

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

Э.А.Абдыкеримова

DELPHI ОРТАСЫНДА БАҒДАРЛАМАЛАУ

Оқу құралы

Ақтау, 2010

ӨӘЖ 004.4 (075)

ББК 32.923я7

А 14

Пікір жазғандар:

1. ф.-м.ғ.д., профессор Ж.А.Тусупов – М.Х.Дулати атындағы Тараз мемлекеттік университеті
2. п.ғ.к., доцент Ш.Т.Шекербекова – Абай атындағы Қазақ Ұлттық педагогикалық университеті

А 14 Абдыкеримова Э.А.

Delphi ортасында бағдарламалау: Оқу құралы. – Ақтау, 2010. – 181 бет.

ISBN 978-601-226-071-7

Оқу құралында бағдарламалау курсы бойынша Borland Delphi 7 ортасында бағдарламалау негіздерін қамтитын теориялық мағлұматтар мен практикалық есептерді шешу үшін бағдарлама құрудың жолдары және әрбір тақырыптың соңында өзіндік жұмысқа арналған тапсырмалар, бақылау сұрақтары келтірілген.

Жоғары оқу орындарының студенттеріне, оқытушыларына, орта мектептің информатика пәні мұғалімдері мен оқушыларына арналған.

ӨӘЖ 004.4 (075)

ББК 32.923я7

Ш.Есенов атындағы Каспий мемлекеттік технологиялар және инжиниринг университетінің ғылыми кеңесінің шешімімен баспаға ұсынылған.

ISBN 978-601-226-071-7

© Ш.Есенов атындағы КМТ және ИУ, 2010

АЛҒЫ СӨЗ

Delphi – бұл әр түрлі мазмұндағы сан қилы мақсаттарды жүзеге асыруға арналған қосымшаларды дайындайтын, қуатты жұмыс ортасының бірі.

Delphi-де қолданбалы бағдарламалар мен қосымшаларды құру IDE (Integrated Development Environment) жұмыс жасау ортасында жүзеге асады. IDE – бағдарламашымен арадағы байланысты орнатып, оған қажетті басқарушы элементтерден тұратын бірнеше терезелерді қамтиды. IDE жұмыс ортасының көмегімен қосымшалардың интерфейстік бөлігін жобалап, сонымен бірге басқарушы элементтермен байланысқан бағдарламалық кодты жазуға болады.

Қазіргі кезге дейін Borland Delphi ортасында бағдарламалауға арналған көптеген оқу құралдары, әдістемелік құралдар жазылған. Бірақ олар негізінен тәжірибелі қолданушыларға арналған. Әдетте оларда тілдің нақты жүзеге асырылуында бағдарламалауды енді ғана бастаған студенттерге бірқатар қиындықтар туғызатын ерекшеліктері бар.

Студенттер бағдарламалау іскерлігін үйрену үшін нақты есептер шығарып, бағдарламасын жаза білуі қажет. Ол үшін бағдарламалау тілі мен оны құру ортасын меңгеру керек. Объектілі бағдарланған бағдарламалау жүйелерінде бағдарламалаудағы негізгі ерекшелік алдымен бағдарламалау синтаксисі мен компоненттерді пайдалануды үйренуде болып табылады.

Бұл ұсынылып отырған оқу құралында объектілі бағдарланған бағдарламалау жүйелерінің негіздері берілген. Delphi ортасында бағдарламалауға қажетті теориялық материалдар мен мысалдар келтірілген. Әрбір тақырыптың соңында теориялық материалдарды бекітуге арналған бақылау сұрақтары мен тапсырмалар берілген.

Зертханалық жұмыс №1. Delphi ортасымен танысу

Жұмыстың мақсаты: Delphi бағдарламалау ортасының негізгі терезелерімен таныстыру.

Бағдарламалық дидактикалық қамтылуы: Дербес компьютерлер, Windows XP операциялық жүйесі, Borland Delphi 7 бағдарламасы, «Бағдарламалау» пәнінен оқу-әдістемелік кешен.

Delphi бағдарламалау ортасын іске қосу бірнеше тәсілмен жүзеге асырылады:

1. Windows жүйесінің басты мәзірінен Delphi 7 тікелей қосылады:

Пуск → Программы → Borland Delphi 7 → Delphi 7.

2. Егер жұмысшы үстелінде Delphi 7 таңбашасы (ярлык) болса, онда тышқанның сол жақ батырмасы арқылы екі рет шертіп іске қосуға болады.

Delphi ортасын шақырғаннан кейін ортаның жұмысын басқаратын алты негізгі терезе шығады (1-сурет):

1) Негізгі терезе (басты мәзір, саймандар тақтасы және компоненттер тақтасы), оның тақырыбы – Delphi 7 – Project1 - деп аталады;

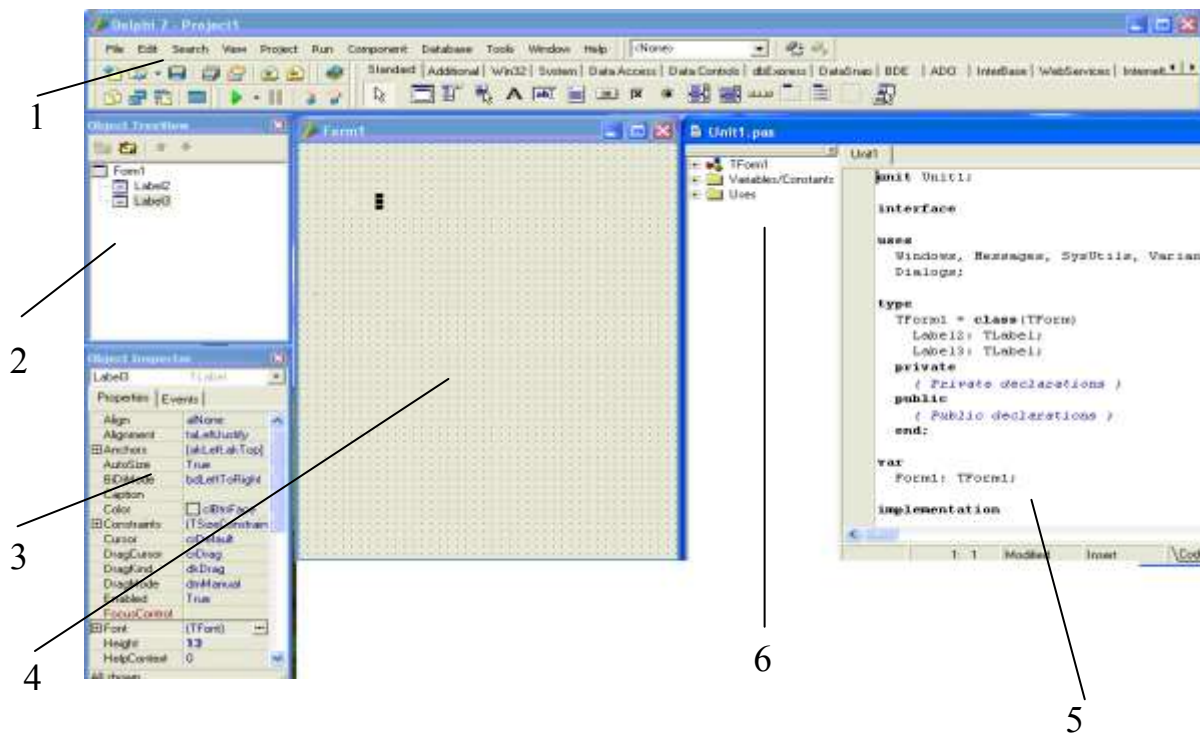
2) Объектілер құрылымы орналасқан терезе (Object TreeView);

3) Объектілер инспекторы терезесі (Object Inspector);

4) Форма терезесі (Form1);

5) Бағдарлама коды орналасқан терезе (Unit1.pas);

6) Броузер (көру) терезесі (код терезесінің сол жағында орналасқан).



1-сурет. Delphi ортасының интерфейсі

Негізгі терезе. Негізгі терезе құрылатын бағдарламаның жобаларын басқаратын негізгі қызметтерді атқарады. Бұл терезе экранда барлық уақытта болады және ең жоғарғы бөлігінде орналасады. Негізгі терезеде Delphi ортасының бас мәзірі, пиктографиялық командалық батырмалар жиынтығы, компоненттер палитрасы орналасқан.

Басты мәзір. Басты мәзір деп Delphi терезесінің жоғарғы жағында орналасқан мәтін жолын айтады. Басты мәзір командалары арқылы жобаларды жасауға, ашуға, басуға, сақтауға болады; белгілі бір аймақты алмасу буферіне қиып алу, көшірмесін алу, кірістіру іспетті стандартты операциялар орындалады, бұл бағдарлама фрагменттеріне ғана емес, кез келген объектіге де қолданылады; процедурадағы, модульдегі немесе бүтін жобадағы мәтін фрагментін іздеуге; бағдарламаны тексеріп, орындауға және т.б. әрекеттер орындауға болады

Саймандар тақтасы.

New items – объектілерді таңдау.

Open – алдын-ала бар файлды ашу.

Save (Ctrl+S) – белсенді құжатты сақтау.

Save All (Shift+Ctrl+S) – жобаның барлық файлдарын сақтау.

Open Project (Ctrl+F11) – бағдарламаның алдын-ала құрылған жобасын шақыру.

Add file to Project (Shift+F11) – жобаға файл қосу.

Remove file from Project – жобадан файлды жою.

Help contents – анықтаманы іске қосу.

View Unit (Ctrl+F12) – тізімнен модуль таңдау.

View Form (Shift+F12) - тізімнен форма таңдау.

Toggle Form /Unit (F12) – форма терезесінен бағдарлама коды терезесіне және кері алмасу.

New Form – жаңа форма құрады және оны жобаға қосады.

Run (F9) – бағдарламаны тексереді және орындайды.

Pause – дұрыс орындалмаған бағдарламаға үзіліс жасау.

Trace Into (F7) – шақырылған қосалғы бағдарламаларға қадам бойынша ену.

Step over (F8) – шақырылған қосалғы бағдарламалардың жұмысын елемей қадам бойынша ену.

Пиктографиялық тетіктер орналасқан саймандар тақтасының конфигурациясын өзгерту үшін негізгі мәзірден View/Toolbars/Customize командасын немесе саймандар тақтасының контексті мәзірінен Customize командасын таңдаймыз.

Бұл терезеде пиктографиялық тетіктердің орналасуын алмастыруға, жоюға, қосуға, жасыруға, көрсетуге болады.

Toolbars жапсырмасы саймандар тақтасымен жылдам орындалатын тетіктерді басқаруға арналған.

Commands жапсырмасы тетіктерді таңдауға және оларды саймандар тақтасына орналастыру мүмкіндігін береді.

Options жапсырмасы арқылы, Show Tooltips - жылдам орындалатын пернелердің жарлығын (курсорды оған ұстап тұрғанда) көрсетуді немесе Show Shortcut keys on tooltips - бұл жарлықтарда сәйкес келетін пернелер тобын бейнелеуді іске қосу, не ажыратуға болады.

Компоненттер тақтасы. Компонент деп белгілі бір қасиеті бар, программист формалар терезесінде орналастыра алатын қандай да бір басқару элементін түсінеміз. Компоненттер тақтасы негізгі терезенің оң жақ бөлігінде орналасқан және қажетті компоненттерді жылдам іздеуді қамтамасыз ететін ашылатын беттері бар (2-сурет). Компоненттердің көмегімен бағдарламаның сұлбасы жасалады, жалпы жағдайда экранда көрінетін терезелер, батырмалар, таңдаулар тізімі және т.б.



2-сурет. Компоненттер тақтасының терезесі

Беттердің орналасу тәртібін өзгерту, атын өзгерту, жою, пиктограмма қосу немесе кейбір пиктограммаларды басқа бетке жылжыту үшін Palette Properties диалогтық терезесі қолданылады. Ол диалогтық терезені ашу үшін **Component/Configure Palette** командасын немесе компоненттер политрасының кез-келген жерінде тышқанның оң жағындағы контексті мәзірдің Properties (Қасиеттер) командасы қолданылады.

Delphi-ді іске қосқанда Standard беті белсенді болып тұрады.

Мысал ретінде, компоненттер тақтасының бетшелерінде орналасқан компоненттердің қасиеттеріне тоқталайық:

Label компоненті  - форма бетіне мәтін шығаруға арналған компонент. Компоненттің қасиеттері мәтіннің орналасуы мен түрін анықтайды:

Name – компонент аты.

Caption – компонентте орналасатын мәтін.

Top – шығару өрісінің жоғарғы шекарасынан форманың жоғарғы шекарасына дейінгі қашықтық.

Left - шығару өрісінің сол жақ шекарасынан форманың сол жақ шекарасына дейінгі қашықтық.

Width – шығару өрісінің ені.

Height – шығару өрісінің биіктігі.

WordWrap – ағымдағы жолға сыймайтын сөздің автоматты түрде келесі жолға көшетінін көрсетеді. AutoSize қасиетінің мәні False болуы тиіс.

Alignment - өрістің ішіндегі мәтінді туралау тәсілін береді.

Transparent – фондағы шығарылатын мәтінді басқарады.

Color – мәтін шығатын фонның түсін көрсетеді.

Visible – мәтіннің көрінуін немесе көрінбеуін басқарады.

Font – мәтінді шығару үшін пайдаланылатын қаріп.

Edit компоненті -  - символдар жолын енгізуге және жөндеуге арналған.

Name – компонент аты.

Text – енгізу және жөндеу өрісінде орналасатын мәтін.

Top – компоненттің жоғарғы шекарасынан форманың жоғарғы шекарасына дейінгі қашықтық.

Left – компоненттің сол жақ шекарасынан форманың сол жақ шекарасына дейінгі қашықтық.


Width - өрістің ені.

Height - өрістің биіктігі.

Visible – компоненттің көрінуін немесе көрінбеуін басқарады.

Enabled – енгізу өрісінде мәтінді өзгерту мүмкіндігін шектеу үшін пайдаланылады.

Font – мәтінді шығару үшін пайдаланылатын қаріп.

Button компоненті –  оқиғаға жауапты қалыптастыратын командалық батырма болып табылады.

Name – компонент қасиетіне кіру мүмкіндігін алу үшін пайдаланылатын компонент атауы.

Caption – батырманың үстіне орналасатын мәтін.

Top – батырманың жоғары шекарасынан форманың жоғары шекарасына дейінгі қашықтық.

Left - батырманың сол жақ шекарасынан форманың сол жақ шекарасына дейінгі қашықтық.

Width – батырманың ені.


Height – батырманың биіктігі.

Visible – батырманың көрінуін немесе көрінбеуін басқарады.

Enabled – батырманың пайдаланылу мүмкіндігін көрсетеді, егер оның қасиетінің мәні ақиқат болса, онда батырма пайдаланылады, кері жағдайда пайдаланылмайды.

Hint – командалық батырмаға курсор келгенде тышқанның көрсеткішінің қасында нұсқау-мәтін пайда болады.


Show Hint – егер мән ақиқат болса, нұсқау көрінеді, кері жағдайда көрінбейді.

Мемо компоненті –  бірнеше жолдан тұратын, мәтінді жөндеу элементі болып табылады. Экранға нәтижені шығару үшін, түсіндірме шығару үшін пайдаланылатын компонент.

Name – компонент қасиетіне кіру мүмкіндігін алу үшін пайдаланылатын компонент атауы.

Text – Мемо өрісінде орналасатын мәтін.

Lines - өріс мазмұнынан сәйкес келетін жолдар жиыны. Жолға бару нөмір бойынша жүзеге асады.

RadioButton компоненті -  - топтағы басқа батырмалардың таңдалуына байланысты анықталатын тәуелді батырма болып табылады. Бұл батырманың негізгі қасиеттері:

Name – компонент қасиетіне кіру мүмкіндігін алу үшін пайдаланылатын компонент атауы.

Caption – батырманың оң жағына орналасатын мәтін.

Checked – батырманың сыртқы күйін анықтайды: егер батырма таңдалса, онда Checked «ақиқат» мәнін, кері жағдайда «жалған» мәнін қабылдайды.


Top – жалаушаның жоғары шекарасынан форманың жоғары шекарасына дейінгі қашықтық.

Left - жалаушаның сол жақ шекарасынан форманың сол жақ шекарасына дейінгі қашықтық.

Width – түсіндірме мәтін шығаратын өрістің ені.

Height – түсіндірме мәтін шығаратын өрістің биіктігі.

Font – түсіндірме мәтінді шығару үшін пайдаланылатын қаріп.

ComboBox компоненті -  - Combobox компоненті тізімдер редакторына пернетақтадан мәліметтерді енгізіп, бағдарламаның орындалу барысында енгізілген тізімнен бір элементті таңдау мүмкіндігін береді.

Name – компонент қасиетіне кіру мүмкіндігін алу үшін пайдаланылатын компонент атауы.

Text – енгізу өрісінде орналасатын мәтін.

Items – тізім элементтері – жолдар жиыны.

Count – тізім элементтерінің саны

ItemIndex – тізімде таңдалған элемент нөмірі.

Sorted – тізімге кезекті элементті қосқаннан кейінгі автоматты сұрыптау қажеттілігін көрсетеді.

DropDownCount – ашылған тізімде бейнеленетін элементтер өлшемі.


Top – компоненттің жоғары шекарасынан форманың жоғары шекарасына дейінгі қашықтық.

Left - компоненттің сол жақ шекарасынан форманың сол жақ шекарасына дейінгі қашықтық.

Width – компоненттің ені.

Height – компоненттің биіктігі.

Font – тізім элементтерін шығару үшін пайдаланылатын қаріп.

CheckBox компоненті -  - тәуелсіз ауыстырып қосқыш батырманы сипаттайды, бұл батырманың көмегімен пайдаланушы өзінің «иә» немесе «жоқ» деген тәрізді шешімін көрсете алады.

Name – компонент қасиетіне кіру мүмкіндігін алу үшін пайдаланылатын компонент атауы.

Caption – жалаушаның оң жағына орналасатын мәтін.

Checked – жалаушаның сыртқы күйін анықтайды: егер жалауша таңдалса, онда Checked «ақиқат» мәнін, кері жағдайда «жалған» мәнін қабылдайды.


Top – жалаушаның жоғары шекарасынан форманың жоғары шекарасына дейінгі қашықтық.


Left - жалаушаның сол жақ шекарасынан форманың сол жақ шекарасына дейінгі қашықтық.

Width – жалаушаның ені.

Height – жалаушаның биіктігі.

Font – түсіндірме мәтінді шығару үшін пайдаланылатын қаріп.

RadioGroup компоненті –  - пайдаланушының таңдауына тәуелді болатын батырмалар тобының жұмысын сипаттайды. Бір батырманы таңдау қалған батырмалардың жұмысын тоқтатады, яғни таңдалған батырмаға сәйкес әрекет орындалады.

ListBox –  - бұл компонент керекті элементті таңдауға мүмкіндік беретін тізім болып келеді.

Name – компонент қасиетіне кіру мүмкіндігін алу үшін пайдаланылатын компонент атауы.

Items – тізім элементтері – жолдар жиыны.

Count – тізім элементтерінің саны

ItemIndex – тізімде таңдалған элемент нөмірі.

Sorted – тізімге кезекті элементті қосқаннан кейінгі автоматты сұрыптау қажеттілігін көрсетеді.


Top – тізімнің жоғары шекарасынан форманың жоғары шекарасына дейінгі қашықтық.

Left - тізімнің сол жақ шекарасынан форманың сол жақ шекарасына дейінгі қашықтық.

Width – тізім өрісінің ені.

Height – тізім өрісінің биіктігі.

Font – тізім элементтерін шығару үшін пайдаланылатын қаріп.

StringGrid компоненті –  - бұл компонент ақпаратты кесте түрінде бейнелеу үшін пайдаланылады. Кесте белгіленген жұмыс аймағынан тұрады. Белгіленген аймақ жұмыс аймағының жол және баған атауларын шығару және тышқанның көмегімен олардың өлшемін басқару қызметін атқарады. Белгіленген аймақ басқа түспен ерекшеленген және оған пернетақтадан ақпарат енгізуге болмайды. Қасиеттері:

FixedRows – белгіленген аймақтың жолдар саны

FixedCols - белгіленген аймақтың бағандар саны

RowCount – жұмыс аймағы жол ақпараттарынан тұрады

ColCount - жұмыс аймағы баған ақпараттарынан тұрады (оны бағдарламалық жолмен де, тышқанның немесе пернетақтаның да көмегімен өзгертуге болады).

Бағдарламадағы ақпаратпен қатынас Cells[Acol, AROW:integer]: string қасиеттерінің көмегімен жүзеге асырылады. Мұндағы, Acol - кестенің баған нөмірі, ал AROW - жол нөмірі, ал нөмірлеу нөлден басталады.

Top – кесте өрісінің жоғары шекарасынан форманың жоғары шекарасына дейінгі қашықтық.

Left - кесте өрісінің сол жақ шекарасынан форманың сол жақ шекарасына дейінгі қашықтық.

Width – кесте өрісінің ені.


Height – кесте өрісінің биіктігі.

Font – кесте ұяшықтарының мазмұнын шығару үшін пайдаланылатын қаріп.

Options.goEditing – кесте ұяшығының мазмұнын жөндеу мүмкіндігін көрсетеді. Егер «ақиқат» мәні тағайындалса, жөндеуге рұқсат етіледі, кері жағдайда рұқсат етілмейді.

Options.goTab – курсорды кестенің келесі ұяшығына <Tab> клавишын пайдаланып көшіру мүмкіндігін көрсетеді. Егер «ақиқат» мәні тағайындалса, жөндеуге рұқсат етіледі, кері жағдайда рұқсат етілмейді.

Options.GoAlwaysShowEditor – компоненттің жөндеу режимінде болу мүмкіндігін көрсетеді.

Image компоненті –  - форма бетіне заты .bmp, .jpg, .ico болып келген графикалық файлдарды шығару үшін пайдаланылады. Бұл компоненттің төмендегідей қасиеттері бар:

Picture – компонент өрісінде бейнеленетін суретті көрсетеді.


Proportional – суреттің сапасын бұрмаламай автоматты түрде масштабтау белгісі. Масштабтау орындалу үшін Autosize қасиетінің мәні False болуы тиіс.

Stretch – компоненттің шынайы өлшеміне сәйкес суретті автоматты түрде масштабтау белгісі. Егер компоненттің өлшемі суреттің өлшеміне сәйкес келмейтін болса, суреттің сапасы бұзылады.

