғарыдағы анықтамалардан, әрбір алгоритм алғашқы мәліметтерді пайдаланып, іздеген нәтижеге жеткізетін әрекеттер тізбегін көрсетуші ереже деген пікір туады. Мұндай әрекеттер тізбегінің орындалуы - алгоритмдік процесс, әрбір әрекет - оның қадамы, әрбір нұсқау (жарлық) - алгоритм командасы, ал атқарушы жүзеге асыра алатын командалар жиыны - атқарушының командалар жүйесі деп аталады.

Алгоритм мен бағдарламаға байланысты ЭЕМ-нің мынадай жұмыс ерекшеліктері болады:

1. Есепті шығару жолы алгоритм түрінде өрнектелуі қажет;

2. Алгоритм бағдарламаға айналдырылуы тиіс;
3. Бағдарлама машина жадына енгізіліп, ретімен орындалуы керек.

Алгоритмнің математикалық анықтамасы ХХ ғасырдың 30 жылдарында үш типтегі модельдер түрінде алынды:

- 1) Есептелетін (рекурсивті) функциялар
- 2) Шектелген немесе шектелмеген автоматтар теориясы
- 3) Марковтың нормальды алгоритмдері

Бұл бір–біріне тәуелсіз тәсілдердің барлығы бір–біріне эквивалентті. Алгоритм ұғымын формальдаудың негізгі мақсаты әр түрлі математикалық есептердің алгоритмдік шешілетіндігі туралы мәселелерді шешуге алып келеді. Яғни есепті шешуге алып келетін алгоритм құруға бола ма деген сұраққа жауап беру.

8.2 Алгоритмнің қасиеттері

1. Алгоритммен өрнектелген есептеу процесі жеке әрекеттерге жіктелген тізбек болуы тиіс. Осы пайда болатын мазмұндалу алгоритмдік процестің үзілісті құрылымын түзетін, бір–бірінен ажыратылған нұсқаулар (командалар) тізбегінен тұрады. Бір команданың талабын орындамай тұрып, келесі командаға көшуге болмайды. Алгоритмдік жарлықтардың бұл қасиетін дискреттілік (үздіктілік) қасиет деп атайды.

2. Алгоритмде көп мазмұнды командалар болмауы тиіс. Алгоритмнің атқарылуы кезінде алгоритмді құрастырушы қарастырмаған қандай да бір шешім қабылдау қажеттігі тумауы тиіс. Алгоритмдік мазмұндалудың бұл қасиеті алгоритмнің айқындылық қасиеті деп аталады.

3. Алгоритм нақты бір есепті ғана шығаруға бола құрастырылмайды. Ол берілген типтегі есепті толық шығаруға арналады. Алгоритмнің осындай бейімділігі әр түрлі бастапқы мәліметтерді пайдалану мүмкіндігін қамтамасыз етеді. Алгоритмнің мұндай қасиетін ортақтылық қасиет деп атайды.

4. Алгоритмнің негізгі қасиеттерінің бірі – оның нәтижелілігі. Бұл қасиеттің мәнісі – алгоритмнің барлық командаларын дәл атқарғанда шектеулі қадамнан кейін процесс тоқталуы және есептің сұрағына қандай да бір жауап алынуы тиіс.

5. Іс жүзінде алгоритм нақты атқарушыға арналып жазылады. Алгоритм жазу үшін атқарушының қандай нұсқауларды түсінетіндігін және орындай алатындығын білу керек. Атқарушыға түсінікті нұсқаулар жиынтығы атқарушының командалар жиыны деп аталады. Алгоритм атқарушыға толығынан түсінікті болуы керек. Алгоритмнің бұл қасиеті түсініктілігі деп аталады.

8.3 Алгоритмдеу. Алгоритмді бейнелеу тәсілдері мен құрылымдық негіздері

Есептегіш машинаға арналған бағдарламаның негізі - есепті шешу алгоритмі болып табылады. Алгоритм бағдарламаға қарағанда жалпы ұғым

болғандықтан, есептегіш машинаға арналған бағдарлама дегеніміз – бағдарламалау тілінің ережелеріне сай, сол ЭЕМ-нің нақты мүмкіндіктері мен ерекшеліктері ескеріле отырып жазылған алгоритм. Демек, есептегіш машинаның жәрдемімен есеп шығару процесінің мазмұны – алгоритмдеу. Есепті шешу алгоритмі айқын болған жағдайда, оны нақты есептегіш машинаның ерекшеліктерін ескере отырып, бағдарламалау тілінде жазып шығуға болады.

Алгоритм түрліше жолмен мазмұндалады. Алгоритмдерді мазмұндаудың жиі қолданылатын әдістеріне тоқталайық:

Алгоритмді формула түрінде бейнелеу. Математикалық формулалар мен оларды жазу ережелерінің жиынтығы ерекше алгоритмдік тіл болып табылады. Бұл тіл кейбір есептеу алгоритмдерін өрнектеу үшін қолданылады. Мысалы, конустың толық бетінің ауданы:

$$S = \pi \cdot R^2 + \pi \cdot R \cdot l \quad (2) \text{ формуласы арқылы анықталады.}$$

мұндағы, R - конустың табанының радиусы, l - ұзындығы.

Алайда, алгоритмді формула арқылы өрнектеу кезінде алгоритмнің айқындылық (орындалатын әрекеттердің нақтылығы) қасиеті сақтала бермеуі мүмкін. Мысалы, жоғарыдағы формула бойынша конустың толық бетінің ауданын анықтау ретін адам өзі анықтайды. Ол алдымен конустың табан ауданын тауып, оған бүйір бетінің ауданын қосуы немесе керісінше орындауы да мүмкін. Қалай орындағанда да алгоритм түпкі мақсатқа жеткізеді, ең бастысы математикалық амалдардың орындалу реті сақталса болғаны.

Алгоритмді кесте түрінде бейнелеу. Алгоритмді кесте түрінде өрнектеу, есептеу алгоритмдерін құрастыру кезінде жиі пайдаланылады. Есептеудің аралық нәтижелерін кестеге түсіру, есептеу алгоритмнің қадамдарын орындау болып табылады. Мысалы, (2) формула бойынша есептеудің кестелік алгоритмі кестеде көрсетілгендей болуы мүмкін. Бұл кестеде конустың толық бетінің ауданын R мен l -дің әр түрлі мәндері үшін есептеудің барлық кезеңдері келтірілген.




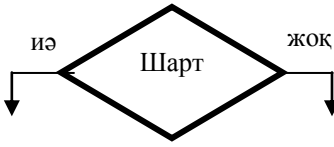


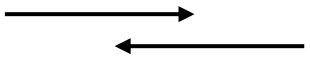
R	l	πR	πR^2	πRl	S
6,5	8,4	20,41	132,66	171,44	304,10
9,2	7,6	28,88	265,76	219,54	485,31
5,1	3,8	16,01	81,671	60,853	142,52

Алгоритмді кесте түрінде жазу, бір өрнектің мәнін әр түрлі бастапқы мәліметтер үшін есептеу кезінде қолайлы болады.

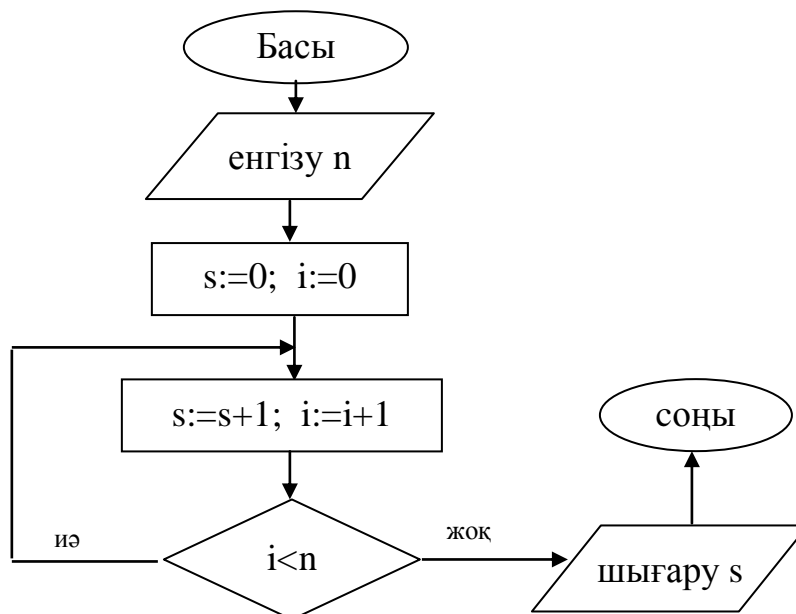
Алгоритмді блок-схема түрінде бейнелеу. Алгоритмнің әрбір әрекеті түрлі жазық геометриялық фигуралар түрінде бейнеленіп, олардың арасындағы байланыстар сызықтар мен бағдарлаушылар (стрелкалар) арқылы көрсетілуі мүмкін. Алгоритмнің мұндай жазылуы блок-схема деп аталады. Блок-схема алгоритмнің құрылымын көрнекті түрде кескіндеуге мүмкіндік береді.

Блок-схема тұрғызуда пайдаланылатын шартты белгілеулердің тізбесі Мемлекеттік стандарт бойынша анықталған. Төмендегі кестеде (1-кесте) жиі қолданылатын бірқатар шартты белгілеулер көрсетілген.

1-кесте. Блок-схемадағы шартты белгілеулер

Геометриялық пішім	Оның атқаратын жұмысы
	Алгоритмдерді бастау және аяқтау
	Мәліметтерді енгізу мен нәтижені шығару
	Математикалық өрнектерді есептеу
	Шартты тексерудің нәтижесінде есептеу процесі бағыттың бірі бойынша жалғастырылады
	Қайталануды ұйымдастыру
	Қосалқы бағдарламаларға кіру және шығу
	Блоктарды бір-біріне байланыстырады

Мысалы, 1-ден n-ге дейінгі натурал сандардың қосындысын есептеу алгоритмін блок-схема түрінде көрсетейік:



Алгоритмді сөзбен баяндап жазу. Алгоритмнің жазылуының кең тараған түрі - сөзбен баяндап жазу (табиғи тіл). Бұл алгоритм күнделікті пайдаланатын сөздер мен алгоритмдік символдарды қолданып, орындалу реттері көрсетіліп жазылған жеке-жеке нұсқаулар жиынтығынан тұрады.

Беймәлім әрекетті жазба түрде баяндау, мысалы, кір жуу машинасын қалай пайдалану керектігі туралы жазба алгоритмді сөзбен баяндау болып табылады. Сол сияқты, қаламен таныс емес адамға жөн сілтеу, қалааралық телефон - автоматпен сөйлесу ережесі, т.б. - алгоритмді сөзбен баяндаудың мысалдары. Алгоритмді сөзбен баяндаудың бір ерекшелігі, ол кез келген алгоритмді, тіпті есептеу алгоритмдерін де мазмұндауға мүмкіндік береді.

Мысалы:

$y = \frac{2}{3 \cdot t - 1}$ формуласы бойынша t -нің кез келген мәні үшін y -тің мәндерін

есептеу алгоритмін құрастыру.

Бұл үшін төмендегідей жарлықтар тізбегін орындау қажет:

- 1) t -нің берілген мәнін оқы;
- 2) t -ны 3-ке көбейт;
- 3) нәтижені 1-ге азайт;
- 4) 2-ні үшінші әрекеттің нәтижесіне бөл;
- 5) нәтижені y деп белгіле.

Осы жарлықтар тізбегі берілген есепті шешу алгоритмі болып табылады. Мұнда атқарушыға y -тің мәнін есептеуге арналған бастапқы формуланы білу міндетті емес. Тиісті нәтиже алу үшін оған тек көрсетілген жарлықтардың орындалу ретін қатаң сақтап отыру жеткілікті.

Қарастырылған алгоритмде біз мынадай бір ережені сақтадық: алгоритмнің бірінші жолы орындалған соң, келесі 2-ші нөмірлі жолдағы нұсқау орындалады. Ең үлкен 5-ші нөмірлі нұсқау орындалған соң алгоритмді атқару аяқталады.

Алайда бұл ереже тек сызықтық алгоритмдер үшін ғана дұрыс болып табылады. Тармақталу командасы жағдайында процестің орындалу реті келесі нөмірлі жарлыққа емес, кез келген нөмірлі жарлыққа ауыса алады. Блок-схема кескінделген алгоритмдерде еш қиындық жоқ, қажетті ауысулардың бағытын бағдар сызықтарды пайдаланып көрсетуге болады. Сөзбен баяндап жазу кезінде бұл үшін арнайы командалар пайдаланылады. Оларды шартты және шартсыз өту командалары деп атайды.

Шартсыз өту командасы мына түрде жазылады:

1-ге өт,

мұндағы 1 - алгоритмдегі командалардың бірінің нөмірі.

Шартсыз өту командасы бойынша нөмірі көрсетілген жолдағы команда орындалады, немесе басқару 5-ші нөмірлі жолға беріледі.

Шартты өту командасының жазылу түрі мынадай:

егер P болса 1-ге өт,

мұндағы P - тексерілетін шарт, 1 - шарт сақталған жағдайда басқару берілетін команданың нөмірі. P шартты есептеу алгоритмі жағдайында бір

немесе бірнеше айнымалыдан тұратын өрнек, математикалық емес алгоритм жағдайында логикалық тұжырым болуы мүмкін. Қалай болғанда да, шарт орындалған жағдайда басқару нұсқауда көрсетілген жолға беріледі де, сақталынбаған жағдайда басқару шартты өту командасынан кейінгі жолға беріледі.

$$y = \begin{cases} x^2, & \text{егер } x \leq 1 \\ x - 1, & \text{егер } x > 1 \end{cases} \quad \text{формула бойынша } y\text{-тің мәнін табу алгоритмін жазалық:}$$

1. x -тің берілген мәнін оқы;
2. егер $x \leq 1$ болса, онда 5-ке өт;
3. x -тен 1-ді шегер;
4. 6-ға өт;
5. x -ті x -ке көбейт;
6. нәтижені y деп белгіле;
7. соңы.

Егер екінші жолдағы шарт қанағаттандырылмаған болса, келесі, үшінші жолдағы команда орындалады. Бұл жағдайда қажеті жоқ бесінші жолда “аттап” өту үшін, төртінші жолға шартсыз өту командасы жазылған. Жетінші жолдағы “соңы” нұсқауы алгоритмнің атқарылуын тоқтатады.

Алгоритмдік тілде бейнелеу. Алгоритмді жазу үшін қандай да болмасын бір тілді пайдалану қажет. Жалпы алғанда тіл - белгілі бір ақпаратты өрнектеу және жеткізу құралы. Бұл мағынада адамдар арасындағы қарым-қатынас тілі, математика тілі, адамдар мен автоматтар арасындағы қатынас тілі, т.б. тілдер туралы айтуға болады. Алгоритмді жазу үшін пайдаланылатын тілдің сипаты осы алгоритмді пайдаланушының мүмкіндігімен анықталады.

Алгоритмдік тіл - құрылымы нақтыланған, бірыңғай және дәл жазылатын арнайы символдар мен ережелер жүйесінен тұрады.

Алгоритмдік тіл бірқатар қатаң талаптарға сай болуы керек:

- 1) командаларды тұжырымдау кезінде атқарушыға белгілі операциялардың орындалуын ғана талап ету;
- 2) командаларды тек қана белгілі бір ережелерге сай қолдану;
- 3) ережеден тыс әрекеттерді пайдаланған жағдайда атқарушы алгоритмді түсінбейді де, оны атқарудан бас тартады.

Мысалы, a және v екі айнымалыны көбейту операциясын математика тілінде бірнеше түрде жазуға болады: 1) av ; 2) axv . Ал алгоритмдік тілде, бұл операция $a*v$ деп жазылады.

Мысалы, $n!$ мәнін есептеудің табиғи тілде жазылатын алгоритмі мен блок-схемасын құру керек ($n! = 1*2*...*n$).

1. Табиғи тілде:

Алгоритм факториал

Басы

n мәнін енгізу;

k -ға 1-ді меншіктеу; r -ға 1-ді меншіктеу.

Цикл басы

r -ға $r*k$ мәнін меншіктеу;

k мәнін 1-ге өсіру.

Егер $k \leq n$ болса, қайталау

Цикл соңы

$n!$ мәнін (r) басып шығару

Соңы.

2. Алгоритмдік тілде:

Алг факториал

1. енгізу n

2. $k:=1$; $r:=1$

3. $r:=r*k$

4. $k:=k+1$

5. егер $k \leq n$ өту 3

6. шығару r

7. соңы

Алгоритмнің бастапқы жолы (Алг факториал) алгоритм тақырыбы деп аталады. Мұнда жазылған Алг – қызметші сөз, факториал – алгоритмге берілген атау. Бұдан соң берілген нұсқаулар:

1-ші жол - есептелінетін $n!$ үшін n айнымалысының мәнін ЭЕМ жадына енгізу; 2-ші жол - k , r айнымалыларына 1 санын меншіктеу;

3-5-ші жолдар - цикл бойынша r ($n!$) мәнін есептеу циклі:

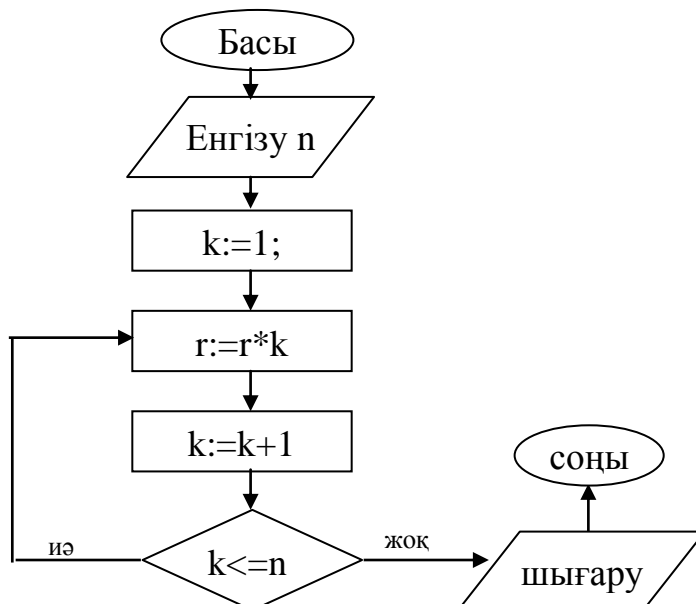
3-ші жолда әр жолы r айнымалысына бастапқы берілгендерден ($k=1$; $r=1$) бастап, $r*k$ көбейтіндісі меншіктеледі;

4-ші жолда k айнымалысының мәні 1-ге өсіріліп, k -ға меншіктеледі.

Ал, 5-ші жол – k -ның жаңа мәнін тексеру командасы. Ол $k \leq n$ болса, есептеуді k -ның жаңа мәнімен қайталау үшін басқаруды 3-ші жолға қайтарып береді, әйтпесе алгоритмнің орындалуы 6-шы жолға өтеді;

6-шы жол - $n!$ мәнін басып шығару; 7-ші жол – есептеуді аяқтау.

Осы алгоритмнің блок-схема түрінде құрылуы төмендегі суретте көрсетілген.



Пайдалануға рұқсат етілген символдардың жиынтығы алгоритмдік тілдің алфавиті деп аталады.

Алгоритмдік тілдің алфавиті мен пайдаланылатын символдар:

Латын, орыс алфавиттері мен араб цифрлары: A, B, C, D, ... , a, b, c, d, ... , А, Б, В, Г, ..., а, б, в, г, ... , 1, 2, 3, ...; * - көбейту; / - бөлу; ** - дәрежелену; + - қосу; - - алу таңбалары; := - меншіктеу белгісі; >, <, >=, <=, <> - қатынас таңбалары; . - нүкте (нүкте - бұл нақты санның бөлшек бөлігін бүтін бөлігінен ажырату үшін пайдаланылатын белгі) және т.б.

Алгоритмдік тілдің алфавитінде мағынасы мен қолдану тәсіліне алдын-ала түсініктеме беріліп қойылған қызметші сөздер деп аталатын символдар да бар. Басқа сөздерден бөліп көрсету үшін әдетте олардың астын сызып қояды немесе қалың қаріппен жазады. Қызметші сөздер:

1. Алгоритм тақырыбы: **алг, арг, нәт**
2. Алгоритмнің басталуы мен соңы: **басы, соңы**
3. Тармақталу командасы: **егер-онда-әйтпесе-бітті**
4. Таңдау командасы: **таңдау, жағдай ... жағдай-әйтпесе-бітті**
5. Қайталану (цикл) командасы:
 - а) n рет циклі: **n рет**
 - б) “Әзірше” циклі: **цб** (цикл басы), **әзір** (әзірше), **цс** (цикл соңы)
 - в) “Үшін” циклі: **цб, үшін, дейін, қадам, келесі, цс - цикл соңы**
- 6) Құрама шарттар: **және, немесе, емес**
- 7) Функция (айнымалы): **мән**
- 8) **Өту n,**
- 9) **Егер p өту n (p-шарт)**

Мұндағы, өту n - шартсыз өту операторы деп аталады. Ол - алгоритм командаларының орындалу ретін тоқтатып, келесі орындалуды n жолға беру жөніндегі нұсқау. Егер p өту n - шартты өту операторы деп аталады, ол «p шарт орындалса, басқаруды n - жолға беру деген нұсқау.

Қызметші сөздер атқарушының командалар жүйесіне кіретін командаларды жазу үшін қолданылады. Егер алгоритмде атқарылу үшін ешқандай шарт қойылмаса, онда оның командалары жай командалар деп аталады. Қызметші сөздерді пайдалану алгоритмнің жазылуын көрнекі етеді, әрі әр алгоритмдердің жазылу үлгісі бірыңғай болады.

Алгоритмдік тілде жазылған әр алгоритмнің атауы яғни тақырыбы болуы керек. Алгоритмнің атауын бөліп көрсету үшін оның алдына **алг** (алгоритм) қызметші сөз жазылады. Ал алгоритмнің басталуы мен аяқталуын көрсету үшін оның командалары **басы** және **соңы** қызметші сөздерінің арасына жазылады. Командалар біртіндеп жазылады және сол жазылу реті бойынша орындалады. Командалардың осы тізбегі - серия деп аталады. Мәтіндік мәндер тырнақшаларға алынып жазылады. Алгоритмдік тілде жазылған алгоритмнің жалпы түрі мынадай:

Алг <алгоритмнің атауы> (типтері көрсетілген шамалар тізімі)

Арг аргументтер атауларының тізімі

Нәт нәтижелер атауларының тізімі

Басы

Алгоритмнің денесі (командалар тізбегі)

Соңы

Осы айтылған **алг.**, **арг.**, **нәт** қызметші сөздерінен тұратын алгоритмнің бөлігі алгоритм тақырыбы деп аталады. Ал **басы** және **соңы** қызметші сөздері алгоритм денесінің басталғанын және алгоритмнің аяқталғанын көрсетеді. Осы екі сөздің арасындағы командалар сериясы алгоритм денесі деп аталады. Денеді ретімен кездесетін бірнеше командаларды бір жолға орналастыру да мүмкін. Бұл кезде олар нүктелі үтір белгісімен ажыратылуы тиіс. Ал шамалар тізімі мен аргументтер, нәтижелер аттары үтір арқылы бөлініп жазылады.

Алгоритмдік тілде алгоритмді басқару құрылымдары деп аталатын негіздік төрт құрылым бар:

1. Тізбектелу;
2. Тармақталу;
3. Таңдау;
4. Қайталану (цикл);

Алгоритм осы құрылымдарды пайдаланып құрылады.

Алгоритмдерді сипаттауда орындаушыға шамаларға қатысты әрекеттерді орындау және осы әрекет нәтижесі ретінде жаңа шаманы құру ұйғарылады. Алгоритм тілінде шамаларға қатысты әрекеттерді белгілеу үшін “:=” командасы пайдаланылды. Оны меншіктеу белгісі деп атайды.

8.4 Тьюринг және Пост машинасы көмегімен «алгоритм» ұғымын анықтау

4-ші дәрісте Тьюринг және Пост машиналары цифрлы автоматтар мысалы ретінде қарастырылған. Бұл машиналар толығымен детерминделген универсалды орындаушылар болып табылады. Олардың көмегімен алғашқы деректер енгізілгеннен кейін нәтижені «оқуға» болады. Тьюринг және Пост машиналарында орындалатын есептеулерге шектеулер бар ма деген сұраққа Пост былайша жауап берген: «егер кез-келген бойынша нәтижеге әкелетін жалпы әдіс болса ғана программа құруға берілген есептердің шешімі болады».

Постың анықтамасы алгоритм ұғымына және осы алгоритмді цифрлы автомат көмегімен шешуге болатындығына әкеледі.

8.5 Марковтың нормальды алгоритмдері

Алгоритм ұғымын формальдау үшін орыс математигі А.А.Марков ассоциативтік есептеулерді қолдануды ұсынды. Мұнда алгоритм ауыстырымдар жүйесі түрінде беріледі, олар қандай символдарды ауыстыру керек екендігін және бұл ауыстырулар қандай ретпен жүретіндігін көрсетеді. Мұндай тәсілді А.А.Марков 50-ші жылдары ұсынды, ол нормальды алгоритм (Марков өзі алгоритм деп атаған) ұғымын енгізді.

Ассоциативті есептеудің кейбір ұғымдарын келтіреміз. А алфавиті (әр түрлі символдардың шектелген тобы) берілсін. Оны құрайтын символдарды әріптер деп атаймыз. Алфавиттің кез–келген шектелген тізбегін (сызықтық қатары), осы алфавиттегі сөзі деп атаймыз. Сөздегі символдар саны оның ұзындығы деп аталады. Ұзындығы нөлге тең сөз бос деп аталады.

S сөзі q сөзінің ішкі сөзі деп аталады. Егер q-ді $q=rst$ түрінде көрсетуге болса, мұндағы r және t - сол алфавиттегі кез–келген сөз.

А алфавитінде N және M екі сөзін қарастырамыз. Егер N M-нің бөлігі болса, онда N M-ге кіреді деп аталады. Қандай да бір алфавитте N-M, S-T,... ауыстыруларының шектелген жүйесі берілсін. Мұндағы N,M,S,T... осы алфавиттегі сөздер. N-M кез–келген ауыстыруын k сөзіне былайша қолдануға болады. Егер K-ге N сөзі бір немесе бірнеше рет кіретін болса, онда оның кез–келгені M сөзімен ауыстырыла алады және керісінше.

А алфавитіндегі алгоритм деп тиімді есептелетін функцияны айтамыз. Оның анықталу облысы ретінде А алфавитіндегі барлық сөздер жиынтығындағы қандай да бір ішкі жиынды айтамыз. Және мәні ретінде А алфавитіндегі сөздер болып табылады.

Дәріс №9. Қолданбалы есептерді шешу алгоритмдерін талдау қағидалары

Дәріс жоспары:

- 9.1 Сызықтық құрылымды алгоритм
- 9.2 Тармақталған құрылымды алгоритм
- 9.3 Қайталанатын құрылымды алгоритмдер
- 9.4 Массив

Алгоритмнің машиналық тілдегі бейнесі программа деп аталады. Ал программа жазуға арналған тілдерді программалау тілдері дейді. Сондықтан машиналық тілдер ең алғашқы программалау тілдері болып есептеледі.

Машиналық тілде программаларды жазу адам үшін өте қиын. Себебі, бір жағынан барлық адамдар мен нұсқаулардың екілік бейнелерін, кодтарын есте сақтау керек, екіншіден кез келген алгоритмді осы машиналық амалдар мен нұсқаулар арқылы өрнектеп білу қажет.

Берілгендер, нәтиже және олардың адрестері екілік сандар арқылы беріледі. Тармақтау нұсқауын жазу үшін басқару беретін адресті қолмен есептеу қажет. Сол сияқты бұрын істелген көптеген жұмыстарды басынан бастап қайтадан жасау керек. Мұның бәрі бірден көзге көрінбейтін көптеген қателер жіберуге соқтырады. Жазылған программалар оқуға және түсінуге қолайсыз болғандықтан, олардың ішіндегі жіберілген қателерді тауып түзету өте көп уақытты және шыдамдылықты қажет етеді. Сондықтан адамдар өздерінің жұмыстарын жеңілдету үшін программа жазуға ыңғайлы басқа жасанды тілдер ойластыра бастады.

Жасанды тілді компьютер өзінің машиналық тіліне адамның қатысуынсыз автоматты түрде аударатындай етіп жасайды. Аударуды программа орындайды.

Ондай программаны ағылшын тілінде «транслятор» (аудару) деп атайды, ал аудару процесін «трансляция» дейді. Сондықтан программаны жасанды тілді пайдаланып жазған кезде компьютердің жұмысы екі кезеңнен тұрады:

1. Жасанды тілден машиналық тілде трансляциялау;

2. Машиналық тілдегі программаны орындау;

Осы кезеңдегі көрсетілген жұмысты орындаудың екі түрлі әдісі бар:

1. Компиляция әдісі – алдымен жасанды программа түгелдей машиналық тілге аударылады, сонан кейін бұл программа басынан орындалады.

2. Интерпретация әдісі – жасанды тілдегі программаның әрбір әрекеті (нұсқау немесе амалы) жеке алдын-ала аударылмастан бірден машиналық тілде орындала бастайды. Яғни, компиляция әдісінде жасанды тілдегі программа тек қана бір рет қарастырылады және аударылған программаны бірнеше рет орындауға болады, ал интерпретация әдісінде әр орындар алдында жасанды тілдегі программаны қайтадан қарастыру қажет.

Сонымен осындай жасанды тілдер де компьютерге түсінікті болады, сондықтан оларды да біз программалау тілдері дейміз.

Қазіргі кезде компьютердің көмегімен әр түрлі есептерді шығаруға мүмкіндік беретін сан алуан программалау тілдері бар. Әр тілдің кемшілігі де, жетістігі де бар. Мысалы, кейбір тілде программаны жазу оңай болғанымен оны орындау өте көп уақытты талап етеді немесе жадыдан көп орын алады.

Жалпы, программалау тілдерін мынадай белгілері, қасиеттері бойынша жіктеуге болады:

1. Тілдің деңгейіне байланысты;

2. Тілдің деңгейіне байланысты, яғни қандай есептерді шешуге мүмкіндік беретініне байланысты;

3. Тілдің моделіне байланысты, яғни құрылу принципіне байланысты.

Ең төменгі деңгей машиналық тілдерде болады деп қабылданады. Сондықтан басқа тілдердің деңгейі деп олардың машиналық тілдерге жақындық өлшемін айтады: тіл неғұрлым машиналық тілге жақын болса, соғұрлым оның деңгейі төмен болады және керісінше. Ең жоғарғы деңгей табиғи тілдерде болады. Сонымен, бірінші кластағы тілдер екі топтан тұрады:

- деңгейі төмен машинаға байланысты тілдер

- деңгейі жоғары машинаға байланысты тілдер

9.1 Сызықтық құрылымды алгоритм

Блоктары ретімен орындалатын алгоритм сызықтық алгоритм деп аталады немесе денесі тек ретімен орындалатын командалардан тұратын алгоритм сызықтық алгоритм деп аталады.

1-мысал. Аргумент x -тің берілген мәні

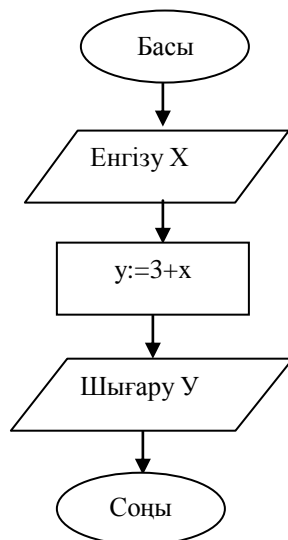
үшін $y=3+x$ функциясының мәнін есептеу алгоритмін түзу

Алгоритмдік тілдің қызметші сөздері арқылы жазу және блок-схемасы

Алг функция (нақ x , нақ y)

Арг x

Нәт у
Басы
 енгізу X
 $y:=3+x$
 шығару Y
Соңы



9.2 Тармақталған құрылымды алгоритм

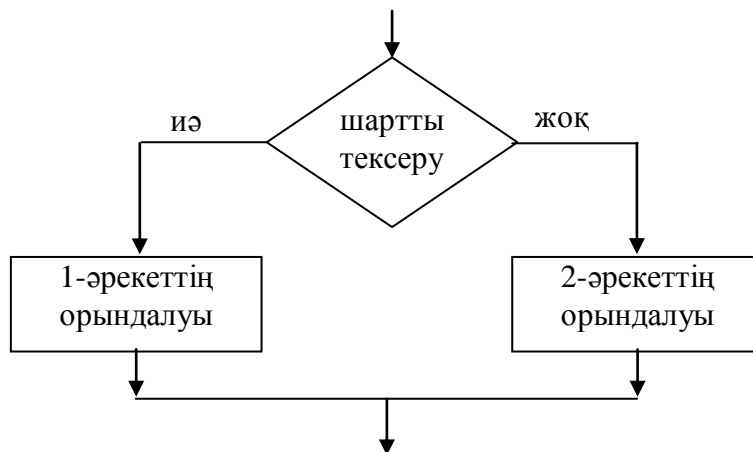
Берілген шарттың орындалуына байланысты бір тармақ бойынша орындалатын есептеу процестері бар алгоритмдер **тармақталған алгоритмдер** деп аталады.

Тармақталған алгоритмдерде тармақталу және таңдау командалары қолданылады.