Center – көлденең бағытта компонент өрісіндегі суреттің орналасуын анықтайды. Егер қасиеттің мәні жалған болса, онда сурет оң жаққа жиек тартылып орналасады, ал қасиеттің мәні ақиқат болса, онда сурет ортаға орналасады.

Visible – форма бетіне компонент немесе сәйкес суреттің бейнеленуін анықтайды.

Canvas – графиканы шығаратын бетті сипаттайды.

Timer –  - On Timer оқиғалар тізбегін генерациялауды қамтамасыз етеді.

Name – компонент аты. Компонентке кіру үшін пайдаланылады.

Interval – On Timer оқиғаларын генерациялау периоды. Миллисекунд арқылы беріледі.

Enabled – On Timer оқиғаларын егер мәні ақиқат болса генерациялайды, ал мәні жалған болса жұмысты тоқтатады.

Animate –  - кадрлары AVI файлында орналасқан, қарапайым дыбыссыз анимацияны шығару үшін пайдаланылады.

Name – компонент қасиетіне кіру мүмкіндігін алу үшін пайдаланылатын компонент атауы.

FileName – компоненттің көмегімен бейнеленетін анимация орналасқан AVI файлының аты.

StartFrame – анимация бейнесі басталатын кадр нөмірі.


StopFrame – анимация бейнесі аяқталатын кадр нөмірі.

Activate – анимация кадрларын бейнелеу процесін екпінді жасау белгісі.

Color – анимация өтетін компонент фонының түсі.

Transparent – анимацияны бейнелеу кезінде «мөлдір» түсті пайдалану.

Repetitions – анимация бейнесін қайталау мөлшері

MediaPlayer –  - видеороликтерді, дыбыстарды және дыбысы бар анимацияларды орындауға мүмкіндік береді.

Name – компонент қасиетіне кіру және плеерді басқару мүмкіндігін алу үшін пайдаланылатын компонент атауы.


DeviceType – құрылғы типі.

FileName – бейнеролик немесе дыбыстық фрагменті орналасқан файл аты.

AutoOpen – бағдарламаны шақырғаннан кейін бейнеролик файлы немесе дыбыстық фрагменттің автоматты ашылу белгісі.

Display – бейнеролик орналасқан компонентті анықтайды.

Visiblebuttons – компоненттің көрінетін батырмаларын анықтайды.

UpDown –  - ішкі айнымалы есептегішті белгілі бір шамаға өзгертуге мүмкіндік беретін екі батырманы сипаттайды.

Name – компонент қасиетіне кіру және плеерді басқару мүмкіндігін алу үшін пайдаланылатын компонент атауы.

Position – есептегіш. Up батырмасын басқанда есептегіш 1-ге артады, ал Down батырмасын басқанда 1-ге кемиді.

Min – Position қасиетінің өзгеру аралығының төменгі шекарасы.

Max – Position қасиетінің өзгеру аралығының жоғарғы шекарасы.

Increment – екі батырманың бірін басқанда Position қасиетінің мәнінің өзгерісін көрсетеді

Associate – Position қасиетінің мәнінің идентификаторы ретінде пайдаланылады.

Orientation – батырманың орналасуын көрсетеді, батырмалар тік немесе көлденең орналасуы мүмкін.

Формалар терезесі. Әрбір жаңа жоба үшін автоматты түрде форма терезесі беріледі және ол Form1 деп аталады немесе жаңа форманы құрастыру File → New → Form опциялары арқылы орындалады. Басында бұл терезе бос болады. Дәлірек айтсақ, ол Windows-тің интерфейстік элементтері үшін стандартты жүйелік мәзірді шақыру батырмаларынан, терезені үлкейту, кішірейту, жабу батырмаларынан, тақырып жолынан және қоршаған шегінен тұрады. Бұл терезенің жұмыс аймағы координаттық тордың нүктелерімен толтырылған, ол формадағы компоненттердің орналасуын реттеу қызметін атқарады. Бұл нүктелерді алып тастау үшін Tools/Environment Options мәзір командаларының көмегімен, Preferences қойымтасындағы Display Grid ауыстырып қосқышындағы жалаушаны алып тастау қажет.

Типі TForm болып келген объект, бағдарламаның негізі болып табылады. Форманың төмендегідей қасиеттері бар.

Name – форма аты. Бағдарламада форма аты форманы басқару үшін және форма компонентіне кіру үшін пайдаланылады.

Caption – тақырып мәтіні жазылады.

Top – форманың жоғарғы шекарасынан экранның жоғарғы шекарасына дейінгі қашықтық.

Left – форманың сол жақ шекарасынан экранның сол жақ шекарасына дейінгі қашықтық.

Width – форманың ені.

Height – форманың биіктігі.

BorderStyle – шекараның түрі, ол жұқа болуы немесе болмауы мүмкін.

BorderIcons – терезені басқару батырмалары. Қасиеттің мәні бағдарлама жұмысы барысында пайдаланушының қандай терезені басқару батырмаларымен жұмыс істей алатынын анықтайды.

Icon – жүйелік менюді шығару батырмасын сипаттайтын сұхбат терезенің тақырыбындағы белгі.

Color – фонның түсі.

Font – қаріп. Форма бетіндегі болатын компоненттердегі пайдаланылатын қаріпті (шрифті) анықтайды.

Canvas – графика шығаруға арналған бет.

Объектілер тармағының терезесі. Бұл терезе екпінді формада немесе мәліметтердің екпінді модулінде орналасқан жеке компоненттер арасындағы байланысты көрнекі бейнелеуге арналған. Терезедегі кез келген компонентті белгілеу формалар терезесіндегі сәйкес компонентті екпінді жасайды және объектілер инспекторы терезесінде осы компоненттің қасиеті бейнеленеді. Екі рет тышқанмен белгілеу Code Insight механизмінің жұмыс істеуіне әкеледі, ал OnClick оқиғасын өңдеу үшін дайындалғанды код терезесіне орналастырады.

Объектілер инспекторының терезесі. Объектілер инспекторы объектілер қасиетін орналастыру мүмкіндігін береді және бағдарлама құру кезінде әдістерді тағайындайды. Объектілер инспекторы терезесі екі жапсырмадан тұрады: Properties (қасиет) және Events (оқиға). Properties беті – компоненттің керекті қасиеттерін тағайындау қызметін атқарады, Events беті – қандай да бір оқиғаға компоненттің жауабын анықтайды.

Properties жапсырмасында екі баған бар. Сол жағындағы бағанда өңделіп отырған компонент үшін жобалау кезінде қажет болатын барлық қасиеттердің тізімі орналасқан. Оң жақ бағанда сол қасиеттердің мәнін өзгертуге болады.

Events жапсырмасының сол жағындағы бағанда компонент оқиғасын өңдеушілер тізімі келтірілген.

Бағдарламалар коды терезесі. Код терезесі – бағдарлама мәтінін құрып, жөндеуге арналған. Бұл мәтін арнайы ереже бойынша құрылады және бағдарламаның жұмыс алгоритмін сипаттайды. Мәтінді жазу ережелерінің жиынтығы бағдарламалау тілі деп аталады. Delphi жүйесінде Object Pascal бағдарламалау тілі пайдаланылады.

Ең алғашында Код терезесінде толыққанды Windows терезесі ретінде бос форманы басқаруды қамтамасыз ететін алғашқы шағын мәтін шығады. Жобамен жұмыс барысында программист оған бағдарламаны басқаруды жүзеге асыратын алғашқы толықтырулар жасайды.

Тіпті қарапайым бағдарлама құру үшін бағдарлама кодын құрып өзгертуге, жөндеуге тура келеді. Жаңа жобаны ашқаннан кейін төмендегідей жолдар көрінеді:

```
unit Unit1;  
interface  
uses  
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,  
  Dialogs, StdCtrls;  
type  
  TForm1 = class(TForm)  
  private  
    { Private declarations }  
  public  
    { Public declarations }  
  end;  
var  
  Form1: TForm1;  
implementation  
{$R *.dfm}  
end.
```

Delphi бұл жолдарды жаңа форма үшін код терезесіне автоматты түрде қояды. Код терезесі бағдарламаның жұмысы басталғаннан кейінгі жағын қарастырады. Ал форма терезесі оның сыртқы көрінісі. Екі терезе бір бірімен тығыз байланысты, Delphi ортасы

Unit Unit1;

және

Implementation

жолдарының арасына қажетті жолдарды қояды.

Әзір бұл мәтін бөлігіне өзгеріс енгізбейміз. Кейінірек бағдарламаның мәтін терезесіне

{SR.DFM}*

және

End.

жолдарының арасына қажетті мәтінді енгіземіз.

Код терезесімен бірге әдетте, Code Explorer браузер терезесі де екпінді болады, ол терезеде кодтың көп жолдары болған жағдайда керекті элементті іздеуді жеңілдету үшін пайдаланылады.

Браузер терезесіндегі элементті тышқанмен екі рет белгілеу сәйкес объектіні сипаттау үшін код терезесіне курсорды орналастырады немесе оны алғашқы сақтау кезінде орындалады. Аса күрделі емес оқу бағдарламаларын құруда браузер терезесін жауып қоюға болады. Браузер терезесін шығару үшін View Explorer терезесін таңдаймыз.

Форманың қасиеттерін өзгерту. Әрбір жаңа форма құрғанда модуль құрылады. Бағдарламаны компиляциялау кезінде Delphi заты .PAS, .DFM және .DCU болып келген файлдарды әрбір модуль үшін құрады: .PAS файл – бағдарлама коды терезесіндегі мәтін көшірмесінен тұрады; .DFM файлында форма терезесінің мазмұны сақталады; ал .DCU файлдары компилятормен құрылады және ол компоновщиктің жұмысы үшін қажетті базаны береді. Одан кейін ол заты .EXE болып келген біртұтас шақырылатын файлға түрленеді.

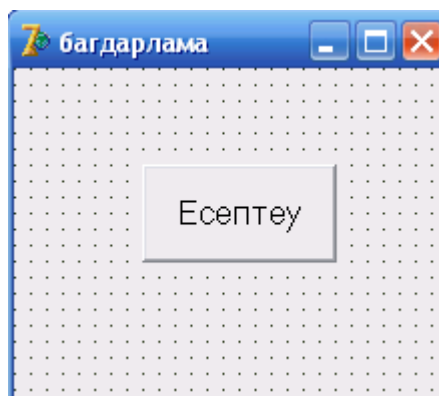
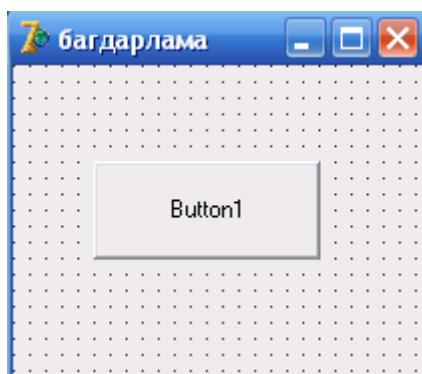
Енді бағдарламаны түрлендіріп көрейік, алдымен терезе тақырыбын өзгертейік. Алғаш Delphi жүйесіне кіргеннен кейін, терезе тақырыбы форма тақырыбына сәйкес келеді. Тақырыпты өзгерту үшін Объектілер инспекторы терезесіне бару қажет. Form1 бағдарламасының терезесін жабыңыз. Объектілер инспекторы терезесінің Caption жолын тышқанмен белгілеңіз. Осы жолдың оң жақ бағаны мәтінмен ерекшеленеді және ерекшеленгеннің оң жағында жыпылықтаған курсор болады. Пернетақтаны кириллицаға ауыстырып, «бағдарлама» деп жазыңыз, одан кейін F9-ды басыңыз. Бағдарламаның бұл орындалуы – тақырыбы «бағдарлама» деп аталатын терезе құру. Дәл осылай формалардың кез келген қасиетін өзгертуге болады.

Формаға жаңа компоненттер орналастыру. Формаға жаңа компонент орналастыру үшін алдымен орналасатын компонентті таңдау қажет, мысалы Standard бетінің Button батырмасын орналастыру үшін компоненттер жинағындағы Standard белгісін сырт еткізіп, бетті екпінді күйге келтіру керек. Button батырмасының кескінін ажырату үшін тышқанды баспай тұрып бетте орналасқан белгілердің үстінен жылжытып көрген сәтте компоненттердің аты шығып тұрады. Қажетті компонентті таңдап, курсорды форма аймағының кез-келген жеріне апарып, тышқанның сол жақ батырмасын жібермей басып тұрып курсорды жылжытып, қажетті мөлшерде тышқан батырмасын босатамыз. Сол кезде форма бетінде Button1 элементі пайда болады.

Формада орналасқан әр компоненттер өзінің мекен-жайымен, мөлшерімен, түсімен т.с.с. анықталады. Компоненттің формадағы орнын, мөлшерін форма терезесінде бірден өзгертуге болады, ол үшін компонентті тышқанның сол жақ батырмасымен белгілеу керек (кішігірім төртбұрыштылар пайда болады). Енді белгіленген компонентті тышқанның сол жақ батырмасымен басып тұрып, форма аймағының кез-келген жеріне жылжытуға

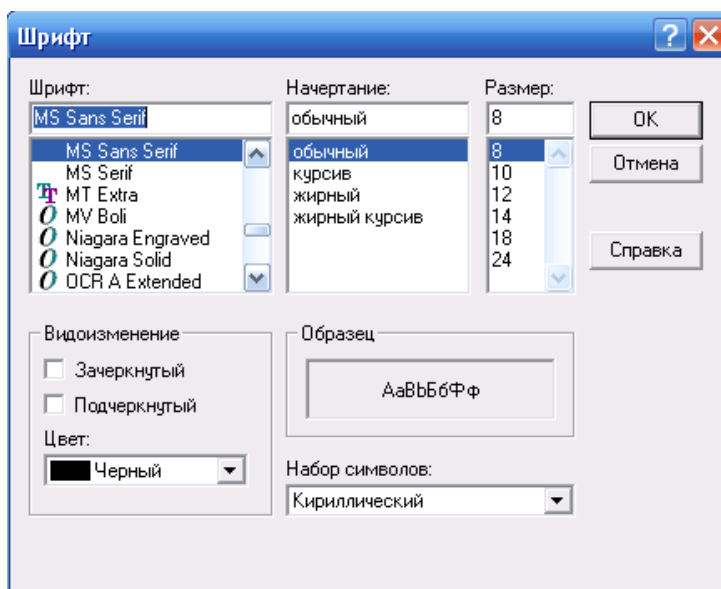
болады. Мөлшерін өзгерту үшін тышқанды элементті қоршап тұрған кез-келген төртбұрышқа орналастырсаңыз, пайда болған қос бағытты тік сызықты тышқанның сол жақ батырмасын басып тұрып жылжытуға болады. Компонент жою үшін оны белгілеп, Delete пернесін басса жеткілікті.

Жаңа компоненттің стандартты аты Button1, батырмадағы жазудың да аты осындай болады. Объектілер инспекторы терезесіндегі Caption жолының көмегімен бұл жазуды оңай өзгертуге болады, «Есептеу» деп енгіземіз. Бұл енгізілген жазу формадағы компоненттен көрінеді.



3-сурет. Формаға Button батырмасын орналастыру

Енді жазудың түсі мен қаріпін өзгертейік. Ол үшін объектілер инспекторы терезесіндегі Font қасиетін тышқанмен белгілейміз, батырманың көмегімен жолдың оң жақ бөлігінен қаріпті түзететін сұхбат терезесі ашылады.



Размер (Өлшемі) тізімінен қаріптің өлшемін 24, Цвет (Түс) тізімінен қажетті түсті таңдаймыз. ОК батырмасымен терезені жабамыз.

Оқиғаға жауапты қалыптастыру

OnClick оқиғасын өңдеуші. Тышқанның сол жақ батырмасын басқанда жұмыс істеп тұрған бағдарламада OnClick оқиғасы пайда болады. Әзір бұл оқиға бағдарламамен ешқандай өңделмейді, сондықтан батырманы басу

ешқандай нәтиже бермейді. Батырма басылғанда жауап беруді жүзеге асыру үшін Object Pascal тілінде бағдарлама фрагментін жазу қажет, ол оқиғаны өңдеуші деп аталады. Бұл фрагментте программист батырма басылғанда бағдарламаның не істейтінін көрсететін мәтіндік жолдар тізбегі жазылады. Фрагмент Object Pascal-дің арнайы қосалқы бағдарламасы – процедура түрінде өрнектеледі.

Delphi OnClick оқиғасын өңдеу процедурасына дайындықты өз бетімен жасау үшін компонентті тышқанмен жылдам екі рет белгілеу қажет. Нәтижесінде Код терезесінде төмендегідей мәтін фрагменті болады:

```
Procedure TForm1.Button1Click (Sender:Tobject);  
    Begin  
  
    End;
```

Мұндағы, Procedure сөзі компиляторға қосалқы бағдарлама - процедураның басы жөнінде хабарлайды. Delphi ортасында сондай-ақ, қосалқы бағдарлама – функция пайдаланылуы мүмкін, бұл жағдайда Function сөзі пайдаланылады. Одан кейін: TForm1.Button1Click процедураның аты жазылған. Атау құрама сөзден тұрады: TForm1 – класс атауы; Button1Click процедураның атауы.

Delphi ортасында класс деп өзіне ұқсас көшірме құрудың үлгісі қызметін атқаратын, аяқталған бағдарлама фрагментін айтады. Программист класс құрғаннан кейін оны бірнеше бағдарламаға немесе бір бағдарламаның бірнеше жеріне қоюға болады. Delphi-дің құрамына Borland коопорациясының программистері құрған стандартты кластар деп аталатын бірнеше жүз кластар кіреді. Стандартты кластар жиынтығы осы бағдарламалау жүйесінің қуатты мүмкіндіктерін анықтайды. Әрбір компонент белгілі бір класқа жатады, ал формаға қойылатын компоненттердің барлық нақты көшірмелері сандық индексі бар класс атауын алады.

Пайдаланылып отырған Delphi ортасында кластардың барлық атауларын T әрпінен бастау келісілген. Сөйтіп, Tform атауы Tform стандартты класс үлгісі бойынша құрылған кластың атын білдіреді. Код терезесіндегі мәтіннің басына назар аударсаңыздар, төмендегідей жолдар көрінеді:

```
Type Tform1=class(Tform)  
    Button1:Tbutton;  
    Label1:Tlabel;  
    Procedure Button1Click (Sender: Tobject);  
    Private  
    {Private declarations }  
    Public  
    {public declarations }  
    End;  
    Var  
    Form1:Tform1;
```


TForm1=class(form) жолы TForm стандартты класынан туған (үлгі бойынша құрылған) TForm1 жаңа класын анықтайды. Ал, Form1:TForm1 жолы осы кластың Form1 атты көшірмесін құрады. Стандартты TForm класы бос Windows терезесін сипаттайды, сонымен бірге TForm1- белгі және батырма компоненттері қойылған терезені сипаттайды. Бұл компоненттерді сипаттайтын жолдар:

```
Button1: TButton;  
Label1: TLabel;
```

Мұндағы Button1 компоненті TButton стандартты класының экземпляры; ал Label1 компоненті TLabel стандартты класының экземпляры.

TForm1.Button1 Click процедура атауынан кейін дөңгелек жақшада параметрді шақыру сипатталған:

```
Sender: TObject
```

«Sender» атты параметр TObject класына жатады. Sender параметрі Delphi ортасына барлық жағдай үшін қойылған: оның көмегімен Button1Click қосалқы бағдарламасы OnClick оқиғасы қандай компонент құрғанын анықтап береді. Жол толығымен

```
Procedure TForm1.Button1 Click (Sender: TObject);
```

процедураның тақырыбы деп аталады. Object Pascal тілінде «;» белгісі тілдің сөйлемінің аяқталғандығын компиляторға хабарлайды. Бағдарлама мәтіні жеке сөйлемдерден құралады. Әрбір сөйлемнің соңына нүктелі үтір қойылады. Төмендегі үш жол процедура денесін анықтайды:

```
Begin
```

```
End;
```

Begin сөзі компиляторға процедураның жұмыс алгоритмін сипаттайтын сөйлемдер тізбегінің бастамасы екені туралы белгі береді, ал End сөзі осы тізбектің соңын білдіреді. Біздің жағдайда процедура денесінде ешқандай сипаттама жоқ. Delphi процедура жазуға дайындықты құрып берді. программист процедура денесін қажетті сөздермен толтырады. Button1 батырмасын әрбір басқан сайын басқару процедура денесіне беріледі, яғни begin және end сөздерінің арасына оқиғаға жауап ретінде орындалатын бағдарлама фрагменті жазылады. Яғни, Delphi-де бағдарлама құру процесі екі кезеңге бөлінеді: форма құру және кодтау.

1-есеп. Формада «Іске қосу» батырмасын басқанда, Label1 компонентінде «Бұл менің Delphi тілінде бірінші рет енгізген бағдарламам!» деген сөйлем шығатын бағдарлама құрыңыз.

Форма құру барысында бірінші кезекте Caption қасиетінің мәнін өзгертуге болады. Біздің мысалымыз бойынша Form1 мәтінін Алғашқы есеп деп өзгертейік. Ол үшін Object Inspector терезесіндегі Caption жолында Алғашқы есеп сөзін енгізіңдер.

Форманың биіктігін және енін Object Inspector терезесіндегі Height және Width қасиеттерінің мәндерін өзгерту арқылы қоюға болады.

Қасиеттер	Меншіктелетін мәндері
Form1	Алғашқы есеп
Height	260
Width	600
Font.size	12 (қаріп мөлшері)
Color	clLime (форма түсі ашық жасыл түс береді)

Формаға Standard бетінен Label және Button компоненттерін орналастыру үшін, компонент палитрасында сол компоненттің шарт белгісінде тышқанның сол жақ батырмасын басып, таңдап алып, сосын формада компоненттің жоғарғы сол жақ бұрышы орналасатын орынға курсорды қойып, тышқан батырмасын бір рет басу. Нәтижесінде формаға стандарт өлшемді компонент пайда болады. Компоненттердің қасиеттері төмендегідей:


Компоненттер	Қасиеттері	Меншіктелетін мәндері
Label1	Caption	Бос
	Font	Times New Roman; өлшемі – 16; қарайтылған; түсі – қызыл.
Button1	Caption	Іске қосу

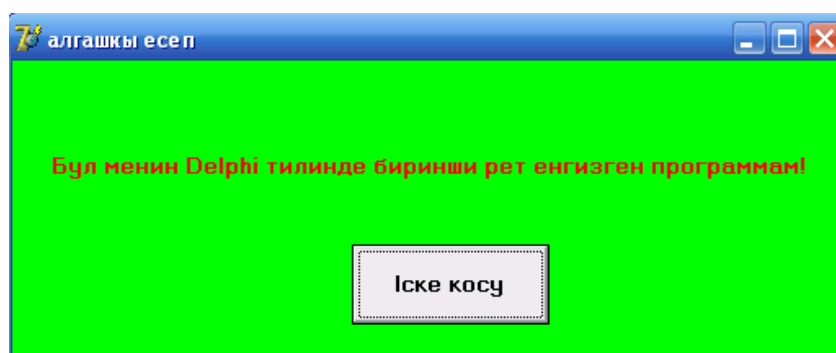
Форма қосымшасын құрып болған соң бағдарлама мәтінін жазуға кірісуге болады. «Іске қосу» батырмасына екі рет шерту арқылы, бағдарлама коды терезесіне өтіндер. Сонда OnClick оқиғасы процедурасының дайындамасы көрінеді. Процедура денесін енгізіндер.

```

procedure TForm1.Button1Click(Sender: TObject);
begin
  Label1.Caption:= 'Бұл менің Delphi тілінде бірінші рет енгізген
бағдарламам'
end;
end.

```

Бағдарламаны орындау үшін мәзір жолынан Run командасын таңдаймыз немесе F9 функционалды пернесін басамыз. Және де саймандар тақтасынан  Run(F9) батырмасын басамыз. Run(F9) командаларын орындағанда 3 саты орындалады: компиляция, жинақтау, бағдарламаның орындалуы.



Құрған бағдарламаны File – Save As командалары арқылы өзіңіздің бумаңызда сақтаңыз. Ол бумада сіз құрған жобаның файлдары бар. Осы файлдармен танысайық:

- *.dpr – жоба файлы.
- *. pas – бағдарламалық модуль файлы.
- *. dfm - форма файлы.
- *. dcu- компиляцияланған бағдарламалық модуль файлы.
- *. res- ресурстар файлы.
- *. cfg- жоба баптауларының компиляцияланған файлы.
- *. dof – жоба баптауларының мәтіндік файлы.
- *. dpr, *. pas, *. dfm – жобаның, модульдің резервтік көшірмелерінің файлы.
- *. exe- қосымшаның орындалатын файлы.

Бақылау сұрақтары:

1. Delphi ортасының жұмысын басқаратын неше негізгі терезе бар?
2. Негізгі терезенің атқаратын қызметі қандай және ол қандай бөлімдерден тұрады?
3. Форма терезесінің міндеті не?
4. Компонент дегеніміз не?
5. Формаға компоненттер қалай орналастырылады?
5. Объектілер инспекторы терезесі қандай қызмет атқарады?
7. Бағдарламалар коды терезесінің қызметі қандай және ол форма терезесімен қалай байланысады?
8. Бағдарламалар коды терезесі ашылғанда Delphi ортасы қандай жолдарды автоматты түрде шығарады?
9. OnClick оқиғаны өңдеушінің қызметі қандай?
10. OnClick оқиға өңдеушісіне байланысты код код терезесіндегі шығатын мәтінге сипаттама беріңіз ?
11. Класс дегеніміз не және ол бағдарламалар коды терезесінде қай бөлімде сипатталады?

Зертханалық жұмыс №2. Сызықтық алгоритмдерді бағдарламалау

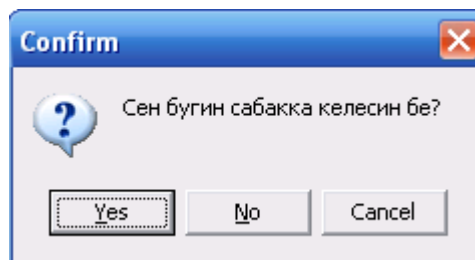
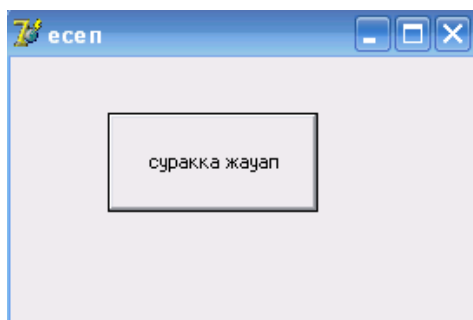
Жұмыстың мақсаты: Delphi ортасында сызықтық алгоритмдерді бағдарламалап, есептер шығарып үйрену дағдыларын қалыптастыру және өзіндік тапсырмаларды орындау.

Бағдарламалық дидактикалық қамтылуы: Дербес компьютерлер, Windows XP операциялық жүйесі, Borland Delphi 7 бағдарламасы, «Бағдарламалау» пәнінен оқу-әдістемелік кешен.

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

1-есеп. Сұраққа жауап беретін бағдарлама

```
procedure TForm1.Button1Click(Sender: TObject);
begin
    MessageDlgPos('Сен бугин сабакка келесин бе?', mtConfirmation,
mbYesNoCancel, 0, 200, 200);
end;
end.
```



2-есеп. Форма терезесінде «Информатика мамандығы» деген мәтіннің қаріп өлшемін үлкейтетін, кішірейтетін және мәтіннің орнын ауыстыратын бағдарлама.

Бағдарламаға қажетті компоненттер: Label1; Button1; Button2; Button3; Button4;

Компоненттер	Қасиеттері	Меншіктелетін мән
Form1	Caption	бағдарлама
	Color	clMenuHighlight
Label1	Caption	Информатика мамандығы
	Font	өлшемі – 10; қарайтылған; қызыл
	Alignment	taCenter
	AutoSize	False
Button1	Caption	үлкейту
Button2	Caption	кішірейту
Button3	Caption	жылжыту
Button4	Caption	шығу

```
procedure TForm1.Button1Click(Sender: TObject);
begin
    label1.Font.Size:=Label1.Font.Size+4;
end;
```

```
procedure TForm1.Button2Click(Sender: TObject);
begin
    label1.Font.Size:=Label1.Font.Size-4;
```

```

end;

procedure TForm1.Button3Click(Sender: TObject);
begin
  label1.Left:=label1.Left+5;
  label1.Top:=label1.Top+5;
end;

procedure TForm1.Button4Click(Sender: TObject);
begin
close
end;

end.

```



3-есеп. Екі бүтін санның қосындысын есептеп, нәтижені экранға шығаратын бағдарлама.

Форма құру барысында бірінші кезекте Caption қасиетінің мәнін өзгертуге болады. Біздің мысалымыз бойынша Form1 мәтінін Қосынды деп өзгертейік. Ол үшін Object Inspector терезесіндегі Caption жолында Қосынды сөзін енгізіңдер.

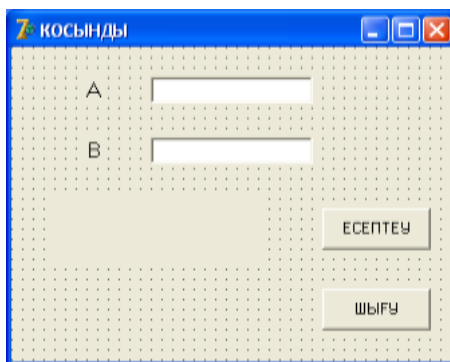
Осындай әрекетпен, форманың биіктігін және енін анықтайтын Height және Width қасиеттерінің мәндерін қоюға болады. Компоненттер үшін өзгертін қасиеттерді былай сипаттауға болады:

Компонент	Қасиеттері	Меншіктелетін мән
Form1	Caption	Қосынды
	Height	250
	Width	330
	Font.size	12

Екі бүтін санның қосындысын есептеу бағдарламасы пайдаланушыдан берілгендерді – екі бүтін санды енгізуді талап етеді. Берілген айнымалы редакциялау өрісінде пернетақтадан енгізілуі керек. Сондықтан формаға Edit компонентін қосу керек. Жиі пайдалынатын компоненттер Standart қосымшасында болады. Осы Standart қосымшасынан нәтижені шығару үшін Label компонентін, екі санның қосындысын есептеуге Button командалық

батырмасын рет-ретімен таңдап, оларды формаға орналастыру. Формаға компонентті қосу үшін, компонент палитрасында сол компоненттің шарт белгісінде тышқанның сол жақ батырмасын басып, таңдап алып, сосын формада компоненттің жоғарғы сол жақ бұрышы орналасатын орынға курсорды қойып, тышқан батырмасын бір рет басу. Нәтижесінде формаға стандарт өлшемді компонент пайда болады. Компоненттердің қасиеттерін төмендегідей орнату керек:

Компоненттер	Қасиеттері	Меншіктелетін мән
Edit1	Text	Бос
Edit2	Text	Бос
Label1	Caption	А
Label2	Caption	В
Label3	Caption	Бос
Button1	Caption	Есептеу
Button2	Caption	Шығу



Форма қосымшасын құрып болған соң бағдарлама мәтінін жазуға кірісуге болады. Есептеу батырмасына екі рет шерту арқылы, бағдарлама коды терезесіне өтіндер. Сонда OnClick оқиғасы процедурасының дайындамасы көрінеді. Процедура денесін енгізіңдер.

```

Procedure TForm1.Button1Click(Sender:Tobject);
  Var a,b,s: integer;
  Begin
    a:=StrToInt(Edit1.text);
    b:= StrToInt(Edit2.text);
    s:=a+b;
    Label3.caption:='ҚОСЫНДЫ='+IntToStr(s);
  end;
end.

```

Шығу батырмасының бағдарламасын, осындай тәсілмен Button2 объекті үшін өңдеу процедурасын былай құруға болады:

```

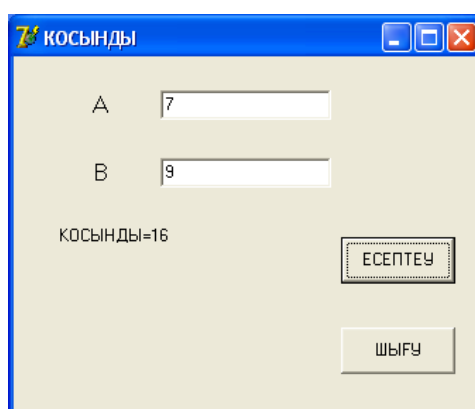
Procedure TForm1.Button2Click (Sender: TObject)
begin
  Form1.Close;
end;
end.

```

Бағдарламаны орындау командасын (F9) басқанда, бағдарлама компиляцияланып, жүйе Қосынды тақырыбы жазылған форманы экранға шығарады.

Курсорды Edit1, Edit2 өрістеріне қойып, а, b мәндерін енгізу (7 және 9 сандарын).

Енді формадағы Есептеу батырмасын басу. Сонда Label3 өрісіне нәтиже шығарылады.



Бағдарламаның орындалуын аяқтау үшін Шығу батырмасына шертіндер. Жобаны сақтау керек.

Жобаны сақтау үшін File - Save All жарлықтарын орындаймыз. Егер жоба алғаш рет сақталып отырса, онда Delphi бірінші модульді сақтауды береді, сондықтан Save Unit1.As терезесі экранға шығады. Терезенің файл аты өрісіне модуль атын енгізіп, Сақтау батырмасын басу керек. Модуль .pas тіркеуі бойынша сақталынады да, экранда жобаны сақтауды сұрайтын терезе көрінеді. Файл аты өрісіне жоба атын енгізіп, Сақтау батырмасын басу. Жоба файлы .dpr кеңейтілуімен сақталады. Модуль және жоба файлдарының аттары әр түрлі болғаны дұрыс.

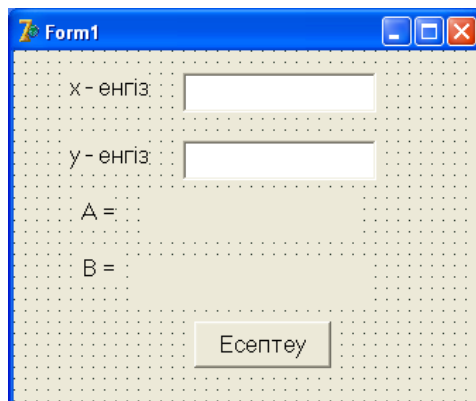
4-есеп. Берілген формула бойынша есептеулер жүргізетін бағдарлама.

$$A = \frac{x^3 + \sin(xy) + y^3}{\cos(\ln x + 1)},$$

$$B = \frac{2}{x} + \sqrt{y + \sin x} + \frac{1}{y}$$

1-әдіс.

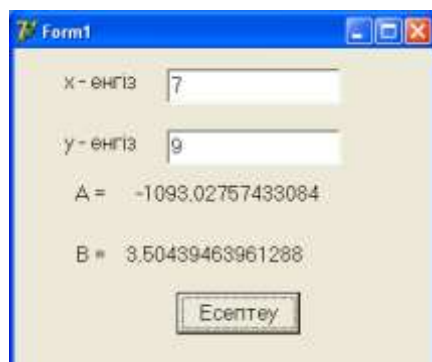
Формаға Label1, Label2, Label3, Label4, Label5, Label6, Edit1, Edit2, Button1 компоненттерін орналастыру. Формада көрсетілгендей компоненттердің қасиеттерінің мәндерін қойып шығу.



Процедура бағдарламасын жазу. Ол үшін Есептеу батырмасына екі рет басу, сонда бағдарлама коды терезесі шығады. Келесі процедура денесін енгізу.

```
procedure TForm1.Button1Click(Sender: TObject);
var x,y,a,b: real;
begin
  x:=strtofloat(edit1.text);
  y:=strtofloat(edit2.Text);
  a:=(sqrt(x)*x+sin(x*y)+sqrt(y)*y)/cos(ln(x)+1);
  b:=2/x+sqrt(y+sin(x))+1/y;
  label5.Caption:=floattostr(a);
  label6.Caption:=floattostr(b);
end;
```

Бағдарламаны орындау командасын (F9) басқанда, формадағы Label5, Label6 өрістеріне нәтиже шығарылады.



2-әдіс.

4-есепті екінші әдіспен шығарып көрсетейік. Бұл әдісте InputBox енгізу терезелерін және ShowMessage шығару терезесін пайдаланамыз.

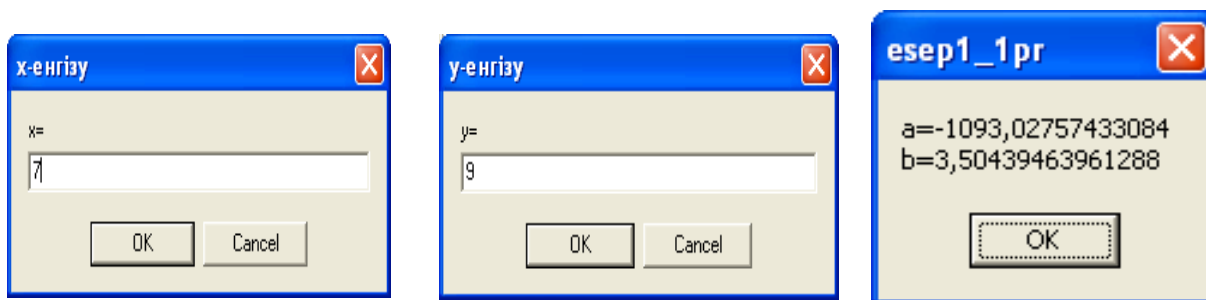
Жаңа формаға екі рет шерту, бағдарлама коды терезесі шығады. Бағдарлама кодын жазамыз.


```

procedure TForm1.FormCreate(Sender: TObject);
var x,y,a,b:real; x1,y1:string;
begin
  x1:=InputBox('x-енгізу','x=','');
  y1:=InputBox('y-енгізу','y=','');
  x:=StrToFloat(x1);
  y:=StrToFloat(y1);
  a:=(sqr(x)*x+sin(x*y)+sqr(y)*y)/cos(ln(x)+1);
  b:=2/x+sqrt(y+sin(x))+1/y;
  ShowMessage('a='+floattoStr(a)+#13
              +'b='+floattoStr(b));
end;

```

Бағдарламаны орындау командасын (F9) басқанда, InputBox терезесі экранға шығады. Бірінші терезеге x-тің мәнін енгізу. Екінші терезеде y-тің мәнін енгізу. OK батырмасын басып отыру керек. Нәтиже ShowMessage терезесіне шығады.



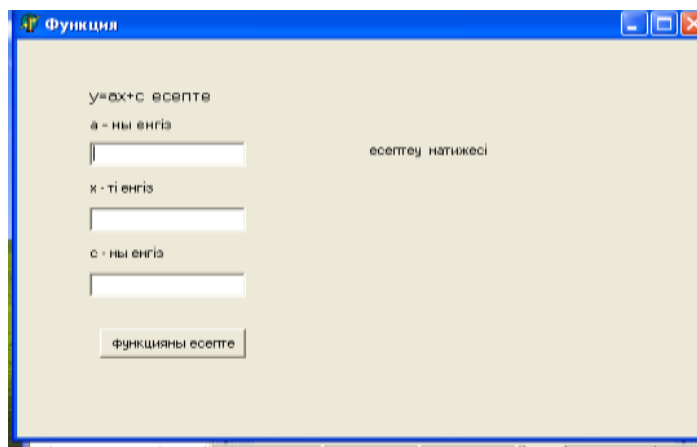
5-есеп. $y=ax+c$ функциясының мәнін есептейтін бағдарлама.

Ол үшін аргумент – бастапқы берілгендерді және нәтиже - нені анықтау қажеттігін білуіміз керек. Бұл мысалда a , c , x – бастапқы берілгендер, ал y – нәтиже .

$a=5$, $c=10$ және $x=3$ үшін функцияның мәнін есептеуді Delphi бағдарламалау ортасында жүргізейік. Формаға компоненттер палитрасының Standard бетінде орналасқан Edit, Label, Button компоненттерін орналастырып сәйкес мәндерін меншіктейміз. Бастапқы мәліметтерді, яғни a , c , x айнымалыларының мәндерін Edit1, Edit2, Edit3 компоненттерінің өрістеріне енгізіп, ал шешімін Label5 компонентінің өрісіне шығарайық. Есеп шешімін Button1 компонентінің OnClick оқиғасында ұйымдастырамыз.

Форма мен оның компоненттерінің қасиеттерін төмендегідей орнату қажет.

Компоненттер	Қасиеттері	Меншіктелетін мән
Form1	Caption	Функция
	Name	Form1
Edit1	Text	тазалау
Edit2	Text	тазалау
Edit3	Text	тазалау
Label1	Caption	$y=ax+c$ есептеу
Label2	Caption	а-енгіз
Label3	Caption	х-ті енгіз
Label4	Caption	с-ны енгіз
Label5	Caption	тазалау
Button1	Caption	өрнекті есепте



Төменде Button1 компонентінің OnClick оқиғасының коды келтірілген.

```

procedure TForm1.Button1Click(Sender: TObject);
var a,x,c,y:real; // айнымалыларды сипаттау бөлігі :a,x,c айнымалылары
нақты типті деп сипатталған
Begin // процедура басы
a:=strtofloat(edit1.Text); //edit1-ге енгізілген тексттің мәні түрленгеннен
кейін а-ға меншіктеледі
x:=strtofloat(edit2.Text); //edit2-ге енгізілген тексттің мәні түрленгеннен
кейін х-ке меншіктеледі
c:=strtofloat(edit3.Text); //edit3-ке енгізілген тексттің мәні түрленгеннен
кейін с-ға меншіктеледі
y:=(a*x+c);
label5.Caption:='жауабы '+ #13+'y='+floattostr(y); // нәтижені символға
айналдырып label5-тің өрісіне жазылады.
end;

```

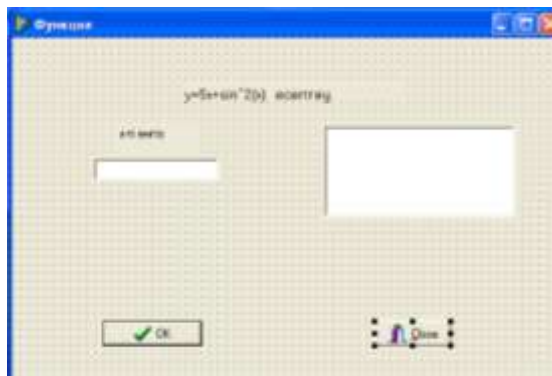
6-есеп. $y=5x+\sin^2x$ функциясының мәнін есептейтін бағдарлама.

Бұл мысалда x – нақты сан, ал y – нәтиже .

$x=3,14$ мәні үшін функцияның мәнін есептеуді Delphi бағдарламалау ортасында жүргізейік. Формаға компоненттер палитрасының Standard бетінде орналасқан Edit, Label, Memo және Additional бетіндегі Bitbtn компоненттерін орналастырып, сәйкес мәндерін меншіктейміз. Бастапқы мәліметті, яғни x айнымалысының мәнін Edit1 компонентінің өрісіне енгізіп, ал шешімін Memo1 компонентінің өрісіне шығарайық. Есеп шешімін Button1 компонентінің OnClick оқиғасында ұйымдастырамыз.

Форма мен оның компоненттерінің қасиеттерін төмендегідей орнату қажет.

Компоненттер	Қасиеттері	Меншіктелетін мән
Form1	Caption	Функция
Form1	Name	Form1
Edit1	Text	тазалау
Label1	Caption	$y=5x+\sin^2x$ есептеу
Label2	Caption	x -енгіз
Bitbtn1	Caption	OK
Bitbtn1	Kind	bkOK
Bitbtn2	Caption	&Close
Bitbtn2	Kind	bkClose
Memo1	Lines	тазалау



"✓ok" батырмасын екі рет шертіп бағдарлама коды терезесіне көшіңіз. OnClick оқиғасы процедурасының дайындамасы көрінеді.

Төменде BitBtn1 компонентінің OnClick оқиғасының коды келтірілген.

```
procedure TForm1.BitBtn1Click(Sender: TObject);
var x,y:real;
begin
edit1.SetFocus; // енгізу фокусы edit1 атты компонентке беріледі
```

```

x:=strtofloat(edit1.text);
y:=5*x+sqr(sin(x));
memo1.Clear; // memo1-гі шығару өрісін тазалау
memo1.Lines.add('y=5x+sin^2(x)-тін ');
memo1.Lines.add('x='+edit1.Text+' болғанда');
memo1.Lines.add('жауабы: '+y=floattostr(y) ); // memo1 өрісіне тексті
және функция мәнін символға айналдырып шығару
end;

```

7-есеп. Спортсменнің берілген ара қашықтықтағы жүгіру жылдамдығын анықтайтын бағдарлама.