Тармақталу командасы шарттың орындалуына байланысты екі әрекеттердің бірін орындауға нұсқау береді. Ол «егер» қызметші сөзінен басталып, «бітті» сөзімен аяқталады.

а) Тармақталу құрылымының толық түрі:

егер шарт
онда 1-әрекет
әйтпесе 2-әрекет
бітті



Бұл тармақталу командасы былай орындалады:

- 1) **егер** қызметші сөзінен кейінгі шарт тексеріледі
- 2) егер шарт орындалса, **онда** қызметші сөзінен кейінгі 1-әрекет орындалады. Бұл жағдайда **әйтпесе** қызметші сөзінен кейінгі 2-әрекет орындалмайды, келесі жолға өтеді.

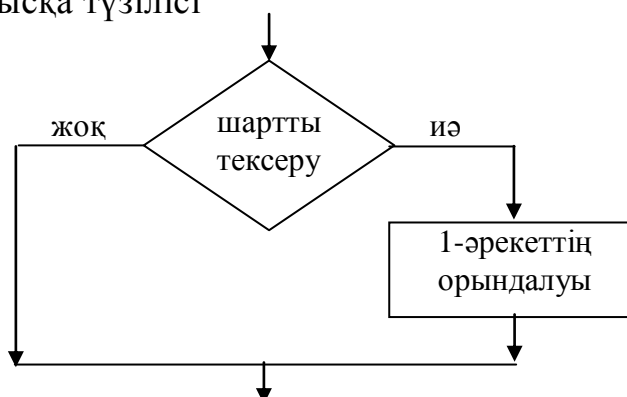
3) егер шарт орындалмаса, онда **әйтпесе** қызметші сөзінен кейінгі 2-әрекет орындалады, бұл жағдайда **онда** қызметші сөзінен кейінгі 1-әрекет атқарылмайды, келесі жолға өтеді.

б) Тармақталу құрылымының қысқа түзілісі

егер шарт

онда 1-әрекет

бітті



Бұл екі түрлі үлгінің бір-бірінен айырмашылығы кейбір есептерді шығарған кезде біз тек ақиқаттық жағдайды ғана қарастырамыз. Сондықтан мұндай жағдайда команданың қысқа түзілісін пайдалану жеткілікті.

Егер есептің шығарылуындағы шарт үшеу болса, онда бір тармақтану командасы екінші тармақтану командасының ішінде пайдаланылады. Осы жағдайда тармақталу командасының алгоритмдік тілдегі түзілісі мынадай болады:

егер шарт

онда 1-әрекет

әйтпесе

егер шарт

онда 2-әрекет

әйтпесе 3-әрекет

бітті

бітті

Бұл жағдайда үш түрлі варианттың тек біреуі ғана таңдалынылады. Команданың жазылуындағы «**әйтпесе егер**» деген сөз тіркесі есептің шығарылу шешімін түсінуді қиындатады. Кейде қойылатын шарттар үштен де көп болады. Бұл жағдайда да егер командасын пайдаланып та шығаруға болады. Бірақ мұндай құрылым, жоғарыда айтқандай, түсінуді одан да әрі қиындата түседі. Сондықтан мұндайда таңдау командасы пайдаланылады.

2-мысал. Аргумент x -тің берілген мәні үшін функцияның (y) мәнін есептеп табу алгоритмін құрыңыз.

$$y = \begin{cases} \operatorname{tg} x^2 + \sin x, & \text{егер } x > \frac{\pi}{2} \\ \cos^2 x - \operatorname{ctg} x, & \text{егер } x \leq \frac{\pi}{2} \end{cases}$$

Алг функция (нақ x , нақ y)

Арг x

Нәт y

Басы

Енгізу π

Енгізу x

Егер $x > \pi/2$

Онда $y := \operatorname{tg}x^2 + \sin x$

Әйтпесе $y := \cos^2 x - \operatorname{ctg}x$

Бітті

Шығару y

Соңы

3-мысал. Алгоритмдік тілде екі санның ең үлкенін табуға программа құру

Алг ЕСҮ (арг a, в, нәт x)

арг a, в

нәт x

басы

енгізу a, в

егер $a \geq в$

онда $x := a$

әйтпесе $x := в$

бітті

шығару x

соңы

Көп жағдайлы, яғни екі жағдайдан артық болған тармақталу командасы таңдау командасы деп аталады. Оның алгоритмдік тілдегі жазылу үлгісі мен блок-схемасы:

таңдау

жағдай 1-шарт: 1-әрекет

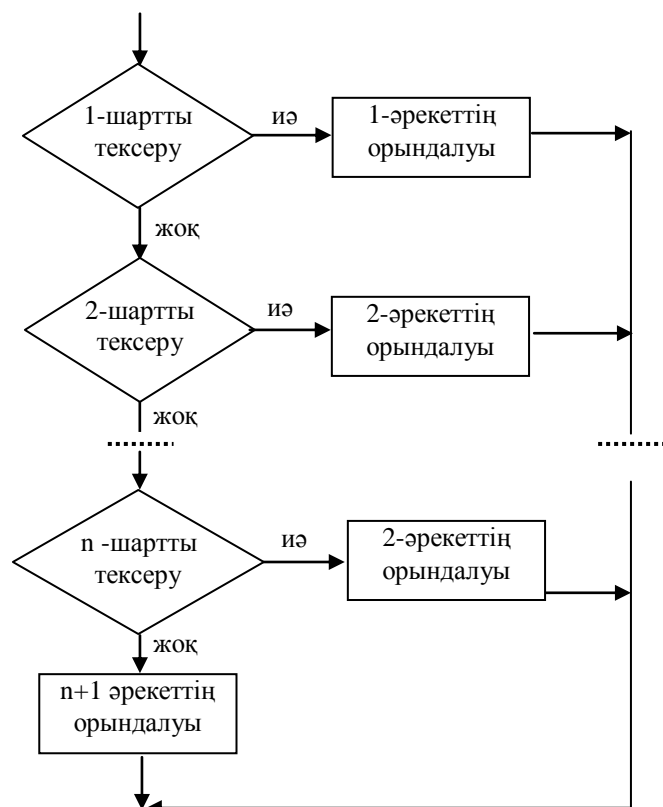
жағдай 2-шарт: 2-әрекет

.....

жағдай n-шарт: n-әрекет

әйтпесе n+1-әрекет

бітті



Алгоритмдік тілде шарттың да екі түрі бар:

1) Бір мезетте бір ғана шартқа тәуелді болып келетін жай шарт: $>$, $>=$, $<$, $<=$, $<>$ қатынас таңбаларымен біріктірілген екі логикалық өрнек.

2) Бірнеше шарт бір мезгілде тексерілетін болса, ондай шартты құрама шарт дейді. Құрама шарт жағдайында да есептің екі шешімі болады: ақиқат және жалған. Жай шарттардан құрама шарттарды құрастыру үшін мынадай қызметші сөздер пайдаланылады: **және**, **немесе**, **емес**.

4-мысал. Балабақшадағы балаларды туған жылы бойынша топтарға бөлуге алгоритмдік тілде бағдарлама құрыңыздар.

Алг топқа бөлу (**арг бүт** x, **нәт лит** топ)

арг баланың туған жылы X берілген

нәт топ

басы

енгізу x

таңдау

жағдай x=1988 : топ:=«яслиге»

жағдай x=1987 : топ:=«кіші жастағы топқа»

жағдай x=1986 : топ:=«орта жастағы топқа»

жағдай x=1985 : топ:=«үлкен жастағы топқа»

жағдай x=1984 : топ:=«даярлық тобына»

әйтпесе топ:=«мектеп жасында»

бітті

шығар топ

соңы

9.3 Қайталанатын құрылымды алгоритмдер

Көп рет қайталанып есептелетін командалар тізбегі бар алгоритмдер **қайталанатын (циклдік) алгоритмдер** деп аталады. Қайталанып орындалатын командалар тізбегін қайталану денесі деп атайды. Ол **цб** (цикл басы) және **цс** (цикл соңы) қызметші сөздерінің ортасында орналасады. Қайталанудың денесі қайталану саны қанша болса, сонша рет орындалады.

Қайталану саны үшін кез-келген бүтін санды алуға болатындықтан, ол сан "0" немесе "теріс" те сан болуы мүмкін. Егер қайталану саны "0" немесе "теріс" сан болса, қайталану бірде-бір рет орындалмайды. Ал ЭЕМ бірден **цс** командасынан соң тұрған командаға өтіп кетеді.

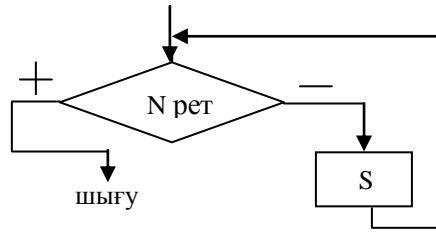
Қайталану командасының төрт түрлі құрылымы бар: «N рет», «Әзірше», «Дейін», «Үшін».

«N рет» қайталану командасы

Көп жағдайларда алгоритмнің ішінде бірнеше командалар сериясы түрлерін өзгертпестен әлденеше рет қайталануы мүмкін.

«N рет» циклінің құрылымы:

цб n рет
S
цс



Қайталану саны N болғанша қайталану (цикл) денесін S қайталап орындау.

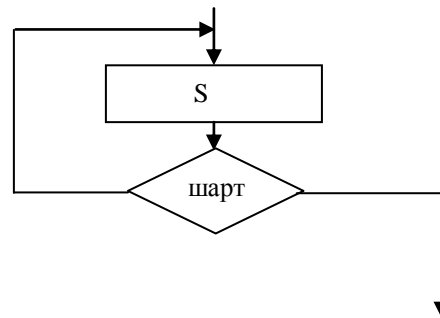
«Дейін» қайталану командасы

Шарттың сақталуы аяқталғанға дейін S-ті қайталап орындау.

Мұндай қайталану түрінде шарттың орындалмаған жағдайында ғана цикл (қайталану) денесі қайталанады.

«Дейін» циклінің құрылымы:

цб
S
дейін шарт
цс



5-мысал. Жалпы мүшесі $a=k/(k^3+5)$ болатын тізбектің алғашқы n мүшесінің қосындысын табу керек ($k=1,2,3,\dots$)

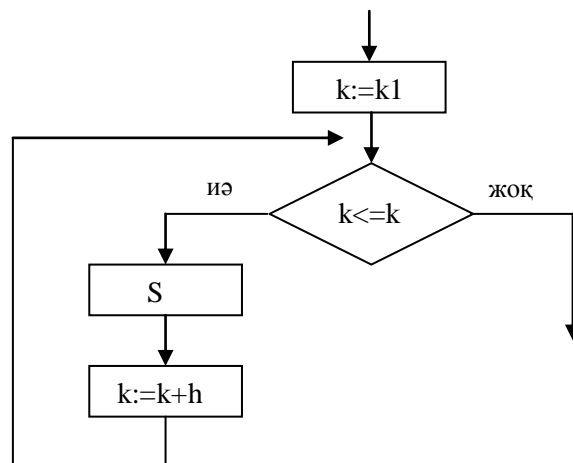
Алг тізбек (**бүт** n, **нақ** a, **нақ** S)

бер n, a
нәт S
басы бүт K
енгізу n
K:=1; S:=0

цб
a:=k/(k**3+5)
S:=s+a
K:=k+1
Дейін k<=n

цс
шығару S

соңы



«Үшін» қайталану командасы

«Үшін» циклінің құрылымы:

цб
k үшін k=k1 ден k2 дейін h қадам
S (цикл денесі)

цс

К - циклдің басқарушы айнымалысы немесе цикл параметрі деп аталады.

k1 - цикл параметрінің бастапқы мәні

k2 - цикл параметрінің соңғы мәні

h – цикл параметрінің өзгеру қадамының мәні

h=1 болса, h қадамды циклге енгізбей кетуге болады.

«Үшін» циклін - параметрлі қайталау циклі деп те атайды.

1. егер $k1 < k2$ болып $h > 0$ болса, цикл денесі $k > k2$ шарт орынды болғанға дейін қайталанып орындалды. Мұнда циклдің әрбір қадамынан кейін цикл параметрінің мәні h-қа артады.

2. егер $k1 > k2$ болып $h < 0$ болса, цикл денесі $k > k2$ шарт орынды болғанға дейін қайталанып орындалады. Мұнда циклдің әрбір қадамынан кейін цикл параметрінің мәні h-қа кеміп отырады.

3. $k1 < k2$ болып $h = 1$ болса, цикл денесі $k1 < k2$ шарт орынды болғанға дейін қайталанып орындалады. Мұнда циклдің әрбір қадамынан кейін цикл параметрінің мәні 1-ге артады.

Алгоритмдік тілдегі **үшін** қайталану командасының бір кемшілігі i параметрінің міндетті түрде бүтін болуы және цикл қадамының 1-ге ғана тең болуы.

6-мысал. a[1:10] кестенің барлық элементтерінің орташа арифметикалық мәнін есептеу алгоритмін түзіңіз

Алг нақ орташа арифметикалық мәні (**нақ** кес a[1:10])

басы

нақ s, **бүт** i

s:=0

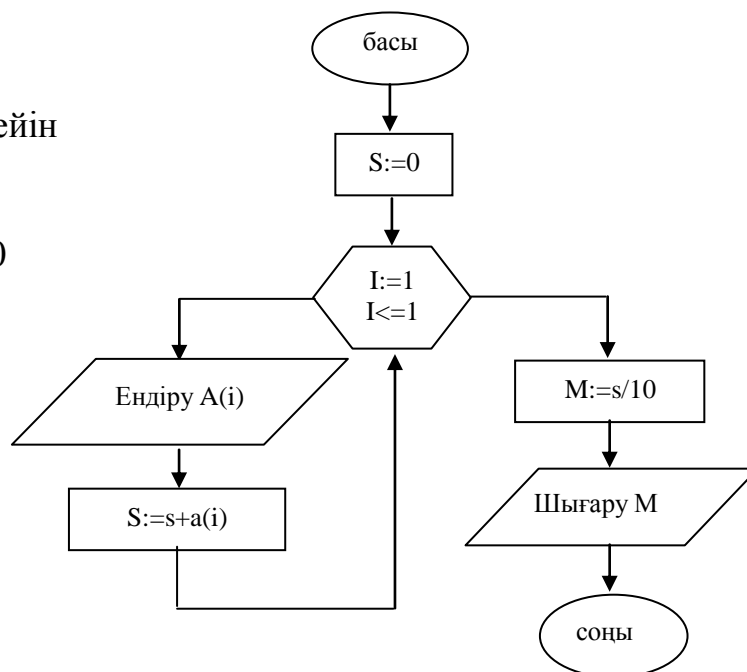
цб i үшін 1 ден 10 дейін

s:=s+a[i]

цс

шығару OA:=s/10

соңы

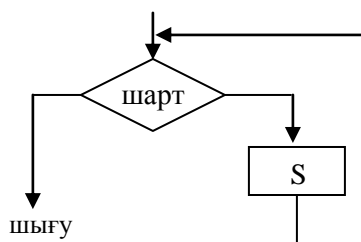


Әзір» қайталану командасы

«Әзір» циклінің құрылымы:

цб

әзір шарт
S (цикл денесі)
цс



Әзір қайталану командасын орындау кезінде ЭЕМ төмендегі әрекеттерді қайталайды:

- а) **әзір** қызметші сөзінен кейін жазылған шартты тексереді;
- ә) осы шарт сақталып тұрғанда, ЭЕМ цикл денесін (**цб** және **цс** сөздерінің арасындағы командаларды) орындай береді.
- б) ал егер шарт сақталмаса, онда ЭЕМ цикл денесін орындауды тоқтатады да **цс** қызметші сөзінен соң тұрған командаларды орындауға өтеді.

Шарт дегеніміз - екі шама арасындағы салыстыруды көрсететін қатыс. Осы қатыстың нәтижесі (ақиқат және жалған) алгоритмнің қайталану командасының орындалу-орындалмауын анықтайды.

7-мысал. Екі оң бүтін санның ең үлкен жалпы бөлгішін есептеу алгоритмін түзіңіз.

алг ЕҮЖБ(**нат** m,n)

басы

нат x, y

x:=m

y:=n

әзір $x \neq y$

цб

егер $x > y$ **онда** $x := x - y$

әйтпесе $y := y - x$

бітті

цс

шығару x

соңы

Итерациялық қайталанулар

Практикада циклді аяқтауда қандай да болмасын бір логикалық шарт арқылы берілетін жағдайлар жиі кездеседі. Мысалы, a_1, a_2, \dots тізбегінің шегін ерс дәлдікпен табу керек болсын (eps (эпсилон) – нәтижені есептеуде мүмкін болатын қате, дәлдік шамасы). Бұл үшін n-нің шексіздікке ұмтылғанында тізбектің шегі болуы тиіс. Осы шарт орындалатын жағдайда a_n мәні $|a_n - a_{n-1}| \leq eps$ теңсіздігі қанағаттанғанға дейінгі есептеледі де a_n мәні қажетті шек ретінде қабылданады. Мұндай циклді – итерациялық цикл деп атайды (iteratio – қайталау).

8-мысал. Жалпы мүшесі $a_k = 1/k^2$ болатын тізбек шегін ерс дәлдікпен есептеу және мәндері ерс-тен кем емес элементтерінің қосындысын табу керек ($k=1,2,\dots$ сәйкес қатардың жинақтылығы математика курсынан белгілі)

Алг Қосынды (**нақ** a, ерс, **нақ** S)
басы **бүт** k
енгізу ерс
 K:=1; s:=0; a:=1
цб **әзір** $\text{abs}(a) \geq \text{eps}$
 S:=s+a; k:=k+1; a:=1/K**2
цс
шығару a, s
соңы

9.4 Массив

Алгоритмдік тілде бір типті шамалардың бір ғана ат беріліп реттелген тізбегі массив деп аталады. Массивпен жұмыс істеу үшін оның әр элементінің мәні белгілі болуы тиіс. Массивтің бүтін немесе нақты сандардан тұратын сандық және символдық типтері бар. Мысалы, сандық кесте, вектор координаттары, матрица (төртбұрышты сандық кесте), әрқайсысы бір элемент түрінде алынған символдар тізбектерінің жиынтығы (текстер) массивтер қатарына жатады (символдық массив элементіне енетін символдар саны 255-тен аспауы тиіс). Массивтің атауы да сандық айнымалының атауы сияқты белгіленеді. Кестеде сызықтық (бір өлшемді) нақты сандар массиві келтірілген. Мұндағы: a - массив аты, a[k] - массив элементтері ($k=1,2,\dots,n$), кестенің төменгі жағында массив элементтерінің мәндері берілген: a[1]=3,5; a[2]=-4,7; т.с.с

Кестеде берілген n элементтен тұратын a массивінің сипатталу түрі мынадай (элемент мәндері – нақты сандар): **нақ** кес a[1:n]

a[1]	a[2]	...	A[n]
3.5	-4.7		7.85

Массивте әр элемент нөмірін (k) элемент индексі деп, элементтің өзін индекстелген айнымалы деп атайды.

9-мысал. $a_0x^n + a_1x^{n-1} + \dots + a_{n-1}x + a_n$ көпмүшелігінің мәнін есептеу керек.

Көпмүшелікті мынадай түрге

$(\dots((a_0x + a_1)x + a_2)x + \dots + a_{n-1})x + a_n$ түрлендіріп алып, еселіктерін бір өлшемді массив элементтері ретінде қарастыруға болады. Математикада мұндай өрнекті Горнер схемасы деп атайды. Үшін циклі бойынша мұның мәнін есептеу алгоритмі:

Алг Горнер (**бүт** n, **нақ** x, **нақ кес** a [0:n], **нақ** y)

басы **бүт** n

енгізу n, x, a[0]; y:=a[0];

цб k үшін 1 бастап n дейін

енгізу a[k]

y:=y*x+a[k]

цс

шығару y

соңы

10-мысал. Сызықтық a[1:n] нақты сандар кестесінің ең кіші элементін табу алгоритмін құру керек.

Алг Min элементі табу (**бүт** n, **нақ кес** a[1:n])

басы **бүт** k, **нақ** min

енгізу n, a[1]

Min:=a[1]; k:=2

цб **әзір** k≤n

енгізу a[k]

егер a[k]<min

онда min:=a[k]

бітті k:=k+1

цс

шығару min

соңы

Екі өлшемді массив матрица деп аталады. Элементтері нақты сандар болып келген n жолдан, m бағаннан тұратын A матрицасының сипатталу түрі мынадай:

нақ кес A[1:n, 1:m], а оның элементтерінің жазылу түрі: A[k,j]. Мұнда k - қатар нөмірі; j - баған нөмірі.

Матрицаны ЭЕМ жадына ендіру үшін екі параметрлі цикл ішінде цикл пайдаланады. (Мұндай циклдерді бір-біріне салынған циклдер деп те атайды)

Дәріс №10. Бағдарлама құрылымы. Операторлар бөлігі

Дәріс жоспары.

10.1 Бағдарлама құрылымы

10.2 Бағдарламаның сипаттау бөлімі

10.3 Turbo Pascal тілінің операторлар бөлігі

10.3.1 Меншіктеу операторы

10.3.2 Мәліметтерді енгізу және шығару операторлары

10.3.3 Шартсыз көшу операторы

10.3.4 Құрама операторлар

10.3.5 Шартты көшу операторы

10.3.6 Таңдау операторы

10.3.7 Қайталану операторлары (For, While, Repeat)

Қазіргі кезде программалау тілінің түрлері көп. Солардың ішіндегі ең танымалысы Паскаль программалау тілі. Өйткені компьютерлік сауаттылық пен программалауды алғашқы кезеңде үйретуге ең қолайлы тіл. Паскаль тілі алгоритмдік тіл ішіндегі кеңінен тараған тілдердің бірі болып табылады. Қарастырылатын Паскаль прораммалау тілі

Есептің қойылымы. Математикалық моделін тұрғызу. Алгоритмін құру. Есепті компьютерде шешу. Проектіні тексеру.

ЭЕМ-н көмегімен есептерді шешуге дайындау процесінде бірнеше кезеңді ерекшелеуге болады:

1. Математикалық модель құруды қамтитын есептің қойылуы (есептің математикалық қойылуы).

2. Есепті шешудің алгоритмін жазу.

3. Прорамманы құрастыру.

4. Программаны жөндеу.

5. ЭЕМ-де есепті шешу және нәтижелерді талдау.

Алғашқы деректердің, есеп шарттарының және оны шешу мақсаттарының дәл сипатталуы есептің математикалық қойылуы деп аталады.

Есеп қойылғанда алғашқы және қажетті деректерді анықтап, атап өту керек, қандай нәтижелер және қандай жағдайда алу қажет екенін айқын түсіну керек. Зерттеу объектісі математикалық терминдерде сипатталады, алғашқы деректер мен қажетті нәтижелер арасындағы байланыс белгіленеді (көбіне формулалардың көмегімен), бұл нақты объектіні зерттеу математикалық зерттеуде түйістіреді.

Егер есеп математикалық болса, формулалары, шешу әдісі, математикалық қойылуы айқын болса, бұл кезеңнің орындалуы қажет болмауы да мүмкін. Іс жүзінде ЭЕМ айтарлықтай күрделі есептерді шешу үшін жиі қолданылады, сондықтан есептердің қойылу кезеңі едәуір күш салуды, уақытты талап етуі мүмкін. Мұндай есептердің мысалы оқу кестесін құрастыру, өнеркәсіп өндірісінің дамуы нәтижесінде ауа райы өзгеруін болжау, табиғи ресурстарды тиімді пайдалану, экология мәселелері бола алады. Бұл жерде математикалық символиканы барлық уақытта пайдалану шарт емес, бірақ ойды математикаға тән қатандықпен жеткізу қажет. Есептің дәл және нақты қойылуы – есепті жартылай шешу деген сөз.

10.1 Бағдарлама құрылымы

Turbo Pascal тілінде бағдарлама екі бөліктен тұрады:

1) мәліметтерді сипаттау (бейнелеу);

2) алгоритмдік амалдарды сипаттау немесе операторлық бөлік.

Мәліметтер сипаттау арқылы жазылса, амалдар оператор арқылы жазылады.

Синтаксистік жағынан Turbo Pascal тілінде жазылған бағдарламаны екі бөлікке бөлуге болады:

- а) бағдарлама тақырыбы;
- ә) блок.

Бағдарламаның тақырыбында: бағдарламаның аты және пайдаланылатын стандартты файлдардың аттары беріледі. Блокта: бейнелеулер және операторлар беріледі.

Бағдарламаның тақырыбын көрсету үшін PROGRAM қызмет сөзі пайдаланылады. Бағдарламаның тақырыбы жалпы түрде былай жазылады:

PROGRAM <программаның аты>;

Бағдарламаның тақырыбын “;” символымен аяқтайды.

Бағдарламаға (операторларға) түсініктеме беру үшін түсініктемені “(” және “)” белгілерінің арасына кез келген жолдан, немесе кез келген қадамнан бастап жазуға болады. Мысалы: (*үшбұрыштың ауданы*)

Блок: 1. Белгілерді сипаттау бөлігі.

2. Тұрақтыларды сипаттау бөлігі.

3. Типтерді анықтау бөлігі.

4. Айнымалыларды сипаттау бөлігі.

5. Функцияларды және процедураларды сипаттау бөлігі.

6. Операторлар бөлігі.

Turbo Pascal тілінде бағдарламаның жалпы құрылымын былай көрсетуге болады:

PROGRAM аты (пайдаланатын файлдар аттарының тізімі);
сипаттау бөлігі

BEGIN

оператор бөлігі

END.

Сонымен, Turbo Pascal-да жазылған бағдарламаның жалпы құрылымы:

PROGRAM ... ; {бағдарлама тақырыбы}

Uses ... ; {стандартты және пайдаланушылар модульдерін қосу}

Label ... ; {белгілерді сипаттау бөлімі}

Const ... ; {тұрақтыларды сипаттау бөлімі}

Type ... ; {жаңа типтерді сипаттау бөлімі }

Var ... ; {айнымалыларды сипаттау бөлімі }

Procedure ... ; {процедураларды сипаттау бөлімі}

Function ... ; {функцияларды сипаттау бөлімі }

BEGIN {негізгі бағдарлама басы}

1 - оператор;

2 - оператор;

.....

n – оператор

END.

Бұл бөлімдер бір-бірінен нүктелі үтір арқылы ажыратылады. Бағдарламада бұл көрсетілген бөліктердің бәрі бірдей бар болуы шарт емес.

10.2 Бағдарламаның сипаттау бөлімі

Белгілерді сипаттау

Белгі ретінде кез келген бүтін оң сан, символ, символдар тіркесі пайдаланылады. Белгі операторды немесе бағдарламаның бөлігін табу үшін пайдаланылады. Белгі оператор алдында орналасады да, қос нүкте арқылы ажыратылады. Қажетті операторға, бағдарламаның бөлігіне оралу белгі арқылы жүзеге асырылады. Белгіні пайдалану үшін ол алдын ала белгі бөлігінде LABEL (белгі) қызмет сөз арқылы хабарлануы қажет. Белгі ретінде таңбасыз бүтін сандар және атаулар пайдаланылады.

Белгіні хабарлау түрі төмендегідей: **Label** сан, сан, ... сан, символ;

Тұрақтыларды сипаттау

Тұрақты (константа) деп бағдарламаның барысында мәндері өзгеріссіз қалатын шамаларға айтылады. Тұрақтыға өзіміз бағдарламаның орындалу барысында бірден сандық мән берсек те, сипаттау бөлімінде атау түрінде бейнелеп алып, мән берсек те болады.

Тұрақтыны бейнелеу түрі: **const** тұрақты атауы = мәні;

Тұрақтылар integer, real, boolean, char түріндегі мәндерді қабылдай алады.

Бір бөлімде бірнеше тұрақты жазуға болады, бірақ олар “;” арқылы ажыратылып жазылуы қажет.

Сандық тұрақтылардың мәні нақты және бүтін сандар болады. Символдық тұрақтылар апостроф ішіндегі таңбалармен жазылады. Мысалы, ‘B’, ‘S’, т.с.с. Логикалық түрдегі тұрақтылар – TRUE (ақиқат) немесе FALSE (жалған) мәндерінің біреуіне ие болады. Мысалы:

const K = 23; (*бүтін тип *)

V = 1.8E – 3; (*нақты тип*)

P = 3.14; (*нақты тип*)

S = ‘F’; (*символдық тип*)

L = TRUE; (*логикалық тип*)

Айнымалыларды сипаттау

Бағдарламаның орындалу барысында әр түрлі мәндерді қабылдай алатын шамалар - айнымалылар деп аталады. Олар атаулармен белгіленіп, әрбір уақытта белгілі бір мәнге ие болады. Бағдарламада пайдаланылатын кез келген айнымалы, бағдарламаның айнымалылар бөлігінде сипатталуы қажет. Айнымалыларды сипаттау VAR (variable – айнымалы) қызмет сөзінен басталады. Ол компьютердің айнымалы мәндері енгізілетін ұяшықтарды типіне сәйкес белгілеп, сақтап қоюы үшін қажет.

Айнымалыны сипаттау түрі: **Var** айнымалы атауы: тип;

Мысалы, VAR K: INTEGER; (*K – айнымалысының типі бүтін*).

Бір мезгілде бірнеше айнымалыны сипаттауға болады.

Var a1, a2: integer; a3: byte; a4: real; sm: char

Мұндағы, нүктелі үтір (;) – бір типті айнымалылар сипатталған соң қойылатын символ.

Turbo Pascal-да INTEGER типті айнымалыны REAL типі арқылы сипаттауға да болады. Керісінше, нақты типті айнымалыны INTEGER типі арқылы сипаттауға болмайды. Ал мәндері INTEGER типті болатын айнымалыларды бір-біріне бөлуде шығатын нәтиже REAL типі арқылы сипатталуы тиіс.

Айнымалылар индексті болуы да мүмкін. Индекспен берілген айнымалылар массив деп аталады.

10.3 Turbo Pascal тілінің операторлар бөлігі

Берілген есепті шешу мақсатында орындалатын іс-әрекеттердің машинаға түсінікті тілде жазылуын операторлар дейді. Операторлар бөлімі бағдарлама тақырыбынан соң жазылатын міндетті бөлім. Ол бағдарлама денесі делінеді, яғни бағдарламаның орындалу бөлімі. Ол BEGIN-нен басталып, END-пен аяқталады да, олардың аралығына нүктелі үтір арқылы ажыратылған орындалушы операторлар жазылады. END-тің соңына нүкте қойылуы тиіс. Ол бағдарламаның аяқталу белгісі. Егер бағдарламаға айнымалылар енгізілген болса ғана, VAR бөлімі оларды сипаттау үшін қажет. Түсінуге ыңғайлы болуы үшін бір қатарға көбінесе бір оператор жазылады. Экран бетіне көбірек мәлімет орналастыру кезінде немесе қағазға бағдарлама мәтінін басып аларда орын үнемдеу үшін бір қатарға бірнеше оператор жазуға болады. Қатарлар арасына не соңына жүйелі жақшаға { } немесе (* *) түсініктеме сөздер жазып отыру керек. Оларды қазақша немесе орысша әріптермен жазуға болады.

Бағдарламаға айнымалылар және операторлар бөлімдерінен басқа бөлімдердің енгізілуі міндетті емес. Олар қажеттілігіне қарай енгізілуі тиіс.

Turbo Pascal тілінің операторларын қарапайым және күрделі операторлар деп екі топқа бөледі.

Қарапайым операторлар тек бір оператордан тұрады, құрамына басқа оператор енбейді. Қарапайым операторларға: меншіктеу, көшіру, бос, енгізу және шығару операторлары жатады.

Ал күрделі операторлардың ішіне бірнеше қарапайым операторлар еніп тұруы мүмкін. Күрделі операторларға: шартты көшіру, қайталану (цикл), таңдау, жалғастыру және т.б. операторлар жатады. Бұл оператор тұтас бір оператор қызметін атқарады. Бағдарламада операторлар орналасқан реті бойынша орындалады.

Операторлар орындайтын қызметтеріне байланысты мынадай топтарға бөлінеді:

- меншіктеу операторы;
- енгізу-шығару операторлары;
- басқару операторлары;
- функция мен процедураларды анықтау операторлары.

Енді осы келтірілген операторлардың жазылу жолдарын, олардың ерекшеліктерін қарастырамыз.

10.3.1 Меншіктеу операторы

Turbo Pascal-дың негізгі операторының бірі – меншіктеу операторы. «Мән меншіктең деген бұйрықты орындайтын операторды меншіктеу операторы дейді. Ол айнымалының мәндерін өзгерту үшін пайдаланылады. Меншіктеу операторының жазылу түрі:

<айнымалы атауы> := <арифметикалық өрнек>

мұндағы: “ := ” – меншіктеу белгісі.

Меншіктеу операторы орындалғанда өрнектің мәні есептелініп, табылған мән айнымалы арқылы сақтау үшін бөлінген ЭЕМ-нің жедел жадысының ұяшығына жіберіледі. Ал меншіктеу операторының оң жағындағы айнымалы міндетті түрде алдын-ала анықталған, яғни сандық мәндерді қабылдаған болуы керек.

Жалпы жағдайда арифметикалық өрнек: тұрақты, айнымалы атауы, функция, өрнек түрлерінің бірінде берілуі мүмкін.

Меншіктеу операторы тек арифметикалық өрнектер үшін ғана емес, логикалық және символдық берілгендер үшін де пайдаланылады.

Мысалы, $A := B \text{ and } C$, егер мұнда B – ақиқат, C – жалған болса, онда A жалған мән қабылдайды, A – ақиқат мән қабылдау үшін B және C ақиқат болуы қажет.