Бағдарламаға қажетті компоненттер: Label1; Label2; Label3; Label4; Edit1; Edit2; Button1; Button2.

```

procedure TForm1.Button1Click(Sender: TObject);
var
  dist: integer; // // арақашықтық метрмен
  t: real; // уақыт, бөлшек сан түрінде
  min: integer; // уақыт, минутпен
  sek: integer; // уақыт, секундпен
  v: real; // Жылдамдық
begin
  // алғашқы мәліметтерді енгізу алаңынан алу
  dist := StrToInt(Edit1.Text);
  t := StrToFloat(Edit2.Text);
  // алдын-ала түрлендірулер
  min := Trunc(t); // минут – бұл t санының бүтін бөлігі
  sek := Trunc(t*100) mod 100; // секунд – бұл t санының бөлшек бөлігі
  // есептеу
  v := (dist/1000) / ((min*60 + sek)/3600);
  // Нәтижені шығару
  label4.Caption := 'Арақашықтық: '+ Edit1.Text + ' м' + #13 +
  'Уақыт: ' + IntToStr(min) + ' мин ' + IntToStr(sek) + ' сек ' + #13 +
  'Жылдамдық: ' + FloatToStrF(v,ffFixed,4,2) + ' км/сағ';
end;
end.

```

Button1Click функциясы жылдамдықты есептейді және нәтижені Label4 алаңына шығарады. Алғашқы берілгендер Edit1, Edit2 редакциялау алаңдарында текст түрінде енгізіледі. Осы текст түріндегі жолды санға айналдыру үшін StrToInt және StrToFloat функциялары пайдаланылады. Олар сәйкесінше dist айнымалыға бүтін, ал t айнымалыға бөлшек сан ретінде меншіктеледі.

Trunc функциясы санның бөлшек бөлігін алып тастайды.

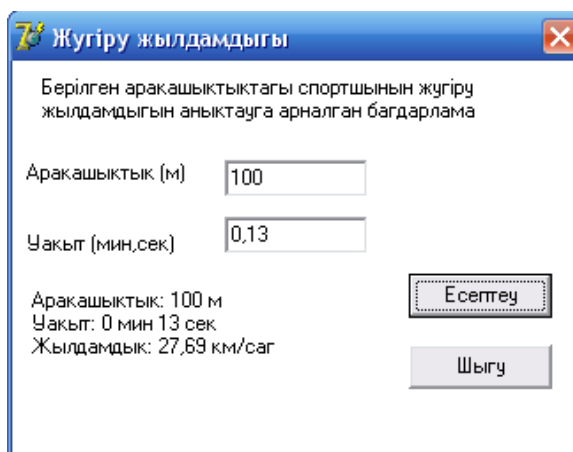
$\text{Trunc}(t*100) \bmod 100$,

t саны 100–ге көбейтіледі, алынған санның бүтін бөлігін 100-ге модуль бойынша бөлінеді. Нәтиже қалдық болады.

Жылдамдықтың мәні label4 алаңына мәнді Caption қасиетіне меншіктеу арқылы шығарылады. Сандарды жолға айналдыру үшін IntToStr және FloatToStr функциялары пайдалынылады.

Шығу батырмасын басқанда бағдарлама жұмысын аяқтау керек. Ол үшін Button2 батырмасына екі рет шерту арқылы бағдарлама кодының терезесіне өтіп, былай жазу керек:

```
// Шығу
Procedure TForm1.Button2 Click (Sender: TObject);
begin
Form1.Close;
end;
```



8-есеп. Бір тәулікте, аптада, жылда қанша секунд бар екенін анықтайтын бағдарлама.

Формаға бір батырма орналастырайық. Осы батырмаға есептің шығару жолын жазамыз және есептің жауабы да осы батырманы шерткенде шығады.

Жұмыстың барысы:

Delphi-ді қосқанда автоматты түрде қосылатын бір рет Button батырмасын орналастырыңдар

Алдын-ала жоба файлы және модуль файлы сақтап алыңдар:

File/Save Project As..., File Save As..., File/Save Al...,

Форманың өлшемін былай өзгертуге болады:

- Форманың жиегін тышқанмен басып жібермей отырып қажетті жерімізге дейін тасымалдаймыз;

- Форманың қасиеттерінде оның ені мен биіктігін көрсету керек

Форманың объектілерді түзету үшін Edit/Align командасын пайдаланамыз. Осы батырманы басу әрекетін өңдейтін процедураға дайындық пайда болады.

```
Procedure TForm1.But1Click(Sender: TObject);
```

Begin

end;

Айнымалыларды сипаттау бөлімінде r,t,s,p,k - айнымалыларын былай сипаттаймыз.

var r, t: integer;

s, p, k: integer;

Сонда бағдарлама:

Procedure TForm1.Button1Click (Sender: TObject);

Var r,t: integer;

s, p, k: integer;

Begin

R:=StrToInt(InputBox('r-ді енгіз', ''));

T:=StrToInt(InputBox('t-ді енгіз', ''));

S:=r*t;

P:=r*7*t;

K:=r*365*t;

ShowMessage ('s-күннің мәні'+inttostr(s)+'#13+'аптаның мәні'+ inttostr(p)
+#13+ 'жылдың мәні'+ inttostr(k))

end; end.

F9 батырмасын басқанда, құрылған форманың терезесі пайда болады. Формада r және t-ға мән беріп, батырманы қайта бассақ есептің жауабы шығады.

Тапсырмалар:

Формулалар бойынша есептеулер жүргізетін бағдарлама құрындар:

1. $B = \frac{2}{x} + \sqrt{y + \sin x} + \frac{1}{y}$

2. Шардың көлемін анықтау $V = 4 \pi r^3$ ($\pi = 3.14$)

3. $R = \sqrt{xy \sin 2x + \frac{e^{-2x}(x+y)}{x^2 + y^2}}$

4. Шеңбердің радиусының өлшемі берілген, оның ауданын $S = \pi r^2$ анықтау қажет.

5. $A = \sqrt{\frac{(n^3 - m^2)}{t} + (\cos n + t)}$

6. Тіктөртбұрыштың a, b қабырғалары берілген, оның ауданын $S = a * b$ және периметрін $P = 2 * (a + b)$ анықтау

7. Негізі үшбұрыш болатын пирамиданың көлемін анықтау $V = \frac{S h}{3}$

мұндағы S - үшбұрыштың ауданы, h - пирамида биіктігі

8. $B = tg \sqrt{n} + ctg \sqrt{2m} + |n - 3m|$

Бақылау сұрақтары:

1. Компоненттер деп нені түсінеміз?
2. Delphi ортасындағы негізгі компоненттер орналасқан беттерді атаңыз?
3. Пайдаланушымен қарапайым сұхбат ұйымдастыру үшін қандай беттен қандай компоненттерді пайдалануға болады?
4. Мәліметтерді енгізу үшін қандай беттен қандай компонент пайдаланылады?
5. Терезеге түсіндірме жазу үшін қандай компонент пайдаланылады?
6. Орындалған нәтижені экранға шығару үшін қандай компонент пайдаланылады?
7. Нәтижені есептеу үшін қандай компоненттер пайдаланылады?
8. Сурет, радиобатырмалар қою үшін қандай компоненттер пайдаланылады?
9. Көп жолдық форматталған мәтін редакторы қандай қызмет атқарады және оның MEMO редакторынан айырмашылығы неде?
10. Мәліметтерді шығаруды ұйымдастыруға қандай компоненттерді пайдалануға болады?

Зертханалық жұмыс №3. Тармақталу алгоритмдерін бағдарламалау

Жұмыстың мақсаты: Delphi ортасында тармақталушы алгоритмдерді бағдарламалау дағдыларын қалыптастырып, есептер шығару және өзіндік тапсырмаларды орындау.

Бағдарламалық дидактикалық қамтылуы: Дербес компьютерлер, Windows XP операциялық жүйесі, Borland Delphi 7 бағдарламасы, «Бағдарламалау» пәнінен оқу-әдістемелік кешен.

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

Шартты оператордың құрылымы төмендегідей:

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

Мұндағы: If, then, else - қызметші сөздер (IF – егер; THEN – онда; ELSE - әйтпесе).

<шарт> - логикалық типті қандай да бір өрнек;

<1-оператор> және <2-оператор> - Object Pascal-дің кез келген операторлары. Мұндағы: 1 және 2-операторлардың өзі қарапайым немесе құрама оператор болуы мүмкін

Шартты оператордың орындалу алгоритмі төмендегідей:

Егер, **IF** сөзінен кейін жазылған шарт сақталса (ақиқат болса), онда **THEN** сөзінен кейін жазылған 1-оператор орындалады, 2-оператор

орындалмайды. Шарт сақталмаса (жалған болса), онда керісінше <1-оператор> орындалмай, **ELSE** сөзінен кейінгі 2-оператор орындалады.

IF сөзінен соң жазылатын шартты жазу үшін Object Pascal тілінде өрнектерді салыстыру: >, >=, <, <=, < > белгілерін пайдаланады. Күрделі логикалық өрнектер and, or, not логикалық операцияларының көмегімен орындалады. Мысалы, if (a>b) and (b<>0) then... .

IntToStr (k) - k бүтін санын жолға айналдыратын функция.

StrToFlat (k) – k жолдық шамасын нақты санға айналдыратын функция.

FloatToStrF(k,ffgeneral,5,2) - k шамасын жолдық шамаға түрлендіреді. Мұндағы ffgeneral сөзі нақты санның форматын көрсетеді; 5- нәтижеге берілетін барлық орындар саны, ал 2 – үтірден кейінгі орындар саны.

InputBox (*'терезе тақырыбы'*, *'нұсқау: енгізілетін мәндер'*,”). Мұндағы - *терезе тақырыбы* - сұхбат терезесінің тақырыбы, *нұсқау* - пайдаланушы енгізген түсіндірме, ал *енгізілетін мәндер* - енгізу өрісіне енгізуге болатын мәндер тізімі. Бұл функцияның орындалуы нәтижесінде экранға енгізу өрісі бар сұхбат терезе пайда болады.

ShowMessage (s) - экранға хабарламамен бірге командалық ОК батырмасымен шығаратын процедура

1-есеп. Цельсий шкаласында берілген температура мәнін Фаренгейт, Кельвин және Ренкин шкалаларындағы сәйкес мәндеріне түрлендіретін бағдарлама.

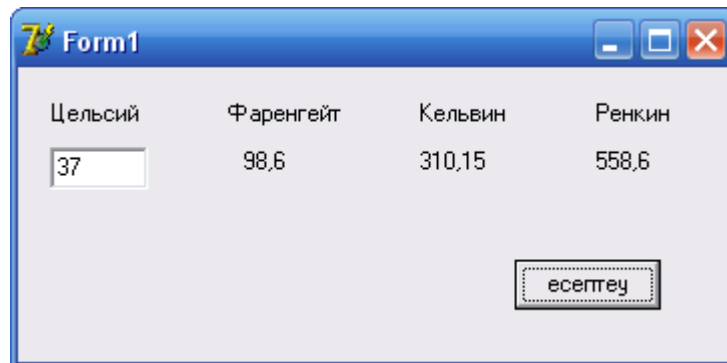
```
unit Unit1;
interface
uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls;
type
  TForm1 = class(TForm)
    Label1: TLabel;
    Label2: TLabel;
    Label3: TLabel;
    Label4: TLabel;
    Label5: TLabel;
    Label6: TLabel;
    Label7: TLabel;
    Edit1: TEdit;
    Button1: TButton;
    procedure Button1Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;
var
```



```

Form1: TForm1;
implementation
{$R *.dfm}
procedure TForm1.Button1Click(Sender: TObject);
var Celsius: Extended;
begin
Celsius:=StrToFloat(Edit1.Text );
label5.Caption :=FloatToStr(1.8*Celsius+32);
label6.Caption := FloatToStr(Celsius+273.15);
label7.Caption := FloatToStr(1.8* Celsius+492);
end;
end.

```



2-есеп. 1000 теңгеден жоғары сомада зат сатып алғанда берілетін жеңілдікті есептейтін бағдарлама.

```

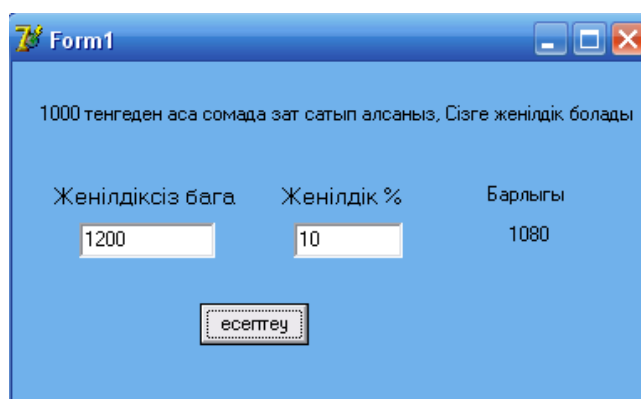
unit Unit1;
interface
uses
Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
Dialogs, StdCtrls;
type
TForm1 = class(TForm)
Label1: TLabel;
Edit1: TEdit;
Label2: TLabel;
Edit2: TEdit;
Label3: TLabel;
Button1: TButton;
Label4: TLabel;
Label5: TLabel;
procedure Button1Click(Sender: TObject);
private
{ Private declarations }
public
{ Public declarations }
end;

```

```

var
  Form1: TForm1;
implementation
{$R *.dfm}
procedure TForm1.Button1Click(Sender: TObject);
var cena,skidka: Extended;
begin
  cena:=StrToFloat(Edit1.Text );
  skidka:=strtofloat(edit2.Text);
  if cena<1000 then
    label3.Caption :=FloatToStr(cena)
  else label3.caption:=floattostr(cena-(cena/skidka));
  end;
end.

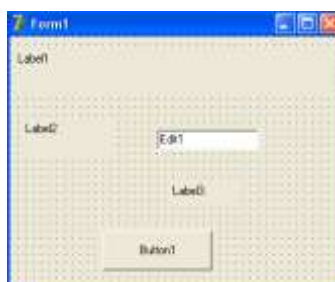
```



3- есеп. Жыл мезгілдерін Ай бойынша табатын бағдарлама.

Бағдарламаға қажетті компоненттер: Label1; Label2; Edit1; Label3; Button1;

Компоненттер	Қасиеттері	Меншіктелетін мән
Form1	Caption	Form1
Form1	Name	Form1
Label1	Caption	Жыл мезгилдерин ай бойынша табу
Label2	Caption	Айды санмен енгиз
Label3	Caption	Тазалау (бос)
Edit1	Text	тазалау
Button1	Caption	есептеу



```

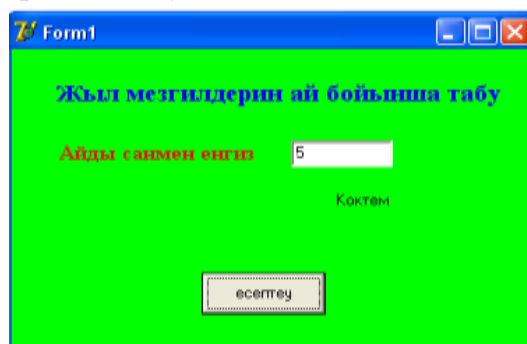
unit Unit1;
interface
uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls;
type
  TForm1 = class(TForm)
    Label1: TLabel;
    Edit1: TEdit;
    Label2: TLabel;
    Button1: TButton;
    Label3: TLabel;
    procedure Button1Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;
var
  Form1: TForm1;
implementation
{$R *.dfm}

```

```

procedure TForm1.Button1Click(Sender: TObject);
Var m:integer;
begin
M:=StrToInt(Edit1.Text);
if (m=12) or (m=1) or (m=2) then Label3.Caption:='Кыс'
else
if (m=3) or (m=4) or (m=5) then Label3.Caption:='Коктем'
else
if (m=6) or (m=7) or (m=8) then Label3.Caption:='Жаз'
else
if (m=9) or (m=10) or (m=11) then Label3.Caption:='Кыз'
else
if m>12 then Label3.Caption:='Мундай ай жок!';
end;
end.

```



4-есеп. Функция мәнін есептейтін бағдарлама.

$$y = \begin{cases} \sin(\cos x) + \sqrt{x^3 + 4}, x > 0,9 \\ \cos(\sin x) + \sqrt[3]{|x| + 2}, x < 0,9 \end{cases}$$

Бағдарламаға қажетті компоненттер: Edit1, Edit2, Button1, Button2

```
procedure TForm1.Button1Click(Sender: TObject);
var x, y: real;
begin
x:=str tofloat (Edit1 text);
if x >=0.9 then y: = sin (cos(x)) + sqrt (exp(3*ln(x))+4)
else y: = cos (sin(x)) + exp (1/3*ln(abs(x))+2);
Edit2, text: = Floattostr (y);
Memo1. lines.Add('y = ' + Floattostr(y));
end;
```

5-есеп. Қысқаша көбейту формуласына арналған анықтамалық бағдарлама.

Қысқаша көбейту формуласының жинақты түрде жазылуын RadioButton компонентінің Checked қасиетіне жазып, егер RadioButton1 ауыстырып қосқыш шертілген болса, яғни шарт ақиқат болса, онда қысқаша көбейту формуласының ашып жазылуын Memo1 компонентінің Lines қасиетінің мәні ретінде экраннан көрсету керек.

Формаға компоненттер палитрасының Standard бетінде орналасқан RadioButton, Memo және Additional бетіндегі BitBtn компонентін орналастырып сәйкес мәндерін меншіктейміз. Есеп шешімін BitBtn1 компонентінің OnClick оқиғасында жүзеге асырамыз.

Форма мен оның компоненттерінің қасиеттерін төмендегідей орнату қажет.

Компоненттер	Қасиеттері	Меншіктелетін мән
Form1	Caption	Қысқаша көбейту формуласы
	Name	Form1
RadioButton1	Caption	(a+b)^2
RadioButton2	Caption	(a-b)^2
RadioButton3	Caption	a^2-b^2
RadioButton4	Caption	(a+b)^3
BitBtn1	Kind	bkOK
BitBtn2	Kind	bkClose
Memo1	Lines	тазалау



батырмасын екі рет шертіп бағдарлама коды терезесіне көшіңіз, OnClick оқиғасы процедураның дайындамасы көрінеді. Төменде BinBtn1 компонентінің OnClick оқиғасының коды келтірілген.

```
unit Unit1;
interface
uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, Buttons;
type
  TForm1 = class(TForm)
    Memo1: TMemo;
    RadioButton1: TRadioButton;
    RadioButton2: TRadioButton;
    RadioButton3: TRadioButton;
    RadioButton4: TRadioButton;
    BitBtn1: TBitBtn;
    BitBtn2: TBitBtn;
    procedure BitBtn1Click(Sender: TObject);
    procedure BitBtn2Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;
var
  Form1: TForm1;
implementation
{$R *.dfm}

procedure TForm1.BitBtn1Click(Sender: TObject);
begin
  if RadioButton1.Checked then
  begin
    Memo1.Lines.Clear;
    Memo1.Lines.Add('a^2+2*a*b+b^2');
  end;
  if RadioButton2.Checked then
  begin
    Memo1.Lines.Clear;
    Memo1.Lines.Add('a^2-2*a*b+b^2');
  end;
  if RadioButton3.Checked then
  begin
```

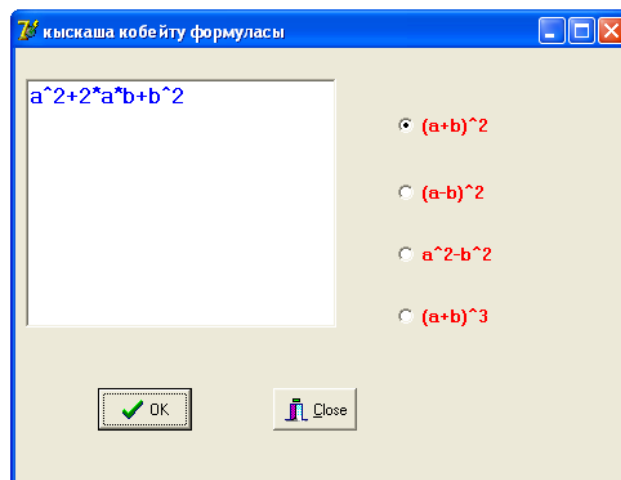
```

Memo1.Lines.Clear;
Memo1.Lines.Add('(a+b)(a-b)');
end;
if RadioButton4.Checked then
begin
Memo1.Lines.Clear;
Memo1.Lines.Add('a^3+3*a^2*b+3*a*b^2+b^3');
end;
end;

procedure TForm1.BitBtn2Click(Sender: TObject);
begin
close
end;

end.

```



6-есеп. Телефонмен сөйлесу құнын есептейтін бағдарлама.

Бағдарламаға қажетті компоненттер: Edit1; Edit2; Button1; Label1; Label2; Label3;

Есептеу батырмасына шерткенде бағдарлама есептеуді жүзеге асырады.

```

procedure TForm1.Button1Click(Sender: TObject);
const
    P = 10; // бір минут сөйлесу құны 10 тенге
    B = 0.2; // 20 пайыз жеңілдік
var
    T:Real; // сөйлесу ұзақтығы
    A:integer; // аптаның күні
    S:real; // сөйлесу құны
begin
    T:=StrToFloat(Edit1.Text);
    A:=StrToInt(Edit2.Text);

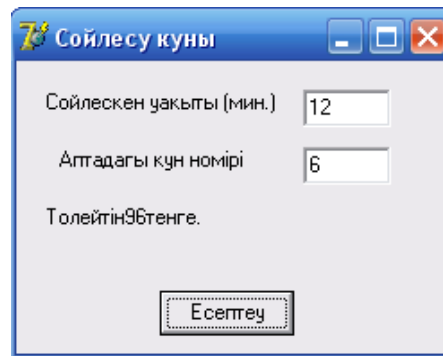
    // сөйлесу құнын есептеу

```

```

S:= P*T;
// Егер күн сенбі және жексенбі болса, онда сөйлесу құны 20 пайыз
жеңілдікпен
if (A = 6) OR (A = 7)
then S:=S*(1 -B);
// есептеу нәтижесін шығару
label3.caption:='Төлейтін'+FloatToStr(S)+'тенге.';
end;

```



7-есеп. Бағдарламалау кезінде көбінесе екіден көп тармақтан таңдауды жүзеге асыруға тура келеді. Келесі бағдарламада сіздің салмағыңызды және бойыңыздың ұзындығын енгізуді сұрайды. Салмақтың оптималды мәнін есептейді және оны нақты салмақпен салыстырады да сәйкес жауапты экранға шығарады. Салмақтың тиімді мәні былай есептелетіндігі бізге белгілі:

Бой ұзындығы (см) - 100

Бағдарламаға қажетті компоненттер: Edit1, Edit2, Label1, Label2, Label3, Button1.

```

unit Unit1;
interface
uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
  StdCtrls;
type
  TForm1 = class(TForm)
    Label1: TLabel;
    Label2: TLabel;
    Edit1: TEdit;
    Edit2: TEdit;
    Button1: TButton;
    Label3: TLabel;
  procedure Button1Click(Sender: TObject);
  private
    { Private declarations }
  public

```

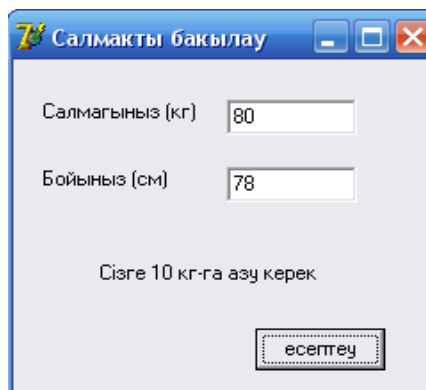
```

    { Public declarations }
end;

var
  Form1: TForm1;
implementation
{$R *.DFM}

procedure TForm1.Button1Click(Sender: TObject);
var
  w: real; {салмақ}
  h: real; {ұзындық}
  opt: real; {оптимальды салмақ}
  d: real; { оптимальды салмақтан ауытқу }
begin
  w:=StrToFloat(Edit1.text);
  h:=StrToInt(Edit2.Text);
  opt:=h-100;
  if w=opt
  then
    Label3.caption:= 'Сіз жақсы формадасыз'
  else
    if w<opt
    then
      begin
        d:=opt-w;
        Label3.caption:= 'Сізге ' + FloatToStr(d) + ' кг - ға толу керек ' ;
      end
    else
      begin
        d:=w-opt;
        Label3.caption:= 'Сізге ' + FloatToStr(d) + ' кг - ға азу керек ' ;
      end;
    end;
end;
end.