Символдық мән әрқашанда апостроф “ ’ ” ішіне алынып жазылады. Мысалы, $A := \text{“B”}$; $A1 := \text{“7”}$;

Меншіктеу операторын пайдаланғанда айнымалы атауының типі мен қабылдайтын мәнінің типі бірдей болуы қажет. Мысалы, егер $\text{VAR } A: \text{REAL}$ болып, оператор былай $A := '7'$ жазылса, онда ол дұрыс болмайды, себебі A -ның типі нақты, ал мән символдық.

Кейде нақты түрдегі айнымалыға бүтін сан мәнін меншіктеуге болады. Онда бүтін сан нақты санға айналып кетеді.

10.3.2 Мәліметтерді енгізу және шығару операторлары

Берілген мәндерді айнымалыға жазу үшін меншіктеу операторын пайдалануға болады. Мысалы, $A := 15$; $B := 20$ т.с.с. Бірақ бұл жағдайда бағдарлама әмбебап болмайды. Бағдарламаны әмбебап етіп жасау үшін айнымалылардың мәндері өзгертін түрде жасау қажет. Ол үшін енгізу операторы READ (READ – ағылшынша оқу дегенді білдіреді) пайдаланылады.

Берілгендерді енгізу – бағдарламадағы алғашқы берілгендерді компьютердің жедел жадысына өңдеуге жіберу болып табылады.

Енгізу операторының жалпы жазылу түрі: READ (a1, a2, ..., an); мұндағы, a1, a2, ..., an – мәні пернетақтадан енгізілуге тиісті айнымалылар тізімі. Айнымалылар тізімінде үтір арқылы айнымалылар жазылады, ал олардың мәндерін пернетақтадан енгізу көзделеді. Тізімдегі айнымалылар санынан енгізілген мәндер саны кем болмауы керек.

Енгізілген айнымалылар мәні типіне сәйкес болуы қажет. Бүтін айнымалыға бүтін, нақты айнымалыларға нақты мәндер берілуі қажет.

Айнымалы бейнеленгенде нақты ретінде бейнеленсе, оған нақты да, бүтін де мән беруге болады, себебі нақты сан құрамына бүтін сан енеді. Айнымалылардың сандық мәндері бір-бірінен бос орын арқылы ажыратылып жазылады. Сандар арасына бір немесе бірнеше бос орын қалдыруға болады, себебі сан енгізілгенде бос орын еске алынбайды. Тізімде көрсетілген айнымалылардың сандық мәнін пернетақтадан енгізіп болғаннан кейін, Enter (енгізу) пернесі басылады.

Мысалы, енгізу операторы READ(A,B,C); түрінде берілсе, параметрлерге мәндер былай беріледі: 7 _ 9 _ 8.5. Бұл жағдайда A=7, B=9, C=8.5 мәндер қабылдайды.

Есептеулер нәтижесін жауапқа шығару үшін WRITE операторы қолданылады (WRITE – ағылшынша жазу дегенді білдіреді). Шығару операторының жалпы жазылу түрі: WRITE (a1, a2, ..., an); мұндағы, a1, a2, ..., an – сандық мәні экранға шығарылатын айнымалылар тізімі. Шығарылатын параметрлер бір-бірімен үтір арқылы ажыратылып, WRITE қызмет сөзінен соң, дөңгелек жақша ішіне жазылады.

Кейде READ, WRITE операторларының соңына LN (line – жол) жалғауын қосып READLN, WRITELN түрінде жазу қажет.

Мысалы, READLN (a, b);
WRITELN (x:6:3);

Мұндай операторлар орындалғанда курсор экранда өзінің тұрған орнынан келесі жолдың басына көшеді де, кезекті оператордың орындалуы экранда осы жерден бастап көріне бастайды. Егер келесі жолды бос күйінде тастап кету қажет болса бағдарламаға соңғы орындалатын LN жалғауы оператордан соң параметрсіз WRITELN операторын енгізсе болғаны.

10.3.3 Шартсыз көшу операторы

Turbo Pascal тілінде негізінен операторлар ретімен орындалады. Олардың бұл орындау тәртібін өзгерту үшін көшіру операторы **GO TO** пайдаланылады. Шартсыз көшіру операторының жазылу түрі: GO TO <белгі>

Бұл оператор бойынша, белгісі көрсетілген операторға көшіріледі. Белгі ретінде таңбасы 1-9999 дейінгі бүтін сан немесе символ пайдаланылады. Әдетте, шартсыз көшіру операторы бағдарламаның кейбір бөліктерін орындамай өтіп кету үшін пайдаланылады.

10.3.4 Құрама операторлар

Бірнеше операторлардың бірігуінен шыққан операторды құрама оператор дейді. Бұл операторлар begin (басы) және end (соңы) қызмет сөздері арасында жазылады. Құрама оператордың жазылу түрі:

BEGIN
1-оператор;
2-оператор;
.....

n-1 - оператор;

n-оператор

END;

Құрама оператор бүтін бір оператор қызметін атқарады, сондықтан оны бағдарламаның кез келген бөлігіне қоюға болады. Құрама оператордың құрамында өзінің құрама операторы да болуы мүмкін.

BEGIN-нің соңына, END-нің алдындағы операторға нүктелі үтір ";" қойылмайды.

Құрама операторларға шартты көшіру, таңдау және қайталау операторлары жатады.

10.3.5 Шартты көшу операторы

Тармақталуы бар процестерді ұйымдастыру үшін шартты оператор пайдаланылады. Тармақталу белгілі бір шарттың орындалуы немесе орындалмауына тәуелді басқарылады.

Шартты көшіру операторының жазылу түрі:

IF <шарт> THEN <1-оператор> ELSE <2-оператор>

мұндағы: 1 және 2 операторлардың өзі қарапайым немесе құрама оператор болуы мүмкін (IF – егер; THEN – онда; ELSE - әйтпесе).

Егер, IF сөзінен кейін жазылған шарт сақталса (ақиқат болса), онда THEN сөзінен кейін жазылған 1-оператор орындалады, 2-оператор орындалмайды. Шарт сақталмаса ELSE сөзінен кейінгі 2-оператор орындалады.

IF сөзінен соң жазылатын шартты жазу үшін Turbo Pascal тілінде өрнектерді салыстыру: =, >, >=, <, <=, <> белгілерін пайдаланады.

Бір мезгілде бір емес, бірнеше шартты, яғни құрама шартты жазу үшін логикалық амалдарды пайдалануға болады.

Шартты операторларда құрама операторды пайдалануға болады:

IF логикалық өрнек THEN

BEGIN

1-оператор;

2-оператор;

.....

n-1- оператор;

n- оператор;

END

ELSE

BEGIN

1-оператор;

2-оператор;

.....

n-1- оператор;

n- оператор;

END.

Turbo Pascal тілінде де шартты оператордың толық емес түрін пайдалануға болады.

IF шарт (логикалық өрнек) THEN оператор

10.3.6 Таңдау операторы

Берілген өрнектің мәніне тәуелді бір немесе бірнеше оператор орындалатын жағдайда таңдау операторы CASE пайдалынады. Оның Turbo Pascal-дағы құрылымы мынадай:

CASE m OF

1-тұрақты: 1-оператор;

2-тұрақты: 2-оператор;

.....

n-тұрақты: n-оператор

END;

CASE - жағдай; OF - солар, төмендегілер;

m – REAL типінен басқа кез келген скалярлық типте берілген айнымалы не өрнек.

Бағдарламаның орындалуы кезінде алдымен m айнымалысының мәні жадқа енгізіліп, ол осы тұрақтылармен салыстырылады. Егер мән олардың бірімен тең болып шықса, онда осы тұрақтыға сәйкес оператор орындалады да, CASE операторын орындау аяқталады. Егер m-нің мәні сұрыптаушы тұрақтылардың ешқайсысымен тең болып шықпаса, CASE операторы орындалмай тастап кетіледі, басқару таңдау операторынан кейінгі операторға беріледі. Оны жүйенің хабарлауы үшін CASE операторын аяқтайтын end-тің алдына else writeln ('Қате. Мұндай сұрыптаушы табылған жоқ') операторын енгізіп қоюға болады.

10.3.7 Қайталану операторлары (For, While, Repeat)

Берілген есепті шешуде алгоритмнің кейбір бөліктері немесе алгоритмдік тілдің бірнеше оператор тобы бірнеше рет қайталанып орындалуы мүмкін. Осындай процестерді ұйымдастыру үшін қайталану операторлары пайдалынады. Turbo Pascal тілінде қайталану процесін үш түрлі жолмен ұйымдастыруға болады:

1. Параметрдің мәні бойынша қайталауды ұйымдастыру.

2. Алдын ала берілген шарт бойынша қайталауды ұйымдастыру;

3. Келесі шарт бойынша қайталауды ұйымдастыру;

Бұл қайталаудың қай-қайсысын алсақ та, олар қайталану операторының көмегімен жүзеге асырылады. Қайталану операторлары құрама операторлардың қатарына жатады.

Параметрдің мәні бойынша қайталауды ұйымдастыру (For)

Параметрдің мәні бойынша қайталауды ұйымдастыру (For) цикл параметрі өзінің алғашқы мәнінен соңғы мәніне дейін бірлік қадаммен өзгеру барысында

циклге кіретін бір немесе бірнеше операторларды қайталап орындау қызметін атқарады.

Цикл барысында оның параметрінің мәні біртіндеп өсетін болса, TO түйінді сөзі қолданылады. Тек бір оператор қайталанылатын кездегі оның жазылу түрі:

```
FOR i:=m1 TO m2 DO <оператор>;
```

мұндағы FOR - үшін, TO - дейін, DO - орындау деген мағынаны беретін қызмет сөздері; i - циклдың параметрі. Ол міндетті түрде бүтін типтегі айнымалы болуы қажет; m1, m2 - цикл параметрінің бастапқы және соңғы мәндері, олар бүтін сандар болуы немесе бүтін мән беретін арифметикалық өрнектер болуы мүмкін. $m1 < m2$ болғанда - <оператор> бірнеше рет қайталанып орындалады.

Егер циклде бір ғана емес бірнеше операторлардан тұратын құрмалас оператор қайталанатын болса, онда операторлардың жазылу түріне BEGIN, END түйінді сөздері кіреді.

```
FOR i:=m1 TO m2 DO
```

```
  BEGIN
```

```
    <1 - оператор>;
```

```
    <2 - оператор>;
```

```
    .....
```

```
    <n - оператор>;
```

```
  END;
```

Егер $m1 > m2$ болса, яғни қадамы «-1» болып келетін кезде FOR операторының жазылу түрі өзгешелеу болады:

```
FOR i:=m1 downto m2 DO <оператор>;
```

мұндағы: downto – түйінді сөз «кері қарай» болып аударылады. BEGIN және END түйінді сөздерімен қоршалған құрмалас оператор қайталанатын жағдайда бұл оператордың жазылуы төмендегідей болады:

```
FOR i:=m1 downto m2 DO
```

```
  BEGIN
```

```
    <1 - оператор>;
```

```
    <2 - оператор>;
```

```
    .....
```

```
    <n - оператор>;
```

```
  END;
```

Оператордың орындалу тәртібі:

а) параметрдің мәні m2-мен салыстырылады, егер ол m2-ден кіші болса, онда циклдің денесі бір рет қайталанылады. Ал ол m2-ден үлкен болса, онда басқару цикл денесінің соңындағы операторға беріледі де, қайталану аяқталады;

ә) әр қайталанғанда, параметрдің алғашқы мәніне бір қосыла отырып, қашан параметрдің ағымдағы мәні m2-ден артық болғанша циклдың денесі қайталана береді.

Егер m1 және m2 арифметикалық өрнектер болса, онда олардың мәндері циклдың орындалу алдында бір-ақ рет орындалады.

Алдын ала берілген шарт бойынша қайталану операторы (WHILE)

Алдын ала берілген шарт бойынша қайталану көп жағдайларда қайталаудың саны белгісіз болған жағдайда пайдаланылады. Бұл оператор WHILE - әзір және DO - орындау қызмет сөздерінің көмегі арқылы орындалады. Қайталанушы процесс қойылған шартты тексеру арқылы жүзеге асады. Яғни, шарт ақиқат болса цикл қайталанса да, шарт жалған болса циклдан шығу орындалады.

Алдын ала берілген шарт бойынша қайталану операторының (қайталану денесі бір ғана оператордан тұрса, онда оны операторлық жақшасыз жазуға болады) жалпы жазылу түрі:

WHILE <шарт> DO <цикл денесі>;

Мұнда шарт орнында логикалық өрнек тұруы да мүмкін. Цикл денесі – берілген шартқа тәуелді бірнеше рет қайталанып орындалатын оператор.

Бұл оператор былай орындалады: алдымен шарт тексеріледі, егер шарт орындалса, онда циклдің денесі орындалады, одан соң және шарт тексеріледі, және тағы сол сияқты, бұл процесс қашан шарт сақталудан қалғанша қайталанылады. Егер шарт сақталмаса, басқару операторлық жақшаның (END) соңында тұрған операторға көшіріледі, цикл денесі бірден-бір рет орындалмайды. Басқаша айтқанда, логикалық өрнектің мәні қашан жалған болғанша, қайталаудың денесі қайталап орындала береді.

Егер қайталанудың денесі бірнеше оператордан құралса, онда олар BEGIN-END қызмет сөздерінің арасына алынып жазылады.

Келесі шарт бойынша қайталау операторы (REPEAT)

Бұл оператор көп жағдайларда қайталаудың саны белгісіз болған жағдайда пайдаланылады. Оның жазылу түрі:

REPEAT

 қайталау денесі

UNTIL логикалық өрнек

Мұндағы REPEAT - қайтала, UNTIL - соған дейін қызмет сөздері.

Орындалуы: қашан логикалық өрнектің мәні ақиқат болғанша қайталану денесі қайталанса береді. Бұл қайталануда алдымен қайталау денесі орындалады да, одан соң шарт тексеріледі. Оператордың соңы UNTIL қызмет сөзімен бітетін болғандықтан, мұнда операторлық ашылатын және жабылатын жақшалар орнына жазылатын BEGIN және END қызмет сөздері жазылмайды. Егер қайталану денесінде көшіру операторы болып, ол қайталану сыртындағы оператордың белгісіне басқаруды берсе, онда қайталанудың орындалуы ары қарай жалғаспайды.

Күрделі қайталаулар

Күрделі қайталанушы процесстерді ұйымдастыруда бір цикл операторы құрамында екінші бір цикл операторы болуы мүмкін. Бұл жағдайда бірінші цикл операторы - сыртқы цикл, ал оның құрамындағы екінші оператор - ішкі цикл деп аталады. Сыртқы және ішкі циклді ұйымдастыруда мына шарт орындалады: ішкі циклдің барлық операторы сыртқы цикл денесіне толық енеді.

Дәріс №11. Массив. Екі өлшемді массивтер

Дәріс жоспары:

11.1 Массив

11.2 Екі өлшемді массивтер

11.1 Массив

Массив - бір атауға біріктірілген айнымалылардың реттелген тізбегі. Массивпен жұмыс істеу үшін оның әр элементінің мәні белгілі болуы тиіс. Массив құрылымдық типке жатады. Құрылымдық типтегі айнымалылардың стандартты типтегі айнымалылардан айырмашылығы мынау: олар бірнеше элементтерден құралады, ал стандартты типтер бір ғана элементтен құралады. Массивтің символдық бүтін және нақты сандардан тұратын сандық және символдық типтері бар. Мысалы, сандық кесте, вектор координаттары, матрица (төртбұрышты сандық кесте), әрқайсысы бір элемент түрінде алынған символдар тізбектерінің жиынтығы массивтер қатарына жатады (символдық массив элементіне енетін символдар саны 255-тен аспауы тиіс). Массивтің атауы да сандық айнымалының атауы сияқты белгіленеді.

Массив элементін пайдаланғанда оның атынан кейін тік жақшалардың ішінде индексі (индекс айнымалы, өрнек болуы да мүмкін) көрсетілуі тиіс:

Элемент индексі	1	2	3	4	5	6	7	8	9	10
Массив элементі	A[1]	A[2]	A[3]	A[4]	A[5]	A[6]	A[7]	A[8]	A[9]	A[10]

Егер бағдарламада массив пайдаланылатын болса, онда ол айнымалы бөлігінде (Var) немесе тип бөлігінде (Type) сипатталуы қажет:

Var <массив атауы>: <массив типінің атауы>;

Type <массив типінің атауы>= **array** [индекс типі] **of** элемент типі;

мұндағы: Массив типінің атауы - массив элементтерінің жиынын сипаттайды;

Индекс типі - тізбектелген немесе шектелген типтерді көрсету;

Элемент типі - массив элементтерінің типін көрсету.

Мысалы:

VAR A : ARRAY[1..5] OF REAL

Массив атауы 5 элементті массив Нақты тип

Массивтерді дұрыс жазуға мысалдар келтірейік:

VAR

MAS:ARRAY[1..N] OF REAL;

Жыл: ARRAY[қаңтар..желтоқсан] OF INTEGER;

L:ARRAY[жол] OF BOOLEAN;

M1:ARRAY[CHAR] OF шеңбер;

Егер бірнеше массив бірдей базалық және индекстік типте болса, онда оларды сипаттағанда бір массив ретінде үтір арқылы бөліп жазуға болады. Мысалы:

VAR A,B,C: ARRAY[1..10] of REAL;

Массивтің кез-келген бір элементіне нәтиже беру үшін, меншіктеу операторы қолданылады.

Массив атауы [индекс]:= нәтиже

Мысалы:

а) A массивінің бірінші элементіне 10 санын меншіктеу үшін - A[1]:=10;

ә) B массивінің бесінші элементіне 20 санын меншіктеу үшін - B[5]:=20;

б) B массивінің үшінші элемент нәтижесін экранға шығару үшін - writeln (B[3]);

в) A массивінің бесінші элементіне нәтижені пернелер тақтасы көмегімен енгізу үшін - readln (A[5]);

г) B массивінің бесінші және оныншы элементтерінің қосындысын Sum айнымалысына меншіктеу үшін - Sum:= B[5]+ A[10];

Бағдарламаның операторлық бөлігінде массив элементтеріне мынадай амалдар қолданылады:

A[3]:=C[4]+5;

S:=S+X[K];

P:=X[3*I+1]; және т.с.с.

Массив элементтерін енгізу немесе шығару үшін бағдарламаның мына үзінділерін пайдалануға болады:

а) енгізу:

FOR i:=1 to n do

read(A[I]);

б) шығару:

for i:=1 to n do

write(A[I], ' ');

11.2 Екі өлшемді массивтер

Егер массив атауында бір ғана индекс болса, онда ол массивті бір өлшемді, ал екі индекс болса - екі өлшемді және т.с.с. n индекс болса, n өлшемді массив дейді. Бір өлшемді массив вектор элементтері, ал екі өлшемді массив (n жолдан, m бағаннан тұратын) матрица деп аталады.

Екі өлшемді массивті Var бөлімінде сипаттаудың жалпы түрі:

Var <массив атауы>: array[1..n, 1..m] of элемент типі;

Екі өлшемді массивті Type бөлімінде сипаттаудың жалпы түрі:

Type <массив типінің атауы>= array [1..n, 1..m] of элемент типі;

мұндағы: 1..n – массив жолының (қатарының) алғашқы және соңғы мәні; 1.. m – массив бағанының алғашқы және соңғы мәні.

Мысалы, Var бөлімінде екі өлшемді массивті сипаттау:

Var

A: array [1 ..n, 1.. m] of integer;

Түр бөлімінде екі өлшемді массивті сипаттау:

Түр

GR= array [1..n, 1..m] of integer;

Матрица элементтерін енгізу, оларды қалпын сақтап экранға шығару циклдерінің жазылу түрлері:

- 1) for k:=1 to n do
for j:=1 to m do
read (a[k, j]);
- 2) for k:=1 to n do
begin writeln;
for j:=1 to m do
write (a[k, j] : p : q) end.

Матрицаны экранға қалпын сақтап шығару үшін сыртқы циклге қатысты операторлар (ішкі цикл мен writeln) бір құрама оператор түрінде қарастырылуы тиіс.

Дәріс №12. Алгоритмдердің тиімділігі мен күрделілігіне анализ жасау

Алгоритмдер математикалық логика – алгоритмдер теориясына жалғасатын ғылыми пәндер математика мен информатика арасындағы шектеулерді жүйелі зерделеу объектісі болып табылады.

Жағдайдың ерекшелігі мынада, ЭЕМ-да іске асыру үшін алгоритмдер әзірлеуге ұйғарылған практикалық міндеттерді шешу кезінде және сонымен қатар практикада ақпараттық технологияларды пайдалану кезінде, негізінен бұл ұғымның жоғары формальдылығына сүйенбеген жөн. Сондықтан алгоритм ұғымының мәнін мазмұнды түсіндіру және оның негізгі қасиеттерін қарастыру негізінде алгоритмдермен және алгоритмдеумен танысқан дұрыс болады. Мұндай тәсіл кезінде алгоритмдеу берілген тілдік құралдар шеңберінде белгілі бір практикалық тәсілдердің, ұтымды ойлаудың ерекше өзіндік дағдыларының жиынтығы ретінде алға шығады. Ақпаратты өлшеуде мұндай жағдай мен жоғарыда қарастырылған тәсілдің арасында ұқсастықты жүргізуге болады: “кибернетикалық” тәсіл кезіндегі жіңішке математикалық құрулар компьютермен практикалық жұмыс кезінде барынша қарапайым “көлемді” тәсілді қолдану кезінде аса қажет емес.

“Алгоритм” сөзінің өзі IX ғасырдың ұлы математигі Әл-Хорезмидің атының латын тіліндегі algorithmi жазылуынан шыққан, ол арифметикалық амалдарды орындау ережесін қалыптастырған. Бастапқыда алгоритм ретінде көп мәнді санды төрт арифметикалық амалды орындау ережесі арқылы түсінді. Мысалы, көпразрядты екі бүтін санды қосу үшін мынадай ережені орындау керек:

- 1) екі көпразрядты бүтін сандардың мәнін анықтау;
- 2) осы екі санды бірінің астына бірін разрядтарын сәйкестендіріп жазу;

3) егер осы сандардың біреуінің үлкен разрядтары жетіспесе, оны нөлдермен толтыру;

4) ең кіші разрядқа қосу амалын орындау және келесі разрядты қарастыруға көшу;

Бұл жағдайда, келесі үлкен разрядқа өтетін бірлік пайда болса, онда оны еске сақтап қою керек;

5) барлық разрядтар біткенше оңнан жылжи қарастырып, есте сақталған бірлікті ескеріп және пайда болған жаңа үлкен разрядқа өтетін бірлікті қайтадан еске сақтай отырып әрбір разрядқа қосу амалын орындау;

6) нәтиже ретінде барлық разрядтарда қосу амалы орындалғаннан кейін, шыққан санды аламыз және есте сақталған бірлік болса, онда оны нәтиженің ең үлкен разрядының мәні ретінде есептейміз.

Бұл жазылған ережемен екі көпразрядты бүтін санды қосу үшін орындаушының қазақ тілін білуі және бір разрядты сандарды қосатын қабілеті қажет.

Енді ондық сан деген ұғымды білетін, екі көп разрядты бүтін санды көбейте алатын және қазақ тілін түсінетін қабілеті бар деп есептеп, кез келген екі ондық санды көбейтуге болатын ережені құрастырайық.

1. Екі ондық санның мәндерін анықтау;

2. Осы сандардың ондық белгісін көрсететін үтірді ескермей, оларды бірінің астына бірін бүтін көбейткіштер ретінде жазу;

3. Көбейткіштерді көбейтіп, көбейтіндіні табу;

4. Берілген екі ондық сандардағы үтірдің оң жағындағы таңбалардың санын қосу;

5. Нәтиже ретінде үтірден кейінгі таңбалардың саны 4-қадамнан шыққан санға тең болатындай етіп жазылған 3-қадамнан шыққан көбейтіндіні аламыз.

Осындай арифметикалық амалдарды орындау ережелерін алгоритм деп түсінсе, кейін келе “алгоритм” деген сөз түрлі математикалық есептерді шешу ережелерін белгілеуге қолданыла бастады.

Мысалы, грек ғалымы Евклид құрастырған екі натурал санның ең үлкен ортақ бөлгішін табатын төмендегі ережені *Евклид алгоритмі* деп атайды.

1) екі натурал санды алып, олардың мәндерін анықтау;

2) егер екі сан өзара тең болса, онда нәтиже ретінде осы сандардың кез келгенін алу керек немесе осы сандардың үлкенін анықтау қажет;

3) үлкен санды үлкен сан мен кіші санның айырмасына алмастыру керек;

4) 2-қадамнан бастап, қайталау.

“Алгоритм” ұғымын нақтылаудың бірнеше тәсілдері белгілі:

- ақырлы және ақырсыз автоматтар теориясы;

- есептелетін (рекурсивті) функциялар теориясы;

- Черчтің λ есептеуі.

Алгоритмнің асимптотикалық күрделілігі

Алгоритмге анализ жасай отырып, қойылған есепті берілген алгоритммен шешу қанша уақыт алатынын есептеуге болады. Әрбір қарастырылатын алгоритм үшін ұзындығы N болатын кіріс деректері массивінде берілген есеп

қаншалықты жылдам шешілетінін бағалай аламыз. Мысалы біз N шамадан тұратын тізімді өсу реті бойынша сұрыптау алгоритмі неше салыстыруды қажет ететіндігін бағалай аламыз, немесе өлшемі NxN екі матрицаны көбейту үшін неше арифметикалық амал қажет екендігін есептей аламыз. Мысалы, төрт шаманың үлкенін табудың екі алгоритмін қарастырамыз:

```

1. l=a
   if b>l then
       l=b
   end if
   return a
if c>l then
    l=c
end if
if d>l then
    l=d
end if
return l.
else
2. if a>b then
    if a>c then
        if a>d then
            return a
        else
            return d
        end if
    if c> d then
        return c
    else
        return d
    end if
end if
if b>c then
    if b>d then
        return b
    else
        return d
    end if
else
    if c>d then
        return c
    else
        return d
    end if
end if
end if

```

Бұл алгоритмдердің әрбіреуінде үш салыстыру жасалады. Бірінші алгоритмді оқу жеңіл және түсінікті, бірақ компьютерде орындалуда олардың күрделілік деңгейі. Уақыты жағынан алғанда бұл екі алгоритм бірдей, бірақ екінші алгоритм көбірек жазуы қажет. Егер сандар немесе символдар салыстырылса, бұның маңызы ерекше болмайды, ал басқа типті деректермен жұмыс істегенде оның маңызы өте зор болуы мүмкін. Қазіргі кездегі бағдарламалау тілінің көпшілігінде үлкен және күрделі объектілерді немесе жазбаларды салыстыру немесе операторларын анықтауға болады. Бұл жағдайларда уақытша айнымалыларды қолдану көп орынды қажет етеді. Алгоритмдердің тиімділігіне анализ жасағанда бізді алдымен уақыт

қызықтырады, ал жады маңызды болған жағдайларда да қажет кезінде қарастырамыз.

Дәріс №13. Іздеу, таңдау, сұрыптау, рекурсивті алгоритмдер

Дәріс жоспары:

- 13.1 Массивтің элементтерін жағалап іздеу
- 13.2 Массивтің элементтерін екіге бөліп іздеу
- 13.3 Элементтерді таңдау арқылы сұрыптау
- 13.4 Элементтерді алмастыру арқылы сұрыптау
- 13.5 Массивті индекстері арқылы сұрыптау
- 13.6 Элементтерді енгізу тәсілімен сұрыптау
- 13.7 Біріктіру тәсілімен сұрыптау
- 13.8 Екі өлшемді массивтер
- 13.9 Тура қосулар көмегімен сұрыптау
- 13.10 Тура алмастыру көмегімен сұрыптау (көпіршікті әдіс)
- 13.11 Рекурсивті алгоритмдер

Тізімдегі ақпаратты іздеу теориялық программалаудың фундаменталды есептерінің бірі. Іздеу алгоритмдерін қарастырғанда программадағы деректер массивтер тізімі түрінде берілген деп есептейміз. Тізімдер сұрыпталған немесе сұрыпталмаған болуы мүмкін.

Сұрыпталмаған тізімде қажетті жазуды іздеу дегеніміз – қажетті элемент табылғанға дейін бүкіл тізімді көріп шығу. Бұл іздеудің қарапайым түрі.

Сұрыпталған тізімде – екілік іздеу жүргізуге болады.

13.1 Массивтің элементтерін жағалап іздеу

Мысалы. Mas бүтін сандар массиві берілген. Оның алғашқы және соңғы теріс элементтерінің idx_1 , idx_2 индекстерін анықтаңдар.

```
Program Prost_1;  
  const n=15;  
  var i, idx1, idx2: integer; p: boolean;  
      Mas: array[1..n] of integer;  
begin  
  Randomize;  
  For i:=1 to n do  
    begin  
      Mas[i]:=Random(10) - Random(10); Write(Mas[i]:4);  
    end;  
  Writeln; p:=true;;  
  For i:=1 to n do  
    if Mas[i]<0  
      then if p then begin idx1:=i; idx2:=i; p:=false; end
```

```

        else idx2:=i;
    if p then Writeln('Теріс сан жоқ!')
        else Writeln('Алғашқысының индексі ', idx1, ', соңғысы- ', idx2);
end.

```

Мысалы. Mas үш орынды натурал сандар массиві берілген. Оның элементтерінің ішінен толық квадрат болатындарын анықтаңдар.

```

Program Prost_2;
const n=20;
var i: integer; p: boolean;
Mas: array[1..n] of integer;
begin
    Randomize; p:=true;
    for i:=1 to n do
        begin
            Mas[i]:=Random(900)+100;
            write(Mas[i]:4);
        end;
    writeln;
    for i:=1 to n do
        if Frac(sqrt(Mas[i]))=0 { Frac(x) - санның бөлшек бөлігі }
            then begin writeln(Mas[i], ' – индексі: ', i); p:=false; end;
        if p then writeln('Толық квадрат жоқ!'); Readln;
    end.

```

13.2 Массивтің элементтерін екіге бөліп іздеу

Егер массив реттелген болса, онда оның берілген үлгіге тең элементін жағалап іздеу арқылы табу тиімсіз болып табылады. Тиімді әдістердің бірі - екіге бөліп іздеу.

Мысалы. Саны n болатын, өсуіне қарай реттелген әр түрлі элементтерден тұратын A сандар массиві берілген. Оның элементтерінің ішінде берілген ulg санына тең болатыны бар ма? Болса, оның индексін анықтаңдар.

```

Program Prost_3;
const n=17;
var A: array[1..n] of integer;
    i, min, max, ort, ulg: integer; p: boolean;
begin
    Randomize;
    For i:=1 to n do
        begin
            if i=1 then A[i]:=Random(8)+1
            else A[i]:=A[i-1]+Random(8)+1;
        end;
    write(A[i]:4);
    Writeln; Write('Үлгіні енгізіңіз: ');

```

```

Readln(ulg); min:=1; max:=n;
p:=false; Repeat
  Ort:=(min+max) div 2;
  if A[ort]=ulg
    then p:=true  else if A[ort]>ulg
      then max:=ort-1  else min:=ort+1;
  until p or (min>max);
  if p then writeln('Элемент индексі: ',ort)  else writeln ( 'Ондай элемент
жок!'); Readln;end.

```

Массивтің элементтерін белгілі бір заңдылықпен орындарын ауыстырып реттеу процесін сұрыптау дейді. Сұрыптаудың мақсаты - сұрыпталған жиындағы элементтерді іздеуді жеңілдету. Мысалы, сандар массивін өсуі, кемуі бойынша сұрыптау, жолдар массивін алфавит бойынша сұрыптау және т.б. Ақпараттық жүйелерде мәліметтерді сұрыптаудың маңызы өте зор. Сұрыптаулар жағдайында сәйкес әдістер екі класқа бөлінген: массивтерді сұрыптау және файлдарды сұрыптау. Кейде оларда ішкі және сыртқы сұрыптаулар деп атайды. Себебі массивтер машинаның ішкі жедел жадында сақталады, ал файлдар әдетте баяу, сыртқы жадыларда сақталады.

Бүгінде сұрыптаудың көптеген тәсілдері белгілі. Олардың ішінен таңдау арқылы, алмастыру арқылы, индекстері арқылы, енгізу арқылы және біріктіру арқылы сұрыптауларды ғана қарастырмақпыз.

13.3 Элементтерді таңдау арқылы сұрыптау

Мысалы. n элементтен тұратын A сандар массиві берілген. Оны таңдау әдісін қолданып элементтерінің өсуі бойынша сұрыптаңдар

```

Program Prost_4;
const n=8;
type MasType = array [1..n] of integer;
var i, idx, k: integer;      A: MasType;
Function MinMas(j: integer): integer;
  var p, min: integer;
  begin  for p:=j to n do
    if (p=j) or (A[p]<A[min]) then min:=p;
    MinMas:=min;  end;
begin  Randomize; Writeln('Берілген сандар массиві:');
For i:= 1 to n do begin  A[i]:=Random(30); Write(A[i]:4); end; For i:=1 to n-1
do
  begin  idx:=MinMas(i);
    k:=A[i]; A[i]:=A[idx]; A[idx]:=k;  end;
  Writeln; Writeln('Өсуіне қарай сұрыптау нәтижесі:');
  For i:=1 to n do Write(A[i]:4);  Readln;end.

```

Бұл программада берілген массив бөлігінің ең кіші элементінің индексі (рет нөмірін) табатын MinMas(j) функциясы пайдаланылған. Функцияның j параметрінің мәні массив бөлігінің бірінші элементінің рет нөмірін (соңғысы n) көрсетеді.