```



8-есеп.. Функция мәнін есептейтін бағдарлама.

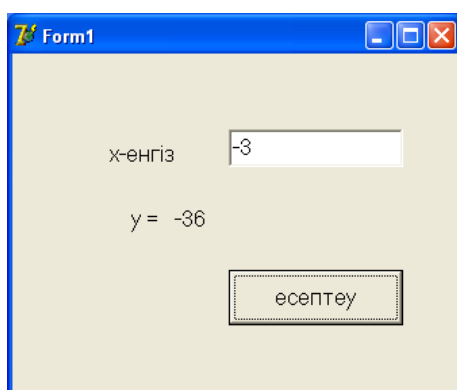
$$y = \begin{cases} -4x^2 + 8, & x < 0 \\ \sqrt{x^2 + 3x + 4}, & x > 1 \\ (x^2 + 3x + 4)^2, & 0 \leq x \leq 1 \end{cases}$$

1-әдіс.

Формада көрсетілгендей компоненттерді орналастырып, әр компоненттердің қасиеттерінің мәндерін қойып шығу.

Процедура бағдарламасын жазу үшін Есептеу батырмасына екі рет басу, сонда бағдарлама коды терезесі шығады. Келесі процедура денесін енгізу.

```
procedure TForm1.Button1Click(Sender: TObject);
var x,y:real;
begin
  x:=strtofloat(Edit1.text);
  if x<0 then y:=-4*x*x
    else if x> 1 then y:=sqrt(x*x+3*x+4)
      else y:= sqrt(x*x+3*x+4);
  label3.Caption:=floattostr(y);
end;
```



2-әдіс.

Бұл әдісте InputBox енгізу терезелерін және ShowMessage шығару терезесін пайдаланамыз.

Жаңа формаға екі рет шерту, бағдарлама коды терезесі шығады. Бағдарлама кодын жазамыз.

```
procedure TForm1.FormCreate(Sender: TObject);
var x,y:real; x1:string;
begin
  x1:=inputbox('x-енгіз ','x=','');
  x:=strtofloat(x1);
  if x<0 then y:=-4*x*x
    else if x> 1 then y:=sqrt(x*x+3*x+4)
```

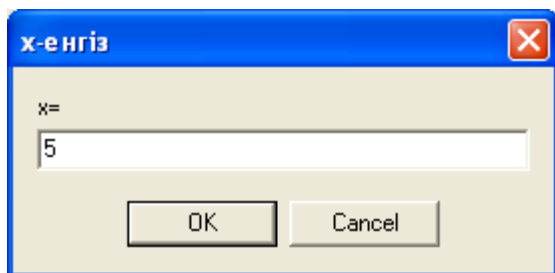
```

else y:= sqr(x*x+3*x+4);
ShowMessage('y='+floattostr(y));
end;

```

Бағдарламаны орындау. Сонда келесі InputBox терезесі экранға шығады. Шыққан терезеде x-тің мәнін енгізіп, ОК батырмасын басу керек.

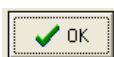
Нәтиже ShowMessage терезесіне шығады.



9-есеп. RadioButton компонентін пайдаланып, физикалық формулалар жөнінде анықтамалық бағдарлама.

Форма мен оның компоненттерінің қасиеттерін төмендегідей орнату қажет.

Компоненттер	Қасиеттер	Меншіктелетін мән
Form1	Caption	Анықтама_физикалық формулалар
	Name	Form1
	Color	clYellow
Memo1	Lines	Тазалау
	Font	TimesNewRoman, қарайтылған, өлшемі – 22, көк
RadioButton1	Caption	Ом заңы
RadioButton2	Caption	Идеал газ күйі
RadioButton3	Caption	Ампер күші
RadioButton4	Caption	Лоренц күші
RadioButton5	Caption	Магнит индукциясы
RadioButton 1-5	Font	TimesNewRoman, қарайтылған, өлшемі – 8, қызыл
BitBtn1	Kind	bkOK
BitBtn2	Kind	bkClose



батырмасын екі рет шертіп бағдарлама коды терезесіне көшіңіз. Төменде BitBtn1 компонентінің OnClick оқиғасының коды келтірілген.

```

unit Unit1;
interface
uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, Buttons;
type
  TForm1 = class(TForm)
    RadioButton1: TRadioButton;
    RadioButton2: TRadioButton;
    RadioButton3: TRadioButton;
    RadioButton4: TRadioButton;
    BitBtn1: TBitBtn;
    BitBtn2: TBitBtn;
    RadioButton5: TRadioButton;
    Memo1: TMemo;
    procedure BitBtn1Click(Sender: TObject);
    procedure BitBtn2Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;
var
  Form1: TForm1;
implementation
{$R *.dfm}

```

```

procedure TForm1.BitBtn1Click(Sender: TObject);
begin
  If radioButton1.Checked then
  Begin Memo1.Lines.Clear;
  Memo1.Lines.Add('I=U/R;');
  end;
  If radioButton2.Checked then
  Begin Memo1.Lines.Clear;
  Memo1.Lines.Add('PV=m*R*T/M;');
  end;
  If radioButton3.Checked then
  Begin Memo1.Lines.Clear;
  Memo1.Lines.Add('F=I*B*I*sina;');
  end;
  If radioButton4.Checked then
  Begin Memo1.Lines.Clear;
  Memo1.Lines.Add('F=q*V*B*sina;');
  end;

```

```

If radioButton5.Checked then
Begin Memo1.Lines.Clear;
Memo1.Lines.Add('B=M/I*S;');
end;
end;

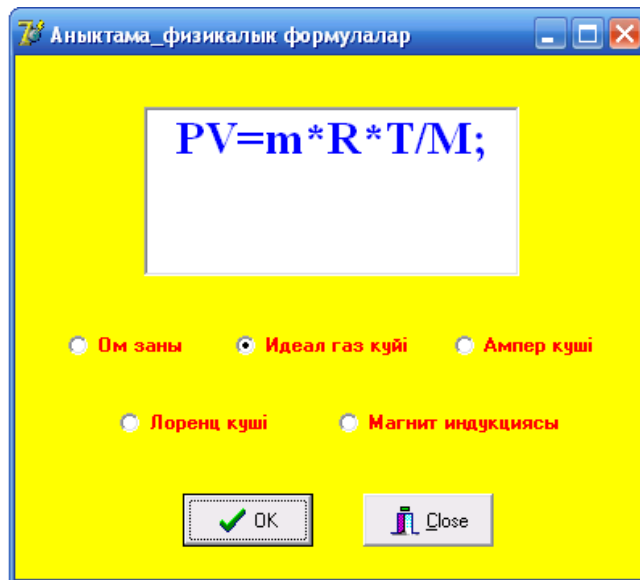
```

```

procedure TForm1.BitBtn2Click(Sender: TObject);
begin
close;
end;

end.

```



10-есеп. Екі санды оқитын, оларды салыстыратын және олардың қайсысы ең кіші екендігі туралы хабар шығаратын, ал егер сандар тең болса, ол туралы да хабар шығаратын бағдарлама.

```

unit Unit1;
interface
uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls;
type
  TForm1 = class(TForm)
    Label1: TLabel;
    Label2: TLabel;
    Button1: TButton;
    Edit1: TEdit;
    Edit2: TEdit;
    procedure Button1Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;
end;

```

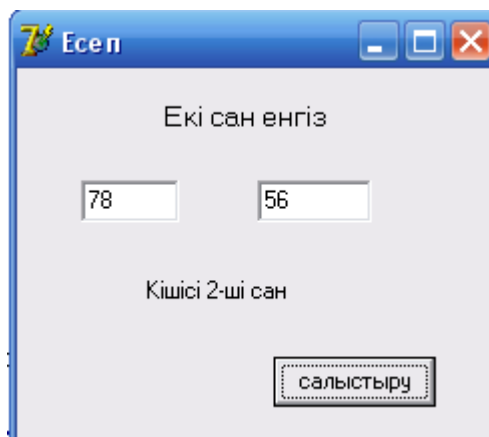
```

var
  Form1: TForm1;
implementation
{$R *.dfm}

procedure TForm1.Button1Click(Sender: TObject);
var first_number, second_number: Real;
begin
  first_number:=StrToFloat(Edit1.Text );
  second_number:=StrToFloat(Edit2.Text );
  if first_number < second_number then
    label2.Caption :='Кішісі 1-ші сан'
  else if first_number = second_number then
    label2.Caption :='Екі сан бір-бірімен тең'
  else label2.Caption :='Кішісі 2-ші сан';

end;
end.

```



11-есеп. Қарапайым калькулятордың іс-әрекетін орындайтын бағдарлама.

Бағдарламаға қажетті компоненттер: Label1, Label2, Label3, Label4, Label5, Edit1, Edit2, Edit3, Button1, Button2, Button3, Button4, Button5, Button6.

```

procedure TForm1.FormCreate(Sender: TObject);
begin
  Edit1.Text := "";
  Edit2.Text := "";
  Edit3.Text := "";
end;

procedure TForm1.Button5Click(Sender: TObject);
begin
  Edit1.Text := "";
  Edit2.Text := "";
  Edit3.Text := "";

```

```

end;

procedure TForm1.Button6Click(Sender: TObject);
begin
    close;
end;

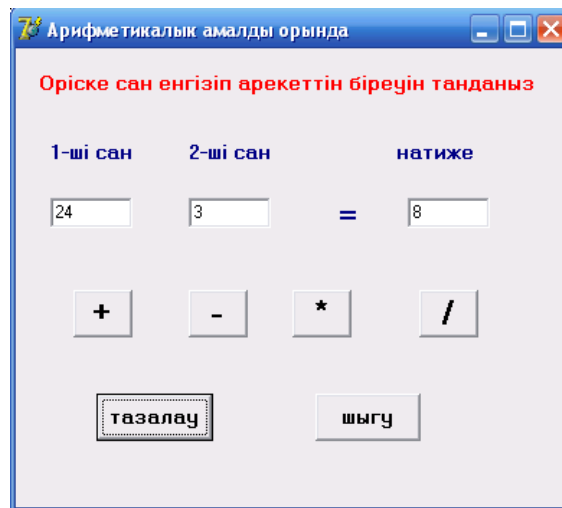
procedure TForm1.Button1Click(Sender: TObject);
begin
    a := StrToFloat (Edit1.Text);
    b := StrToFloat (Edit2.Text);
    c := a + b;
    Edit3.Text := FloatToStr (c);
end;

procedure TForm1.Button2Click(Sender: TObject);
begin
    a := StrToFloat (Edit1.Text);
    b := StrToFloat (Edit2.Text);
    c := a - b;
    Edit3.Text := FloatToStr (c);
end;

procedure TForm1.Button3Click(Sender: TObject);
begin
    a := StrToFloat (Edit1.Text);
    b := StrToFloat (Edit2.Text);
    c := a * b;
    Edit3.Text := FloatToStr (c);
end;

procedure TForm1.Button4Click(Sender: TObject);
begin
    a := StrToFloat (Edit1.Text);
    b := StrToFloat (Edit2.Text);
    if b = 0 then Edit3.Text := 'division by zero'
    else
        begin
            c := a / b;
            Edit3.Text := FloatToStr (c);
        end;
end;
end;

```



12-есеп. Енгізу өрісіне енгізілген санның факториалын есептейтін бағдарлама.

```

unit factor;
interface
uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
  StdCtrls;
type
  TForm1 = class(TForm)
    Label1: TLabel;
    Edit1: TEdit;
    Button1: TButton;
    Label2: TLabel;
    procedure Button1Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;
var
  Form1: TForm1;
implementation
{$R *.DFM}

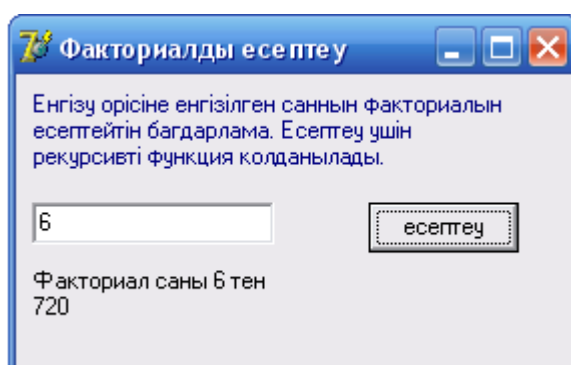
// рекурсивті функция
function factorial(n: integer): integer;
begin
  if n > 1
  then factorial := n * factorial(n-1)
  else factorial:= 1;
end;

```

```

procedure TForm1.Button1Click(Sender: TObject);
var
  k:integer;
  f:integer;
begin
  k := StrToInt(Edit1.Text);
  f := factorial(k);
  label2.caption:='Факториал саны '+Edit1.Text
    +' тен '+IntToStr(f);
end;
end.

```



Тапсырмалар:

Берілген нақты x аргументі үшін y функциясының мәнін есептеуге арналған бағдарлама жазындар.

$$1. y = \begin{cases} -(x^2 \sqrt{x+1}), & x > 0 \\ \sin(x^2 + 1), & x \leq 1 \end{cases}$$

$$2. y = \begin{cases} \sin x^2 + x, & x \leq 1 \\ \frac{5}{x} + 2x + \cos(x+3), & x > 1 \end{cases}$$

$$3. y = \begin{cases} -4x^2 + 8, & x < 0 \\ \sqrt{x^2 + 3x + 4}, & x > 1 \\ (x^2 + 3x + 4)^2, & 0 \leq x \leq 1 \end{cases}$$

$$4. y = \begin{cases} \sqrt{x} + \sin x, & x > 0,4 \\ \sqrt{2-x} + \cos x, & -1 < x < 0,4 \\ e^{5x-1} + \lg x, & x \leq -1 \end{cases}$$

Бақылау сұрақтары:

1. Шартты оператордың қызметі және жазылу форматы қандай?
2. Шартты оператор қалай орындалады?
3. IntToStr функциясының қызметі не?
4. StrToFloat функциясының қызметі не?
5. FloatToStr функциясының қызметі не?
6. InputBox функциясының қызметі не?
7. ShowMessage (s) процедурасының қызметі не?

Зертханалық жұмыс №4. Таңдау командасы

Жұмыстың мақсаты: Delphi ортасында таңдау командасына бағдарлама құрып, есептер шығарып үйрену дағдыларын қалыптастыру және өзіндік тапсырмаларды орындау.

Бағдарламалық дидактикалық қамтылуы: Дербес компьютерлер, Windows XP операциялық жүйесі, Borland Delphi 7 бағдарламасы, «Бағдарламалау» пәнінен оқу-әдістемелік кешен.

Берілген өрнектің мәніне тәуелді бір немесе бірнеше оператор орындалатын жағдайда таңдау операторы пайдалынады.

Таңдау жүргізілетін параметр ретіндегі өрнек үшін кез келген реттелген тип алынады (реттелген типке – integer, char, логикалық типтер жатады).

Таңдау операторының құрылымы төмендегідей:

```
Case <өрнек> of  
<таңдау тізімі> [else <операторлар >]  
end;
```

Мұндағы case, of, else, end – қызметші сөздер (жағдай, одан, әйтпесе, соңы); <таңдау тізімі> - бір немесе бірнеше төмендегідей құрылым:

<таңдау тұрақтысы>: <оператор>;

<оператор> - Object Pascal тілінің кез келген операторы. Таңдау операторының орындалуы төмендегідей:

Алдымен өрнек есептеледі, одан кейін таңдау тізімінен есептелген өрнектің мәніне тең тұрақты ізделінеді. Табылған тұрақтыға сәйкес оператор орындалып, таңдау командасы өзінің жұмысын аяқтайды. Егер таңдау тізімінен есептелген өрнектің мәніне тең тұрақты табылмаса, онда бағдарламаны басқару ELSE операторынан кейін тұрған операторға береді. Кейде else <операторлар> бөлігін жазбауға да болады.

Таңдау тізіміндегі операторлардың кез келгеніне бір емес, бірнеше үтірмен жазылған таңдау тұрақтысы сәйкес келуі мүмкін.

1-есеп. Бұл мысалда қабат нөмірі енгізіледі де осы нөмірге байланысты хабар беріледі. Егер мәні нөлге тең болса шартсыз көшу операторы бойынша Case операторы жұмысын аяқтайды.

Бағдарламаға қажетті компоненттер: Edit1, Memo1, BitBtn1, Button1.

```
Procedure TForm. BitBtn1Click (Sender: TObject);  
label go_end;  
var  
i : integer;  
begin  
i := StrToInt (Edit1.Text);  
case i of  
0 : Goto go_end;  
2, 4, 6, 8 : Memo1.Lines.Add ('Сіз жұп номерлі қабатта тұрасыз');
```

```

1, 3, 5, 7, 9: Memo1.lines.Add ('Сіз тақ номерлі қабатта тұрасыз');
10..150 : Memo1.lines.Add ('Сіз зәулім, көкпен таласқан үйде тұрасыз');
else
Memo1.lines.Add('Сіз жертөледе немесе өте биік үйде тұрасыз');
go_end: end;
end;

```

2-есеп. Таңдау командасын пайдаланып берілген санға екі еселісін, квадратын, кері санын және қарама-қарсы санын анықтайтын бағдарлама.

Берілген санды Edit компоненті арқылы немесе ScrollBar-дың көмегімен енгізіп, ал орындалатын амалдар ComboBox-тің Items қасиеті арқылы таңдалып және ComboBox-тің Itemindex-тің мәніне сәйкес есептеп, нәтижені StaticText компонентінің өрісіне шығарамыз. Есеп шешімін Bitbtn1 компонентінің OnClick оқиғасында ұйымдастырамыз.

Формаға компоненттер палитрасының Standard бетінде орналасқан Edit1, Label1, ScrollBar1, ComboBox1 және Additional бетіндегі Bitbtn компоненттерін орналастырып, сәйкес мәндерін меншіктейміз.

Форма мен оның компоненттерінің қасиеттерін төмендегідей орнату қажет.

Компоненттер	Қасиеттері	Меншіктелетін мән
Form1	Caption	Еселігі, квадраты, кері саны, қарама-қарсы сан
Form1	Name	Form1
Label1	Caption	Санды енгіз
Edit1	Text	тазалау
Label2	Caption	амалды таңда
Combobox1	items	екі еселегенде квадраттағанда кері саны қарама-қарсы сан
StaticText1	Caption	тазалау
Label3	Caption	санды таңда
ScrollBar1	min	-100
ScrollBar1	max	100
Bitbtn1	Kind	BkOK
Bitbtn2	Kind	BkClose

```

unit Unit1;
interface
uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, Buttons;

```

```

type
  TForm1 = class(TForm)
    StaticText1: TStaticText;
    Edit1: TEdit;
    ComboBox1: TComboBox;
    ScrollBar1: TScrollBar;
    BitBtn1: TBitBtn;
    Label1: TLabel;
    Label2: TLabel;
    Label3: TLabel;
    BitBtn2: TBitBtn;
    procedure BitBtn1Click(Sender: TObject);
    procedure scrollBar1Scroll(Sender: TObject; ScrollCode: TScrollCode;
      var ScrollPos: Integer);
  private
    { Private declarations }
  public
    { Public declarations }
  end;
var
  Form1: TForm1;
  x,y:real;
implementation
{$R *.dfm}

procedure TForm1.BitBtn1Click(Sender: TObject);
begin
  x:=strtofloat(edit1.text);
  case combobox1.ItemIndex of
  0: begin y:=2*x;
    StaticText1.caption:=combobox1.Items[combobox1.Itemindex] +' жауабы'+ '='
+floattostr(y);
    end;
  1:begin y:= sqr(x);
    staticText1.caption:=
    combobox1.Items[combobox1.Itemindex] +' жауабы'+ '=' +floattostr(y);
    end;
  2:begin y:=1/x;
    staticText1.caption:=
    combobox1.Items[combobox1.Itemindex] +' жауабы'+ '=' +floattostr(y);
    end;
  3:begin y:=-x;
    staticText1.caption:=
    combobox1.Items[combobox1.Itemindex]
    +' жауабы'+ '=' +floattostr(y);
  end;
end;

```

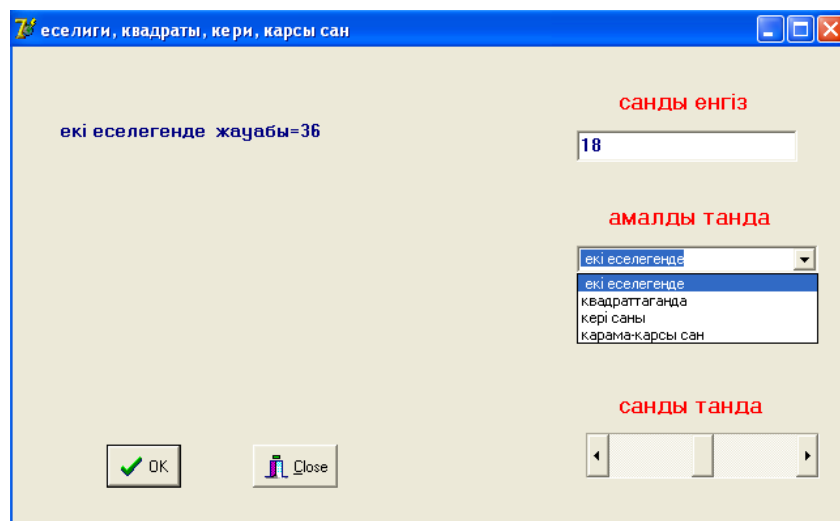
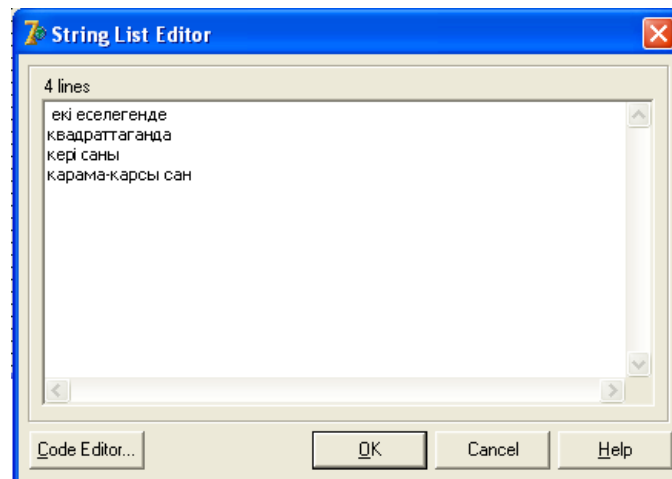
```
end;  
end;  
end;
```

```
procedure TForm1.ScrollBar1Scroll(Sender: TObject; ScrollCode: TScrollCode;  
  var ScrollPos: Integer);  
begin  
  x:=ScrollBar1.Position;  
  edit1.Text:=floattostr(x);  
  
end;
```

```
end;
```

```
procedure TForm1.BitBtn2Click(Sender: TObject);  
begin  
  close  
end;
```

```
end.
```



3-есеп. Таңдау командасын пайдаланып, арифметикалық төрт амалды орындайтын калькулятор бағдарламасы.