13.4 Элементтерді алмастыру арқылы сұрыптау

Мысалы. n элементтен тұратын A сандар массиві берілген. Оны алмастыру әдісін қолданып элементтерінің өсуі бойынша сұрыптаңдар.

```
Program Prost_5;
  const n= 8;
  var   A: array[1..n] of integer;
        i, j, k: integer; p:boolean;
begin  Randomize;
  Writeln('Берілген сандар массиві:');
  For i:= 1 to n do
    begin A[i]:=Random(25); Write(A[i]:4); end;
  Repeat   p:=true;
    For i:=1 to n-1 do
      if A[i]>A[i+1] then begin k:=A[i]; A[i]:=A[i+1]; A[i+1]:=k; p:=false;
end; until p;
  Writeln; Writeln('Өсуіне қарай сұрыптау нәтижесі:');
  For i:=1 to n do Write(A[i]:4); Readln; end.
```

13.5 Массивті индекстері арқылы сұрыптау

Мысалы. n элементтен тұратын A сандар массиві берілген. Массивті индекстері (индекстер массивін жасақтау) арқылы элементтерінің өсуі бойынша сұрыптаңдар.

```
Program Prost_6;
  const n= 8;
  var A, IDX: array[1..n] of integer; i, j, k: integer;
begin  Randomize;
  Writeln('Берілген сандар массиві:');
  For i:= 1 to n do   begin A[i]:=Random(40); IDX[i]:=1; Write(A[i]:4);
end; For i:=n downto 2 do
  For j:=i-1 downto 1 do
    if A[i]<A[j] then  IDX[j]:=IDX[j]+1 else IDX[i]:=IDX[i]+1; Writeln;
Writeln('Өсуіне қарай сұрыптау индекстері:'); For i:=1 to n do Write(IDX[i]:4);
  Readln;end.
```

13.6 Элементтерді енгізу тәсілімен сұрыптау

Бұл тәсілдің мәні массивтің сұрыпталмаған бөлігінен сұрыпталған бөлігіне элементтерді бір-бірлеп енгізу. Енгізілген элемент массив бөлігінің сұрыпталуын бұзбауы қажет. Ол үшін енгізілетін элемент өз орнын тапқанша, сұрыпталған бөлігінің элементтерімен орын ауыстырып отыруы тиіс.

Мысалы. n элементтен тұратын A сандар массиві берілген. Элементтерді енгізу тәсілін пайдаланып массивті элементтерінің өсуі бойынша сұрыптаңдар.

```
Program Prost_7;
  const n= 8;
  var A: array[1..n] of integer;    i, j, k: integer;
begin Randomize;
  Writeln('Берілген сандар массиві:');
  For i:= 1 to n to begin A[i]:=Random(30); Write(A[i]:4); end;
  For i:=2 to n do
    begin j:= i - 1;
      Repeat    if A[j+1]<=A[j]
        then begin k:=A[j]; A[j]:=A[j+1]; A[j+1]:=k; j:=j - 1; end else j:=0;
      until j=0;    end;
  Writeln; Writeln('Өсуіне қарай сұрыптау нәтижесі:');
  For i:=1 to n do Write(A[i]:4);  Readln;end.
```

13.7 Біріктіру тәсілімен сұрыптау

Бұл тәсіл бойынша:

- берілген массив бірнеше бөліктерге (кіші массивтерге) бөлініп, бөлек-бөлек сұрыпталады;
- бірінші және екінші бөліктен сұрыпталған бір бөлік жасақталады; пайда болған бөлік пен үшінші бөлік және т.с.с. біріктіріліп сұрыпталады;
- осы процесс соңғы екі бөлік біріккенге дейін жалғастырылады.
- Массивті бөліктерге бөліп, бөлек-бөлек сұрыптау оқушыға қиындыққа түспейді деп ойлаймыз. Сондықтан, сұрыпталған екі массивті біріктіріп сұрыптау алгоритмін қарастырайық.

Мысалы. Өсуі бойынша сұрыпталған m элементтен тұратын A және n элементтен тұратын B сандар массивтері берілген. A және B массивтерінің элементтерінен, өсуі бойынша сұрыпталған C массивін жасақтаңыз.

```
Program Prost_8;
  const m=6; n=8;
  var A: array[1..m] of integer;
      B: array[1..n] of integer;
      C: array[1..m+n] of integer;
      i, j, k: integer;
begin Randomize; Write('A массиві: ');
```



```

For i:= 1 to m do
  begin    if i=1 then A[i]:=Random(8)+1
           else A[i]:=A[i-1]+Random(9)+1;
           Write(A[i]:4);    end;
  Writeln; Write('B массиві:');
For i:= 1 to n do    begin
  if i=1 then B[i]:=Random(8)+1
  else B[i]:=B[i-1]+Random(9)+1;
  Write(B[i]:4);    end;
i:=1; j:=1; k:=1; While (i<=m) and (j<=n) do
  begin
if A[i]<=B[j]    then begin C[k]:=A[i]; inc(i); end
  else begin C[k]:=B[j]; inc(j); end;
inc(k);    end;
While i<=m do begin C[k]:=A[i]; inc(i); inc(k); end;
While j<=n do begin C[k]:=B[j]; inc(j); inc(k); end;
Writeln; Write('C массиві: ');
For i:=1 to m+n do Write(C[i]:4); Readln;end.

```

13.8 Екі өлшемді массивтер

Мысалы. Өлшемі $n \times m$ болатын A массиві мен оны транспозициялау (жатық жолдарын тік жолдарға айналдыру) нәтижесінде алынған B сандар массивтерін экранға шығару.

```

Program Prost_9;
  const n=5; m=7;
  var i, j, k: integer;  A: array[1..n, 1..m] of integer;
      B: array[1..m, 1..n] of integer;
begin k:=0;
  For i:=1 to n do    begin
    For j:=1 to m do    begin      {Массивті натурал сандармен
inc(k); A[i, j]:=k; Write(A[i, j]:4); толтыру және экранға шығару}
end;    Writeln;    end;
  Writeln; Writeln('Транспозициялау нәтижесі:');
  For i:=1 to m do    begin
    For j:=1 to n do
      begin B[i, j]:=A[j, i]; Write(B[i, j]:4);    {Массивті транспозициялау
end; және экранға шығару}
      Writeln; Writeln;    end; Readln;end.

```

Мысалы. Өлшемі $n \times m$ болатын A бүтін сандар массиві берілген. Массивтің оң сандардан ғана тұратын жатық және тік жолдары бар болса олардың нөмірлерін анықтаңдар

```

Program Prost_10;
  const n=5; m=8;
  var i, j, k: integer; p1,p2: boolean;A: array[1..n, 1..m] of integer;
begin Randomize; For i:=1 to n do
  begin For j:=1 to m do
    begin A[i, j]:=Random(6)-Random(6)+2; {Массивті кездейсоқ
сандармен Write(A[i, j]:4); толтыру және экранға шығару}
    end; Writeln; Writeln; end;
  p1:=true;
  For i:=1 to n do begin p2:=true;
    For j:=1 to m do if A[i, j]<=0 then begin p2:=false; Break; end;
    {тік жолды тексеру}
    if p2 then begin Writeln(i, ' - жатық жол;'); p1:=false; end;
    end; {Нәтижені экранға шығару}
  if p1 then Writeln('Жатық жол жоқ');
  p1:=true; For j:=1 to m do
  begin p2:=true; For i:=1 to n do
    if A[i, j]<=0 then begin p2:=false; Break; end; {жатық жолды тексеру}
    if p2 then begin Writeln(j, ' - тік жол;'); p1:=false; end;
    end; {Нәтижені экранға шығару}
  if p1 then Writeln('Тік жол жоқ'); Readln;end.

```

13.9 Тура қосулар көмегімен сұрыптау

Тура қосулар көмегімен сұрыптаудың негізгі идеясы - жана элементті реттелген тізімге қажетті әсеріне қосу болып табылады. Мұндай сұрыптауда кез келген тізімнің бірінші элементі сұрыпталған деп есептеледі. Екінші элементті бірінші элементтен тұратын тізімнің керек жеріне қосады. Енді берілген тізімнің үшінші элементін реттелген екі элементтен тұратын қажет жеріне қосады. Бұл процесті берілген тізім элементінің барлығы тізімнің сұрыпталған бөлігіне қосылғанша жалғасады.

Қажетті орынды іздеу процесін былай жүргізуге болады: x- ті кезектегі а элементімен салыстырамыз. Одан кейін немесе x бос орынға қосылады немесе а оңға жылжиды және процес солға кетеді. Бұл процес келесі екі шарттың бірі орындалған кезде аяқталады.

- 1) кілті x-тің кілтінен кіші элемент табылды;
- 2) дайын тізбектің сол жағына жетті ;

Мұнда бөгет әдісін қолданған дұрыс, яғни x мәні бар a(o) бөгетін енгіземіз.

Толық алгоритмді келтіреміз:

```

Procedure SORT 1;
VAR E, j : index ; x: item ;
BEGIN
  FOR i: =2 TO n 20
  X: = a [I]; a[o]:=x; j=I;

```

```

While x<a [j-1] do a [j]: =a[ j-1] ; j:=j-1; END;
A[j]: 2x;
END;

```

Тура қосулар көмегімен сұрыптау мысалы:

Бастапқы кілттер	44	55	12	42	94	18	06	67
I=2	44	55	12	42	94	18	06	67
I=3	12	44	55	42	94	18	06	67
I=4	12	42	44	55	94	18	06	67
I=5	12	42	44	55	94	18	06	67
I=6	12	18	42	44	55	94	06	67

13.10 Тура алмастыру көмегімен сұрыптау (көпіршікті әдіс)

Тура алмастыру алгоритмі көршілес элементтердің пар жұбын салыстыру және орнын алмастыруға негізделген, мұндай салыстыру мен алмастырулар барлық элементтер реттелгенше жалғастырылады. Егер массивтерді горизонталь емес вертикаль түрде қарастырсақ ондағы элементтерді судың көпіршіктері деп алуға болады. Оның әрбіреуінің салмағы оның кілтіне сәйкес келеді. Бұл жағдайда әрбір жүрісте бір көпіршік оның салмағына қарай деңгейге көтеріледі.

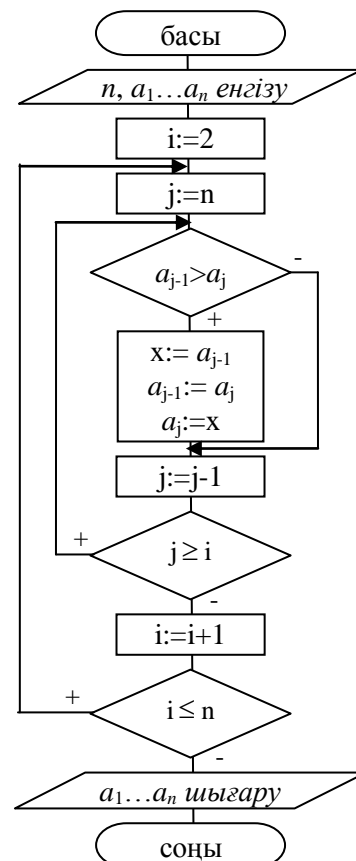
Мысалы. Бұл әдіс бойынша көрші тұрған екі элемент салыстырылып, одан кейін сұрыптау шартына тәуелді орындары алмасады.

I = 1	2	3	4	5	6	7	8
44	06	06	06	06	06	06	06
55	44	12	12	12	12	12	12
12	55	44	18	18	18	18	18
42	12	55	44	42	42	42	42
94	42	18	55	44	44	44	44
18	94	42	42	55	55	55	55
06	18	94	67	67	67	67	67
67	67	67	94	94	94	94	94

```

PROCEDURE BybbleSort;
  VAR i,j, x: integer;
BEGIN
  FOR i: =2 TO n DO
    FOR j: = n TO i DO
      If a[j-1]> a[j] THEN
        begin
          x: = a[ j - 1]; a[j-1]:=a[j]; a[j]: = x
        end
      end
    end
  END;

```



Шейкерлік сұрыптау. Көпіршік сұрыптау алгоритмін жақсарту тәсілі – қандай да бір жүріс процесінде ауыстырулар болды ма, болмады ма соны сақтап отыру болып табылады. Егер соңғы жүрісте ауыстырулар болмаса, онда алгоритмді аяқтауға болады. Алайда, егер ауыстырулар болғандығы туралы фактыны ғана сақтап қоймай, сонымен қатар соңғы ауыстыру орнын (индексін) да сақтаса бұл жақсартуды тағы да жақсартуға болады. Осы k индексінен жоғары орналасқан барлық көрші элементтердің жұптары керекті ретпен орналасқан. Сондықтан қарастыруды осы индекспен аяқтауға болады. Алдын ала анықталған i -дің төменгі шегіне дейін жетпей-ақ қарастыруды осы k индексмен аяқтауға болады. Үшінші жақсарту: тізбектеп қарастырулардың бағытын алмастыру. Осылайша алынған алгоритм «шейкерлік» сұрыптау (ShakerSort) деп аталады.

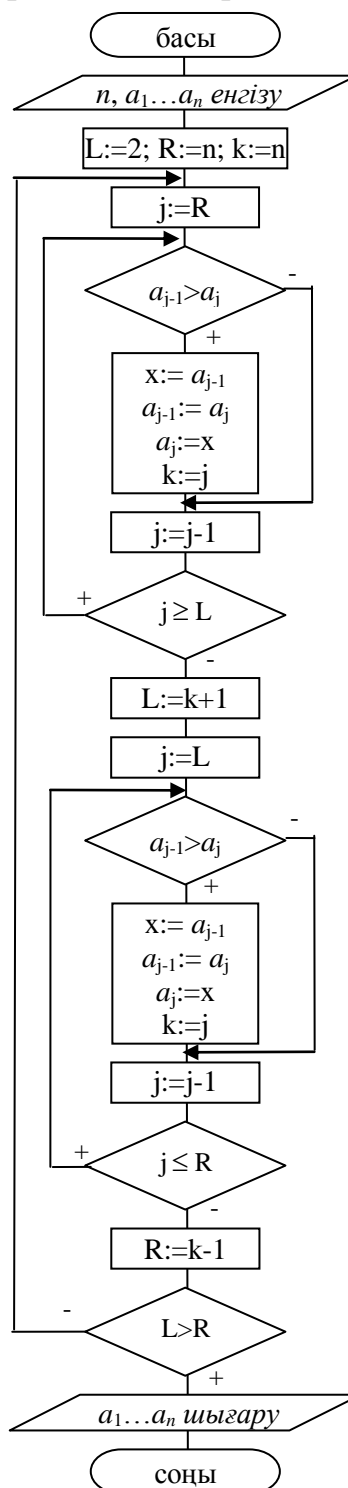
Шейкерлік сұрыптау мысалы

L =	2	3	3	4	4
R =	8	8	7	7	4
Dir =	↑	↓	↑	↓	↑
	44	06	06	06	06
	55	44	44	12	12
	12	55	12	44	18
	94	12	42	18	42
	18	94	18	55	55
	06	18	67	67	67
	67	67	94	94	94

```

PROCEDURE ShakerSort;
  VAR j, k, L, R, x: integer;
BEGIN L: = 2; R: = n; k: = n;
  REPEAT
    FOR j: = R DOWNTO L DO
      IF a[j-1]>a[j] THEN BEGIN
        x: = a[j-1]; a[j-1]:=a[j]; a[j]: = x; k: = j  END;
    L: = k+1;
    FOR j:=L to R do
      IF a[j-1]>a[j] THEN BEGIN
        x:=a[j-1];a[j-1]:=a[j];a[j]:=x;k:=j  END;
      R:=k-1
    UNTIL L>R
  END; { ShakerSort}

```



Бұл сұрыптау алгоритмді жақсартудың жолы – қандай да бір жүріс кезінде алмасулар болған, болмағандығын сақтау. Егер соңғы жүрісте алмасулар болмаса, алгоритмді яқтауға болады. Бұл жақсарту, бірақ оны тағы да жақсартуға болады, егер алмасуды сақтап қана қоймай, сонымен қатар соңғы алмасудың орнын (индексін) сақтау. Осы к индексінен жоғары көршілес элементтердің жұптары реттеліп тұр. Сондықтан қарастыруларды осы индексте аяқтау керек. Үшінші жақсарту: тізбектің бағытын кезекпен қарау керек. Бұндай алгоритмді «Шейкерлік» алгоритм деп атайды.

«Шейкерлік» сұрыптау мысалы

k - алмасу орны

l - алмасудың сол жақ шеті

r - алмасудың оң жақ шеті

Procedure shaver sort;

Var j, k, l, r, x: integer;

Begin

L:=2; r:=n; k:=2n;

Repeat

For j:=r downto l do

If a[j-1]>a[j] then begin

X:=a[j-1]; a[j-1]:=a[j]; a[j]:=x; k:=j; end;

L:=r+1;

For j:=1 to r do

If a[j-1]>a[j] then begin

X:=a[j-1]; a[j-1]:=a[j]; a[j]:=x; k:=j; end;

R:=r-1 until l>r

End;

Мысалы. Массивті толтыру. Массивті мына формула бойынша толтырайық: $c[i]=a*i^2$.

Program summa;

Uses crt;

var c:array[1..100] of word;

i,n:integer; a:integer;

Begin

write('n=');

readln(n); write('a=');

readln(a);

for i:=1 to n do

c[i]:=a*sqr(i);

for i:=1 to n do

writeln('c['i,']=',c[i], ' ');

repeat until keypressed;

end.

Массив элементтерін алмастыру. Бүтін сандардан құрылған екі өлшемді массив берілген. Массив элементтерінің арифметикалық ортасынан кіші болатын барлық элементтерін бүтін мәніне дейін дөңгелектелінген арифметикалық орта мәнімен алмастыратын программа құру. Массив 0-ден 100-ге дейінгі сандармен кездейсоқ түрде толтырылады.

```
Program almastyru;
Uses crt;
var c:array[1..100,1..100] of word;
    i,j,n:integer; a:real;
Begin
write('n=');
readln(n);  a:=0;
Randomize;
  for i:=1 to n do
  for j:=1 to n do begin
c[i,j]:=random(100);  a:=a+c[i,j] end;
  for i:=1 to n do
  for j:=1 to n do
writelnc['i,',j,']=c[i,j],';
  a:=a/n; writeln('arif.orta=',a:5:2);
  for i:=1 to n do
  for j:=1 to n do
if c[i,j]<a then c[i,j]:=round(a);
  for i:=1 to n do
  for j:=1 to n do
writelnc['i,',j,']=c[i,j];
  repeat until keypressed;
end.
```

Массив элементтерін өшіру. Бүтін сандардан құрылған сызықтық массив берілген. Массивтің k-сыншы элементін өшіретін программа құру.

```
Program almastyru;
Uses crt;
var c:array[1..100] of word;
    i,n,l:integer;
Begin
write('n='); {массив өлшемін енгізу}
readln(n);
{массивті толтыру}
Randomize;
  for i:=1 to n do
    c[i]:=random(100);
  for i:=1 to n do {алғашқы массив элементтерін шығару}
```

```

writeln('c['i,']=',c[i], ' ');
writeln('oshiriletin element N=');
read(l);
{көрсетілген элементті өшіру}
  for i:=1 to n do
    c[i]:=c[i+1];
  {көрсетілген элемент өшірілгеннен кейінгі массивті шығару}
for i:=1 to n-1 do
  writeln('c['i,']=',c[i]:4);
  repeat until keypressed;
end.

```

Массивке элементтер қосу. Массивке элемент қосқанда қандай да бір мән қойылатын орыннан бастап, массив элементтері оңға жылжиды. Бұдан массивтің ұзындығы 1-ге артады.

Бүтін сандардан құрылған k-сыншы элементінің орнына массивтің ең кіші элементіне тең мәнді қоятын программа құру.

```

Program vstavka;
Uses crt;
var c:array[1..100] of word;
    i,n,l:integer; min:integer;
Begin
write('n='); {Массивтегі элементтер саны}
readln(n);   {}
write('element koyylatyn oryn=');
readln(l);
Randomize;
  for i:=1 to n do {массивті толтыру}
    c[i]:=random(100);
  for i:=1 to n do {массив элементтерін шығару}
    writeln('c['i,']=',c[i], ' ');
    min:=c[1];
    for i:=2 to n do {ең кіші элементті анықтау}
      if min>c[i] then min:=c[i];
      writeln('min=',min);
      {көрсетілген орынға элемент қою}
      for i:=n+1 downto l do {}
        c[i]:=c[i-1];c[l]:=min;
        writeln('turlengen massiv');
      {алынған массивті шығару}
    for i:=1 to n+1 do
      writeln('c['i,']=',c[i]);
      repeat until keypressed;
end.

```

Массивтерді түрлендіру.

Бүтін оң және теріс сандардан құрылған бір өлшемді массив берілген. Екі жаңа массив құру қажет: біреуінде тек оң сандар мен 0-дер, ал екіншісінде теріс сандар ғана орналасқан болуы қажет.

```
Program turlendiru;
Uses crt;
var mas1,mas2,c:array[1..100] of integer;
    i,n,l,k:integer;
Begin
write('n='); {массив өлшемін енгізу}
readln(n);
for i:=1 to n do {массивті толтыру}
begin
write('c['i,']=');
readln(c[i]) end;
{шығарылатын массивтерді толтыру}
k:=1; l:=1;
for i:=1 to n do
begin
if c[i]>=0 then begin mas1[k]:=c[i]; k:=k+1 end
else begin mas2[l]:=c[i]; l:=l+1
end
end;
writeln('on elemenyuer massivi');
for i:=1 to k-1 do
writeln('mass1['i,']=',mas1[i]);
writeln('teris elementter massivi');
for i:=1 to l-1 do
writeln('mas2['i,']=',mas2[i]);
repeat until keypressed;
End.
```

Массив элементтерін сұрыптау

Массив элементтерін сұрыптауда қойылатын негізгі шарт: массив элементтерін сұрыптаудың таңдалған әдісі жадыны тиімді пайдалануда болып табылады.

Тікелей қосу көмегімен сұрыптау

Бұл әдіс карта ойынында кеңінен пайдаланылады. Элементтер ойша a_1, \dots, a_{i-1} «дайын» тізбек және a_i, \dots, a_n алғашқы тізбек болып бөлінеді. Әрбір қадамда $i=2$ бастап, i -дің мәнін 1-ге арттыра отырып, алғашқы тізбектен i -ші элемент шығарылып тасталынады да, дайын тізбекке барып орналасады. Сөйтіп, ол жаңа орынға қойылады.

Сегіз кездейсоқ таңдалынған сандарды тікелей жалғау көмегімен сұрыптаудың мысалы төмендегідей:

Алғашқы кілттер	44	55	12	42	94	18	06	67
i=2	44	55	12	42	94	18	06	67
i=3	12	44	55	42	94	18	06	67
i=4	12	42	44	55	94	18	06	67
i=5	12	42	44	55	94	18	06	67
i=6	12	18	42	44	55	94	06	67
i=7	06	12	18	42	44	55	94	67
i=8	06	12	18	42	44	55	67	94

Бұл сұрыптаудың алгоритмі төмендегідей:

For i:=2 To n Do

X:=a[i]; {x-ті a[1],...,a[i] арасындағы сәйкес орынға қою}

End;

Шынайы іздеу процесінде тізбектегі салыстырулар мен жылжуды алмастыра отырып, електен өткізу болып табылады. X кезекті a_j элементімен салыстырылады, одан кейін x не бос орынға қойылады, не a_j оңға қарай жылжиды, ал процесс солға қарай жүреді. Електен өткізу процесі төмендегі шарттардың бірі орындалғанда аяқталады:

1. x-тің кілтінен кіші кілтті a_j элементі табылған жағдайда;
2. тізбектің сол жақ шетіне жеткен жағдайда.

Сонымен, бұл алгоритм фрагменті төмендегідей:

```

...
For i:=2 to n do
begin
X:=a[i];a[0]:=x; j:=i;
While x<a[j-1] do a[j]:=a[j-1]; j:=j-1 end;
A[j]:=x
End;
...

```

Мұндай алгоритм орнықты сұрыптау процесін сипаттайды: кілттері бірдей элементтер реті өзгеріссіз қалады. Тікелей жалғау алгоритмін оңай жетілдіруге болады: дайын тізбекке жаңа элемент қойылғаннан кейін өзі реттеледі. Ал, екілік іздеуде салыстыру дайын тізбектің ортасынан басталады, одан кейін қажет бөлу процесі жалғау нүктесі табылғанша жалғаса береді. Мұндай түрлендірілген сұрыптау алгоритмі екілік жалғау әдісі деп аталады.

Тікелей таңдау көмегімен сұрыптау әдісі

Бұл әдіс төмендегідей принципке негізделген:

1. Кілтті кіші элемент таңдалады;
2. ол бірінші элемент a_1 -мен алмастырылады;
3. осы процес қалған $n-1$ элементпен, $n-2$ элементпен және т.с.с. қайталанып, бір ең үлкен элемент қалғанша жалғаса береді.

Бұл әдіспен жұмыс істеу процесі алдыңғы мысалдағы 8 кілтпен жүргізіледі.(2.2-сурет).

Алғашқы кілттер	44	55	12	42	94	18	06	67
i=2	44	55	12	42	94	18	06	67
i=3	12	44	55	42	94	18	06	67
i=4	12	42	44	55	94	18	06	67
i=5	12	42	44	55	94	18	06	67
i=6	12	18	42	44	55	94	06	67
i=7	06	12	18	42	44	55	94	67
i=8	06	12	18	42	44	55	67	94

Оның алгоритмі төмендегідей:

```

For i:=1 to n-1 do
    a[i]..a[n] аралығынан ең кіші индексті k-ға меншіктеу;
    a[i] мен a[k]-ны орындарымен алмастыру;
End;

```

Мұндай әдіс тікелей таңдау деп аталады, бұл әдіс қандай да бір мағынада тікелей жалғауға қарама-қарсы. Тікелей жалғауда әрбір қадамда алғашқы тізбектің тек бір ғана элементі, ал дайын тізбектің қосу нүктесі ізделінетін барлық элементі қарастырылады. Ал, тікелей таңдауда ең кіші кілтті бір элементті іздеу үшін алғашқы тізбектің барлық элементтері қарастырылып, табылған элемент кезекті элемент ретінде дайын тізбекке орналасады. «Тікелей таңдау» әдісімен сұрыптау алгоритмі төмендегідей.

```

Program suruptau2;
Uses crt;
var c:array[1..100] of word;
    i,j,n,r,k:integer;
Begin
write('n='); {массивтің өлшемін енгізу}
readln(n);
{массивті толтыру}
Randomize;
for i:=1 to n do
    c[i]:=random(100);
for i:=1 to n do {алғашқы массивті шығару}
    writeln('c['i,']=',c[i, ' ');
    {массив элементтерін сұрыптау}
for i:=1 to n-1 do
    begin
        k:=i;
        r:=c[i];
        {массив элементтерінің қалған бөлігінен ең кіші элементін іздеу}
for j:=i+1 to n do
if c[j]<r then
begin
k:=j;
r:=c[j]; end;
c[k]:=c[i];

```

```

        c[i]:=r;
    end;
    {сұрыпталған массив элементтерін шығару}
    writeln('suruptalghan massiv');
    for i:=1 to n do
    writeln('c['i,']=',c[i]);
    repeat until keypressed;
end.

```

13.11 Рекурсивті алгоритмдер

Рекурсия – ішкі программаны оның өз ішінде тұрып шақыру. Паскаль тілінде рекурсияны қолдануға рұқсат етілген. Рекурсиялық алгоритм, қойылуының өзі рекурсиялық сипатта болатын кейбір математикалық есептерді шешкенде өте тиімді. Бұл жағдайда есеп кіші өлшемді шағын есептер жиынтығына бөлінеді. Классикалық мысал - $N > 0$ және $0! = 1! = 1$ шарттары үшін $N! = N * (N-1)!$ факториалын есептеу. Мұнда N өлшемді бастапқы есеп N көбейтінді мен $N-1$ шағын есебін шешуге бөлінген. Рекурсиялық алгоритм құру үш этапты қамтиды:

- 1) Есепті параметрлеу;
- 2) Тривиалдық жағдай іздеу және оны шешу;
- 3) Жалпы жағдайды бейнелеу.

Ішкі программа әрбір орындалған сайын оның жергілікті айнымалылары мен ішкі программдан қайту адресі жадының арнайы облысы – *стекте* сақталады. Ішкі программа аяқталып, одан шығу кезінде жадының бұл бөлігі босайды. Рекурсиялық шақыру кезінде локалдық жады босамайды, "қысқы ұйқыға" кетеді, стекте жергілікті айнымалылардың жаңа мәндеріне жаңа облыс және жаңа қайту адресі бөлінеді. Стекке қатынас жасау "соңынан келу – бірінші кету" (Last Input, First Output - LIFO) қағидасы бойынша жүреді, сондықтан ішкі программдан шығу кезінде қайту нүктесі – соңғы шақыру нүктесі болады. Яғни стекті ұйымдастыру рекурсияның жүзеге асырылуын қамтамасыз етеді. Рекурсия – жанама болуы мүмкін: А ішкі программасы В ішкі программасын, ал ол өз кезегінде А ішкі программасын шақырады. Ал бұл Паскальдың қатаң типтендірілген - ішкі программа оған сілтеме жасалмас бұрын сипатталуы тиіс деген принципін бұзады. Бұл жағдайда төменде анықталған ішкі программа тақырыбын алғашқы ішкі программдан жоғары шығару керек, ал ішкі программа денесінің орнына тақырыптан кейін forward қызметші сөзін көрсетеді; (нүктелі үтір - міндетті).

Рекурсия деп өзін өзі анықтайтын немесе құрайтын объект. Рекурсия тек қана математика да ғана емес, күнделікті өмірде де жиі кездеседі. Рекурсия анықтамасы математикада өте үлкен қуатты аппарат. Бірнеше мысалдарға тоқталайық. Натурал сан, ағаш және анықталған функция.

1. Натурал сан:
 - а) 0 натурал сан,
 - б) натурал саннан кейінгі сан да натурал.

2. Ағаш:

а) 0 ағаш (оны "бос ағаш" деп атайды),

б) егер t_1 және t_2 — ағаш болса, онда оның төбесін құрайтын екі төменгі орналасқан ағашта ағаш болады

3 "факториал" $n!$ функциясы (теріс емес бүтін сан):

а) $0! = 1$,

б) $n > 0: n! = n * (n - 1)!$

Рекурсивті алгоритмнің артықшылығы ақырғы тұжырым көмегімен объектілердің шексіз жиынын анықтауға көмектеседі.

Сәйкесінше соңғы рекурсивтік программа көмегімен шексіз есептеулер жүргізуге болады және программада бірдей қайталаулар кездеспейді. Алайда рекурсивті алгоритмдерді негізінен шығарылатын есепте, есептелетін функцияда немесе өңделетін берілгендер құрылымында нақты анықталған рекурсия бар болса. Жалпы жағдайда P рекурсивті программасын S операторлар жиынынан (P -ны қамтымайтын) және P -ның өзінен тұратын қандай да бір P композициясы ретінде өрнектеуге болады:

$$P = P[S, P].$$

Рекурсивті программаларды өрнектеу үшін процедура немесе ішкі программа ұғымын білу қажетті және жеткілікті, себебі олар кез-келген операторға оған қатынайтын ат қоюға мүмкіндік туғызады. Егер қандай да бір P процедурасы өз өзіне нақты сілтеме жасаса, онда оны тура рекурсия деп аталады, егер P процедурасы P -ға тура немесе жанама сілтеме жасайтын басқа Q процедурасына сілтеме жасаса, онда P жанама рекурсия деп аталады. Сондықтан программа мәтіні бойынша рекурсиялық әрқашан айқын көрінбейді.

Негізінен, процедурамен тек осы процедурада анықталған және одан тыс жерде мағынасы болмайтын локальді объектілер жиыны, яғни айнымалылар, константалар, типтер және процедуралар жиыны байланыстырылады. Ондай процедуралардың әрбір рекурсивті орындалуы кезінде байланысқан локальді айнымалылардың жаңа жиыны құрылады.

Дәріс №14. Модельдер. Ақпараттық модель. Математикалық модель. Компьютерлік модель

Дәріс жоспары:

14.1 Модельдер туралы ұғым

14.2 Модельдерді әр түрлі белгілеріне қатысты классификациялау

14.1 Модельдер туралы ұғым

Дәл бүгінгі күні модель туралы сөз қозғамайтын әрі модельдеумен айналыспайтын бірде-бір ғылым, бірде-бір білім саласы жоқ деуге болады.

Қазіргі философия ғылымының модель және модельдеу ұғымына беретін анықтамаларын қарастырып көрейік.

Табиғи және әлеуметтік шындықтың, адам мәдениеті тудырған нәтиженің, концептуальды-теориялық білімнің т.б. белгілі бір көрінісінің аналогын (сұлбасын, құрылымын, таңбалар жүйесін) модель деп айтады.

Модель - бұл зерттеу процесінде әлгі түпнұсқаның аталмыш зерттеу үшін кейбір маңызды типтік белгілерін сақтай отырып, түпнұсқа - объектіні алмастыратын материалдық немесе ойша көзге елестететін объект.

Құрылу тәсілдеріне және құрылымдарына қарай барлық модель материалдық және идеал модель болып екіге бөлінеді.

Материалдық модельге объективті тіршілік ететін, адам қолымен жасалған немесе осы модельдерге тән белгілі қасиеттерге бара-бар адам іріктеп алған модельдер жатады.