Форма бетіне StaticText1, ComboBox1, Edit1, Edit2, Edit3, BitBtn1, BitBtn2 компоненттерін, екі Edit, үш Label және бір Button компонентін орналастырыңыз

Компоненттерге келесі қасиеттерді меншіктеңіз:

Компонент	Қасиеті	Меншіктелетін мән
Form1	Caption	Калькулятор
	Font	Times Kaz, 11
Edit1	Text	Бос ету
Edit2	Text	Бос ету
BitBtn1	Kind	BkOk
BitBtn2	Kind	BkClose
ComboBox1	Text	Бос ету
StaticText	Caption	Бос ету
	Align	AlTop

ComboBox компонентінің Items қасиетін шертіп, ашылған String List Editor терезесіне +, -, *, / операторларының әрқайсысын жеке жолға енгізіңіз.

```

procedure TForm1.BitBtn1Click(Sender: TObject);
Var z,x,y :real;
begin
X:=StrToFloat(edit1.Text);
Y:=StrToFloat(edit2.Text);
case ComboBox1.ItemIndex of
0:begin z:=x+y;
StaticText1.Caption:=Edit1.Text+ComboBox1.Items[ComboBox1.ItemIndex]
+ "+Edit2.Text+="+floatToStrF(z,ffGeneral,5,2);
end;
1:begin z:=x-y;
StaticText1.Caption:=Edit1.Text+ComboBox1.Items[ComboBox1.ItemIndex]
+ "+Edit2.Text+="+floatToStrF(z,ffGeneral,5,2);
end;
2:begin z:=x*y;
StaticText1.Caption:=Edit1.Text+ComboBox1.Items[ComboBox1.ItemIndex]
+ "+Edit2.Text+="+floatToStrF(z,ffGeneral,5,2);
end;
3:begin z:=x/y;
StaticText1.Caption:=Edit1.Text+ComboBox1.Items[ComboBox1.ItemIndex]
+ "+Edit2.Text+="+floatToStrF(z,ffGeneral,5,2);
end;
end;
end;
end;

```

```
procedure TForm1.BitBtn2Click(Sender: TObject);
begin
Close;
end;
end.
```

Тапсырмалар:

1. Елдің атының бірінші әрпін енгізу арқылы ондағы қалалардың санын жауапқа шығарыңдар.
2. Қаланың телефондық кодын енгізу арқылы сол қаланың атын жауапқа шығарыңдар.
3. Апта күндерінің нөмірін енгізу арқылы сол күні болатын сабақтардың санын жауапқа шығарыңдар.
4. Алты түрлі тауар туралы мәлімет бар. Оның біреуінің кодын енгізу арқылы сол заттың бағасын жауапқа шығарыңдар.
5. 0-ден 7-ге дейінгі сандардың біреуін енгізу арқылы олардың ағылшын тілінде аталуын жауапқа шығарыңдар.
10. 1-ден 10-ға дейінгі сандарды енгізу арқылы олардың сөзбен жазылуын жауапқа шығарыңдар.

Бақылау сұрақтары:

1. Таңдау операторының жазылу форматы және атқаратын қызметі қандай?

Зертханалық жұмыс №5. Қайталанатын алгоритмдерге бағдарлама құру (For қайталану операторы)

Жұмыстың мақсаты: Delphi ортасында параметрдің мәні бойынша қайталануды ұйымдастыратын алгоритмдерге бағдарлама құру дағдыларын қалыптастыру және өзіндік тапсырмаларды орындау.

Бағдарламалық дидактикалық қамтылуы: Дербес компьютерлер, Windows XP операциялық жүйесі, Borland Delphi 7 бағдарламасы, «Бағдарламалау» пәнінен оқу-әдістемелік кешен.

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

1. Параметрдің мәні бойынша қайталануды ұйымдастыру.
2. Алдын ала берілген шарт бойынша қайталануды ұйымдастыру;
3. Келесі шарт бойынша қайталануды ұйымдастыру;

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

Параметрдің мәні бойынша қайталануды 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 - орында 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 $m1 > m2$ 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.

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.

1-есеп. 1-ден 50-ге дейінгі және 50-ден 1-ге дейінгі сандарды FOR операторының көмегімен жауапқа шығаратын бағдарлама.

```
procedure TForm1.Button1Click(Sender: TObject);  
var b, a, i: integer;  
begin  
  for i:=1 to 50 do
```

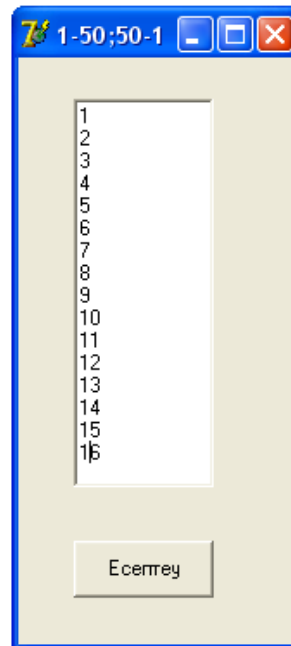
```

begin
a:=i;
Memo1.lines.Add(Inttostr(a));
end;
for i:=50 downto 1 do
begin
b:=i;
Memo1.lines.Add(Inttostr(b));
end;

end;

end.

```



2-есеп. Бір жетінің ішінде жауған жаңбырдың ылғалдылығының мөлшерінің ең көбін және нешінші күні жауғанын параметрлі циклдің көмегімен есептейтін бағдарлама.

Жауған жаңбырдың ылғалдылығының мөлшерін InputBox функциясының көмегімен енгізіп оны St-ға меншіктеп және оларды Memo-ға шығар, ылғалдылығының мөлшерінің ең көбін және нешінші күні жауғанын шарт операторының және параметрлі циклдің көмегімен тауып нәтижесін тағы Memo-ға шығарылады.

Есеп шешімін Bitbtn1 компонентінің OnClick оқиғасында ұйымдастырамыз.

Форма мен оның компоненттерінің қасиеттерін төмендегідей орнату қажет.

Компонент	Қасиеті	Меншіктелетін мән
Form1	Caption	жауған жаңбыр мөлшері
Form1	Name	Form1
Bitbtn1	Caption	есепте
Bitbtn1	Kind	bkOK
Bitbtn2	Kind	bkClose
Memo1	Lines	тазалау

```

procedure TForm1.BitBtn1Click(Sender: TObject);
var a,max:real; st:string;
    i,k:integer;
begin
max:=0;
for i:=1 to 7 do begin

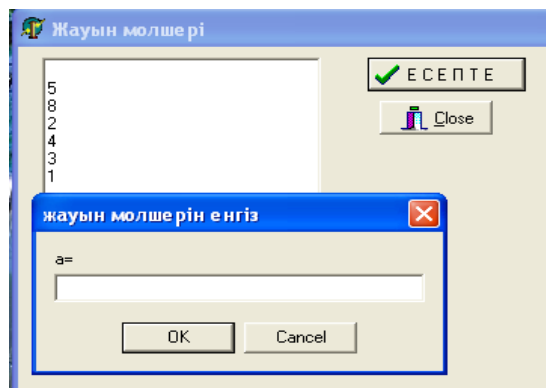
```



```

st:=inputbox('жауын молшерін енгіз','a=',');
Memo1.Lines.Add(st) ;
a:=strtoint(st);
if max<a then begin max:=a;k:=i;
end; end;
Memo1.Lines.Add('Ен мол ылгалдылык '+
inttostr(k)+'- күні '+floattostr(max)+' молшерде жауган' );
end;

```



3-есеп. $S=N!$ факториалды есептеу.

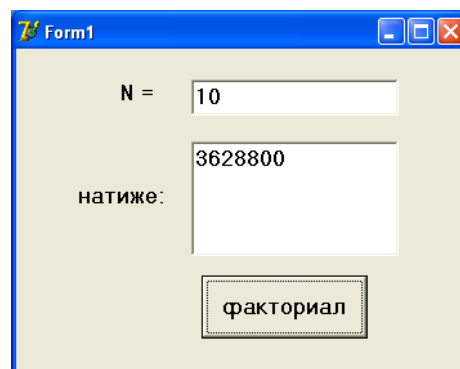
Формаға Label1, Label2, Edit1, Memo1, Button1 компоненттерін орналастырып, қасиеттерінің мәндерін қою.

Факториал батырмасын шертіп, процедура денесін жазу.

```

procedure TForm1.Button1Click(Sender: TObject);
var n:string;
    fac,i:integer;
begin
    fac:=1;
    { N! факториалды есептеу }
    n:=Edit1.text;
    for i:=1 to strtoint(n) do
        fac:=fac*(i);
    memo1.lines.add(inttostr(fac));
end;

```



4-есеп. 1..n сандары берілген. Осы сандардың ішінен жұп және тақ сандардың қосындысын табатын бағдарлама.

Формаға Label1, Edit1, Memo1, Button1 компоненттерін орналастырып, компоненттердің қасиеттерінің мәндерін беру.

Есептеу батырмасын шертіп, процедура денесін жазу.

```

procedure TForm1.Button1Click(Sender: TObject);
var
    n,i:integer;
    sum1,sum2:integer;
    n1:string;
begin
    n1:=Edit1.text;
    n:=strtoint(n1);
    sum1:=0; sum2:=0;
    for i:=1 to n do
    if i mod 2 = 0 then sum1:=sum1+i
        else sum2:=sum2+i;
        memo1.lines.add('jup='+inttostr(sum1));
        memo1.lines.add('tak='+inttostr(sum2));
    end;
end;

```

Бағдарламаны орындау, x-қа мән енгізу (4). Есептеу батырмасын басу, нәтиже төмендегідей формаға шығады.



5-есеп. $S = 1 + 1/2 + 1/3 + \dots + 1/10$ қатардың қосындысын есептейтін бағдарлама

Формаға Label1, Button1 компоненттерін орналастыру.
Формадағы Қосынды батырмасын басып, процедура денесін жазу.

```

procedure TForm1.Button1Click(Sender: TObject);
var
    i:integer;
    elem:real;
    sum:real;
begin
    sum:=0;
    label1.Caption:="";
    for i:=1 to 10 do
    begin
        elem:=1/i;

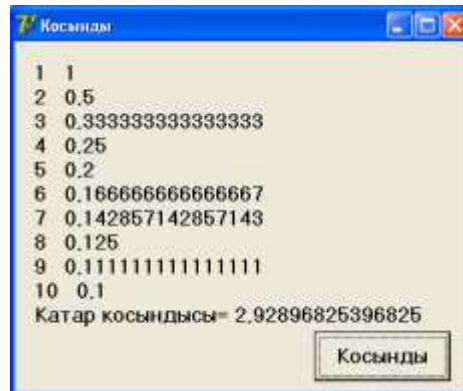
```

```

label1.Caption:=label1.Caption+inttostr(i)+' ' +floattostr(elem)+#13;
sum:=sum+elem;
end;
label1.Caption:=label1.Caption+
    'Қатар қосындысы= '+floattostr(sum);
end;

```

Бағдарламаны орындау, Қосынды батырмасына шерту, сонда қатардың қосындысы label1 өрісіне шығады.



6-есеп. $y = 5x^2 - 7$ функциясының $-1, -0.5, 0, 0.5, 1$ нүктелеріндегі мәнін есептейтін бағдарлама.

```

Procedure TForm1. Button1Click (Sender: TObject);
    var
        y: real; // функция мәні
        x: real; // функция аргументі
        dx: real; // аргумент өсімшесі
        st: string; // кестені бейнелеу
    i: integer; // цикл параметрі
    begin
        st: = '';
        x: = -1;
        dx: = 0.5;
        for i: = 1 to 5 do
            begin
                y: = 5*x*x -7;
                st: = st + FloatToStr(x) + ' ' + FloatToStr(y) + chr (13);
                x: = x + dx;
            end;
        Label1. Caption: = st;
    end;
end;

```

7-есеп. Мына қатардың алғашқы 10 мүшесінің қосындысын есептейтін бағдарлама.

$$1 + \frac{1}{2} + \frac{1}{3} + \dots$$

Procedure TForm1.Button1Click (Sender: TObject);

var

i: integer; { қатар элементінің номері}

elem: real; { қатар элементінің мәні}

summ: real; { қатар элементтерінің қосындысы}

begin

summ: = 0;

label1.Caption: = ' ' ;

for i: = 1 to 10 do

begin

elem: = 1/ i;

label1.Caption: = label1.Caption + IntToStr(i) + ' ' + FloatToStr(elem) + # 13;

Summ: = summ + elem;

end;

label1.Caption: = label1.Caption + ' Қатар қосындысы:' + FloatToStr(Summ);

end;

Тапсырмалар:

1. Берілген өрнектің мәнін табуға бағдарлама құрыңдар

$$S = \frac{4}{3} + \frac{5}{5} + \frac{6}{7} + \frac{6}{9} + \frac{8}{11} + \frac{9}{13} + \frac{10}{15}$$

$$2. Y = \frac{1}{2-3} + \frac{2}{3-4} + \dots + \frac{n}{(n+1)-(n+2)}$$

3. Мына қатардың алғашқы N мүшесінің қосындысын табыңдар:

$$y = 1 + \frac{x}{2} + \frac{x^2}{3} + \frac{x^3}{4} + \dots \text{ мұндағы } X \text{ кез-келген сан.}$$

Бақылау сұрақтары:

1. Object Pascal тілінде пайдаланылатын қайталану операторларының түрлері қандай және айырмашылығы неде?

2. Параметрлі қайталану операторының жазылу форматы қандай және қалай орындалады?

Зертханалық жұмыс №6. Қайталанатын алгоритмдерге бағдарлама құру (WHILE, REPEAT қайталану операторлары)

Жұмыстың мақсаты: Delphi ортасында алдын ала берілген шарт бойынша қайталану, келесі шарт бойынша қайталану операторларына бағдарлама құру дағдыларын қалыптастыру және өзіндік тапсырмаларды орындау.

Бағдарламалық дидактикалық қамтылуы: Дербес компьютерлер, Windows XP операциялық жүйесі, Borland Delphi 7 бағдарламасы, «Бағдарламалау» пәнінен оқу-әдістемелік кешен.

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 - орында 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 орындалмайды. Басқаша айтқанда, логикалық өрнектің мәні қашан жалған болғанша, қайталаудың денесі қайталап орындала береді.

Келесі шарт бойынша қайталану операторы (REPEAT)

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

```
REPEAT  
  <қайталану денесі>  
UNTIL <логикалық өрнек>
```

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

Орындалуы: < қайталану денесіндегі> операторлар кем дегенде бір рет орындалады, егер <шарт> жалған болса, яғни логикалық өрнек мәні False болса, онда Repeat, Until операторы жұмысын тоқтатып, басқаруды келесі операторға береді, кері жағдайда шарт орындалудан қалғанша <қайталану денесі> қайталана береді.

< Repeat - Until> Begin ... end операторлық жақшасына ұқсас, сондықтан Until сөзінің алдына нүктелі үтір (;) қою міндет емес.

Циклдан дұрыс шығуды ұйымдастыру үшін цикл шарты цикл денесін құрайтын операторлардың ішінде өзгеріп отыруы тиіс.

For, while, repeat қайталану операторларын басқаруды жеңілдету үшін Object Pascal тілінің құрамына параметрсіз екі процедура енгізілген:

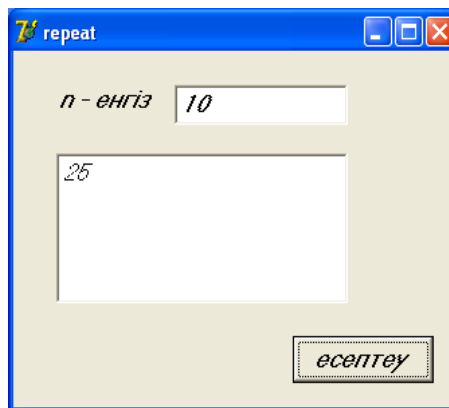
Break – циклдан тікелей шығуды жүзеге асырады; процедураның қызметін басқаруды циклдық оператордан кейінгі операторға беру болып табылады.

Continue – циклдың кезекті қайталануын алдын ала аяқтауды қамтамасыз етеді; басқаруды цикл операторының соңына берумен бара-бар.

Бұл процедуралар шартсыз көшіру GoTo операторын пайдалану қажеттілігінен босатады.

1-есеп. 1-ден 10-ға дейінгі бүтін сандардың тек жұп сандарының қосындысын Repeat операторының көмегімен есептейтін бағдарлама.

```
procedure TForm1.Button1Click(Sender: TObject);
var
    n,i:integer;
    sum1,sum2:integer;
    n1:string;
begin
    n1:=Edit1.text;
    n:=strtoint(n1);
    sum1:=0; i:=1;
    repeat
        sum1:=sum1+i;
        i:=i+2;
    until i>n;
    memo1.lines.add(inttostr(sum1));
end;
```



2-есеп. Келесі процедура Repeat операторының көмегімен 2-ден 50-ге дейінгі жұп сандарды жауапқа шығарады. Цикл денесінде цикл айнымалысының мәнін өзгертетін $i:=i+2$ операторы (командасы) орналасқан. Циклдің аяқталу шарты: $i > 50$.

Бағдарламаға қажетті компоненттер: Memo1, BitBtn1, Button1.

```

Procedure TForm1.BitBtn1Click (Sender: TObject);
var
a, i: integer;
begin
i := 2;
repeat
a := i;
Memo1.Lines.Add(IntToStr(a));
i := i + 2;
until i > 50
end;

```

Бағдарламадан шығу батырмасына (Button1) бағдарлама жазуды ұмытпаңдар (Form1.Close).

3-есеп. Мына қатардың берілген $e \leq |x|$ санынан артық $|y|$ мүшелерінің қосындысын есептейтін бағдарлама.

$$Y = 1 + \frac{x^2}{2!} - \frac{x^4}{3!} + \frac{x^6}{4!} - \dots, \quad \text{мұндағы } |x| \leq 1$$

```

Procedure TForm1.Button1Click (Sender: TObject);
var
x, y, z, e: real;
i: integer;
begin
{ қатар айнымалысы x – ті енгізу }
x := StrToFloat (Edit1.Text);
{ e -есептеу дәлдігін енгізу }
e := StrToFloat (Edit2.Text);
y := 1; z := 1; i := 2;
repeat
z := (z * x * x) / i;
if i mod 2 = 0 then y := y + z
else y := y - z;
i := i + 1;
until abs (z) <= e;
Edit3.Text := floattostr(y); // қатар қосындысы
end;

```

4-есеп. Геометриялық прогрессияның бірінші мүшесі және еселігі белгілі, алғашқы n мүшесінің қосындысын аяқталу шартымен берілген циклдің көмегімен есепте.

Бірінші мүшесін (p), еселігін (q) және мүшелер саны (n)-ді TEdit компоненттері арқылы оқып, ал нәтижесін TМемо-ның көмегімен шығарамыз. Жоғарыдағы R - шарты i-параметрі арқылы тексеріледі, яғни i-ді бірге арттырып, ол n-нен үлкен бе деп тексеріледі. Сонымен қатар TМемо-да жазылатын мәтіннің қаріпінің стилін және түсін өзгерту үшін TGroupBox контейнеріне екі TCheckBox компонентін орналастырайық. Компонент күйін State қасиеті көрсетеді. CbUn checked – жоқ, Cb Checked - иә, cbGrayed - білмеймін. Checked қасиеті қолданушының иә/жоқ типті таңдауын көрсетеді.

Есеп шешімін Bitbtn1 компонентінің OnClick оқиғасында ұйымдастырамыз.

Форма мен оның компоненттерінің қасиеттерін төмендегідей орнату қажет.