Материалдық модельдер объектінің көлемдік қасиеттерін немесе байланыстарын қалыпқа келтіреді немесе бейнелейді. Мұндай модельдердің түпнұсқаға геометриялық жағынан ұқсас болуы міндетті шарт болып табылады. Бұл топқа әр түрлі макеттер, кристалдық торлардың көлемдік модельдері жатады. Мұндай модельдерді мектепте иллюстративтік модельдер деп те атайды.

Материалдық модельдердің көрнекілігі олардың сезім органдарына лайықтылығында ғана емес, сонымен бірге барлық практикалық әрекет (өлшеу, бақылау) объектінің ішкі құрылысына, қозғалыс принциптеріне, оның заңдылықтары мен маңыздылығына да байланысты екендігінде. Материалдық модель көрінбейтін объектілерді көрнекі түрде зерттеуге мүмкіндік береді.

Адам ойындағы физикалық заңдылықтарға, математикалық ережелерге және логикалық ақыл-ойға сүйеніп жасалатын модельді идеал модель дейді. Олар: суреттер, белгілі бір таңбалар, символдар арқылы берілуі мүмкін. Бірақ элементтердің барлық өзгерісі адам миында логикалық, математикалық, физикалық ережелер мен заңдар арқылы өтеді.

Идеал модельді екіге бөлуге болады: түсінік модель және таңбалық модель.

Түсінік модель - объективтік шындықтың адам миындағы бейнесі. Адам бейне арқылы ойлайды. Сондықтан оқушыларда бейнелер қорын толтыру - физиканы оқытудың басты міндеттерінің бірі. Түсінік модель әсіресе микробөлшектерді, тікелей сезім мүшелері арқылы қабылдауға болмайтын объектілерді зерттеуде керек.

Таңбалық модельдерде объектінің элементтері, олардың қатынастары мен қасиеттері белгілі таңбалардың көмегімен беріледі. Таңбалық модельдердің ерекшелігі - модельдің элементтері мен түпнұсқаның элементтерінің арасында ешбір ұқсастықтың болмауында.

Таңбалық модельдер сезімге - көрнекілік элементтері базасында құрылады. Таңбалық модельдеуде әр түрлі сұлбалар, сызбалар, формулалар, табиғи және жасанды тілдердің алфавитіндегі сөйлемдер модель ретінде қызмет ете алады.

Модель төмендегі жәйттер:

- нақты объект қалай құрылған - оның құрылысы, негізгі қасиеттері, даму заңдылығы мен қоршаған дүниемен өзара әрекеттестігі қандай екендігін тану үшін;

- объектіні не процесті басқаруға үйрену және қойылған мақсатқа жету кезінде басқарудың ең тиімді тәсілдерін айқындау үшін;
- берілген тәсілдерді іске асырудың тура және жанама салдарларына, объектіге әсер ету формаларына болжам жасау үшін қажет.

14.2 Модельдерді әр түрлі белгілеріне қатысты классификациялау

1. Пайдалану саласы бойынша:

- оқулық - көрнекі құралдар, оқыту бағдарламалары, әр түрлі тренажерлар;
- тәжірибелік - кеме моделі (шайқалыс кезінде кеменің орнықтылығын анықтау үшін бассейнде сыналады);
- ғылыми-техникалық - радиоқондырғыларды тексеруге арналған стенд; электрондардың үдеткіші; жайтартқыш;
- ойындық - әскери, экономикалық, спорттық, іскерлік және басқа ойындар;
- имитациялық - шынайы объектіге әлдебір әрекеттің әсерін зерттеу мен бағалау үшін сан рет қайталанатын, болмаса бір-біріне ұқсас, бірақ әр басқа жағдайға қойылған көптеген объектілермен бір мезгілде өткізілетін тәжірибе (эксперимент).

2. Уақыт факторын ескеруі бойынша модельдер статикалық және динамикалық болып екіге бөлінеді.

Статикалық - объектінің ағымдағы жағдайдағы “бір сәттік кесіндісін” береді;

Динамикалық - объектінің уақыт ішіндегі өзгеруін көруге мүмкіндік береді.

3. Ғылым салалары бойынша: экономикалық; экологиялық; ғарыштық; әлеуметтік, т.б.

4. Объектіні түйсіну тәсілі бойынша: материалдық; ақпараттық.

Аталмыш топтардың атаулары модельдердің неден істелгенін көрсетеді. Материалдық (заттық, физикалық) модельдер түпнұсқаның геометриялық және физикалық қасиеттерін көз алдына келтіреді және әрқашанда шынайы нақты түрде болады.

Материалдық модельдер объектіні, құбылысты не процесті зерттеуге материалдық тұрғыдан (ұстау, иіскеу, көру, есту) келуді жүзеге асырады.

Ақпараттық модельдерді ұстауға не көзбен көруге болмайды, олардың материалдық нақты түрі жоқ.

Ақпараттық модель – объекті мен процестің, құбылыстардың қасиеті мен жағдайын, сондай-ақ оның сыртқы дүниемен өзара байланысын сипаттайтын ақпараттар жиынтығы.

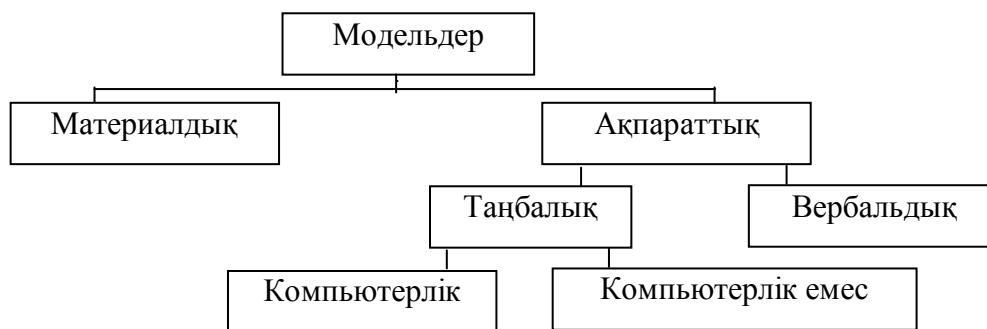
Объекті мен процесті сипаттайтын ақпараттың әр түрлі көлемі мен түсіндіру формасы болуы, әр түрлі құралдармен берілуі мүмкін.

Ақпараттық модельдердің өзін таңбалық және вербальдық (ауызша айтылған) модельдер деп бөлуге болады.

Таңбалық модель – арнайы белгілермен берілген ақпараттық модель, яғни кез келген формальды тіл құралымен берілген. Таңбалық модельдер бізді барлық жерде қоршап алған. Бұл суреттер, мәтіндер, графиктер мен сұлбалар.

Таңбалық модельдер де таңбалы оқудың белгілі ережелері арқылы оқушының санасында объектінің бейнесін жасауға мүмкіндік береді. Таңбалық модельдің ерекшелігі – нәрсенің сезімдік бейнесі тек таңбалардың мәнін ашқан соң ғана пайда болады.

Іске асыру тәсілі бойынша таңбалық модельдерді компьютерлік және компьютерлік емес деп ажыратуға болады.



7-сурет Объектіні түйсіну тәсілі бойынша модельдерді топтастыру

Вербальдық модель - ойша немесе ауызекі формамен берілген ақпараттық модель.

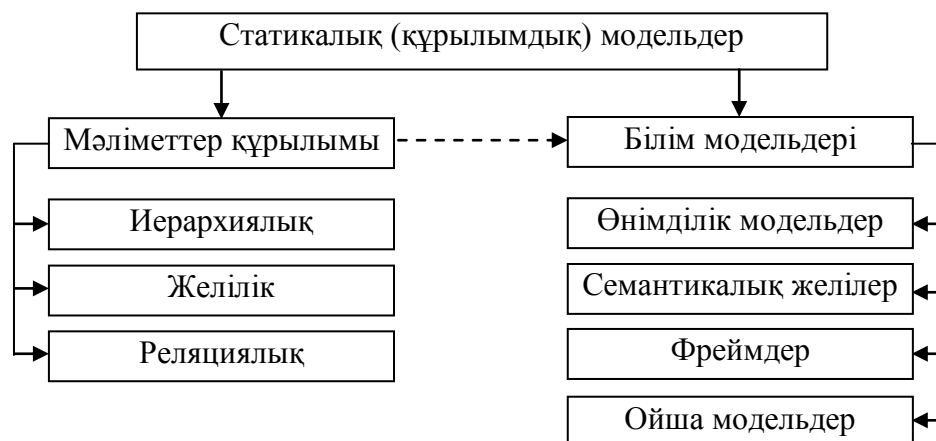
Бұл модельдер ойлау мен тұжырымдау нәтижесінде алынған модельдер. Олар сол ойша күйінде қалуы немесе сөзбен берілуі мүмкін.

Таңбалық және вербальдық модельдер, әдетте, өзара байланысты. Адамның миында туған ойша бейне белгілік формада көрінуі мүмкін. Керісінше, таңбалық модель санада ойша бейнені жасауға көмектеседі.

Мысалы: Аңызға сәйкес, Ньютонның басына құлап түскен алма оның санасында заттардың жерге тартылысы туралы ойдың тууына ықпал етті. Бұл ой тек кейін ғана заң болып шықты, яғни таңбалық формаға ие болды.

Ақпараттық модельдер де статикалық және динамикалық болып бөлінеді.

Статикалық модель объектінің құрылысы мен аумағын көрсетеді, сол себепті оны құрылымдық деп те атайды (8-сурет).



8-сурет. Статикалық модельдің түрлері

Мәліметтер құрылымы - бұл заттық саладағы объектіні, сондай-ақ, оның қасиетін сипаттайтын жекелеген деректер (фактілер). Түйсіну тәсілі бойынша иерархиялық, желілік және реляциялық (кестелік) мәліметтер құрылымы деп ажыратады.

Білім модельдері - бұл заттық саланың осы облыстағы міндеттерді шешуге мүмкіндік беретін анықталған заңдылықтар (принциптер, байланыстар, заңдар). Әр түрлі заттық салалар үшін білім берудің бірнеше модельдері бар. Олардың көпшілігі төмендегі кластарға келіп саюы мүмкін:

- өнімдік модельдер - “егер <шарт>, онда <іс-әрекет>” тәрізді сөйлемдер түрінде білім беруге мүмкіндік жасайтын заңдарға негізделген;

- семантикалық желілер - төбесі - ұғымдар, ал доғалары - олардың арасындағы қатынастарды білдіретін бағдарланған графалар, шешімді іздеу қойылған сұраққа сәйкес келетін желілер фрагментін іздеуге келіп саяды;

- фреймдер - бейнені кескіндеу үшін қажетті формальдандырылған модельдер, абстрактілік бейнелер немесе жағдайлар;

- формальды ойша модельдер - заттық сала аксиомалар жиынтығы түрінде берілген кезде бірінші қатардағы предикаттарды классикалық есептеуге негізделген.

Мәліметтер құрылымы модельге заттық саланың білім бөлігі ретінде енеді.

Динамикалық модель аумақтар жиынтығымен берілген объектінің өзгеруі мен қызмет етуі процесін білдіреді. Динамикалық модель - жүйелердің қызметін бейнелейтін алгоритмдер кешені. Динамикалық модель алгоритмдік түрінде блок-сұлбамен, бағдарламамен, т.б. берілуі мүмкін.

Ақпаратты модельді бейнелеудің ең ыңғайлы формасы - мәтін. Мәтіннің құрылу ережесі пайдаланылатын тілдің грамматикасымен беріледі. Сондықтан, кез келген мәтінді нақты объектінің тілдік модельдері ретінде қарастыруға болады. Арнайы ақпараттық модельді сипаттау үшін формальді тілдер, мысалы бағдарламалау тілдері пайдаланылады. Осылайша бағдарлама мәтіні ақпараттық модель болып табылады. Объектінің құрамын сипаттау үшін математика тілі пайдаланылуы мүмкін. Математикалық формулалар тізбегімен, яғни реттелген математикалық модельмен алгоритмді беруге болады. Сәйкесінше, математикалық модель ақпараттық модельді бейнелеудің бір тәсілі ретінде қарастырылады. Бірақ та "математикалық модель" ұғымының ғылымдағы рөлі жоғары болғандықтан оны жеке қарастыруды талап етеді.

Математикалық модельдер - кең түрде математикалық тәсілдер пайдаланатын таңбалар моделінің маңызды класы. Мысалы, күрделі теңдеулер жүйесін көрсетіп, жұлдыздар қойнауында болатын физикалық процестерді сипаттайтын жұлдыздың математикалық моделін қарастыруға болады. Математикалық модель ретінде объект немесе процестің маңызды қасиеттерін бейнелейтін, математикалық ара қатынастардың жүйесі болып табылатын - формулалар, теңдеулер, теңсіздіктер және т.б. түсіндіріледі. Математикалық модельдеуде біз объектінің нақтылы физикалық қасиетіне және оларда жүретін процестерге ден қоймаймыз, берілген шамалардың шешімге қайта өңделуін ғана қарастырамыз. Математикалық модельдер құру кезінде әркезде

берілгендер арқылы ізделінді шаманы өрнектейтін формуланы табу мүмкіндігі тумады. Осындай жағдайларда қандай да бір дәлдік дәрежесіне жауап бере алатын математикалық әдістер пайдаланады. Әдебиеттерде математикалық модель ұғымына келесі анықтама берілген: "Математикалық модель деп берілген процестерді зерттеу үшін физикалық мәні әр түрлі болса да, ұқсас математикалық өрнектермен бейнеленетін құбылыстарды қарастыру тәсілі".

Есептің математикалық моделін құру үшін:

- 1) математикалық модель негізделінетін болжамдарды бөліп алу керек;
- 2) нені бастапқы берілгендер, нені нәтиже деп есептеу керек, соны анықтау керек;
- 3) нәтиже мен бастапқы берілгендерді байланыстыратын математикалық арақатынастарды (формулалар, теңдеу, теңсіздік және т.б.) жазып алу керек.

Физикада математикалық модельдеу құбылыс пен процесті оқып-үйрену, зерттеудің жетекші тәсілі болып табылады. Қарапайым мысал: $s = g t$ түріндегі математикалық формуланы көріп 9-шы сыныптың оқушысы нақты өмірде бірқалыпты түзу сызықты қозғалыспен істес екенін біледі. Сонда, математикалық модель - объект немесе процестің әр түрлі параметрлерінің байланысын бейнелейтін математикалық формулалар.

Барлық уақытта танып білетін есті адам нақтылықтың объектілерін саналы, ойлы модельдер түрінде өз санасында бейнелеген және олардың үлгілерінен күтілетін нәтижеге әрекет жасаған. Компьютердің пайда болуымен жағдай мүлдем басқаша өзгереді. Адам енді нақтылы объектінің компьютерлік моделін құрып, компьютерге өз білімін ендіреді. Табиғаттың екінші шағылуы адам ойлауынан енді компьютер жадына енеді.

Осы ұғымның әр түрлі әдебиеттерде берілген анықтамаларын қарастырып көрелік.

“Компьютерлік модель” термині А.С.Бешенков, Н.В.Макарова, В.Б.Гисин басшылықтарымен құрылған оқу құралдарында: бірінші жағдайда, орындаушыға есептелініп құрылған компьютерде еліктірілген есептің моделі ретінде, екінші жағдайда, “...компьютерлік модель деп бағдарламалық орталар құралдарымен жүзеге асатын модель айтылады” деген анықтама беріледі, ал соңғы автор компьютерлік модельдеуді модельденетін жүйе туралы ақпаратты компьютерде бейнелеу деп түсіндіреді. Компьютерлік модель компьютер жадында электромагниттік түрде бар болғандықтан, яғни ол материалды формальды модель болып табылады, оның анықтамасын келесі түрде беруге болады: компьютерлік модель – мәліметтер жиынтығы және оларды өңдеуге арналған бағдарламалар, өз кезегінде бағдарламалар да, мәліметтер де компьютер жадында сақталады.

Дәріс №15. Модельдеу. Модельдерді құрудың негізгі кезеңдері. Математикалық модельдеу және есептеу эксперименті. Компьютерлік модельдеу

Дәріс жоспары:

15.1 Модельдеу. Ақпараттық модельдеу туралы ұғым

15.1.1 Модельдерді құрудың негізгі кезеңдері

15.3 Компьютерлік модельдеудің кезеңдері

15.1 Модельдеу. Ақпараттық модельдеу туралы ұғым

Модельдеу деп “таным объектілерін олардың құрылым модельдері арқылы зерттеу” әдісін айтады.

Құрылған модель көмегімен түпнұсқаның құрылымы мен қасиеттерін зерттеу процесі модельдеу деп аталады.

Модельдеу объективті әлемді тану әдістерінің бірі болып табылады. Сондықтан да таным теориясы таным әрекетінің заңдылықтары мен мән-маңызы туралы ілім ретінде модельдеу процесінде де орын алады. Таным объективті шындықты адам санасында нақты бейнелегенде ғана ғылыми дәрежеге көтеріле алады.

Танымның әдістемелік құралы ретінде модельдеуге қойылатын маңызды гнесеологиялық талап: санада зерттелінетін объектіге бара-бар көрініс жасау, сондай-ақ модельдеудің таным әдісі ретіндегі маңызын, оның өзіне тән ерекшеліктерін есепке алу болып табылады.

Ғылыми таным процесінде модельдеу - бірқатар гнесеологиялық, сондай-ақ болжағыштық әрі баға беруші қызметтер атқарады.

Модельдеу барысында зерттелуші объектінің моделі сол объектіні бара-бар бейнелеуде өзінің гнесеологиялық қызметін нәтижелі орындауы үшін төмендегідей қажетті жағдайлар жасалуы керек:

- модельді қабылдаудың нәтижелерін өңдеу барысында ойлау процесінің белсенділігін, оның динамикалы қызметін қамтамасыз ету (бұған ізделініп отырған нәтижені ғылыми тұрғыда алдын-ала болжап көре білу мүмкіндігі арқылы, объектіні зерттеу мақсаттарын нақты белгілей білу арқылы қол жеткізуге болады);

- зерттелініп отырған құбылыстар, процестер арасындағы маңызды байланыстарды (себеп-салдарды) анықтауға қол жеткізу;

- зерттелініп отырған жүйелер (объектілер) құрылымының бірдейлігін және ұқсастығын қамтамасыз ету;

- бір заңға бағынатын объектілер арасындағы байланыстарды салыстыру үшін негіз боларлық нәрселерді табу.

Модельдеудің тағы бір қажетті гнесеологиялық алғы шарты - модель көмегімен зерттелуші объект туралы деректі мен дерексіздің бірлігін модельдік көріністе қамтамасыз етуге мүмкіндік беретін жағдайларды туғызу болып табылады.

Модель танып білушіге нақты материалды модельдеу барысында алынған бір-біріне кереғар емес, қатаң ғылыми интерпретация мен түсінік беруі керек. Модельді теориялық аппарат түсіндірмесі түрінде ғана емес, эксперименттік мәліметтер түсіндірмесі ретінде де пайдалану модельдеу қызметін кеңейтеді және оны зерттелуші құбылысты немесе процесті түсіндіруші модельге айналдырады.

Таным процесінде модельдеу алдын-ала болжам жасау қызметін де атқара алады. Бұл дегеніміз қандай да бір объектінің моделін зерттеу барысында әр түрлі қызмет жағдайындағы осы объектінің одан әрі дамуын болжау мүмкіндігі деген сөз.

Білім алудың таным процесі ретіндегі ерекшеліктері оқушылардың таным күштері мен қабілеттері (ес, назар, ойлау, сөйлеу) дамуының жас ерекшеліктеріне байланысты анықталады. Осыған орай модельдеудің гнесеологиялық аспектілерінің дидактикалық қырлары ашылады.

Модельдеу процесінің құрылымын талдау модельдеудің әдіснамалық маңызы мен гнесеологиялық табиғатын зерттеу барысында үлкен қызығушылық туғызады.

Кез келген ғылымда модельдеу процесі міндеттерді белгілеуден басталады, ал ол мақсаттарды шешудің өзі модельді пайдалану арқылы жүзеге асуы керек. Мұнда, алдымен, зерттеу кезінде көздеген мақсатқа жеткізуі мүмкін қасиеттер мен байланыстарды айқындап алу керек. Іс барысында алдағы зерттеудің пәні болып табылатын зерттелуші объектінің қасиеттері мен байланыстарының жиынтығы айқындалады. Шындықты толығырақ тануға тырыса отырып, нақты объектінің сипаттамалары мен белгілерінің біразынан абстракциялану керек.

Модельдеудің екінші кезеңі модельді, яғни зерттеу нәтижесінде түпнұсқа туралы қажетті мәліметтер алуға мүмкіндік беретін (зерттеуге оңайлары арасынан) объектіні жасаудан (таңдаудан) тұрады.

Түпнұсқаға орынбасар іздеу интуиция арқылы, немесе қатаң логикалық ойлау негізінде жүргізілуі мүмкін. Модель ретінде түпнұсқаға ортақ белгісі бар ұқсас объектіні таңдайды. Олардың қасиеттері мен байланыстары да ұқсас болғаны жөн. Модельді іздестіріместен, таңдамастан бұрын алдын-ала оның жалпы сұлбасын елестетіп, түсініп алу қажет. Модельге түпнұсқа қасиеттерін берудің қай кезеңі нақты жағдайларда мақсатқа сай келеді? - деген сұрақ модельдеудегі маңызды мәселелердің бірі болып табылады.

Модельге түпнұсқаның аса көп қасиеттерін бір мезгілде енгізу зерттеуді қиындатуы мүмкін, мұндай жағдайда біз белгісіз параметрлер мен белгісіз функциялардан тұратын күрделі теңдеулерге шырмалып қаламыз.

Ал, егер керісінше, тым қарапайым модель құрастырсақ, онда біз көп кешікпей мұндай модель құбылыстың одан арғы барысын біздің талаптарымызды қанағаттандырушы дәрежеде болжай алмайтындығын көреміз. Бір модельдің әр түрлі элементтерін конструкциялаудағы әр түрлі ұқсастықтарды пайдалана білу мүмкіндігі де модельдеудің жалпы маңызды бір ерекшелігі болып табылады.

Модельдеудің келесі кезеңі - модельді зерттеу. Бұл кезеңде модель зерттеу объектісіне айналады. Барлық әрекет модельдің өзінде жүргізіледі және

ол тұрақты ақпарат алуға, оның даму заңдылықтарын, қасиеттері мен байланыстарын анықтауға бағытталады.

Бірақ та, модельді зерттеу оның өзі үшін емес, сайып келгенде басқа бір объекті (түпнұсқа) туралы мәлімет беруге қабілетті болғандығы үшін ғана қажет. Сөйтіп, модельді зерттеу дегеніміз модельденуші объектіні зерттеу болып шығады.

Модельді зерттеудің нақты нәтижесі түпнұсқаны тану міндетін шешу арқылы анықталады. Бұл модель мен түпнұсқа арасындағы нақты ұқсастықтардың болуынан және таным процесіндегі модельдің көмекші рөлінен келіп шығады. Модельді зерттеу әдістемесі мен түпнұсқаны зерттеу әдістемесі арасындағы сәйкестік жалпы ортақ сипатқа ие. Бұл модельді зерттеу барысында түпнұсқаны алмастыру қызметін атқара отырып, түпнұсқадан тыс жерде өзі жеке заңдылықтары мен қасиеттері бар объекті екендігінен келіп шығады.

Модель мен түпнұсқа сипаттары арасындағы қиын айырмашылықтар, түпнұсқада зерттеулерді модель көмегімен орындауға мүмкіндік береді. Мұндай айырмашылықтар бұрын қолданылмаған әдіс-тәсілдерді модельді зерттеуде қолдануға жағдай жасайды. Әр түрлі уақыт интервалын қолдану және тәжірибелерді бірнеше рет қайталап жасау мүмкіндігінің болуы модельдеудің маңызды бір құндылығы болып табылады.

Осының нәтижесінде, бұрын тек бақылауға ғана қолайлы болып келген объектілер модельдеу арқасында эксперимент жасау объектісіне айналады.

Модельдеудің келесі кезеңі - модель көмегімен алынған ақпаратты түпнұсқаға көшіру кезеңі.

Модельдеудің эвристикалық маңызы модельді зерттеу барысында табылған жаңа қасиетті немесе байланысты модельге де түпнұсқаға да ортақ бірқатар қасиеттердің болуы негізінде, түпнұсқаға әкеп таңуға болатындығы арқылы көрінеді.

Модельдеу әдісі арқылы алынған кейбір ақпарат кей жағдайларда нақты сенімді емес, ықтимал сипаттамаларға ие болады. Модельдеу нәтижесінің нақтылығына бірнеше факторлар әсер етеді.

Модельдеу процесі модель көмегімен алынған объект туралы мәліметтердің шындыққа сай екендігін тексеруімен және оларды осы объекті туралы ақпараттар жүйесіне енгізумен аяқталады.

Модельдеу технологиясы зерттеушіден мәселелерді айқындау мен мақсат қою іскерлігін, зерттеу нәтижелерін болжауды, ойға қонымды баға беруді жүргізуді, модельдерді құру үшін қажетті басты және екінші дәрежелі факторларды бөле білуді, математикалық теориялар мен функционалдық тәуелділіктерді таңдауды, тапсырманы компьютерлік жүйені пайдаланумен шешуді, компьютерлік эксперименттерге талдау жасауды талап етеді.

Модельдеу дағдылары адам үшін оның күнделікті іс-қызметінде де өте маңызды. Олар күн тәртібін, оқуды, жұмысты орынды жоспарлау мен таңдау бар кезде тиімді варианттарды таңдауға, әр түрлі өмірлік мәселелерді сәтті шешуге көмектеседі.

Құрылу тәсілдері мен құрылымдарына қарай модельдеуді материалдық және идеал модельдеу деп екіге бөледі.

Зерттелетін процестер мен құбылыстардың қасиеттерін ұқсас теориялардың негізінде модельден объектіге кезекті көшіру көмегімен зерттеу жол беретін (әдетте, зертханалық жағдайда) кезде шынайы объектіге оның үлкейтілген немесе кішірейтілген көшірмесі салыстырылатын модельдеуді материалдық деп атаймыз.

Мысалы, астрономияда - планетария, архитектурада - ғимараттар макеті, ұшақ жасауда - ұшу аппараттарының модельдері.

Объекті мен модельдердің материалдық ұқсастығына емес, логикалық ойша ұқсастығына негізделген модельдеуді идеал модельдеу деп атаймыз. Идеал модельдеуді екіге бөлуге болады: таңбалық және математикалық.

Таңбалық модельдеу - модель ретінде қайсыбір түрдің: сызбалардың, графиктердің, сұлбалардың, формулалар мен таңбалар жиынтығының таңбалық қайта өзгеруін пайдаланатын модельдеу түрі.

Математикалық модельдеу - объектіні зерттеу математика тілінде бейнеленген модельдер арқылы іске асырылатын модельдеу түрі. Мысалы: Ньютонның заңдарын математикалық формулалар арқылы бейнелеу мен зерттеу.

Жоғарыда айтылғандарды ескере отырып, модельдеу процесін бірнеше кезеңдердің жүйелілігі түрінде түсінуге болады, олар:

объект→модель→модельді зерттеу→объекті жайлы білім

Модельдеу процесінің басты мақсаты түпнұсқаға анағұрлым барабар модельді таңдау және зерттеу нәтижелерін түпнұсқаға көшіру болып табылады.

Ақпараттық модельдеу деп объектінің моделін немесе құбылыстың өту барысын формальды түрде жазып сипаттауды айтамыз. Ақпараттық модельдеуді интуитивтік модельдеу, бейнелік модельдеу және бейнелік-таңбалық деп бөледі.

Интуитивтік модельдеу - бұл объект туралы ойша көз алдына елестету.

Бейнелік модельдеу - бұл табиғи тілмен немесе салынған суретпен бейнеленген, сезімдік көрнекі бейнелер көмегімен түпнұсқаның қасиетін білдіру. Ақпаратты тасушы адамнан тыс болуы да мүмкін. Мысалы: фотосурет, кинофильмдер, ауызша әңгімелер, көптеген физикалық модельдер: Резерфорд пен Бор ұсынған атом моделі, молекула-кинетикалық теория, идеал газ моделі. Ғылыми модельдеу кезінде ұғым бәрінен бұрын сөзбен не суреттермен кодталатындықтан, модельдеудің бұл түрін тағы да иконикалық немесе вербальдық деп те атайды. Біздіңше, вербальдық және иконикалық модельдеу кодтау тәсілі бойынша (түйсіну тәсілі бойынша) жасалған бейнелік модельдеудің түрлерінің біріне жатады деп есептеу мүмкін тәрізді.

Бейнелік-таңбалық ақпараттық модельдеу қайсыбір түрдің: сұлба, график, сызба, жоспар, карта, т.б. таңбалық бейнелерін пайдаланады. Мысалы: мектеп картасы, пәтер жоспары, сайлаушылар дауысы қатынасының бағаналық диаграммасы, ұғымның семантикалық желісі, шежіре, алгоритмнің блок-сұлбасы, топтастыру сұлбасы. Глобусты шынайы бір объектідегі екі модельдің жиынтығы: физикалық дене түріндегі Жер шарына ұқсас материалдық модель

және осы дененің үстіне әлдене орналастырылған ақпараттық бейнелік-таңбалық модель ретінде қарастыруға болады.

Таңбалық модельдеу шартты белгілерді, арнаулы символдарды, әріптер мен сандарды пайдаланады және таңдап алынған белгілермен операция жасау заңдылықтарының жиынтығын қарастырады. Мысалы: тілдің жүйесін суреттеудің жалпы сұлбасы немесе соның әлдебір жүйесі, физикалық не химиялық формулалар, математикалық теңдеулер, музыка теориясы, ноталық жазу, т.б. Бұдан бейнелік-таңбалық модельдеу бейнелік және таңбалық модельдердің арасындағы аралық модель екендігі және әр түрлі дәрежеде әрбір нақты модель үшін екеуінің де белгі-сипатын сақтайтыны көрінеді.

“Компьютерлік модельдеу” терминімен академик А.А. Самарский ендірген есептеу эксперименті, яғни зерттеп отырған объектінің математикалық моделін компьютердің көмегімен талдау мен құруға негізделген зерттеу технологиясы тікелей байланысты.

Процестер мен құбылыстарды компьютерлік модельдеу модельденетін объекті туралы есепті шешу үшін пайдаланылады. Компьютерлік модельдеуде модельденетін объекті туралы есеп мақсат түрінде, яғни модельдің қалауымыз бойынша күйін алу есебі ретінде ұйымдастырылуы мүмкін. Мақсаттың қойылуы модельдің жұмыс істеуіне - модельді басқаруға мақсатты бағыттың араласуын болжайды.

Компьютерлік модельдеуде элементтер арасындағы байланыс шамалар мен алгоритмдердің, сондай-ақ жекеше жағдайда есептеуіш формулалар көмегі арқылы сипатталады. Байланыстар компьютерлік графика құралдарымен көзбен шолу түрінде беріледі. Жүйені суреттейтін мәліметтер құрылымы сыртқы жағдайларға тәуелді емес және олардың ауысуында өзгермейді. Компьютерлік модельдеуде құбылыстар мен процестердің модельдерін басқару әдетте адамның компьютермен сұхбаты түрінде жүреді.

15.2 Компьютерлік модельдеудің кезеңдері

Құбылысты компьютер көмегімен модельдеуге кіріспес бұрын ізденуші құбылыстың шығу себептерін, шамалар мен олардың арасындағы тәуелділікті, сондай-ақ өтетін процестердің кезеңдерін өзінше айқын түсініп алуы тиіс. Бұл жерде бастапқы кезең - прототип. Ол өмір сүретін немесе жобалайтын объект не процесс болуы мүмкін. Модельдеудің ақырғы кезеңі - объект жөніндегі білім негізінде шешім қабылдау.

Компьютерлік модельдеу – шығармашылық процесс. Оны белгілі бір формаға сыйғызу өте қиын. Себебі, оның мазмұны тереңде, ол адам ойында жатыр. Анағұрлым жалпы түрде оны төмендегідей кезең-кезеңмен түсіндіруге болады (9-сурет):

Нақты тапсырманы шешу кезінде бұл сұлба бірқатар өзгерістерге ұшырауы мүмкін: қайсыбір блок алынып тасталады не жетілдіріледі, әлдебірі - қосылады. Кезеңдердің мазмұны қойылған тапсырма мен модельдеу мақсаттарына орай айқындалады.

Модельдеудің негізгі кезеңдерін толығырақ қарастырайық. Мысалы, қандай да бір физикалық құбылысты сипаттайтын есеп берілсін.

1-кезең. Есептің қойылуы.

Есеп астарынан шешуге тиісті әлдебір мәселе түсініледі. Есепті қою кезеңінде: есептің жазбасы; модельдеу мақсатын анықтау; объект немесе процесті талдау қажет.

Есептің жазбасында процесс түсінікті болуы үшін есептің шарты әдеттегі тілде беріледі. Бұл жерде бастысы - модельдеу объектісін анықтап, нәтижесінде не алынатынын білу, түсіну.



9-сурет. Компьютерлік модельдеу кезеңдері

Модельдеу мақсаты.

1. Объектіні талдау. Бұл кезеңде модельденуші объектінің негізгі қасиеттерін, элементтері мен олардың арасындағы байланыстарды айқын ажыратып алады.

2. Белгілі қасиеттерімен объектілерді құру. Бұл кезеңде нақты объект қалай жасалған, оның құрылымы, негізгі қасиеттері, даму және қоршаған әлеммен өзара қарым-қатынас заңдары қандай екенін түсіну.

3. Объектіге ықпал жасаудың тікелей және жанама салдарын анықтап болжай білу, дұрыс шешім қабылдау.

4. Объектіні басқарудың тиімділігі. Бұл кезеңде объектіні басқаруды үйрену және қойылған мақсаттар мен критерийлер бойынша осы басқарудың ең тиімді әдістерін анықтау.

2-кезең. Модельді жасау.

1. Ақпараттық модель. Модельдер заттық әлемнің объектілерінің ең бір негізгі белгілерін, қасиеттері мен жағдайын, қатынастарын білдіруі тиіс. Объект жайлы толық ақпаратты нақ солар береді. Бұл кезеңде кез келген формадағы қарапайым объектілердің қасиеттері, жағдайы, іс-әрекеті, т.б. сипаттары анықталады. Мұнда ақпараттың нақты объектіден оқшаулануы жүреді, маңызды ақпарат бекітіліп, елеусіз алынып тасталады. Объектінің іс-әрекеті немесе процестің барысына бағынышты шамалар (кірістік шамалар) және де модель жасау нәтижесінен алынуға тиісті шамалардың (шығыстық шамалар) тізімі жасалады. Бір есеп үшін елеусіз ақпарат, басқасы үшін маңызды болуы мүмкін және одан айырылу есептің дұрыс шешілмеуіне әкелетінін немесе қажет шешімді алуға мүмкіндік бермейтінін атап өту маңызды. Елеусіз ақпараттың тіркелуі қосалқы қиындықтар әкеледі, шешімге жетелейтін жолда кедергілер жасайды. Кірістік параметрлерді олардың шығыстық параметрлерге әсерінің маңыздылығы бойынша бөлу маңызды.

Ақпараттық модель объектіні ешқашан да толық сипаттамайды. Бір ғана объекті үшін әр түрлі ақпараттық модель құруға болады.

Ақпараттық модельді құру модельді жасау кезеңінің бастапқы кезеңі болып табылады. Талдау кезінде бөліп алынған объектілер аумағының бүкіл мәліметі маңыздылығының өсу ретімен орналастырылады және модельдеу мақсатына сәйкес модельді жеңілдетуге алып келеді.

2. Таңбалық модель. Модельдеу процесіне кіріспес бұрын, адам қағазға алдын-ала сызбалардың болмаса сұлбалардың моделін түсіреді, есептеу формулаларын шығарады, яғни қайсыбір таңбалық формада ақпараттық модель құрады, ол не компьютерлік, не компьютерлік емес болуы мүмкін.

3. Компьютерлік модель. Компьютерлік модель - бұл бағдарламалық орта құралдарымен іске асырылған модель.

Компьютерлік модельдерге (модельдеуге) зерттеу жүргізуге мүмкіндік беретін көптеген бағдарламалық кешендер бар. Әрбір бағдарламалық ортаның өз құралдары бар және олар ақпараттық объектілердің белгілі бір түрлерімен жұмыс істеуге мүмкіндік береді.

3-кезең. Компьютерлік эксперимент.

Жаңа конструкторлық зерттеулерге өмір беру үшін, жаңа техникалық шешімдерді өндіріске енгізу үшін немесе жаңа ойларды тексеру үшін эксперимент қажет. Бұдан біраз уақыт бұрын мұндай тәжірибені не сол үшін әдейі жасалған қондырғыларда зертханалық жағдайда, не табиғи түрде, яғни оны жан-жақты сынап, заттың нағыз үлгісінде өткізу мүмкін болатын.

Есептеу техникаларының дамуымен зерттеудің жаңа, ғажап тамаша әдісі пайда болды, ол - компьютерлік тәжірибе. Компьютерлік тәжірибе модельмен

жұмыс істеудің қайсыбір жүйелілігін, компьютерлік модельді пайдаланушының мақсатты бағыттағы іс-әрекетінің жиынтығын қамтиды.

1. Модельдеу жоспары. Бұл кезеңде модельденіп жатқан объект туралы ақпарат құрылымданып, компьютерде өңделуге дайындалады, модельдің абстрактілі сипаттау тұжырымдамасынан нақты толыққан тұжырымдамасына өту қажет. Модельдеу тұжырымдалған кезде оны зерттеу әдісі пайда болады.

2. Модельдеу технологиясы. Бұл шығармашылық және қиын жобаланатын процесс. Қазіргі кезде құрылымдық және объектіге бағытталған бағдарламалаудың амалдары кеңінен таралған. Бағдарламалау тілін таңдау әдетте программистің тәжірибесіне, стандарттық бағынқы бағдарламалар және ашық кітапханалар болуына байланысты. Кейбір жағдайларда есептеулерді дайын бағдарламалық өнімдерді (мысалы, электронды кестелерді немесе арнайы математикалық пакеттерді) пайдаланып өткізу ыңғайлы. Бағдарлама құрастырғаннан кейін оның көмегімен кездесетін қателерді түзету мақсатында қарапайым тестілік есептер (жауабы алдын-ала белгілі) шешіледі.

4-кезең. Модельдеу нәтижелерін талдау.

Модельдеудің соңғы кезеңі - алынған нәтижелерді жан-жақты талдау негізінде түзілуі тиіс шешім қабылдау. Бұл кезең шешуші – сіз зерттеуді не жалғастырасыз, не тоқтатасыз.

Егер ЭЕМ-де алынған кейбір объектілердің (процестердің) сипаттамалары эксперимент нәтижелерімен нақтылықтың берілген дәрежесінде сәйкестенсе, модель нақты процеске тепе-тең. Модель нақты процеске сәйкес келмеген жағдайда алдыңғы кезеңдердің біріне қайтып ораламыз. Қайтудың мүмкін нүктелері сұлбада көрсетілген: модельдеу процесінде қайсыбір маңызды факторлар алынып тасталған немесе елеусіз факторлар тым көп алынған. Осыған орай модельді нақтылау қажет. Зерттеу әдісін таңдау онша сәтті болмаған, сондықтан оның неғұрлым күрделісін және нақтысын қолдану қажет. Өзгерістерді енгізгеннен кейін технологиялық тізбек бөлігінен тағы өтеміз де, қолайлы нәтижелер алғанша қайталай береміз.

Егер нәтижелер эксперименттік деректерге немесе біздің интуициялық деректерімізге сай болса, бағдарлама бойынша есептеулер жүргіземіз. Нәтижелер жинақталып өңделеді.

Оқыту үрдісінде демонстрациялық модельдер жиі пайдаланылады. Олардың міндеті – белгілі бір ғылыми мәселелерді толық меңгеру үшін қолайлы жағдайлар жасау. Сондықтан мұндай модельдерді пайдаланғанда түпнұсқаның күрделілік дәрежесін, сонымен қатар психологиялық және дидактикалық кезеңдерді есепке алу керек. Демонстрациялық модельдер белгілі бір зерттелетін заттың, құбылыстың қажетті элементтерін бөліп алуға мүмкіндік береді. Техникалық объектіде көрінбейтін тораптарды көруге, процестің кинематикасын баяу түрде көруге, зерттелетін процестің белгілі бір күйін тоқтатып қойып көруге, процесті қанша болса, сонша рет қайталап көруге, сезім арқылы қабылдауға болмайтын объектілерді көрнекі түрде түсінуге, мектепте жасауға мүмкін емес техникалық объектілер мен құбылыстарды зерттеуге мүмкіндік береді.

Әр уақытта модельдің түпнұсқаға қаншалықты сәйкес келетіндігін анықтап отыру керек. Бұдан оқушылар құбылыстар мен объектілер туралы мағлұматымыз салыстырмалы екендігіне сенімін арттырып, сонымен бірге табиғатты тануда бірте-бірте шындыққа келетінімізді түсінетін болады. Тану процесі шексіз болады деген сенімге келеді.

Қорытындылай келе, А.Ф.Иоффенің модельдеу туралы айтқан мынадай пікірін келтірейік: “Құбылыстарды оқыту үрдісінде модельдеудің зор пайдасы бар. Дәл жасалған модель белгілі дерек жөніндегі қорытындыны оңайлатады. Және ғылымды ілгері бастыруға мүмкіндік береді, жаңа тәжірибелер жасауға жағдай туғызады. Сонымен бірге модель құбылыстың қарапайым түрін, сұлбасын ғана көрсетеді”.

ТЕСТ

Информатиканың бағыттары ...

- A) теориялық информатика, бағдарламалау, жасанды интеллект, ықтималды логика, математикалық логика, табиғаттағы информатика
- B) теориялық информатика, бағдарламалау, математикалық логика, жасанды интеллект, информациялық жүйе, формалды логика
- C) информациялық жүйе, теориялық информатика, кибернетика, ықтималды логика, қоғамдағы информатика
- D) теориялық информатика, бағдарламалау, табиғаттағы информатика, математикалық логика, есептеуіш техника
- E) теориялық информатика, кибернетика, бағдарламалау, жасанды интеллект, информациялық жүйе, қоғамдағы информатика, есептеуіш техника, табиғаттағы информатика

Келесі тұжырымдардың қайсысының информатикаға қатысы жоқ

- A) ақпараттарды компьютерлер көмегімен түрлендіру және компьютерлердің қолдану ортасымен өзара әрекеттесу процесстерімен байланысты адам қызметінің саласы
- B) әр түрлі – техникалық, биологиялық, әлеуметтік және т.б. жүйелердегі басқарудың жалпы принциптері туралы ғылым
- C) компьютерлік ақпараттық жүйелер негізінде кез-келген объектілерді басқару процесстеріне ақпараттық жасаудың методологиясын құрумен айналысады
- D) ақпараттық процесстердегі заңдылықтарды (жинақтау, өңдеу, тарату) зерттеумен айналысады
- E) адам қызметінің әр түрлі саласындағы ақпараттық модельдердің байланыстарын құрумен айналысады

«Информация» термині ..

- A) Француздың түсіндіру, баяндау, білу деген ұғымдарды білдіретін informatio сөзінен шыққан
- B) Латынның түсіндіру, баяндау, білу деген ұғымдарды білдіретін informatio сөзінен шыққан
- C) Гректің түсіндіру, баяндау, білу деген ұғымдарды білдіретін informatio сөзінен шыққан
- D) Испанияның түсіндіру, баяндау, білу деген ұғымдарды білдіретін informatio сөзінен шыққан
- E) Немістің түсіндіру, баяндау, білу деген ұғымдарды білдіретін informatio сөзінен шыққан

«Информатика» термині ..

- A) Француздың информация және математика сөздерінен шыққан
- B) Ағылшынның информация және математика сөздерінен шыққан
- C) Гректің информация және техника сөздерінен шыққан
- D) Француздың информация және автоматика сөздерінен шыққан
- E) Латынның информация және электроника сөздерінен шыққан

Теориялық информатика нені зерттемейді:

- A) жалпы ақпараттар теориясы
- B) цифрлы автоматтар теориясы
- C) ақпараттандыру құралдары
- D) алгоритмдер теориясы
- E) ақпараттық модельдеу

Информатика іс-әрекеттің негізгі ... бағыттарын қамтитын ғылыми-техникалық пәндердің кешені

- A) екі
- B) алты
- C) төрт
- D) он алты
- E) сегіз

Техникада информация деп -

- A) бағыт алу, екпінді түрде әрекет ету, басқару үшін пайдаланылатын, яғни белгілі бір жүйені сақтау, дамыту, жетілдіру мақсатында қолданылатын білімнің бөлігін айтады
- B) қабылдау, сақтау, түрлендіру және басқаларға беру әрекеттері
- C) таңбалар немесе сигналдар түрінде берілетін кез-келген хабарларды айтады
- D) нақты дүние бейнесінің мәліметтер, хабарлар арқылы айтылуы
- E) қажет болған кез келген мәліметті немесе хабарды айтады

Кибернетикада информация деп -

- A) бағыт алу, екпінді түрде әрекет ету, басқару үшін пайдаланылатын, яғни белгілі бір жүйені сақтау, дамыту, жетілдіру мақсатында қолданылатын білімнің бөлігін айтады
- B) қабылдау, сақтау, түрлендіру және басқаларға беру әрекеттері
- C) таңбалар немесе сигналдар түрінде берілетін кез-келген хабарларды айтады
- D) нақты дүние бейнесінің мәліметтер, хабарлар арқылы айтылуы
- E) қажет болған кез келген мәліметті немесе хабарды айтады

Психология, физиология, лингвистика т.б. ғылымдар тоғысында тұрған күрделі мәселелерді шешуге мүмкіндік беретін информатика саласы

- A) кибернетика
- B) есептеу техникасы
- C) табиғаттағы информатика
- D) жасанды интеллект
- E) ақпараттық жүйе

Әр түрлі күрделі жүйелердегі ақпарат ағынын талдау, оларды тиімділеу, құрылымдау, сақтау принциптері мен ақпарат іздеуге байланысты сұрақтарды шешуге бағытталған информатика бөлімі

- A) кибернетика
- B) есептеу техникасы
- C) ақпараттық жүйе
- D) жасанды интеллект
- E) табиғаттағы информатика

Ақпараттық жүйе дегеніміз

- A) қойылған мақсатқа жетуде ақпаратты сақтау, өңдеу және шығару үшін қолданылатын құралдар, әдістер және тұлғалардың өзара байланысқан тобы
- B) кез келген объект, ол біруақытта бірбүтін және қойылған мақсатқа жету үшін әртүрлі элементтердің жиынтығы ретінде қарастырылады
- C) ақпаратты топтастыру және кодтаудың бірыңғай жүйесінің, құжаттандырудың бірыңғай жүйелерінің, ұйымдағы айналып жүретін ақпараттық ағындар схемаларының, сондай-ақ деректер базасын құру методологиясының жиынтығы
- D) ақпараттық жүйенің жұмыс істеуі үшін қажетті техникалық құралдар жиынтығы, сондай-ақ осы құралдар мен технологиялық процесстер үшін сәйкес құжаттар
- E) ақпараттық жүйелердің мақсаттары мен мәселелерін, сондай-ақ техникалық құралдардың жиынтығының дұрыс қызмет атқаруы үшін қажетті математикалық әдістер, модельдер, алгоритмдер және программалар жиынтығы

Ақпараттық жүйенің математикалық және программалық жасауы дегеніміз

- А) қойылған мақсатқа жетуде ақпаратты сақтау, өңдеу және шығару үшін қолданылатын құралдар, әдістер және тұлғалардың өзара байланысқан тобы
- В) кез келген объект, ол біруақытта бірбүтін және қойылған мақсатқа жету үшін әртүрлі элементтердің жиынтығы ретінде қарастырылады
- С) ақпаратты топтастыру және кодтаудың бірыңғай жүйесінің, құжаттандырудың бірыңғай жүйелерінің, ұйымдағы айналып жүретін ақпараттық ағындар схемаларының, сондай-ақ деректер базасын құру методологиясының жиынтығы
- Д) ақпараттық жүйенің жұмыс істеуі үшін қажетті техникалық құралдар жиынтығы, сондай-ақ осы құралдар мен технологиялық процесстер үшін сәйкес құжаттар
- Е) ақпараттық жүйелердің мақсаттары мен мәселелерін, сондай-ақ техникалық құралдардың жиынтығының дұрыс қызмет атқаруы үшін қажетті математикалық әдістер, модельдер, алгоритмдер және программалар жиынтығы

Ақпараттық жүйенің техникалық жасауы дегеніміз

- А) қойылған мақсатқа жетуде ақпаратты сақтау, өңдеу және шығару үшін қолданылатын құралдар, әдістер және тұлғалардың өзара байланысқан тобы
- В) кез келген объект, ол біруақытта бірбүтін және қойылған мақсатқа жету үшін әртүрлі элементтердің жиынтығы ретінде қарастырылады
- С) ақпаратты топтастыру және кодтаудың бірыңғай жүйесінің, құжаттандырудың бірыңғай жүйелерінің, ұйымдағы айналып жүретін ақпараттық ағындар схемаларының, сондай-ақ деректер базасын құру методологиясының жиынтығы
- Д) ақпараттық жүйенің жұмыс істеуі үшін қажетті техникалық құралдар жиынтығы, сондай-ақ осы құралдар мен технологиялық процесстер үшін сәйкес құжаттар
- Е) ақпараттық жүйелердің мақсаттары мен мәселелерін, сондай-ақ техникалық құралдардың жиынтығының дұрыс қызмет атқаруы үшін қажетті математикалық әдістер, модельдер, алгоритмдер және программалар жиынтығы

Ақпараттық технология бұл ...

- А) объект, процесс немесе құбылыстың (ақпараттық өнім) күйі туралы жаңа сапа туралы ақпарат алу үшін деректерді жинау, өңдеу және жеткізу (алғашқы ақпараттар) құралдары мен әдістерінің жиынтығын қолданатын процесс
- В) дербес компьютер және телебайланыс құралдарын қолданатын қолданушының жұмысының "достық" интерфейсі бар ақпараттық технология
- С) қолданушы қойған мақсатқа жетуге мүмкіндік беретін белгілі бір типті компьютер үшін бір немесе бірнеше өзара байланысқан программалық өнімнің жұмыс технологиясы
- Д) ұйымның алдыңғы қатарлы мақсаттарын жүзеге асыруда шешім қабылдауды қолдайтын компьютерлік ақпараттық жүйе
- Е) ақпараттық жүйелердің мақсаттары мен мәселелерін, сондай-ақ техникалық құралдардың жиынтығының дұрыс қызмет атқаруы үшін қажетті математикалық әдістер, модельдер, алгоритмдер және программалар жиынтығы

$x(t)$ функциясымен сипатталатын және $x(t)$ функциясының мәндері тек t_i , $i = 0 \pm 1 \pm 2, \dots$ аргументінің мәндерінің жиынында анықталған, ал $x(t_i)$ шамасы (x_{\min}, x_{\max}) интервалында кез келген мәнді қабылдай алатын сигналдардың түрлері:

- А) үздіксіз аргументтің үздіксіз функциясы
- В) дискретті аргументтің үздіксіз функциясы
- С) үздіксіз аргументтің дискретті функциясы
- Д) дискретті аргументтің дискретті функциясы
- Е) дискретті-үздіксіз аргументтің дискретті-үздіксіз функциясы

$x(t)$ функциясымен сипатталатын және $x(t)$ функциясының және t аргументінің мәндері x_1, x_2, \dots, x_k және t_1, t_2, \dots, t_k сандар қатарын құрайтын, (x_{\min}, x_{\max}) және $(-T, T)$ интервалдарын толтыратын сигналдардың түрлері мынаған сәйкес:

- A) үздіксіз аргументтің үздіксіз функциясы
- B) дискретті аргументтің үздіксіз функциясы
- C) үздіксіз аргументтің дискретті функциясы
- D) дискретті аргументтің дискретті функциясы
- E) дискретті-үздіксіз аргументтің дискретті-үздіксіз функциясы

Уақыт бойынша кванттау, немесе дискреттеу деп

- A) t үздіксіз аргументті $x(t)$ сигналын t_i аргументті $x(t_i)$ дискретті сигналына түрлендіру амалын
- B) $x(t_i)$ сигналдарының мәндерінің үздіксіз жиынын $x_k, k = 0, 1, \dots, (m - 1); x_k \in (x_{\min}, x_{\max})$ мәндерінің дискретті жиынына түрлендіретін амалды
- C) $x(t)$ үздіксіз сигналын x және t координаттары бойынша дискретті сигналына түрлендіретін амалды
- D) x және t координаттары бойынша дискретті сигналды $x(t)$ үздіксіз сигналына түрлендіретін амалды
- E) $x_k, k = 0, 1, \dots, (m - 1); x_k \in (x_{\min}, x_{\max})$ дискретті мәндердің жиынын $x(t_i)$ сигналының мәндерінің үздіксіз жиынына түрлендіретін амалды айтамыз

Ақпаратты өлшеудің аддитивті өлшемі (Хартли өлшемі):

- A) $I_{cp} = -\sum_{i=1}^k p_i \log_2 p_i$, мұндағы p_i - i -ші оқиғаның ықтималдығы
- B) $I(g) = \log_2 N = n \log_2 q$, N санының берілген ұзындығы және тереңдігі
- C) $I_{цел} = \log_2 p_1 - \log_2 p_0 = \log_2 \frac{p_1}{p_0}$, мұндағы p_0 және p_1 — мақсатқа жетудің бастапқы (ақпарат алынғанға дейінгі) және соңғы (ақпарат алынғаннан кейінгі) ықтималдықтары
- D) $I_{nf} = \log_2 [1/m(i)] = -\log_2 m(i)$, мұндағы $m(i)$ - i оқиғасының мазмұндылығының өлшемінің функциясы
- E) $I_{cp} = \frac{n_1}{N} (-\log_2 p_1) + \frac{n_2}{N} (-\log_2 p_2) + \dots + \frac{n_k}{N} (-\log_2 p_k)$, мұндағы N тәжірибелердің мүмкін нәтижелерінің саны, ондағы k нртүрлі типтері, ал i -ші нәтиже n_i рет қайталанады

Оқиғаның энтропиясы келесі формуламен анықталады

- A) $I_{cp} = -\sum_{i=1}^k p_i \log_2 p_i$, мұндағы p_i - i -ші оқиғаның ықтималдығы
- B) $I(g) = \log_2 N = n \log_2 q$, N санының берілген ұзындығы және тереңдігі
- C) $I_{цел} = \log_2 p_1 - \log_2 p_0 = \log_2 \frac{p_1}{p_0}$, мұндағы p_0 және p_1 — мақсатқа жетудің бастапқы (ақпарат алынғанға дейінгі) және соңғы (ақпарат алынғаннан кейінгі) ықтималдықтары
- D) $I_{nf} = \log_2 [1/m(i)] = -\log_2 m(i)$, мұндағы $m(i)$ - i оқиғасының мазмұндылығының өлшемінің функциясы
- E) $K = 2^I$, мұндағы K тәжірибелердің мүмкін нәтижелерінің саны

Ақпаратты өлшеудің семантикалық өлшемі:

- A) $I_{cp} = -\sum_{i=1}^k p_i \log_2 p_i$, мұндағы p_i - i -ші оқиғаның ықтималдығы
- B) $I(g) = \log_2 N = n \log_2 q$, N санының берілген ұзындығы және тереңдігі

С) $I_{цел} = \log_2 p_1 - \log_2 p_0 = \log_2 \frac{p_1}{p_0}$, мұндағы p_0 және p_1 — мақсатқа жетудің бастапқы (ақпарат

алынғанға дейінгі) және соңғы (ақпарат алынғаннан кейінгі) ықтималдықтары

Д) $K = 2^I$, мұндағы K тәжірибелердің мүмкін нәтижелерінің саны

Е) $I_{cp} = \frac{n_1}{N}(-\log_2 p_1) + \frac{n_2}{N}(-\log_2 p_2) + \dots + \frac{n_k}{N}(-\log_2 p_k)$, где N мұндағы K тәжірибелердің мүмкін нәтижелерінің саны, оның k әртүрлі типті, ал i -ші нәтиже n_i рет қайталанады

Кедергісі жоқ байланыс каналдары бойынша ақпараттарды жіберу жылдамдығы мына формуламен анықталады:

А) $c = \lim_{T \rightarrow \infty} \log_2 k / T$

В) $v' = \lim_{T \rightarrow \infty} \frac{H(i) - H_i(i)}{T}$, мұндағы $H(i)$ — хабар көзінің энтропиясы; $H_i(i)$ — қабылданатын жақтағы хабар энтропиясы

С) $c = F_m \log_2(1 + W_c / W_u)$, мұндағы F_m — каналдың жиілік жолағы (Гц); W — сигналдың орташа қуаттылығы

Д) $v = \lim_{T \rightarrow \infty} (I / T)$

Е) $c = \lim_{T \rightarrow \infty} \frac{I_{\max}}{T}$

Кедергісі бар байланыс каналдары бойынша ақпараттарды жіберу жылдамдығы мына формуламен анықталады:

А) $c = \lim_{T \rightarrow \infty} \log_2 k / T$

В) $v' = \lim_{T \rightarrow \infty} \frac{H(i) - H_i(i)}{T}$, мұндағы $H(i)$ — хабар көзінің энтропиясы; $H_i(i)$ — қабылданатын жақтағы хабар энтропиясы

С) $c = F_m \log_2(1 + W_c / W_u)$, мұндағы F_m — каналдың жиілік жолағы (Гц); W — сигналдың орташа қуаттылығы

Д) $v = \lim_{T \rightarrow \infty} (I / T)$

Е) $c = \lim_{T \rightarrow \infty} \frac{I_{\max}}{T}$

Тьюринг машинасын қандай түрде көрсетуге болады?

- А) ақырлы автомат
- В) КБ-тіл
- С) грамматика
- Д) жүйе
- Е) алгоритмдік тіл

Жүйе түрінде қандай машинаны көрсетуге болады?

- А) ақырлы автомат
- В) Пост машинасы
- С) автомат
- Д) Тьюринг машинасы
- Е) Пост-Тьюринг машинасы

→ m командасын орындағаннан кейін Пост машинасының лентасының күйі қандай болады

	бастапқы	командадан кейін
A)		
B)		
C)		
D)		
E)		

← m командасын орындағаннан кейін Пост машинасының лентасының күйі қандай болады

	бастапқы	командадан кейін
A)		
B)		
C)		
D)		
E)		

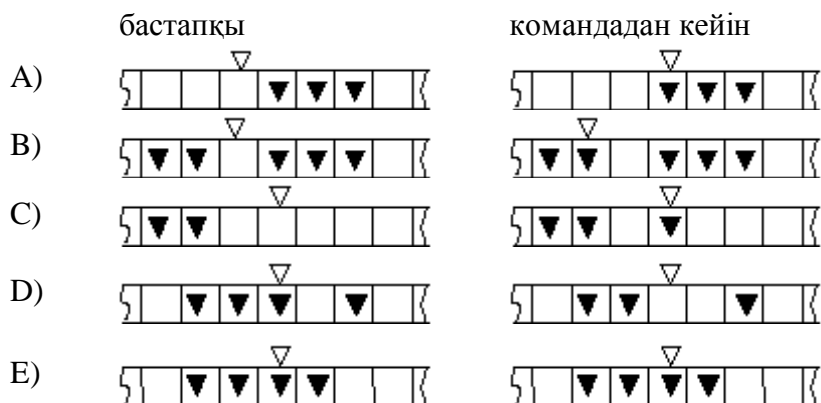
M m командасын орындағаннан кейін Пост машинасының лентасының күйі қандай болады

	бастапқы	командадан кейін
A)		
B)		
C)		
D)		
E)		

S m командасын орындағаннан кейін Пост машинасының лентасының күйі қандай болады

	бастапқы	командадан кейін
A)		
B)		
C)		
D)		
E)		

Тоқта n командасын орындағаннан кейін Пост машинасының лентасының күйі қандай болады



Пост машинасының автоматын іске қосқанда төмендегі жағдайдың қайсысы туындамайды

- A) автомат орындалмайтын командаға дейін орындап жетті (бос емес ұяшыққа белгіні жазу, бос ұяшықта белгіні өшіру); программаның орындалуы тоқтатылады, автомат тоқтайды, нәтижесіз тоқтату болады
- B) автомат тоқта командасына дейін жетті, программа орындалды деп есептеледі, нәтижесі бар тоқтату болады
- C) автомат нәтижелі де, нәтижесіз де тоқтатуға жетпеді, шексіз жұмыс істеу болады (автомат «тұрып қалады»)
- D) автомат лента аяғына дейін жетті; программа жұмысы тоқтайды, автомат тоқтайды, нәтижесіз тоқтау болады
- E) автомат лентада белгіні жазуды жүзеге асырды және келесі команданы орындауға көшеді

Тьюринг машинасының қай жерінде жазу орындалады?

- A) жады;
- B) лента;
- C) процессор
- D) қатар.
- E) бастиекте

Эмулятор қалай жұмыс жасайды?

- A) клавиатурадан екі сан енгізіледі - регуляторлық тілдердің бастапқы және қайтарған күйлері
- B) клавиатурадан екі сан енгізіледі - регуляторлық тілдердің жіберілген және қайтарған күйлері
- C) клавиатурадан үш сан енгізіледі - бастапқы нөмір регуляторлық тілдердің жіберілген және қайтарған күйлері
- D) клавиатурадан үш сан енгізіледі - бастапқы нөмір күйлері
- E) клавиатурадан үш сан енгізіледі бастапқы нөмір күйлері

Эмулятор программасы қалай саналады?

- A) бастапқы лентаның құрамы пробел арқылы жазылып, әрбір екі элементтен тұрады
- B) бастапқы лентаның құрамы және ережелер жиынтығы пробел арқылы жазылып, әрбір 5 элементтен тұрады
- C) соңғы лентаның құрамы және ережелер жиынтығы пробел арқылы 5 элементтен тұрады
- D) бастапқы лентаның құрамы пробел арқылы жазылып, әрбір үш элементтен тұрады
- E) соңғы лентаның құрамы және ережелер жиынтығы пробел арқылы 3 элементтен тұрады

Пробелдік лента символы қандай символмен белгіленеді?

- A) асты сызылған
- B) слэш
- C) төменгі сызылуы
- D) бөлшек
- E) жоғары сызылуы

Тьюринг машинасының жұмыс істеу процесінде не байқалады?

- A) жұмысты басынан бастау;
- B) жұмысты тоқтату;
- C) кіріс жолын сырғыту;
- D) жолдан шығу.
- E) жұмысты екінші жолдан сырғыту

Тьюринг машинасының конфигурациясы дегеніміз

- A) $\{0, 1, \dots, N-1\}$ алфавитінің символдары
- B) операциялық орындайтын құрылғының $\{q_1, \dots, q_r\}$ дискретті күйлерінің жиыны
- C) sDq символдарының үштігі, мұндағы s – машинаның сыртқы алфавитінің символы, q – машинаның ішкі алфавитінің символы және D – қозғалысты сипаттайтын алфавиттің символы, яғни $\{R, L, S\}$ жиыны
- D) qa түріндегі тізбек, мұндағы a - лентаның құрамы, q – бастиектің ағымдағы күйі, ал оның орны және арасындағы көрінетін ұяшықты нұсқайды
- E) q_0a түріндегі тізбек, яғни бұл жағдайда бастиек лентаның сол жақ шетіне қозғалған

Тьюринг машинасының сыртқы алфавиті дегеніміз

- A) $\{0, 1, \dots, N-1\}$ алфавитінің символдары
- B) операциялық орындайтын құрылғының $\{q_1, \dots, q_r\}$ дискретті күйлерінің жиыны
- C) sDq символдарының үштігі, мұндағы s – машинаның сыртқы алфавитінің символы, q – машинаның ішкі алфавитінің символы және D – қозғалысты сипаттайтын алфавиттің символы, яғни $\{R, L, S\}$ жиыны
- D) qa түріндегі тізбек, мұндағы a - лентаның құрамы, q – бастиектің ағымдағы күйі, ал оның орны және арасындағы көрінетін ұяшықты нұсқайды
- E) q_0a түріндегі тізбек, яғни бұл жағдайда бастиек лентаның сол жақ шетіне қозғалған

Тьюринг машинасының ішкі алфавиті дегеніміз

- A) $\{0, 1, \dots, N-1\}$ алфавитінің символдары
- B) операциялық орындайтын құрылғының $\{q_1, \dots, q_r\}$ дискретті күйлерінің жиыны
- C) sDq символдарының үштігі, мұндағы s – машинаның сыртқы алфавитінің символы, q – машинаның ішкі алфавитінің символы және D – қозғалысты сипаттайтын алфавиттің символы, яғни $\{R, L, S\}$ жиыны
- D) qa түріндегі тізбек, мұндағы a - лентаның құрамы, q – бастиектің ағымдағы күйі, ал оның орны және арасындағы көрінетін ұяшықты нұсқайды
- E) q_0a түріндегі тізбек, яғни бұл жағдайда бастиек лентаның сол жақ шетіне қозғалған

Тьюринг машинасының келтірілген функционалды схемасы қандай есепті шешеді

a	0	1	2	3	4	5	Δ
q	$z1S$	$z2S$	$z3S$	$z4S$	$z5S$	$q0L$	$z1S$

- A) бестік санау жүйесіндегі екі санды қосу
- B) бестік санау жүйесіндегі екі санды азайту
- C) бестік санау жүйесіндегі санға бірді қосу
- D) бестік санау жүйесіндегі саннан бірді азайту
- E) ондық санау жүйесіндегі санды бестік санау жүйесіне ауыстыру

Тьюринг машинасының келтірілген функционалды схемасы қандай есепті шешеді

A	q
Δ	$z \Delta S$
a	qbR
b	qaR

- A) бірінші таңбаны сөз соңына ауыстыру
- B) соңғы таңбаны сөз басына ауыстыру
- C) a символын b символымен ауыстыру
- D) b символын a символымен ауыстыру
- E) a символын b символымен ауыстыру және b символын a символымен ауыстыру

Пост машинасы үшін келтірілген программа көмегімен қандай есеп шешіледі

1. M^2
2. $\rightarrow 3$
3. M^4
4. тоқта

- A) бос лентаға екі белгі жазу
- B) бастиекті солға бірінші бос позицияға дейін қозғалту
- C) бірнеше санның біреуіне бірді қосу
- D) екі көрші ұяшыққа анализ жасау (кіріс ұяшықтары) және үшінші ұяшықты өзгерту (шығыс ұяшығы). Егер кіріс ұяшықтары белгіленген болса, онда шығысында белгі өшіріледі; егер кіріс ұяшықтарының ең болмағанда біреуінде белгі болмаса, онда шығыс ұяшықта белгі қойылады
- E) лентада кез келген ара қашықтықта жазылған a және b екі санын қосу

1930 жылдардың ортасында алгоритмдер теориясында кімдердің жұмыстары басылымға шықты?