Компоненттер	Қасиеттер	Меншіктелетін мән
Form1	Caption	геометриялық прогрессия
Form1	Name	Form1
Label1	Caption	бірінші мүшесін енгіз
Edit1	Text	тазалау
Label2	Caption	еселігін енгіз
Edit1	Text	тазалау
Label3	Caption	мүшесінің санын енгіз
Bitbtn1	Kind	bkOK
Bitbtn2	Kind	bkClose
Memo1	Lines	тазалау
GroupBox1	Caption	редакторды келтіру
CheckBox1	Caption	“курсив” стилі
CheckBox2	Caption	“қызыл” түс

```

unit Unit1;
interface
uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, Buttons;
type
  TForm1 = class(TForm)
    Edit1: TEdit;
    Edit2: TEdit;
    CheckBox1: TCheckBox;
    CheckBox2: TCheckBox;
    BitBtn1: TBitBtn;
    BitBtn2: TBitBtn;
    Label1: TLabel;
    Edit3: TEdit;
    Memo1: TMemo;
  end;

```



```

procedure BitBtn1Click(Sender: TObject);
procedure CheckBox1Click(Sender: TObject);
procedure CheckBox2Click(Sender: TObject);
procedure BitBtn2Click(Sender: TObject);
private
  { Private declarations }
public
  { Public declarations }
end;
var
  Form1: TForm1;
implementation
{$R *.dfm}

```

```

procedure TForm1.BitBtn1Click(Sender: TObject);
var p,q,i,n,s:integer;
begin
  p:=strtoint(edit1.text);
  q:=strtoint(edit2.text);
  n:=strtoint(edit3.text);
  i:=1;
  repeat
    s:=s+p;
    p:=p*q;
    i:=i+1;
  until i>n;
  memo1.Clear;
  memo1.Lines.Add('Геометриялык прогрессиянын ' + edit3.text + '
мушесінің косындысы s='+inttostr(s));
end;

```

```

procedure TForm1.CheckBox1Click(Sender: TObject);
begin
  if CheckBox1.State=cbChecked then
    memo1.Font.Style:=[fsitalic]
  else memo1.Font.Style:=[];
end;

```

```

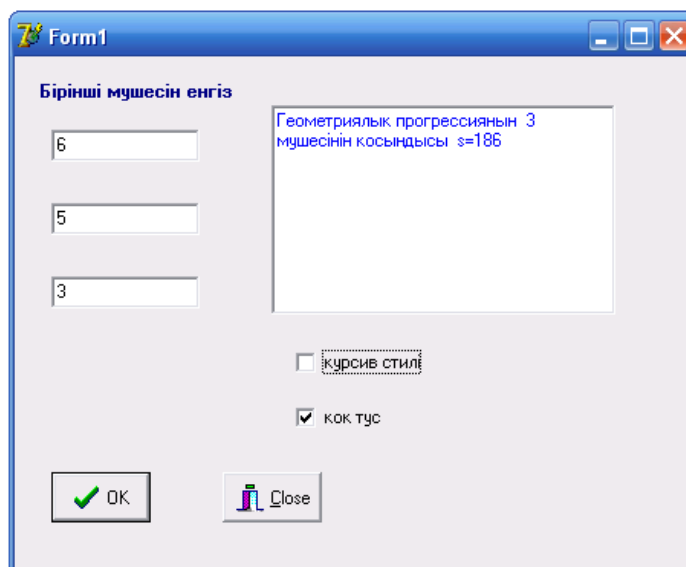
procedure TForm1.CheckBox2Click(Sender: TObject);
begin
  if CheckBox2.State=cbChecked then
    memo1.Font.Color:=clBlue
  else memo1.Font.Color:=clLime;
end;

```

```

procedure TForm1.BitBtn2Click(Sender: TObject);
begin
  close
end;
end.

```



5-есеп. π санының мәнін берілген дәлдікпен есептеңдер. Есептеу алгоритмінің негізінде мына қатардың

$$1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \frac{1}{9} + \dots$$

қосындысы $\frac{\pi}{4}$ мәніне қатардың мүшелерінің саны үлкейген сайын жақындай түседі.

Қатардың n -ші мүшесі $\frac{1}{2n-1}$ формуласымен есептелінеді және минус бірге көбейтіледі, егер n жұп болса, оның жұптығын n -ді 2-ге бөлгендегі қалдықты тексеру арқылы анықтаймыз. Қатардың келесі мүшесінің мәні берілген дәлдіктен аз болғанда есептеу тоқтайды.

```

Procedure TForm1. Button1Click (Sender: TObject);
  var
    pi: real;    // есептелінетін ПИ мәні
    t: real;     // есептеу дәлдігі
    n: integer; // қатар мүшесінің номері
    elem: real; // қатар мүшесінің мәні
begin
  pi := 0; n := 1;
  t := StrToFloat (Edit1.text);
  elem := 1; // циклді бастау үшін
while elem >= t do
  begin
    elem := 1 / (2 * n - 1);

```

```

        if n mod 2 = 0 then pi: = pi - elem;
    else pi: = pi + elem;
        n: = n + 1;
        end;
    pi: = pi * 4;
    label1.Caption: = 'ПИ саны' = 'FloatToStr(pi)+ # 13
        + ' Қатардың ' + Inttostr(n) + ' мүшесін қостық.';
end;

```

6-есеп. 1-ден n-ге дейінгі жұп сандардың квадраттарының қосындысын жалғасу шартымен берілген цикл операторының көмегімен есептейтін бағдарлама.

Берілген n-санын енгізу үшін енгізу терезесін пайдаланамыз. Енгізу терезесі Delphi-дің стандартты InputBox функциясының терезесі. Бағдарламада InputBox функциясын пайдалану командасының жазылу үлгісі: < айнымалы >:= InputBox (< тақырып >), < түсініктеме >, < мәні >)

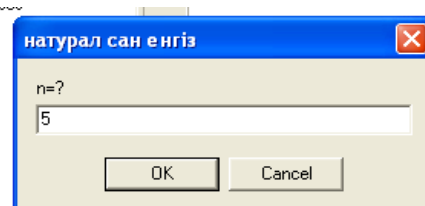
Мұндағы: *айнымалы* - мәні функция терезесіне енгізілетін жолдық типті айнымалының атауы. InputBox функциясының мәні әр кезде жолдық (String) типті. Мәні меншіктелетін st айнымалысы бағдарламада st:string түрінде сипатталуы тиіс;

тақырып - енгізу терезесінің тақырыбы ретінде жазылатын мәтін;

түсініктеме - енгізу терезесінің ішінде жазылатын түсініктеме мәтін;

мән - функция терезесі көрінген кезде оның енгізу өрісінде көрінетін мәтін. Әдетте оны бос етіп қалдыруға болады. Мысалы, бағдарламада n-санын st-ға меншіктеуді InputBox функциясын пайдаланып, мына түрде жазуға болады: st:=InputBox(' натурал сан енгіз ', 'n=?', '');

Команданың орындалу барысында көрінетін терезе суретте көрсетілген.



Ал, нәтижені бірнеше қатар жолға шығаратындай етіп, Label1-дің қасиеттерінің көмегімен экранға шығарамыз. Ол үшін Label1-дің WordWrap қасиетіне True мәнін меншіктесе болғаны. Есеп шешімін Form1-дің OnCreate оқиғасында ұйымдастырамыз.

Бағдарламаның коды:

```

procedure TForm1.FormCreate(Sender: TObject);
    var s,i,n: integer; st: string;
begin
    st:=InputBox(' натурал сан енгіз ', 'n=?', ' ');

```

```

n:=strtoint(st);
s:=0; i:=2;
while i<=n do begin
s:=s+sqr(i);i:=i+2;end;
Label1.Caption:=st+'-ге дейінгі жұп сандардың квадраттарының
қосындысы '+ ' s'+IntToStr(s);
end;
end.

```

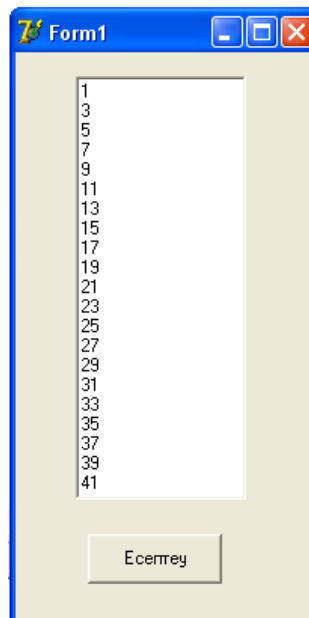
7-есеп. While операторының көмегімен 1-ден 50-ге дейінгі тақ сандарды жауапқа шығаратын бағдарлама.

Бағдарламаға қажетті компоненттер: Memo1; Button1.

```

procedure TForm1.Button1Click(Sender: TObject);
var
a, i: integer;
begin
i:=1;
while i<=50 do
begin
a:=i;
Memo1.Lines.Add(Inttostr(a));
i:=i+2;
end;
end;
end.

```



8-есеп. ListBox компонентінде берілген санды 30 шамаға арттырып отыратын бағдарлама.

Бағдарламаға қажетті компоненттер: ListBox1; Button1.

```

procedure TForm1.Button1Click(Sender: TObject);
Var i:integer;
S:real;
begin
i:=0;
while i<2000 do
begin
i:=i+100;

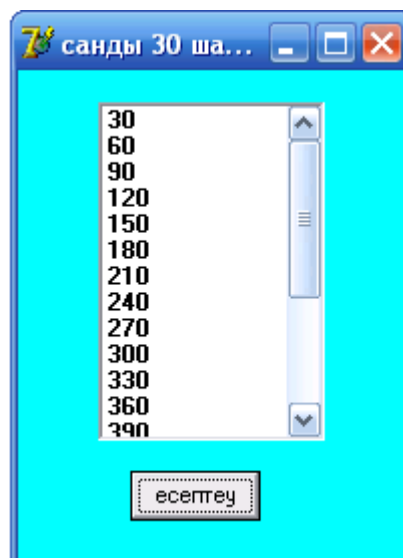
```

```

S:=i*0.3;
ListBox1.items.Add(floattostr(S));
end;
end;

end.

```



Тапсырмалар:

1.
$$S = \frac{\cos x}{1} + \frac{\cos 2x}{4} + \dots + \frac{\cos nx}{n^2}$$

2.
$$P = \sqrt{3 + \sqrt{6 + \sqrt{9 + \dots + \sqrt{3(n-1) + \sqrt{3}}}}$$

3.
$$\frac{1}{1 \cdot 2 \cdot 3} + \frac{1}{2 \cdot 3 \cdot 4} + \dots + \frac{1}{n(n+1)(n+2)}$$

4.
$$1 + \frac{1}{2^4} + \frac{1}{3^4} + \dots + \frac{1}{n^4}$$

5. Натурал сандар қатарының жұп және тақтарының көбейтіндісін есептеңдер.

6. Мына қатардың берілген $e \leq |x|$ санынан артық $|y|$ мүшелерінің қосындысын есептеңдер: $y = 1 + \frac{x^2}{2!} - \frac{x^4}{3!} + \frac{x^6}{4!} - \dots$ мұндағы $|x| \leq 1$.

Бақылау сұрақтары:

1. Алдын ала берілген шарт бойынша қайталану операторы (WHILE) қандай жағдайда пайдаланылады.

2. Әзір қайталану операторының жазылу форматы қандай және қалай орындалады?

3. Дейін қайталану командасының жазылу форматы қандай және қалай орындалады?

4. Қандай жағдайларда WHILE, FOR, REPEAT операторларын пайдаланған қолайлы?

Зертханалық жұмыс №7. Мәлімет типтері

Жұмыстың мақсаты: Object Pascal-дың мәліметтер типтерінің құрылымымен танысу және мәліметтерді саналатын тип, аралық тип, мерзім-уақыт типі ретінде сипаттап, есептерге бағдарлама құру дағдыларын қалыптастыру және өзіндік тапсырмалар орындау.

Бағдарламалық дидактикалық қамтылуы: Дербес компьютерлер, Windows XP операциялық жүйесі, Borland Delphi 7 бағдарламасы, «Бағдарламалау» пәнінен оқу-әдістемелік кешен.

Object Pascal тілінде кез келген мәліметтер өзінің типімен сипатталады. Мәліметтер типі объект қабылдай алатын мүмкін мәндер жиынтығын анықтайды. Сонымен қатар, мәліметтер типі дербес компьютердің жадындағы мәліметтердің ішкі берілу форматын да анықтайды. Object Pascal тілі мәліметтердің тармақталған құрылымымен сипатталады.

Қарапайым типке реттелген, нақты типтер және мерзім-уақыт типтері жатады. Реттелген типтер мүмкін болатын мәндерінің шектелгендігімен ажыратылады. Бұл мәндерді белгілі бір ретпен реттеуге болады, сонымен қатар мәннің реттік номерін бүтін санмен беруге болады.

Нақты типтердің де, нақты санның ішкі берілу форматымен анықталатын мәндерінің мөлшері көрсетіледі. Бірақ нақты типтердің мүмкін мәндерінің мөлшерінің көптігі соншалық, оны олардың әрқайсысын бүтін санмен белгілеу мүмкін емес.

Мерзім-уақыт типі мерзім мен уақытты белгілеуге арналған. Оны сақтау үшін нақты формат пайдаланылады.

Реттелген типке: бүтін, логикалық, символдық, саналатын және аралық тип жатады. Аталған типтердің әрқайсысына X өрнегінің мәнінің реттік нөмірін беретін $ORD(X)$ функциясын қолдануға болады.

Бүтін типтер үшін $ORD(X)$ функциясы X -тің мәнінің өзін береді. $ORD(X)$ функциясын логикалық, символдық және саналатын типтерге қолдану: логикалық тип үшін 0 мен 1 аралығындағы бүтін оң санды, символдық тип үшін: 0 мен 255 аралығындағы сандарды, ал саналатын типтер үшін: 0 мен 65535 аралығындағы сандарды береді. Аралық тип негізгі реттік типтің барлық қасиеттерін сақтайды, сондықтан оған $ORD(X)$ функциясын қолдану сол типтің қасиетіне тәуелді болады.

Реттелген типтерге төмендегідей функцияларды қолдануға болады:

$PRED(X)$ функциясы реттелген типтің алдыңғы мәнін береді, яғни $ORD(X-1)$ -дің реттік нөміріне сәйкес мән немесе $ORD(PRED(X))=ORD(X)-1$;

$SUCC(X)$ функциясы $ORD(X)+1$ -дің реттік нөміріне сәйкес келетін реттелген типтің келесі мәнін береді, яғни $ORD(SUCC(X))=ORD(X)+1$.

Бүтін типтер. Бүтін тұрақтылар - кез келген нүктесіз жазылған бүтін сан. Олар теріс немесе оң таңбалы болуы мүмкін.

1-кесте. Мәліметтердің бүтін типтерінің сипаттамалары

Тип	Мәндер аралығы	Көлемі (байт)
Byte	0...255	1
ShortInt	-128...+127	1
Smallint	-32768...+32767	2
Word	0...65535	2
Integer	-2147483648...+2147483647	4
LongInt	-2147483648...+2147483647	4
LongWord	0...4294967295	4
Int64	$-9*10^{18} \dots 9*10^{18}$	8
Cardinal	0...2147483647	4

Мысалы, тегі бүтін болатын айнымалыны көрсету үшін integer, byte және т.б. қызмет сөзі пайдаланылады. Айнымалыларды белгілеу үшін оларға атау береді.

Бейнелену түрі:

атау, атау, ... , атау: integer

атау, атау, ... , атау: byte

Нақты типтер. Мәндері бүтін типтермен салыстырылған реттелген типтерден нақты типтердің ерекшелігі нақты санның ішкі форматына тәуелді қандай да бір дәлдікпен еркін алынған санды анықтайды.

2-кесте. Мәліметтердің нақты типтері

Тип	Мәндер аралығы	Цифрлар саны	Көлемі (байт)
Real	$5,0*10^{-324} \dots 1,7*10^{308}$	15...16	6
Single	$1,5*10^{-45} \dots 3,4*10^{38}$	7...8	4
Double	$5,0*10^{-324} \dots 1,7*10^{308}$	15...16	8
Extended	$3,4*10^{-4951} \dots 1,1*10^{4932}$	19...20	10
Comp	$-2^{63} \dots +2^{63}-1$	19...20	8
Currency	922337203685477,5807	19...20	8

Кестеде көрсетілгендей, Object Pascal тілінде нақты сандар 4-тен 10 аралығындағы аралас байтқа дейін орын алады.

Арифметикалық сопроцессор барлық уақытта сандарды Extended форматында өңдейді; ал қалған нақты типтер қажетті мөлшерге дейін нәтижені

қарапайым қысқартылған күйінде беру болып табылады және олар жадыны үнемдеу үшін қолданылады.

Object Pascal тілінде Comp және Currency типтері ерекше орын алады, бөлшек бөлігінің ұзындығы шектеулі нақты сан ретінде қарастырылады:

Comp типінің бөлшек бөлігінің ұзындығы 0 (нөл) разряд, яғни болмайды, ал Currency типінің бөлшек бөлігінің ұзындығы 4 ондық разряд. Бұл аталған екі тип те 19...20 мәнді ондық цифрды сақтайтын үлкен бүтін санды анықтайды.

Қарапайым бағдарламаларда REAL типі пайдаланылады. Ал, real типі бойынша жүргізілетін есептеулерге қосымша процессордың қажеті жоқ.

Нақты сандардың бейнелену түрі:

атау, атау, ... , атау: real

Символдық тип. Символдық типтің мәні ретінде дербес компьютердегі барлық символдардың жиынтығы алынады. Әрбір символға 0..255 аралығындағы кез келген бүтін сан сәйкес келеді. Char типтегі көрсеткіштердің кодын ord функциясымен алуға болады. Символдық тұрақтылар апостроф ішіне алынып жазылады.

Тип	Мәндер аралығы	Көлемі (байт)
AnsiChar	0 - 255	2
WideChar	0 - 65535	2

Символдық айнымалылардың тегін көрсету үшін CHAR қызмет сөзі пайдаланылады. Символдық шамалар былай бейнеленеді:

атау, атау, ... , атау: char

Символдық типтің кеңейтілген түсінігі жолдық тип string болып табылады.

Char типіне қатынас операциялары, сондай-ақ, орнатылған функциялар қолданылады:

Chr(B) – Char типті функция; Byte типті өрнекті символға түрлендіреді;

Uppcase(CH) – Char типті функция; егер CH латынның кіші әрпі болса, онда бас әріпті береді, кері жағдайда CH символының өзін береді. Яғни, кіші регистрдегі әріптерді үлкен регистрдегі әріптерге ауыстырады.

Логикалық типтер. Логикалық типтерге: Boolean, ByteBool, Bool, WordBool және LongBool жатады. Стандартты Паскальда тек Boolean типі анықталған, ал қалған логикалық типтер Windows жүйесімен үйлесетін Object Pascal тіліне енгізілген.

Тип	Көлемі (байт)
Boolean	1
ByteBool	1
Bool	2
WordBool	2
LongBool	4

Логикалық тип - True (ақиқат) немесе False (жалған) мәндерінің бірін қабылдайды.

Логикалық типтегі шамалар былай бейнеленеді:

атау, атау, ... , атау: boolean

Саналатын тип. Саналатын тип өзі қабылдайтын мәндердің саналуымен беріледі. Саналатын тип Type бөлімінде сипатталады. Әрбір мән қандай да бір атаулармен аталып, тізімде дөңгелек жақшаның ішінде орналасады, мысалы,

Type

Ai attari= (kantar, akpan, nauris);

Саналатын типтердің мәндері мен олардың реттік нөмірлерінің арасындағы сәйкестік санау ретіне қарай тағайындалады. Мысалы, тізімдегі бірінші мәннің реттік нөмірі - 0, екіншісі - 1 және т.с.с.

Саналатын типтердің ең жоғарғы мәні – 65536, сондықтан саналатын тип Word бүтін типтің ішкі жиыны тәрізді және мәндері 0,1 және т.б. болатын бүтін тұрақтылар тобы ретінде қарастырылуы мүмкін. Саналатын типке бүтін санды меншіктеуге болмайды.

Кез келген саналатын типтің айнымалысын алдын-ала сипаттамай-ақ хабарлауға болады, мысалы:

Var

Col (black, white, green);

Аралық тип. Аралық тип өзінің базалық типінің ішкі жиыны болып табылады. Аралық тип барлық типтің өзінің ішкі мәндерінің шекаралары арқылы беріледі:

<ең кіші мән > ... <ең үлкен мән>

мұндағы, <ең кіші мән > - аралық типтің ең кіші мәні;

<ең үлкен мән> - оның ең үлкен мәні.

Жазылу форматы:

Type

<тип аты>=A..B;

Var

<айнымалы аты>: <тип аты>;

Object Pascal-дың стандартты кітапханасында аралық типтердің жұмысын сүйемелдейтін екі функция бар:

High(x) – x айнымалысы жататын аралық типтің ең үлкен мәнін береді;

Low(x) – аралық типтің ең кіші мәнін береді.

Күн және уақыт типі. Күн, уақыт типі TdateTime стандартты атаулармен анықталады, ол уақытты және күнді бір мезгілде сақтауға арналған.

Жадыдағы ішкі орналасуы:

Тип	Көлемі (байт)
TdateTime	8

Күнмен және уақытпен жұмыс істеу үшін төмендегідей қосалқы бағдарламалар дайындалады:

1) Function Date : TdateTime; - ағымдағы күнді береді;

2) Function DateToStr (D: TdateTime): String; - күнді символдар жолына түрлендіреді;

3) Function DateTimeToStr(D: TdateTime): String ; - күн мен уақытты символдар жолына түрлендіреді;

4) Function FormatDateTime (Format: String; Value: TdateTime): String; - күн мен уақыт Format параметрінің спецификаторына сәйкес Value параметрінен символдар жолына түрлендіреді;

5) Function Now: TdateTime; - ағымдағы күн мен уақытты береді;

6) Function Time: TdateTime; - ағымдағы уақытты береді;

7) Function TimeToStr (T: TdateTime): String; - уақытты жолға айналдырады.

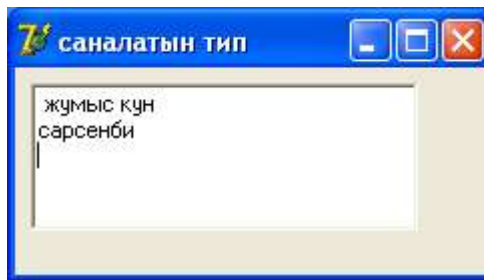
1-есеп. Апта күндерін экранға шығару. Аптаның күндерін саналатын тип ретінде сипаттап, қандай апта күні екендігін Memo1 компонентіне шығаратын бағдарлама.

```
procedure TForm1.FormCreate(Sender: TObject);
type apta=(ds,ss,sr,bs,gm,sb,gg);
var a:apta;
begin
  a:=sr;
  if (ds<=a) and (a<=gm) then memo1.lines.add(' жұмыс күн')
    else memo1.lines.add(' демалыс күн');
  case a of
    ds: memo1.lines.add('дүйсенби');
    ss: memo1.lines.add('сейсенби');
    sr: memo1.lines.add('сарсенби');
    bs: memo1.lines.add('бейсенби');
    gm: memo1.lines.add('жума');
    sb: memo1.lines.add('сенби');
```

```

gg: memo1.lines.add('жексенби');
end;
end;

```

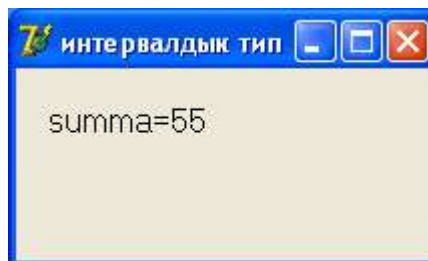


2-есеп. 1-ден 10-ға дейінгі сандардың қосындысын есептеу. 1-ден 10-ға дейінгі сандарды аралық тип ретінде сипаттап, олардың қосындысын Label1 компонентіне шығаратын бағдарлама.