- A) Тьюринг, Черче, Пост;
- B) Евклид, Макаров, Калмагоров;
- C) Пост, Макаров, Калмагоров;
- D) Макаров, Черче, Калмагоров.
- E) Макаров, Черче, Пост

1950 жылдары алгоритм теориясының дамуына кімдердің жұмыстары енгізілді?

- A) Тьюринг, Черче, Ульман;
- B) Калмагоров, Марков, Пост, Тьюринг, Черче;
- C) Евклид, Кудрявцев, Кнут;
- D) Кудрявцев, Кнут, Тьюринг, Черче.
- E) Кнут, Тьюринг, Черче.

q_2 күйіндегі AA кірісіне $a_1a_2a_1a_1a_2$ кіріс тізбегі берілсін. Осыған сәйкес шығыс тізбегі қандай болмақ, егер автоматты кесте түрі мынадай болса

	a_1	a_2
q_1	$q_2, 1$	$q_1, 0$
q_2	$q_1, 0$	$q_2, 1$

- A) 10110;
- B) 01100;
- C) 01001;
- D) 10011.

Е) 10001

q_2 күйіндегі АА кірісіне $a_1a_2a_1a_2a_2$ кіріс тізбегі берілсін. Осыған сәйкес шығыс тізбегі қандай болмақ, егер автоматты кесте түрі мынадай болса

	a_1	a_2
q_1	$q_{2,1}$	$q_{1,0}$
q_2	$q_{1,0}$	$q_{2,1}$

- A) 01011;
- B) 01100;
- C) 11011;
- D) 10011.
- E) 10101

Екі ақырғы автоматтарды қалай атайды?

- A) эквивалентсіз;
- B) жүйелі;
- C) эквивалентті;
- D) регулярлы.
- E) микрокалькулятор

Ақырғы автоматтың ұсыну тәсілдері қандай?

- A) сызықтық (өту диаграмасы);
- B) таблицалық (өту диаграмасы);
- C) символдық (өту диаграмасы);
- D) графикалық (өту диаграмасы), таблицалық;
- E) тармақтық (өту диаграмасы)

Екі ақырғы автоматтар эквивалентті деп аталады, егер

- A) кіріс алфавиттері бірдей болса;
- B) ішкі күй сандары бірдей болса;
- C) бірінші автомат күйі екінші автомат күйіне эквивалентті болса және керісінше;
- D) шығыс алфавиттері бірдей болса.
- E) кіріс және шығыс алфавиттері бірдей болса

Ең алғашқы пайда болған есептеу құралы ... болып есептелінеді.

- A) арифмометр
- B) абак
- C) калькулятор
- D) логика алгебрасы
- E) микрокалькулятор

Абак қазіргі ... ұқсас болған, ағаштан жасалған, санау үшін сүйектер мен тастардың тізбегі пайдаланылған.

- A) бухгалтерлік шотқа
- B) калькуляторға
- C) ойлау тәсілдеріне
- D) компьютерге
- E) есептеу таяқшаларына

1674 жылы қосу және азайту амалдарымен қатар, екі санды көбейту амалын орындай алатын механикалық есептеу машинасын жасаған ғалым ...

- A) Блез Паскаль
- B) *Леонардо да Винчи*
- C) Джон Непьер
- D) Вильгельм Годфрид фон Лейбниц
- E) Томас де Кольмард

... жылы Блез Паскаль 5-таңбалы сандармен арифметикалық амалдар орындай алатын алғашқы есептеуіш машинасын ойлап тапты.

- A) 1644
- B) 1896
- C) 1648
- D) 1642
- E) 1844

Вильгельм Годфрид фон Лейбниц механикалық есептеуіш машинасының жобасын ... жылы дайындап шықты.

- A) 1674
- B) 1962
- C) 1646
- D) 1642
- E) 1647

Аздаған өзгертулер арқылы, 90 жыл көлемінде пайдаланылған бірінші механикалық калькуляторды 1820 жылы кім жасады?

- A) Филипп-Малтус Хан
- B) Блез Паскаль
- C) Вильгельм Годфрид фон Лейбниц
- D) Джон Непьер
- E) Томас де Кольмард

Бірінші механикалық калькулятор ... жылы жасалынды?

- A) 1964
- B) 1962
- C) 1820
- D) 1642
- E) 1800

Чарльз Бэббидждің перфокарталарға жазылған программаларды орындайтын программаланатын қарапайым аналитикалық машинасы ... жылы жасалды.

- A) 1834
- B) 1934
- C) 1836
- D) 1936
- E) 1832

Бэббидж машинасында мәліметтерді енгізу құрылғысын көрсет.

- A) электрондық схема
- B) транзистор
- C) пластина
- D) перфокарта
- E) пернетақта

1623 жылы 7-таңбалы сандармен қарапайым арифметикалық амалдар (қосу, азайту) орындай алатын есептеуіш машинаны жасап шығарған ғалым

- A) Филипп-Малтус Хан
- B) Блез Паскаль
- C) Уильям Шикард
- D) Джон Непьер
- E) Томас де Кольмард

Бэббидж машинасында ақпарат ... көмегімен енгізілді?

- A) транзистормен
- B) перфокарта
- C) схемамен
- D) сымдармен
- E) пернетақта

Перфокартаны пайдалану идеясы ... жылы жүзеге асты.

- A) 1800
- B) 1964
- C) 1961
- D) 1980
- E) 1806

Джорж Буль бульдік алгебра деп аталатын өзінің «Логиканы математикалық талдау» еңбегін ... жылы жариялады.

- A) 1841
- B) 1947
- C) 1847
- D) 1980
- E) 1846

Джорж Бульдің математикалық логика идеяларын ары қарай дамытуға арналған «Ойлау заңдылықтарын зерттеу» атты еңбегі ... жылы жарыққа шықты.

- A) 1841
- B) 1947
- C) 1847
- D) 1980
- E) 1854

Бэббидж компьютердің аналитикалық құрылғысы прототипін және баспа құрылғысы – принтерді ... жылы жасап шығарды.

- A) 1841
- B) 1947
- C) 1847
- D) 1980
- E) 1871

Джон Атанасо Клифорд Берримен бірге ... жылдар аралығында теңдеулерді шешуге арналған алғашқы цифрлық электрондық есептеу машинасын жасаумен айналысты.

- A) 1890
- B) 1937-1942
- C) 1961 -1964
- D) 1980 -1991

Е) 1922-1926

Атанасо-Берри деп аталған бұл жүйеде қазіргі замандағы цифрлық технологиямен қатар ... пайдаланылды.

- А) схемалар
- В) транзисторлар
- С) вакуумдық лампалар
- Д) сымдар
- Е) жартылай өткізгіштер

Джон Атанасо ... жылы компьютердің техникалық моделін жасады.

- А) 1841
- В) 1947
- С) 1847
- Д) 1980
- Е) 1936

Атанасо-Берри деп аталған бұл жүйеде қазіргі замандағы цифрлық технологиямен қатар ... пайдаланылды

- А) лампалар
- В) логикалық схемалар
- С) транзисторлар
- Д) сымдар
- Е) ашық интегралды схемалар

1943 жылы ... немістің «Энигма» атты машинасының шифрланған мәліметін шешетін «Колос» атты алғашқы электрондық компьютер жасады.

- А) Уолтер Броттейн
- В) Лейбниц
- С) Клифорд Берри
- Д) Ален Тьюринг
- Е) М.А.Ньюмен мен Т.Х.Флауерс

Алғашқы компьютерде ... деп аталатын вакуумдық лампалар пайдаланылды.

- А) лампа
- В) транзистор
- С) вакуум
- Д) триод
- Е) логикалық схемалар

Қай жылы 8008 деген 8-разрядты микропроцессор жасалды?

- А) 1947
- В) 1937
- С) 1972
- Д) 1980
- Е) 1976

Қай жылы 8080 микропроцессордың негізінде 8088 микропроцессор жобаланды.

- А) 1973
- В) 1937
- С) 1947
- Д) 1980

Е) 1983

Орталық процессор мен жедел жадының арасындағы жұмыс екі операциядан тұрады:

- А) ақпаратты жазу және ақпаратты оқу
- В) ішкі құрылғылар мен сыртқы құрылғылар арасындағы байланыс
- С) ақпаратты жадыға жазу, жадыдағы ақпаратты оқу
- Д) ақпаратты түрлендіру және ақпаратты сақтау
- Е) ақпаратты жадыға жазу, ақпаратты түрлендіру

Ақпаратты тасымалдау тәсіліне қарай интерфейс ... болып бөлінеді.

- А) параллель және тізбекті
- В) параллель, көлбеулі
- С) тізбекті, өткізгіштік
- Д) өткізгіштік, жұлдызша
- Е) параллель, жартылай тізбекті

Бір байт ақпарат сегіз өткізгішпен өтетіндіктен өткізгіштердің саны ... еселі болады.

- А) беске
- В) бірге
- С) сегізге
- Д) онға
- Е) екіге

Санау жүйесінің түрлері:

- А) позициялы, арифметикалық амалдар
- В) позициясыз, арифметикалық амалдар
- С) позициялы санау жүйесі, санау жүйесінің алфавиті
- Д) арифметикалық амалдар, санау жүйесінің алфавиті
- Е) позициялы, позициясыз

Он алтылық санау жүйесіндегі арифметикалық операцияларды ондық санау жүйесінде есептеп, операция нәтижелерін он алтылық санау жүйесінде жазыңыз: $2E,2+A5,4=$

- А) $31E,6_{p=16}$
- В) $7,6_{p=16}$
- С) $13,6_{p=16}$
- Д) $139,6_{p=16}$
- Е) $D3,6_{p=16}$

$12,5_8+0,43_8=$... сегіздік негіздегі сандардың қосындысы

- А) $12,93_8$
- В) $13,13_8$
- С) $12,48_8$
- Д) $0,170_8$
- Е) $11,11_8$

$351,7_8-23,1_8=$... сегіздік негіздегі сандардың айырымы

- А) $324,6_8$
- В) $328,6_8$
- С) $326,8_8$
- Д) $326,6_8$
- Е) $328,6_{10}$

716,075 және 57,037 сандарының айырмасын сегіздік жүйеде есептеңдер

- A) 659, 38₈
- B) 637,036₈
- C) 657,38₈
- D) 654,38₈
- E) 661,38₈

Ондық санау жүйесіндегі 25 және 5 сандарын екілік негіздегі санау жүйесіне түрлендіріп, бірінші санды екіншісіне бөліңіз

- A) 11₂
- B) 101₂
- C) 111₂
- D) 1010₂
- E) 110₂

24,61₈ және 3,02₈ сандарының көбейтіндісі

- A) 74,74₈
- B) 77,744₈
- C) 76,7442₈
- D) 72,74₈
- E) 74,3222₈

1101₂ және 1100₂ сандарының көбейтіндісі

- A) 1001₂
- B) 100110₂
- C) 101110₂
- D) 1010100₂
- E) 10011100₂

Позициялық емес санау жүйесінде санның жазылуын көрсетіңдер

A) $A_D = D_1 + D_2 + \dots + D_k = \sum_{i=1}^{i=k} D_i$

B) $B_i = q_i B_{i-1}$

C) $A_q = \sum_{i=-m}^{i=n} a_i q^i$

D) $A_q = a_n q^n + \dots + a_1 q^1 + a_0 q^0 + a_{-1} q^{-1} + \dots + a_{-m} q^{-m}$

E) $A_q = a_n \cdot 10^n + \dots + a_1 \cdot 10^1 + a_0 \cdot 10^0 + a_{-1} \cdot 10^{-1} + \dots + a_{-m} \cdot 10^{-m}$

Ондық санау жүйесінде санның жазылуын көрсетіңдер

A) $A_D = D_1 + D_2 + \dots + D_k = \sum_{i=1}^{i=k} D_i$

B) $B_i = q_i B_{i-1}$

C) $A_q = \sum_{i=-m}^{i=n} a_i q^i$

D) $A_q = a_n q^n + \dots + a_1 q^1 + a_0 q^0 + a_{-1} q^{-1} + \dots + a_{-m} q^{-m}$

E) $A_q = a_n \cdot 10^n + \dots + a_1 \cdot 10^1 + a_0 \cdot 10^0 + a_{-1} \cdot 10^{-1} + \dots + a_{-m} \cdot 10^{-m}$

$1 \cdot 8^2 + 2 \cdot 8^1 + 4 \cdot 8^0 + 5 \cdot 8^{-1} + 3 \cdot 8^{-2} + 7 \cdot 8^{-3}$ өрнегімен қандай сан жазылған

- A) 124,537
- B) $1001,1101_2$
- C) 2122_3
- D) $124,537_8$
- E) 21,22

$2 \cdot 3^3 + 1 \cdot 3^2 + 2 \cdot 3^1 + 2 \cdot 3^0$ өрнегімен қандай сан жазылған

- A) 124,537
- B) $1001,1101_2$
- C) 2122_3
- D) $124,537_8$
- E) 21,22

$A = 98$ ондық санын екілік ($q_2 = 2$) санау жүйесіне келтіріңдер

- A) 1100010
- B) 1101010
- C) 1000011
- D) 1001110
- E) 1000000

$A_2 = 1101001$ екілік санын ондық санау жүйесіне келтіріңдер

- A) 105
- B) 97
- C) 215
- D) 33
- E) 15

$A = 0,625$ ондық бөлшекті екілікке келтіріңдер

- A) $0,1010_2$
- B) $0,1100_2$
- C) $0,1001_2$
- D) $0,0110_2$
- E) $0,1110_2$

$A = 0,1011$, $B = -0,0100$ сандарының қосындысы қай қосындылауышта есептелген?

$$\begin{array}{r} [A] = 0,1011 \\ + \\ [B] = 1,1100 \\ \hline [C] = 0,0111 \end{array}$$

- A) тура кодтың екілік қосындылауышында;
- B) қосымша кодтың екілік қосындылауышында;
- C) кері кодтың екілік қосындылауышында;
- D) нормалданған кодтың екілік қосындылауышында;
- E) стандартталған кодтың екілік қосындылауышында.

$A = 0,0101$ және $B = -0,0111$ сандарының қосындысы қай қосындылауышта есептелген?

$$\begin{array}{r} [A] = 0,0101 \\ + \\ [B] = 1,1000 \\ \hline \end{array}$$

$$[C] = 1,1101$$

- A) тура кодтың екілік қосындылауышында;
- B) қосымша кодтың екілік қосындылауышында;
- C) кері кодтың екілік қосындылауышында;
- D) нормалданған кодтың екілік қосындылауышында;
- E) стандартталған кодтың екілік қосындылауышында.

$A = -0,101010$ сандары үшін кері және қосымша кодты табыңдар.

- A) $1,101010$ және $0,101010$;
- B) $1,010101$ және $1,010110$;
- C) $-0,101010$ және $-0,101010$;
- D) $1,111000$ және $1,000111$;
- E) $1,101010$ және $1,101010$.

$A = 0,101010$ сандары үшін кері және қосымша кодты табыңдар.

- A) $1,101010$ және $0,101010$;
- B) $1,010101$ және $1,010110$;
- C) $0,101010$ және $0,101010$;
- D) $1,111000$ және $1,000111$;
- E) $1,101010$ және $1,101010$.

A санының тура кодын көрсетіңдер

- A) $[A] = \begin{cases} A, & \text{егер } A \geq 0 \\ 1 + |A|, & \text{егер } A < 0 \end{cases}$
- B) $[A] = \begin{cases} A, & \text{егер } A \geq 0 \\ q - |A|, & \text{егер } A < 0 \end{cases}$
- C) $[A] = \begin{cases} A, & \text{егер } A \geq 0 \\ q - q^{-n} + A, & \text{егер } A < 0 \end{cases}$
- D) $A_n = m_A q^{p_A}$, мұндағы $q^{-1} \leq |m_A| < 1$
- E) $A = [A]_{\Phi} K_A$, мұндағы $-1 < [A]_{\Phi} < 1$

A санының қосымша кодын көрсетіңдер

- A) $[A] = \begin{cases} A, & \text{егер } A \geq 0 \\ 1 + |A|, & \text{егер } A < 0 \end{cases}$
- B) $[A] = \begin{cases} A, & \text{егер } A \geq 0 \\ q - |A|, & \text{егер } A < 0 \end{cases}$
- C) $[A] = \begin{cases} A, & \text{егер } A \geq 0 \\ q - q^{-n} + A, & \text{егер } A < 0 \end{cases}$
- D) $A_n = m_A q^{p_A}$, мұндағы $q^{-1} \leq |m_A| < 1$
- E) $A = [A]_{\Phi} K_A$, мұндағы $-1 < [A]_{\Phi} < 1$

Логика дегеніміз ...

- A) хабарламаның, берілген белгінің, жадының мазмұны
- B) дәлелді пікір заңдары туралы ғылым
- C) нақты әлемді мәліметтер және хабарламалардың көмегімен көрсету

- D) объектілер мен пікірлердің ақиқаттығын шешуге болатын дәйектемелерді ғана үйрететін ғылым
- E) айқындалған объектілер мен пікірлердің жалғандығын шешуге болатын дәйектемелерді ғана үйрететін ғылым

Ғылыми пән ретінде логиканың нұсқалары ... дараланады

- A) формалды логика, математикалық логика, адамның ойлау нысандары логикасы
- B) формалды логика, диалектикалық логика, информатиканың теориялық негізі логикасы, пікір заңдары логикасы
- C) ықтималды логика, диалектикалық логика, информатиканың теориялық негізі логикасы, ойлау нысандары логикасы
- D) формалды логика, математикалық логика, диалектикалық логика, пікір заңдары логикасы
- E) формалды логика, математикалық логика, диалектикалық логика, ықтималды логика

$x_1 \wedge x_2$ (конъюнкция) логикалық функциясының мәндерінің кестесін көрсетіңдер

A)

$x_1 x_2$	00	01	10	11
f	0	0	0	1

B)

$x_1 x_2$	00	01	10	11
f	0	1	1	1

C)

$x_1 x_2$	00	01	10	11
f	1	1	0	1

D)

$x_1 x_2$	00	01	10	11
f	1	0	0	1

E)

$x_1 x_2$	00	01	10	11
f	0	1	1	0

$x_1 \vee x_2$ (дизъюнкция) логикалық функциясының мәндерінің кестесін көрсетіңдер

A)

$x_1 x_2$	00	01	10	11
f	0	0	0	1

B)

$x_1 x_2$	00	01	10	11
f	0	1	1	1

C)

$x_1 x_2$	00	01	10	11
f	1	1	0	1

D)

$x_1 x_2$	00	01	10	11
f	1	0	0	1

E)

$x_1 x_2$	00	01	10	11
f	0	1	1	0

$x_1 \rightarrow x_2$ (импликация) логикалық функциясының мәндерінің кестесін көрсетіңдер

A)

$x_1 x_2$	00	01	10	11
f	0	0	0	1

B)

$x_1 x_2$	00	01	10	11
-----------	----	----	----	----

f	0	1	1	1
-----	---	---	---	---

C)

$x_1 x_2$	00	01	10	11
f	1	1	0	1

D)

$x_1 x_2$	00	01	10	11
f	1	0	0	1

E)

$x_1 x_2$	00	01	10	11
f	0	1	1	0

$x_1 \equiv x_2$ (тең мәнділі) логикалық функциясының мәндерінің кестесін көрсетіндер

A)

$x_1 x_2$	00	01	10	11
f	0	0	0	1

B)

$x_1 x_2$	00	01	10	11
f	0	1	1	1

C)

$x_1 x_2$	00	01	10	11
f	1	1	0	1

D)

$x_1 x_2$	00	01	10	11
f	1	0	0	1

E)

$x_1 x_2$	00	01	10	11
f	0	1	1	0

$x_1 \oplus x_2$ (2 модулі бойынша қосу) логикалық функциясының мәндерінің кестесін көрсетіндер

A)

$x_1 x_2$	00	01	10	11
f	0	0	0	1

B)

$x_1 x_2$	00	01	10	11
f	0	1	1	1

C)

$x_1 x_2$	00	01	10	11
f	1	1	0	1

D)

$x_1 x_2$	00	01	10	11
f	1	0	0	1

E)

$x_1 x_2$	00	01	10	11
f	0	1	1	0

Логика алгебрасында жалған мән былай белгіленеді:

A) 0

B) 2;

C) 0,1 ;

D) A

E) 1

Логика алгебрасында ақиқаттық мән былай белгіленеді:

- A) 0
- B) 9
- C) 1
- D) 8
- E) 0 және 1

A-ның ақиқаттық мәні нешеге тең?

$$\overline{A} = 1$$

- A) 1
- B) 4
- C) 3
- D) 0
- E) 0 және 1

Келесі өрнектің ақиқаттық мәні нешеге тең?

$$(0 \vee 1) \wedge 1$$

- A) 0
- B) 7
- C) 3
- D) 1
- E) 6

Келесі өрнектің ақиқаттық мәні нешеге тең?

$$(1 \vee 0) \wedge (1 \vee 0 \wedge 1)$$

- A) 3
- B) 1
- C) 4
- D) 0
- E) 6

Схема талдау (анализ) есебінің көмегімен:

- A) Дайын схеманың жұмысын талдайды
- B) Компьютер құрастырылады
- C) Логикалық схема құрастырылады
- D) Логикалық функцияға лайықты схеманы анықтау
- E) үлкен интегралды схемалар құрастырылады

Теріске шығару белгіленеді ...:

- A) \overline{A}
- B) A
- C) a
- D) a емес
- E) A немесе B

A=1, B=0, C=1, болса, келесі өрнектің ақиқаттық мәні нешеге тең: $(A \wedge B) \vee C =$

- A) 1
- B) 2
- C) 3
- D) 0
- E) 6

Екі айтылымның конъюнкциясы ... деп оқылады

- A) "A және A"
- B) "A немесе B"
- C) "A тең B"
- D) "A және B"
- E) "A емес B"

Екі айтылымның дизъюнкциясы ... деп оқылады

- A) "A және B"
- B) "A және A"
- C) "A тең B"
- D) "A емес B"
- E) "A немесе B"

Айтылымның терістеуі ... деп оқылады:

- A) "A және B"
- B) "A және A"
- C) "A емес"
- D) "A немесе A"
- E) "A немесе B"

Екі айтылымның тепе-теңдігін терістеу ... деп оқылады:

- A) "A B-ға тепе-тең емес"
- B) "A және B"
- C) "A емес"
- D) "A B-ға тепе-тең"
- E) "A немесе B"

Екі айтылымның импликациясы ... деп оқылады:

- A) "A және B"
- B) "A және B"
- C) "A емес"
- D) "Егер A онда B"
- E) "A B-ға тепе-тең"

Екі айтылымның үйлесімсіздігі ... деп оқылады:

- A) "A B-ға үйлесімсіз"
- B) "A және B"
- C) "A емес"
- D) "A немесе B"
- E) "A A-ға үйлесімді"

Айтылымның терістеуі ... деп белгіленеді:

- A) $A = B$
- B) \bar{A}
- C) $A \vee B$
- D) $A = B$
- E) $A + B$

Екі айтылымның импликациясы ... деп белгіленеді:

- A) $A \rightarrow B$
- B) $A \wedge B$
- C) $A \vee B$

- D) $A = B$
- E) $A + B$

$A=1, B=0, C=1$, болса, келесі өрнектің ақиқаттық мәні нешеге тең: $(\bar{A} \vee C) \wedge \bar{B}$

- A) A
- B) 1
- C) C
- D) 0
- E) 01

$A=1, B=0, C=1$, болса, келесі өрнектің ақиқаттық мәні нешеге тең: $A / (B \wedge C)$

- A) 2
- B) 0
- C) 1
- D) C
- E) B

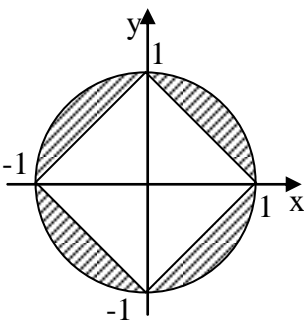
$A=1, B=0, C=1, F=0$ болса, келесі өрнектің ақиқаттық мәні нешеге тең: $(A \vee B) \wedge (C \vee F)$

- A) C
- B) F
- C) A
- D) 1
- E) 0

Логикалық «және» қалай аталады?

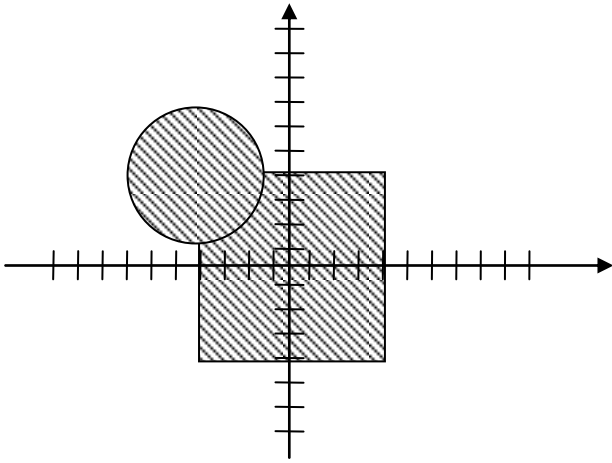
- A) терістеу
- B) дизъюнкция
- C) логикалық айырма
- D) толықтауыш
- E) конъюнкция.

Координаттары x және y болатын нүкте штрихталған облысқа жатқанда, x және y -ке тәуелді болатын ақиқат мәнін қабылдайтын логикалық өрнекті жазыңдар.



- A) $(x^2 + y^2 \leq 1) \wedge (|x| + |y| \geq 1)$
- B) $(x^2 + y^2 \leq 1^2) \vee (|x| + |y| \geq 1)$
- C) $(x^2 - y^2 \leq 1^2) \wedge (|x| + |y| \leq 1)$
- D) $(x^2 - y^2 \leq 1) \vee (|x| + |y| \leq 1)$
- E) $(x^2 + y^2 \geq R^2) \wedge (|x| - |y| \geq 1)$

Координаттары x және y болатын нүкте штрихталған облысқа жатқанда, x және y -ке тәуелді болатын ақиқат мәнін қабылдайтын логикалық өрнекті жазыңдар.



- A) $((x - 4)^2 + (y - 4)^2 \leq 9) \vee ((x \geq -4) \wedge (x \leq 4) \wedge (y \geq -4) \wedge (y \leq 4))$
 B) $((x + 4)^2 + (y - 4)^2 \leq 9) \vee ((x \geq -4) \wedge (x \leq 4) \wedge (y \geq -4) \wedge (y \leq 4))$
 C) $((x - 4)^2 + (y + 4)^2 \leq 9) \wedge ((x \geq -4) \vee (x \leq 4) \vee (y \geq -4) \vee (y \leq 4))$
 D) $((x + 4)^2 + (y + 4)^2 \geq 9) \wedge ((x \leq -4) \vee (x \geq 4) \vee (y \leq -4) \vee (y \geq 4))$
 E) $((x + 4)^2 + (y + 4)^2 \geq 9) \vee ((x \leq -4) \wedge (x \geq 4) \vee (y \leq -4) \wedge (y \geq 4))$

Логикалық алгебраның негізін қалаушы

- A) Пост.
 B) Тьюринг.
 C) Буль.
 D) Колмогоров.
 E) Ньютон.

Алгоритмдік тілде шамалардың үш типін көрсетіңіз?

- A) бүтін, нақты, литерлік;
 B) бүтін, тармақты, литерлік;
 C) нақтылық, тиімділік, шартты;
 D) нақты, сызықтық, тармақтық;
 E) бүтін, нақты, сызықтық

Алгоритмдік тілдегі шамалармен жұмыс жасаудағы басты командалардың бірі

- A) енгізу;
 B) меншіктеу;
 C) шығару;
 D) тармақтау;
 E) сызықтық

Алгоритмдік тілде сандарды жазудың екі түрлі әдісін атаңыз?

- A) меншіктеу, шығару;
 B) қарапайым және күрделі;
 C) қарапайым, экспоненциал;
 D) сызықты және күрделі
 E) меншіктеу, экспоненциал;

Алгоритм ұғымы –

- A) белгілі бір іс-әрекетті орындау үшін көрсетілген нұсқаулар;
 B) белгілі бір іс-әрекетті орындамау үшін көрсетілген нұсқаулар;
 C) Евклид алгоритмінде;

- D) формализация негізінде.
- E) арифметикалық амалдарды орындау ережелері

Геометриялық фигуралардың көмегімен сызылған алгоритм.

- A) сөздік алгоритм;
- B) графикалық алгоритм;
- C) пікірлер алгоритмі;
- D) есептік алгоритм;
- E) сызықтық алгоритм

Алгоритм қасиеттері:

- A) нәтижесіздік, анықтаусыз;
- B) нақтылық, шектеусіз;
- C) нәтижелік, анықтылық, нақтылық, шектеулік, жалпылық;
- D) нәтижелік, анықтылық, нақтылықсыз, шектеусіз ;
- E) үздіктілік, айқындылық, ортақтылық, нәтижелілігі, түсініктілігі

Алға қойылған мақсатқа жету немесе берілген есепті шешу бағытында арнайы ережелер бойынша орындаушыға жинақты түрде берілген нұсқаулар

- A) программа
- B) алгоритм
- C) жүйе
- D) система
- E) команда

Алгоритм сөзі қай ғалымның атының латынша аталуынан шыққан?

- A) Ал-Хорезмий
- B) Беруний
- C) Ибн-Сина
- D) Аль-Фараби
- E) Яссауи

Өңдеу үшін дайындалып, ЭЕМ-ге ендірілетін, компьютерде кодталған түрде пайда болған және шығарылатын

- A) берілгендер
- B) нәтиже
- C) жүйе
- D) операциялық жүйе
- E) айнымалылар

Алгоритм деп ... түсінеміз.

- A) алға қойылған мақсатқа жету барысында, біртіндеп қандай әрекеттер жасау керек екендігін, түсінікті және дәл көрсететін нұсқаулар мен жарлықтар тізбегін
- B) әрекеттер тізбегін
- C) анықталған мақсатқа жету үшін орындаушыға арналған, орындау тәртібі ұйымдастырылмаған түсінікті, ықшамды, шектелмеген нұсқаулар тізбегін
- D) тек адам атқарушыға арналған нұсқаулар тізбегін
- E) нұсқаулар мен жарлықтар тізбегі

Алгоритмнің негізгі қасиеттеріне жатпайды.

- A) үздіктілік
- B) айқындылық

- C) ортақтылық
- D) нұсқаулылық
- E) түсініктілігі

Алгоритмдегі кадамдардың құрылымы жағынан алгоритмдер ... болып бөлінеді.

- A) негізгі, көмекші
- B) сызықтық, тармақталған, қайталану
- C) процедуралық, функциялық
- D) нәтижелі, негізгі, сызықтық
- E) қайталану, тармақталу

Компьютер түсінетін командалар тізбегі

- A) серия
- B) жүйе
- C) программа
- D) операнд
- E) оператор

Алгоритмнің – «басы» қызметші сөзіне дейінгі алғашқы бөлігінің аталуы

- A) алгоритм тақырыбы
- B) алгоритм денесі
- C) программа
- D) алгоритм
- E) дұрыс жауап жоқ

«Басы ... соңы» қызметші сөздерінің аралығында орналасатын командалар тізбегінің аталуы

- A) алгоритм тақырыбы
- B) алгоритм денесі
- C) программа
- D) алгоритм шамасы
- E) дұрыс жауап жоқ

Алгоритмдік тілде алгоритмді басқару құрылымдары деп аталатын негізгі төрт құрылым бар. Осыған жатпайтын

- A) тізбектелу
- B) тармақталу
- C) таңдау
- D) қайталану (цикл)
- E) өңдеу

Блоктары (командалары) ретімен орындалатын алгоритмнің аталуы

- A) тармақталған
- B) циклдық
- C) сызықтық
- D) реттелген
- E) келтірілген

Тексеру нәтижесінде «иә» (true) немесе «жоқ» (false) мәндерін қабылдайтын логикалық өрнектің аталуы

- A) бұйрық
- B) шарт
- C) серия

- D) алгоритм
- E) программа

Алгоритмнің блок-схемасында шартты тексеру блогын өрнектейтін фигура

- A) шеңбер
- B) алтыбұрыш
- C) ромб
- D) параллелограмм
- E) төртбұрыш

Алгоритм блок-схемасында цикл басы мен соңын өрнектейтін фигураның аталуы

- A) шеңбер
- B) алтыбұрыш (модификация)
- C) ромб
- D) параллелограмм
- E) төртбұрыш

Алгоритм блок-схемасындағы информацияны өңдеу блогы

- A) параллелограмм
- B) төртбұрыш
- C) ромб
- D) алтыбұрыш
- E) шеңбер

Тармақталу құрылымын бастайтын команда

- A) егер
- B) алг
- C) басы
- D) бітті
- E) онда

Тармақтау командасында «егер» қызметші сөзінен кейін не келеді

- A) шарт
- B) алгоритм денесі
- C) алгоритм тақырыбы
- D) алг қызметші сөзі
- E) басы қызметші сөзі

Алгоритмдегі қадамдардың құрылымы жағынан алгоритмдер ... болып бөлінеді.

- A) негізгі, көмекші
- B) сызықтық, тармақталған, қайталану
- C) процедуралық, функциялық
- D) нәтижелі, негізгі, сызықтық
- E) қайталану, тармақталу

Алгоритмнің нақтылық (түсініктілігі және анықтылығы) қасиеті ...

- A) ақпаратты өңдеу процесі ретімен жазылған жеке-жеке нұсқаулардан тұратын тізбектен тұруы тиіс және бір қадамдағы команда, яғни бұйрық толық орындалғаннан кейін ғана екінші қадамдағы бұйрықтар орындалуы тиіс.
- B) қарастырып отырған ақпараттың кез-келген алғашқы берілгендеріне бірдейлігі.
- C) алгоритмге енетін берілгендердің түрлі-түрлі болуы

- D) орындаушы алгоритмді түсініп, орындай алатын болуы керек, түрлі түрде түсінілетін нұсқаулар енгізілмеуі тиіс. Орындаушыға алгоритмді орындау үшін басқа нұсқаулар іздеуіне жол қалдырмайтындай етіліп, орындалу реттері дәл көрсетіліп жазылуы
- E) нұсқаулар шексіз көп болмай, мүмкіндігі барынша олардың санын кемітуге әрекет жасалуы керек, алгоритмнің қорытындысында оның нәтижесі болуы тиіс

Алгоритмнің дискреттілік қасиеті ...

- A) орындаушы алгоритмді түсініп, орындай алатын болуы керек, түрлі түрде түсінілетін нұсқаулар енгізілмеуі тиіс. Орындаушыға алгоритмді орындау үшін басқа нұсқаулар іздеуіне жол қалдырмайтындай етіліп, орындалу реттері дәл көрсетіліп жазылуы
- B) нұсқаулар шексіз көп болмай, мүмкіндігі барынша олардың санын кемітуге әрекет жасалуы керек, алгоритмнің қорытындысында оның нәтижесі болуы тиіс
- C) ақпаратты өңдеу процесі ретімен жазылған жеке-жеке нұсқаулардан тұратын тізбектен тұруы тиіс және бір қадамдағы команда, яғни бұйрық толық орындалғаннан кейін ғана екінші қадамдағы бұйрықтар орындалуы тиіс.
- D) алгоритмге енетін берілгендердің түрлі-түрлі болуы
- E) қарастырып отырған ақпараттың кез-келген алғашқы берілгендеріне бірдейлігі.

Алгоритмнің жалпылық (бірдейлігі) қасиеті ...

- A) ақпаратты өңдеу процесі ретімен жазылған жеке-жеке нұсқаулардан тұратын тізбектен тұруы тиіс және бір қадамдағы команда, яғни бұйрық толық орындалғаннан кейін ғана екінші қадамдағы бұйрықтар орындалуы тиіс.
- B) қарастырып отырған ақпараттың кез-келген алғашқы берілгендеріне бірдейлігі.
- C) алгоритмге енетін берілгендердің түрлі-түрлі болуы
- D) орындаушы алгоритмді түсініп, орындай алатын болуы керек, түрлі түрде түсінілетін нұсқаулар енгізілмеуі тиіс. Орындаушыға алгоритмді орындау үшін басқа нұсқаулар іздеуіне жол қалдырмайтындай етіліп, орындалу реттері дәл көрсетіліп жазылуы
- E) нұсқаулар шексіз көп болмай, мүмкіндігі барынша олардың санын кемітуге әрекет жасалуы керек, алгоритмнің қорытындысында оның нәтижесі болуы тиіс

Алгоритмнің нәтижелілік қасиеті ...

- A) ақпаратты өңдеу процесі ретімен жазылған жеке-жеке нұсқаулардан тұратын тізбектен тұруы тиіс және бір қадамдағы команда, яғни бұйрық толық орындалғаннан кейін ғана екінші қадамдағы бұйрықтар орындалуы тиіс.
- B) қарастырып отырған ақпараттың кез-келген алғашқы берілгендеріне бірдейлігі.
- C) алгоритмге енетін берілгендердің түрлі-түрлі болуы
- D) орындаушы алгоритмді түсініп, орындай алатын болуы керек, түрлі түрде түсінілетін нұсқаулар енгізілмеуі тиіс. Орындаушыға алгоритмді орындау үшін басқа нұсқаулар іздеуіне жол қалдырмайтындай етіліп, орындалу реттері дәл көрсетіліп жазылуы
- E) нұсқаулар шексіз көп болмай, мүмкіндігі барынша олардың санын кемітуге әрекет жасалуы керек, алгоритмнің қорытындысында оның нәтижесі болуы тиіс

Машиналық команда -

- A) команда да көрсетілген амалға енетін сан
- B) жоғарғы есепті шешу үшін құрылған алгоритмге енгізілген командалар
- C) машинаға енгізілетін программалар
- D) команда да көрсетілген амалға енетін символ
- E) қарапайым операцияны орындау үшін компьютерге берілетін нұсқаулар

Құрама шарттарда жай шарттарды біріктіруде қолданылатын қызметші сөз

- A) алг
- B) және
- C) басы

- D) соңы
- E) егер

Қайталану командасының түрлері:

- A) «N рет», «Әзірше», «Әдетте», «Цикл басы»
- B) «Дейін», «Үшін», «Цикл басы», «Цикл соңы»
- C) «N рет», «Әзірше», «Дейін», «Үшін»
- D) «Әзірше», «Әдетте», «Дейін», «Үшін»
- E) «N рет», «Әзірше», «Цикл басы», «Цикл соңы»

Циклдің қай құрылымында цикл денесі бір рет те орындалмауы мүмкін

- A) әзірше
- B) n-рет
- C) үшін
- D) дейін
- E) барлығында да

Қайталану командасының қай құрылымында шарттың сақталуы аяқталғанға дейін цикл денесі қайталанады

- A) N рет
- B) Дейін
- C) Әзірше
- D) Әдетте
- E) Үшін

«...» циклін – параметрлі қайталау циклі деп те атайды

- A) «Дейін»
- B) «Әзірше»
- C) «Әдетте»
- D) «N рет»
- E) «Үшін»

Тармақталған алгоритмде қандай команда қолданылады.

- A) Таңдау
- B) Тізбектелу
- C) Өңдеу
- D) Қайталану
- E) Сызықтық

Блок-схема деп ... айтамыз.

- A) алгоритмнің стрелкалармен сипатталып берілуін
- B) алгоритмнің сөзбен жазылуын
- C) программалау тілінің командалары арқылы жазылуын
- D) алгоритмнің көркем сөзбен жазылуын
- E) алгоритмнің бағытталған байланысы бар геометриялық фигуралармен сипатталып берілуін

Бағдарламада өрнектің түрлі мәндерін есептеуге немесе бір типті әрекеттерді бірнеше рет қайталап орындауда пайдаланылатын, алгоритмнің дербес бөлігі

- A) циклдік алгоритм
- B) сызықтық алгоритм
- C) тармақталу алгоритмі

- D) Евклид алгоритмі
- E) қосалқы алгоритм

Алгоритмдік тілде бір типті шамалардың бір ғана ат беріліп реттелген тізбегі ... деп аталады.

- A) көмекші алгоритм
- B) негізгі алгоритм
- C) цикл
- D) массив
- E) матрица

Тармақталған алгоритмдерде ... командалар қолданылады

- A) тармақталу және циклдік командалар
- B) сызықтық және тармақталу командалары
- C) таңдау және циклдік командалар
- D) сызықтық және таңдау командалары
- E) тармақталу және таңдау командалары

Тармақталу командасының қызметші сөздері

- A) алг, арг, нәт, бер
- B) басы, соңы
- C) n рет, цб, цс, әзір, үшін
- D) егер p өту n (p-шарт)
- E) егер, онда, әйтпесе, бітті

Таңдау командасының қызметші сөздері

- A) алг, арг, нәт, бер
- B) таңдау, жағдай ... жағдай, әйтпесе, бітті
- C) n рет, цб, цс, әзір, үшін
- D) егер p өту n (p-шарт)
- E) егер, таңдау, жағдай ... жағдай, онда, бітті

Циклдік алгоритмдер -

- A) орындаушының командалары бірінен соң бірі ілесе тәртібімен орындалатын алгоритм.
- б) орындаушының әрекеттері кейбір шарттарды тексеру нәтижелерімен белгіленетін болса.
- C) көп рет қайталанып есептелетін командалар тізбегі бар алгоритмдер.
- D) кірістік немесе бастапқы деректер.
- E) алгоритмдер мен оларды орындаудың біртекті және дәл жазбаларына арналған белгілер мен ережелер жүйесі.

Тармақталу алгоритмі -

- A) берілген шарттың орындалуына байланысты бір тармақ бойынша орындалатын есептеу процестері бар алгоритмдер.
- B) орындаушының әрекеттері кейбір шарттарды тексеру нәтижелерімен белгіленетін болса.
- C) жеке командалар немесе команда топтары көп рет қайталанатын алгоритм.
- D) кірістік немесе бастапқы деректер.
- E) көп рет қайталанып есептелетін командалар тізбегі бар алгоритмдер.

Тармақталу командасының толық түрі:

- A) егер шарт
онда I-серия
бітті.

- В) егер шарт
 онда I-серия
 әйтпесе II-серия
бітті.
- С) егер шарт
 әйтпесе II-серия
бітті.
- Д) онда I-серия
 әйтпесе II-серия
бітті.
- Е) егер шарт
 онда I-серия
 әйтпесе II-серия

Қандай берілгендер құрылымды болып табылады?

- А) жазбалар, файлдар, жиындар, массивтер.
- В) бүтін сандар, нақты сандар, логикалық, символдық.
- С) тізімдер, стектер.
- Д) графтар, ағаштар.
- Е) логикалық, символдық, жиындар, жазбалар

Қарапайым құрылымды мәлімет типтері

- А) жазбалар, файлдар, жиындар, массивтер.
- В) бүтін, нақты, логикалық, символдық
- С) логикалық, символдық, массивтер.
- Д) тізімдер, стектер, графтар, ағаштар.
- Е) жазбалар, жиындар,

Евклид алгоритмі

- А) Ең үлкен ортақ бөлгішті табу.
- В) Екі санды салыстыру.
- С) Екі санның қосындысын табу.
- Д) Екі санның квадратын есептеу.
- Е) Екі санның ортақ еселігін табу.

Ішкі программаны оның өз ішінде тұрып шақыру -

- А) рекурсия
- В) процедура
- С) функция
- Д) көмекші программа
- Е) қосалқы программа

Байланысқан тізім...

- А) деректердің динамикалық құрылымы, онда әрбір жазу негізгі деректен басқа тізімдегі келесі жазуға сілтеуден тұрады;
- В) деректердің стандартты құрылымы, ондағы барлық элементтер бір-бірімен ортақ номермен байланысқан;
- С) деректердің құрылымдық типі, оның негізгі деректері бір-бірімен қандай-да бір заңдылықпен байланысқан;
- Д) жолдар тізімінен тұратын стандартты компонент;
- Е) көпжолдық мәтінді бейнелеу үшін арналған стандартты компонент.

Біржаққа бағытталған тізім элементі дегеніміз:

- A) екі өрістен (нұсқағыш және деректен) тұратын буын
- B) тек келесі элементке нұсқағыштан тұратын буын
- C) тек деректен тұратын буын
- D) екі өрістен (алдыңғы және келесі элементке нұсқағыштан) тұратын буын
- E) екі нұсқағыштан және деректен тұратын буын

Бұтақ – бұл ...

- A) сызықтық тізім, онда барлық қосу амалдары тізімнің бір шетінде орындалады, ал жою амалдары (және негізінен деректерге қатынау амалдары) екінші шетінде орындалады;
- B) сызықтық тізім, онда барлық қосу және жою амалдары (және негізінен деректерге қатынау амалдары) тізімнің тек бір шетінде орындалады;
- C) сызықтық тізім, онда барлық қосу және жою амалдары (және негізінен деректерге қатынау амалдары) тізімнің екі шетінде де орындалады;
- D) тораптардың шектелген жиыны, олардың біреуі еншілі торап болып табылады, ал қалғандары оның ішкі жиындары болады (олардың да өзінің тораптарының ішкі жиындары болуы мүмкін);
- E) тораптардың шектелген жиыны, ол бір еншілі тораптан және екі қиылыспайтын ішкі жиыннан тұрады (олардың да өзінің тораптарының ішкі жиындарының жұбы болуы мүмкін).

Бинарлық бұтақ – бұл ...

- A) сызықтық тізім, онда барлық қосу амалдары тізімнің бір шетінде орындалады, ал жою амалдары (және негізінен деректерге қатынау амалдары) екінші шетінде орындалады;
- B) сызықтық тізім, онда барлық қосу және жою амалдары (және негізінен деректерге қатынау амалдары) тізімнің тек бір шетінде орындалады;
- C) сызықтық тізім, онда барлық қосу және жою амалдары (және негізінен деректерге қатынау амалдары) тізімнің екі шетінде де орындалады;
- D) тораптардың шектелген жиыны, олардың біреуі еншілі торап болып табылады, ал қалғандары оның ішкі жиындары болады (олардың да өзінің тораптарының ішкі жиындары болуы мүмкін);
- E) тораптардың шектелген жиыны, ол бір еншілі тораптан және екі қиылыспайтын ішкі жиыннан тұрады (олардың да өзінің тораптарының ішкі жиындарының жұбы болуы мүмкін).

Төменде сұрыптаудың қандай әдісі сипатталған?

Алгоритм элементтердің көрші тұрған жұптары үшін салыстыру және орындарын ауыстыруға негізделген және осы процесті барлық элементтер реттелгенше жалғастыру керек.

- A) тура қосу көмегімен сұрыптау;
- B) тура таңдау көмегімен сұрыптау;
- C) көпіршікті сұрыптау;
- D) шейкерлік сұрыптау;
- E) Шелл сұрыптауы.

Төменде сұрыптаудың қандай әдісі сипатталған?

Кілтті кіші элемент таңдалады. Ол бірінші a_1 элементпен ауыстырылады. Одан кейін қалған $n - 1$ элементпен, $n - 2$ элементпен, және т.б., осылайша бір ең үлкен элемент табылғанша.

- A) тура қосу көмегімен сұрыптау;
- B) тура таңдау көмегімен сұрыптау;
- C) көпіршікті сұрыптау;
- D) шейкерлік сұрыптау;
- E) Шелл сұрыптауы.

Төменде сұрыптаудың қандай әдісі сипатталған?

Элементтер ойша “дайын” болған a_1, \dots, a_{i-1} тізбектеріне және берілген тізбектерге бөлінеді. $i = 2$ -ден бастап әрбір қадамда және i –ді әрдайым бірге өсіре отырып берілген тізбектен i –ші элемент алынады да, дайын тізбекке орналастырылады және де қажетті орынға қосылады.

- A) тура қосу көмегімен сұрыптау;
- B) тура таңдау көмегімен сұрыптау;
- C) көпіршікті сұрыптау;
- D) шейкерлік сұрыптау;
- E) Шелл сұрыптауы.

Төменде сұрыптаудың қандай әдісі сипатталған?

Бүкіл тізім элементтері араластырылған ішкі тізімдердің жиынтығы деп қарастырылады. Бірінші қадамда бұл ішкі тізімді тек элементтер жұбы ретінде қарастырылады. Екінші қадамда әрбір топ төрт элементтен қарастырылады. Процессті қайталағанда әрбір ішкі тізімдегі элементтер саны өседі, ал сәйкес ішкі тізімді саны азаяды. Ішкі тізімдерді сұрыптау тура қосу көмегімен сұрыптауды бір рет қолдану арқылы орындалады.

- A) тура қосу көмегімен сұрыптау;
- B) тура таңдау көмегімен сұрыптау;
- C) көпіршікті сұрыптау;
- D) шейкерлік сұрыптау;
- E) Шелл сұрыптауы.

Төменде сұрыптаудың қандай әдісі сипатталған?

Алгоритм екі көрші элементтер жұбын салыстыру және қарастырған тізбектің бағытын алма кезек ауыстырып отыруға негізделген. Соңғы ауыстырудың орнын (индексін) сақтай отырып сұрыпталған және сұрыпталмаған тізімнің сол және оң жақ шетін өзгертеді. Процесс сол және оң жақ шеттері қиылысқанша жалғастырылады.

- A) тура қосу көмегімен сұрыптау;
- B) тура таңдау көмегімен сұрыптау;
- C) көпіршікті сұрыптау;
- D) шейкерлік сұрыптау;
- E) Шелл сұрыптауы.

Төмендегі кестеде (85, 38, 41, 82, 90, 95, 6, 29) тізімін сұрыптаудың қандай алгоритмінің барлық жүрістерінің нәтижесі келтірілген

85	38	41	82	90	95	6	29
6	85	38	41	82	90	95	29
6	29	85	38	41	82	90	95
6	29	38	85	41	82	90	95
6	29	38	41	85	82	90	95
6	29	38	41	82	85	90	95
6	29	38	41	82	85	90	95
6	29	38	41	82	85	90	95

- A) тура қосу көмегімен сұрыптау;
- B) тура таңдау көмегімен сұрыптау;
- C) көпіршікті сұрыптау;
- D) шейкерлік сұрыптау;
- E) Шелл сұрыптауы.

Төмендегі кестеде (21, 77, 44, 22, 6, 52, 61, 46) тізімін сұрыптаудың қандай алгоритмінің барлық жүрістерінің нәтижесі келтірілген

21	77	44	22	6	52	61	46			
6	21	77	44	22	46	52	61	L=2	R=8	k=2
6	21	44	22	46	52	61	77	R=8	L=3	k=8
6	21	22	44	46	52	61	77	L=3	R=7	k=4
6	21	22	44	46	52	61	77	R=7	L=5	k=4

- A) тура қосу көмегімен сұрыптау;
 B) тура таңдау көмегімен сұрыптау;
 C) көпіршікті сұрыптау;
 D) шейкерлік сұрыптау;
 E) Шелл сұрыптауы.

Төмендегі кестеде (44, 55, 12, 42, 94, 18, 06, 67) тізімін сұрыптаудың қандай алгоритмінің барлық жүрістерінің нәтижесі келтірілген

44	55	12	42	94	18	06	67
44	55	12	42	94	18	06	67
12	44	55	42	94	18	06	67
12	42	44	55	94	18	06	67
12	42	44	55	94	18	06	67
12	18	42	44	55	94	06	67
06	12	18	42	44	55	94	67
06	12	18	42	44	55	67	94

- A) тура қосу көмегімен сұрыптау;
 B) тура таңдау көмегімен сұрыптау;
 C) көпіршікті сұрыптау;
 D) шейкерлік сұрыптау;
 E) Шелл сұрыптауы.

Жауаптың қай нұсқасында алгоритмдердің күрделілігінің өсу жылдамдығының функциясы өсу ретімен орналасқан?

- A) 2^n , $\log_2 n$, n^2 , $(\log_2 n)^2$, $n!$;
 B) $n!$, 2^n , n^2 , $\log_2 n$, $(\log_2 n)^2$;
 C) n^2 , 2^n , $n!$, $\log_2 n$, $(\log_2 n)^2$;
 D) $n!$, $(\log_2 n)^2$, $\log_2 n$, n^2 , 2^n ;
 E) $\log_2 n$, $(\log_2 n)^2$, n^2 , 2^n , $n!$.

Жауаптың қай нұсқасында алгоритмдердің күрделілігінің өсу жылдамдығының функциясы өсу ретімен орналасқан?

- A) $\log_2 \log_2 n$, $n - n^2 + 5n^3$, n^3 , \sqrt{n} , n ;
 B) $\log_2 \log_2 n$, \sqrt{n} , n , n^3 , $n - n^2 + 5n^3$;
 C) \sqrt{n} , n , n^3 , $n - n^2 + 5n^3$, $\log_2 \log_2 n$;
 D) $n - n^2 + 5n^3$, \sqrt{n} , n , n^3 , $\log_2 \log_2 n$;
 E) n^3 , n , \sqrt{n} , $\log_2 \log_2 n$, $n - n^2 + 5n^3$.

Жауаптың қай нұсқасында алгоритмдердің күрделілігінің өсу жылдамдығының функциясы өсу ретімен орналасқан?

- A) $n^3 + \log_2 n$, 2^{n-1} , $n \log_2 n$, 6 , $(3/2)^n$;

- B) $6, 2^{n-1}, (3/2)^n, n^3 + \log_2 n, n \log_2 n$;
- C) $6, n \log_2 n, n^3 + \log_2 n, (3/2)^n, 2^{n-1}$;
- D) $(3/2)^n, n \log_2 n, 2^{n-1}, n^3 + \log_2 n, 6$;
- E) $6, n^3 + \log_2 n, 2^{n-1}, n \log_2 n, (3/2)^n$.

Модель дегеніміз не?

- A) объектіні не процесті басқаруға үйрену және қойылған мақсатқа жету кезінде басқарудың ең тиімді тәсілдерін айқындау
- B) зерттеу процесінде әлгі түпнұсқаның аталмыш зерттеу үшін кейбір маңызды типтік белгілерін сақтай отырып, түпнұсқа - объектіні алмастыратын материалдық немесе ойша көзге елестететін объект.
- C) нақты объект қалай құрылған - оның құрылысы, негізгі қасиеттері, даму заңдылығы мен қоршаған дүниемен өзара әрекеттестігі қандай екендігін тану
- D) берілген тәсілдерді іске асырудың тура және жанама салдарларына, объектіге әсер ету формаларына болжам жасау
- E) шынайы объектіге әлдебір әрекеттің әсерін зерттеу мен бағалау үшін сан рет қайталанатын тәжірибе

Құрылу тәсілдеріне және құрылымдарына қарай модельдер ... болып бөлінеді.

- A) таңбалық және идеал модель
- B) идеал және түсінік модель
- C) түсінік модель және таңбалық модель
- D) статикалық және динамикалық модель
- E) материалдық және идеал модель

Уақыт факторын ескеруі бойынша модельдер ... болып бөлінеді.

- A) таңбалық және идеал модель
- B) идеал және түсінік модель
- C) түсінік модель және таңбалық модель
- D) статикалық және динамикалық модель
- E) материалдық және идеал модель

Идеал модель дегеніміз -

- A) объективтік шындықтың адам миындағы бейнесі
- B) объективті тіршілік ететін, адам қолымен жасалған немесе осы модельдерге тән белгілі қасиеттерге бара-бар адам іріктеп алған модель
- C) объектінің элементтері, олардың қатынастары мен қасиеттері белгілі таңбалардың көмегімен берілетін модель
- D) адам ойындағы физикалық заңдылықтарға, математикалық ережелерге және логикалық ақыл-ойға сүйеніп жасалатын модель
- E) берілген тәсілдерді іске асырудың тура және жанама салдарларына, объектіге әсер ету формаларына болжам жасау

Статикалық модель -

- A) объектінің уақыт ішіндегі өзгеруін көруге мүмкіндік береді
- B) объектіні зерттеуге материалдық тұрғыдан келуді жүзеге асырады
- C) объектінің ағымдағы жағдайдағы “бір сәттік кесіндісін” береді
- D) құбылысты не процесті зерттеуге материалдық тұрғыдан келуді жүзеге асырады
- E) адам ойындағы заңдылықтарға, ережелерге және логикалық ақыл-ойға сүйеніп жасалады

Динамикалық модель -

- A) объектінің уақыт ішіндегі өзгеруін көруге мүмкіндік береді
- B) объектіні зерттеуге материалдық тұрғыдан келуді жүзеге асырады
- C) объектінің ағымдағы жағдайдағы “бір сәттік кесіндісін” береді
- D) құбылысты не процесті зерттеуге материалдық тұрғыдан келуді жүзеге асырады
- E) адам ойындағы заңдылықтарға, ережелерге және логикалық ақыл-ойға сүйеніп жасалады

Ұстауға не көзбен көруге болмайтын, материалдық нақты түрі жоқ модель

- A) компьютерлік модель
- B) таңбалық модель
- C) идеал модель
- D) ақпараттық модель
- E) материалдық модель

Модель объектіні түйсіну тәсілі бойынша ... болып бөлінеді

- A) ақпараттық, компьютерлік
- B) ақпараттық, таңбалық
- C) ақпараттық және вербальдық
- D) материалдық, динамикалық
- E) материалдық және ақпараттық

Ақпараттық модельдерді ... деп бөлуге болады.

- A) ақпараттық, компьютерлік модельдер
- B) материалдық, таңбалық модельдер
- C) материалдық және вербальдық (ауызша айтылған) модельдер
- D) таңбалық және вербальдық (ауызша айтылған) модельдер
- E) материалдық, динамикалық

Ақпараттық модель дегеніміз -

- A) объективтік шындықтың адам миындағы бейнесі
- B) объективті тіршілік ететін, адам қолымен жасалған немесе осы модельдерге тән белгілі қасиеттерге бара-бар адам іріктеп алған модель
- C) объектінің элементтері, олардың қатынастары мен қасиеттері белгілі таңбалардың көмегімен берілетін ақпараттар жиынтығының моделі
- D) объекті мен процестің, құбылыстардың қасиеті мен жағдайын, сондай-ақ оның сыртқы дүниемен өзара байланысын сипаттайтын ақпараттар жиынтығы
- E) берілген тәсілдерді іске асырудың тура және жанама салдарларына, объектіге әсер ету формаларына болжам жасау

Таңбалық модель –

- A) объектінің уақыт ішіндегі өзгеруін көруге мүмкіндік беретін модель
- B) ойша немесе ауызекі формамен берілген ақпараттық модель
- C) объектінің ағымдағы жағдайдағы “бір сәттік кесіндісін” береді
- D) кез келген формальды тіл құралымен берілген ақпараттық модель
- E) адам ойындағы заңдылықтарға, ережелерге және логикалық ақыл-ойға сүйеніп жасалатын модель

Вербальдық модель –

- A) объектінің уақыт ішіндегі өзгеруін көруге мүмкіндік беретін модель
- B) ойша немесе ауызекі формамен берілген ақпараттық модель
- C) объектінің ағымдағы жағдайдағы “бір сәттік кесіндісін” береді
- D) кез келген формальды тіл құралымен берілген ақпараттық модель
- E) адам ойындағы заңдылықтарға, ережелерге және логикалық ақыл-ойға сүйеніп жасалатын модель

Математикалық модель –

- A) объектіні не процесті басқаруға үйрену және қойылған мақсатқа жету кезінде басқарудың ең тиімді тәсілдерін айқындау
- B) зерттеу процесінде әлгі түпнұсқаның аталмыш зерттеу үшін кейбір маңызды типтік белгілерін сақтай отырып, объектіні алмастыратын материалдық немесе ойша көзге елестететін математикалық формулалар
- C) объект немесе процестің әр түрлі параметрлерінің байланысын бейнелейтін математикалық формулалар
- D) берілген тәсілдерді іске асырудың тура және жанама салдарларына, объектіге әсер ету формаларына болжам жасау
- E) шынайы объектіге әлдебір әрекеттің әсерін зерттеу мен бағалау үшін сан рет қайталанатын тәжірибе

Компьютерлік модель –

- A) объектіні не процесті басқаруға үйрену және қойылған мақсатқа жету кезінде басқарудың ең тиімді тәсілдерін айқындау
- B) зерттеу процесінде әлгі түпнұсқаның аталмыш зерттеу үшін кейбір маңызды типтік белгілерін сақтай отырып, объектіні алмастыратын материалдық немесе ойша көзге елестететін математикалық формулалар
- C) объект немесе процестің әр түрлі параметрлерінің байланысын бейнелейтін математикалық формулалар
- D) мәліметтер жиынтығы және оларды өңдеуге арналған бағдарламалар
- E) шынайы объектіге әлдебір әрекеттің әсерін зерттеу мен бағалау үшін сан рет қайталанатын тәжірибе

Модельдеу дегеніміз -

- A) құрылған модель көмегімен түпнұсқаның құрылымы мен қасиеттерін зерттеу процесі
- B) зерттеу процесінде әлгі түпнұсқаның аталмыш зерттеу үшін кейбір маңызды типтік белгілерін сақтай отырып, объектіні алмастыратын материалдық немесе ойша көзге елестететін модель
- C) объект немесе процестің әр түрлі параметрлерінің байланысын бейнелейтін математикалық формулалар
- D) мәліметтер жиынтығы және оларды өңдеуге арналған бағдарламалар
- E) шынайы объектіге әлдебір әрекеттің әсерін зерттеу мен бағалау үшін сан рет қайталанатын тәжірибе

Ақпараттық модельдеу -

- A) құрылған модель көмегімен түпнұсқаның құрылымы мен қасиеттерін зерттеу процесі
- B) зерттеу процесінде әлгі түпнұсқаның аталмыш зерттеу үшін кейбір маңызды типтік белгілерін сақтай отырып, объектіні алмастыратын материалдық немесе ойша көзге елестететін модель
- C) объектінің моделін немесе құбылыстың өту барысын формальды түрде жазып сипаттау
- D) мәліметтер жиынтығы және оларды өңдеуге арналған бағдарламалар
- E) шынайы объектіге әлдебір әрекеттің әсерін зерттеу мен бағалау үшін сан рет қайталанатын тәжірибе

Ақпараттық модельдеуді ... деп бөлуге болады.

- A) ақпараттық, компьютерлік модельдеу
- B) материалдық, таңбалық модельдеу
- C) материалдық және вербальдық (ауызша айтылған) модельдеу
- D) интуитивтік, бейнелік және бейнелік-таңбалық модельдеу

Е) материалдық, динамикалық модельдеу

Интуитивтік модельдеу -

- А) құрылған модель көмегімен түпнұсқаның құрылымы мен қасиеттерін зерттеу процесі
- В) зерттеу процесінде әлгі түпнұсқаның аталмыш зерттеу үшін кейбір маңызды типтік белгілерін сақтай отырып, объектіні алмастыратын материалдық немесе ойша көзге елестететін модель
- С) объектінің моделін немесе құбылыстың өту барысын формальды түрде жазып сипаттау
- Д) объект туралы ойша көз алдына елестету.
- Е) шынайы объектіге әлдебір әрекеттің әсерін зерттеу мен бағалау үшін сан рет қайталанатын тәжірибе

Бейнелік модельдеу -

- А) құрылған модель көмегімен түпнұсқаның құрылымы мен қасиеттерін зерттеу процесі
- В) қайсыбір түрдің: сұлба, график, сызба, жоспар, карта, т.б. таңбалық бейнелері
- С) объектінің моделін немесе құбылыстың өту барысын формальды түрде жазып сипаттау
- Д) объект туралы ойша көз алдына елестету.
- Е) табиғи тілмен немесе салынған суретпен бейнеленген, сезімдік көрнекі бейнелер көмегімен түпнұсқаның қасиетін білдіру.

Модельденуші объектінің ақпаратты кодтау тілдерінің бірінде сипатталуы ... деп аталады

- А) ақпараттық модель
- В) уақыттық модель
- С) табиғи модель
- Д) материалдық модель
- Е) бейнеленуші модель

Әдебиеттер

1. Стариченко Б.Е. Теоретические основы информатики: Учебное пособие для вузов. – 2-е изд. перераб. и доп. – М.: Горячая линия-Телеком, 2003. – 312 с.
2. Дж.Макконелл. Программирование. Основы современных алгоритмов. М., Техносфера. - 2004. – 368 с.
3. Бурин Е.А. Введение в основы информатики и вычислительной техники. – Алма-Ата, “Мектеп” - 1989.- 144 с.
4. Информатика. Базовый курс. Учебник для ВУЗов. Под редакцией С.В.Симоновича. 2001, - 640 с.
5. Князева М.Д. Алгоритмика: от алгоритма к программе: КУДИЦ-ОБРАЗ-М., 2006
6. С. Симонович, Г.Евсеев, А.Алексеев. Специальная информатика. Учебное пособие. Москва, 1999 г.
7. Антипов И.Н. Основы информатики и вычислительной техники. Методическое пособие для преподавателей. – М.: Высшая школа – 1991г.- 248с.
8. А.Ю. Гаевский . Информатика. Учебное пособие . 2-е издание . – Киев., Москва . – 2004 г. – 535 с.
9. Бауэр Ф.Л., Гооз Г. Информатика. Вводный курс: В 2-х ч. Пер. с нем. – М., Мир, 1990. – 336 с.

МАЗМҰНЫ

Алғы сөз	3
Дәріс №1. Информатика пәні және мақсаттары. Теориялық информатика	4
Дәріс №2. Ақпарат және оның қасиеттері	16
Дәріс №3. Ақпараттық жүйелер және технологиялар	26
Дәріс №4. Автомат ақпараттық жүйенің негізгі элементі ретінде. Абстрактылы автоматтар	34
Дәріс №5. Сандық автоматтарда ақпаратты өрнектеу. Санау жүйелері	39
Дәріс №6. Екілік қосындылауыштарда арифметикалық амалдарды орындау	46
Дәріс №7. Логикалық алгебраның негізгі ұғымдары	51
Дәріс №8. Алгоритм ұғымы. Алгоритмдер теориясының негізгі ұғымдары	56
Дәріс №9. Қолданбалы есептерді шешу алгоритмдерін талдау қағидалары	66
Дәріс №10. Бағдарлама құрылымы. Операторлар бөлігі	76
Дәріс №11. Массив. Екі өлшемді массивтер	87
Дәріс №12. Алгоритмдердің тиімділігі мен күрделілігіне анализ жасау	89
Дәріс №13. Іздеу, таңдау, сұрыптау, рекурсивті алгоритмдер	92
Дәріс №14. Модельдер. Ақпараттық модель. Математикалық модель. Компьютерлік модель	108
Дәріс №15. Модельдеу. Модельдерді құрудың негізгі кезеңдері. Математикалық модельдеу және есептеу эксперименті. Компьютерлік модельдеу	114
Тест	123
Әдебиеттер	160