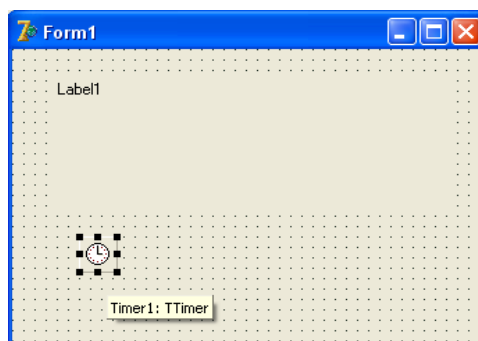
```

procedure TForm1.FormActivate(Sender: TObject);
type san=1..10;
var
  i:san;
  s:integer;
begin
  s:=0;
  i:=1;
  repeat
    s:=s+i;
    i:=i+1;
  until i>10;
  label1.Caption:='summa='+inttostr(s);
end;

```



3-есеп. Терезеде ағымдағы уақытты көрсететін “Электронды сағат”-тың бағдарламасын жазу.



Компонент	Қасиеті	Меншіктелетін мән
Form1	Caption	Электронды сағат
Label1	Caption	тазалау
Timer1	Name	Timer1

```

unit Unit1;
interface
uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, ExtCtrls;
type
  TForm1 = class(TForm)
    Timer1: TTimer;
    Label1: TLabel;
    procedure FormPaint(Sender: TObject);
    procedure Timer1Timer(Sender: TObject);
    procedure FormCreate(Sender: TObject);
    procedure ShowTime;
  private
    { Private declarations }
  public
    { Public declarations }
  end;
var
  Form1: TForm1;
implementation
{$R *.dfm}

procedure TForm1.ShowTime;
var
  Time: TDateTime;
begin
  Time:=Now();
  Label1.Caption:=FormatDateTime('hh:mm:ss',Time);
end;

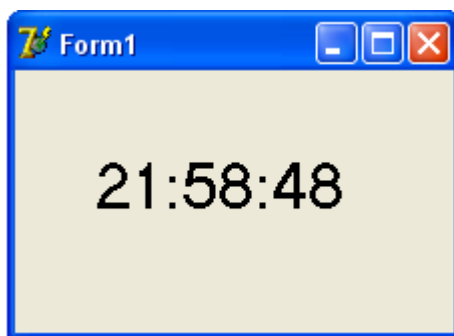
procedure TForm1.FormPaint(Sender: TObject);
begin
  ShowTime;
end;

procedure TForm1.Timer1Timer(Sender: TObject);
begin
  ShowTime;
end;

procedure TForm1.FormCreate(Sender: TObject);
begin
  Timer1.Interval:=1000;
  Timer1.Enabled:=True;

```

```
end;  
end.
```



4-есеп. Терезеде бүгінгі күн нешесі екенін, жылды, аптаның қай күні екенін және ағымдағы уақытты көрсететін “Электронды сағат”-тың бағдарламасы.

Компоненттер	Қасиеттер	Меншіктелетін мән
Form1	Caption	Электронды сағат
Label1	Caption	тазалау
Label2	Caption	тазалау
Timer1	Name	Timer1

```
unit Unit1;  
interface  
uses  
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,  
  Dialogs, StdCtrls, ExtCtrls;  
type  
  TForm1 = class(TForm)  
    Timer1: TTimer;  
    Label1: TLabel;  
    Label2: TLabel;  
    procedure FormPaint(Sender: TObject);  
    procedure Timer1Timer(Sender: TObject);  
    procedure FormCreate(Sender: TObject);  
    procedure ShowTime;  
  private  
    { Private declarations }  
  public  
    { Public declarations }  
  end;  
  
var  
  Form1: TForm1;  
  
implementation  
const  
  stDay: array[1..7] of string[11]=('жексенбі','дүйсенбі','сейсенбі',
```

```

'сәрсенбі','бейсенбі','жума','сенбі');
stMonth: array[1..12] of string[8]=('кантар','акпан','наурыз','сауір',
'мамыр','маусым','шілде','тамыз','қыркүйек','казан','қараша',
'желтоқсан');
{$R *.dfm}

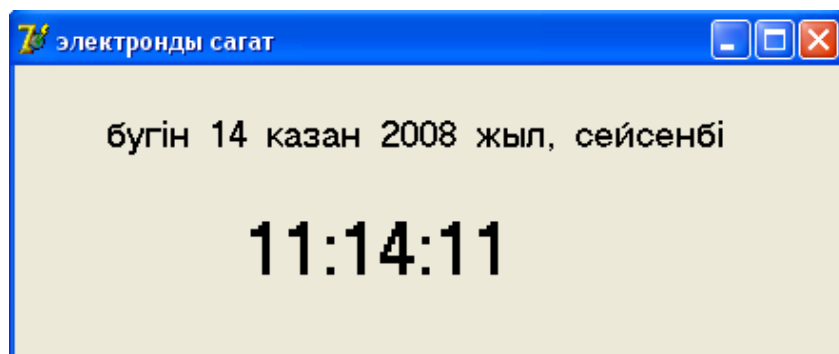
procedure TForm1.ShowTime;
var
Time: TDateTime;
begin
Time:=Now();
Label2.Caption:=FormatDateTime('hh:mm:ss',Time);
end;

procedure TForm1.FormPaint(Sender: TObject);
begin
ShowTime;
end;

procedure TForm1.Timer1Timer(Sender: TObject);
begin
ShowTime;
end;

procedure TForm1.FormCreate(Sender: TObject);
var
Present: TDateTime;
Year, Month, Day: Word;
begin
Present:=Now;
DecodeDate(Present, Year, Month, Day);
label1.Caption:='бүгін '+ IntToStr(Day)+ ' '+ stMonth[Month]+ ' '+
IntToStr(Year)+ ' жыл, '+ stDay[DayofWeek(Present)];
Timer1.Interval:=1000;
Timer1.Enabled:=True;
end;
end.

```



Бақылау сұрақтары:

1. Object Pascal тіліндегі мәліметтер типіне жалпы сипаттама беріндер?
2. Қарапайым типтерге қандай типтер жатады?
3. Реттелген типтерге қандай типтер жатады?
4. ORD(X) және PRED(X) функцияларының қызметі қандай және оларды қандай типтерге бөлуге болады?
5. SUCC(X) функциясының қызметі қандай?
6. Бүтін типтердің қандай түрлері бар және олар қалай белгіленеді?
7. Бүтін типтерге қандай стандарты процедуралар мен функциялар қолданылады?
8. Object Pascal тілінде логикалық типтердің қандай түрлері қолданылады?
9. Символдық типтер қалай белгіленеді және олардың мәндері қалай анықталады?
10. Саналатын типтер қалай белгіленеді және олардың ең жоғарғы мәні қаншаға тең?
11. Аралық тип деп қандай типті айтамыз және олар қалай бейнеленеді?
12. Аралақ типтердің қызметін сүйемелдейтін функцияларды атаңдар және олардың қызметі қандай?
13. Реттелген типтерден нақты типтердің айырмашылығы қандай?
14. Күн және уақыт типі қалай анықталады?
15. Object Pascal тілінде қанша құрылымдық тип бар?

Зертханалық жұмыс №8. Жолдық және символдық типтер

Жұмыстың мақсаты: Символдық ақпараттарды енгізуді және баспаға шығаруды ұйымдастыру, символдық ақпараттарды өңдеу әдістерін Delphi ортасында реализациялау, қарапайым қосымшалар құру дағдыларын қалыптастыру және өзіндік тапсырмалар орындау.

Бағдарламалық дидактикалық қамтылуы: Дербес компьютерлер, Windows XP операциялық жүйесі, Borland Delphi 7 бағдарламасы, «Бағдарламалау» пәнінен оқу-әдістемелік кешен.

Object Pascal-да мәтіндерді өңдеу үшін жолдар қолданылады. Жолдардың төрт түрі болады:

- 1) қысқа жол ShortString немесе String[n] (мұндағы $n \leq 256$);
- 2) Ұзын жол String;
- 3) Кең жол WideString;
- 4) нөлмен аяқталатын жол Pchar. Коды нөл символмен аяқталады.

STRING жолдық типінің ұзындығы айнымалы болып келген жады бөлігін анықтайды, әрбір символ 1 байттан орын алады. Object Pascal тіліндегі символ үшін CHAR типі пайдаланылады. STRING дегеніміз - бұл бірінен соң бірі тізбектеле орналасқан CHAR типті символдар.

Әрбір жолды символдардың бір өлшемді массиві деп қарастыруға болады. Жолдың символдарына оның сол жолдағы рет нөмірі бойынша сілтеме жасауға болады.

Нөлмен аяқталатын жолда оның өлшемі туралы айқын мәлімет болмайды. Сондықтан оның ұзындығын анықтау үшін **Length** немесе **StrLen** стандартты функциялары қолданылады. Ал **SizeOf ()** функциясы бөлінген жадының өлшемін білу үшін қолданылады.

Жолдық типтер үшін жалғастыру операциясы анықталған.

S:= 'Object' + 'Pascal'; // S-тің мәні "Object Pascal";

Сондай-ақ, жолдарды салыстыру белгілері арқылы салыстыруға болады:

"=" – тең, "<" – тең емес, ">" – үлкен, "<" – кіші, ">=" – үлкен немесе тең, "<=" – кіші немесе тең.

Жолдарға қолданылған салыстыру операцияларының нәтижесі логикалық тип болады, ол екі мәнмен сипатталады: True және False. Жолдар байт бойынша солдан оңға қарай салыстырылады. Әрбір символдар жұбы олардың ішкі кадировкасына сәйкес салыстырылады.

Жолдармен жұмыс істеуге арналған процедуралар мен функциялар

AnsiLowerCase(S) - S жолындағы бас әріптерді кіші әріптерге түрлендіреді

Ansi UpperCase(S) - S жолындағы кіші әріптерді бас әріптерге түрлендіреді

Lower Case(S) - Латын алфавитінің бас әріптерін кіші әріптерге алмастырады.

UpperCase(S) – Латын алфавитінің кіші әріптерін бас әріптерге алмастырады.

Concat(S[, S1, S2]) - S жолына аталған S1, S2 жолдарын осы ретпен жалғастырады

Copy(S, I, C) - S жолынан C символдарды I орнынан бастап қиып алады.

Delete(S, I, C) - S жолының C символын I орнынан бастап өшіреді.

Insert(S1, S, I) - S жолына S1 жолды I орнынан бастап қояды.

Length(S) - S жолының ұзындығын береді.

OleStrToStrVar(P, S) – енді (қалың) P жолын String-ке айналдырады.

Pos (S1, S) - S1 жолы S жолының нешінші символынан бастап енетіндігін анықтайды.

StringToOleStr(S) - әдеттегі жолды енді (қалың) жолға айналдырады.

Жолдарды басқа типтерге аударатын функциялар

StrToCurr(S) - S жолының символдарын Currency типіне түрлендіреді.

StrToDate (S) - S жолының символдарын "датаға" аударыады.

StrToDateTime(S) - S жолының символдарын датаға және уақытқа аударуды жүзеге асырады.

StrToFloat(S) - S жолының символдарын нақты санға айналдырады.

StrToInt(S) - S жолын бүтін санға түрлендіреді.

StrToIntDef(S,D) - S жолын бүтінге айналдырады, егер нәтиже бүтін сан болмаса, онда D параметрінде көрсетілген мәнді береді.

StrToIntRange - S жолын [min, max] аралығын қанағаттандыратын бүтін санға айналдырады.

StrToIntTime (S) - S жолын уақытқа айналдырады.

Әр түрлі типтегі мәндерді жолға айналдыратын негізгі функциялар

DateTimeToStr(V) - дата мен уақытты жолға айналдырады.

DateToStr(V) - датаны символдар жолына аударады.

FloatToStr(F) - нақты сандарды символдар жолына айналдырады.

Str(x,S) - кез-келген бүтін немесе нақты санды символдар жолына аудару мүмкіндігін береді.

TimeToStr(V) - уақытты символдар жолына түрлендіреді.

1-есеп. Функциялар мен процедуралардың орындалуын көрсететін бағдарлама.

```
Procedure TForm1.BitBtn1Click (Sender: TObject);
var
S1: String[200]; S2:ShortString; S3:String;
begin
    S1:= 'Ақтау' ; S2:= 'Mangystau' ; S3:= 'tau';
    // кіші әріптерді үлкен бас әріптерге аудару
    Memo1.Lines. Add (AnsiUpperCase (S1));
    // латынның кіші әріптерін үлкен бас әріптерге аудару
    Memo1.Lines. Add (UpperCase (S2));
    // жолдың ұзындығын анықтау
    Memo1.Lines. Add (IntToStr (Length(S1)));
    // 10 рет 'a' символынан тұратын жол құру
    Memo1.Lines. Add (StrhgOfChar ('a', 10)>);
    // ену орнын анықтау S3-тің S2-ге
    Memo1.Lines. Add (IntToStr (POS(S3, S2)));
    // жолдарды біріктіру
    Memo1.Lines. Add (Concat (S1,' ', S2));
end;
```

Орындалу нәтижесі:

АКТАУ

MANGYS

TAU

5

а а а а а а а а а

6

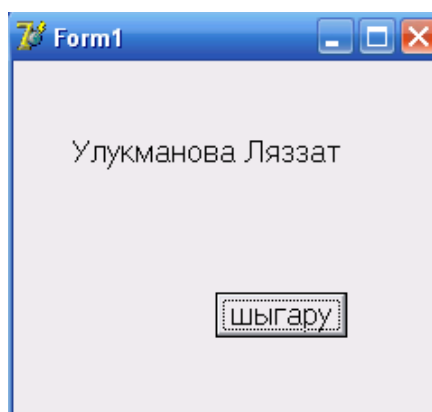
Ақтау Mangystau

2-есеп. Әр түрлі жолдық типтегі айнымалыларды сипаттап және оларға бөлінген байт санын жауапқа шығаратын бағдарлама.

```
Procedure TForm1.BitBtn1Click (Sender: TObject);
var
// ұзындығы 200 символға дейінгі қысқа жол
S1: String [200];
// ұзындығы 255 символға дейінгі қысқа жол
S2: ShortString;
S3: String; // ұзын жол
S4: WideString; // қалың (жалпақ, енді) жол
// нөл – терминал жолға сілтеме
S5: Pchar;
// 500 символға дейінгі нөл – терминал жол
S6: array [0..500] of char ;
begin
Memo1.Lines. Add ('S1='+IntToStr (SizeOf (S1)));
Memo1.Lines. Add ('S2='+IntToStr (SizeOf (S2)));
Memo1.Lines. Add ('S3='+IntToStr (SizeOf (S3)));
Memo1.Lines. Add ('S4='+IntToStr (SizeOf (S4)));
Memo1.Lines. Add ('S5='+IntToStr (SizeOf (S5)));
Memo1.Lines. Add ('S6='+IntToStr (SizeOf (S6)));
end;
```

3-есеп. Өз фамилия мен есімдеріңді терезеге шығаратын бағдарлама.

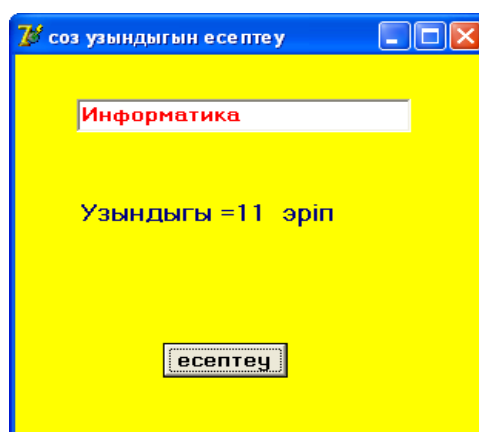
```
procedure TForm1.Button1Click(Sender: TObject);
var s:string;
    fam,at:string;
begin
    fam:='Улукманова';
    at:='Ляззат';
    s:=fam+' '+at;
    label1.Caption:=s;
end;
```



4-есеп. Берілген сөздің ұзындығын анықтайтын бағдарлама.

Компонент	Қасиеті	Меншіктелетін мән
Form1	Caption	Сөз ұзындығын есептеу
	Color	clYellow
Edit1	Text	бос
	Font	MS Sans Serif; қарайтылған; өлшемі – 10; қызыл
Label1	Caption	бос
	Font	MS Sans Serif; қарайтылған; өлшемі – 12; көк
Button1	Caption	есептеу

```
procedure TForm1.Button1Click(Sender: TObject);
Var inf:string;
Begin
Inf:=Edit1.Text;
Label1.Caption:='Ұзындығы'+inttostr(Length(inf))+' әріп';
end;
end.
```



5-есеп. Берілген мәтіннен барлық бос орынды, үтірді және нүктелерді алып тастап, ал қалған символдарды екі реттен қайталап жазу.

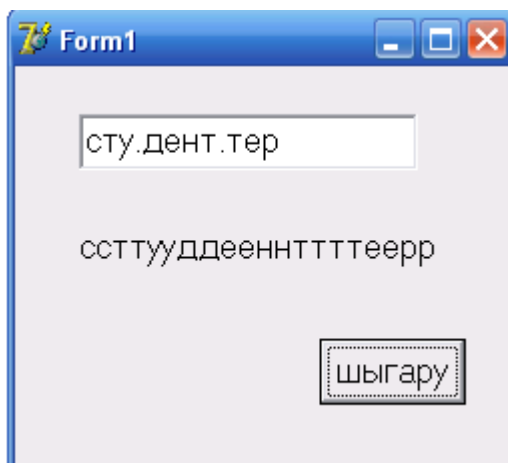
```
procedure TForm1.Button1Click(Sender: TObject);
var a,b,c:string;
    i:integer;
begin
a:=edit1.Text;
b:="";
for i:=1 to Length(a) do
begin
c:=copy(a,i,1);
if (c<>',' ) and (c<>'.' ) and (c<>' ')

```

```

    then b:=b+c+c;
end;
Label1.Caption:=b;
end;

```



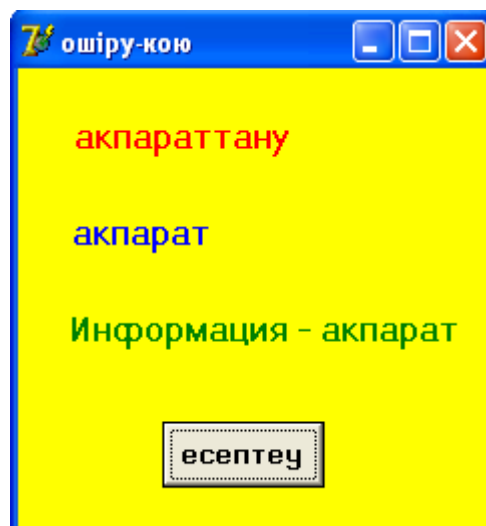
6-есеп. Берілген сөзден кез-келген символдарды өшіру және сөз қою процедурасына мысал.

Компонент	Қасиеті	Меншіктелетін мән
Form1	Caption	өшіру-қою
	Color	c\Yellow
Label1	Caption	бос
	Font	MS Sans Serif; қарайтылған; өлшемі – 12; қызыл
Label2	Caption	бос
	Font	MS Sans Serif; қарайтылған; өлшемі – 12; көк
Label3	Caption	бос
	Font	MS Sans Serif; қарайтылған; өлшемі – 12; жасыл
Button1	Caption	есептеу

```

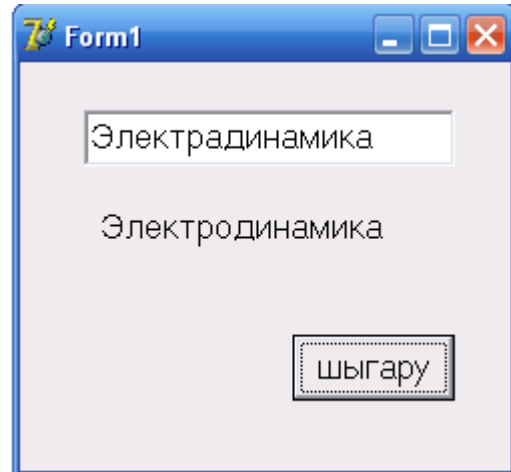
procedure TForm1.Button1Click(Sender: TObject);
Var
inf : string;
Word1 : string[20];
Begin
inf:='акпараттану';
Label1.Caption:=inf;
Delete(inf,8,4);
Label2.Caption:=inf;
Word1:='Информация - ';
Insert(word1,inf,1);
Label3.Caption:=inf;
End;
end.

```



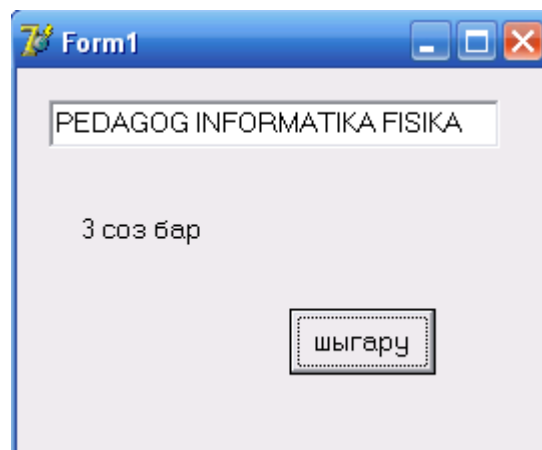
7-есеп. Мәтіннің ‘а’ әрпінің орнына ‘о’ әрпін қойып, терезеге мәтінді қайтадан шығару.

```
procedure TForm1.Button1Click(Sender: TObject);
var a:string;
    i:integer;
begin
    a:=Edit1.Text;
    for i:=1 to Length(a) do
        if copy(a,i,1)='a' then
            begin
                Delete(a,i,1);
                Insert('o',a,i);
            end;
        Label1.caption:=a;
    end;
```



8-есеп. Берілген мәтінде сөздердің санын анықтау.

```
procedure TForm1.Button1Click(Sender: TObject);
var s:string;
    k,i:integer;
begin
    k:=0;
    s:=Edit1.Text;{+' '};
    for i:=1 to Length(s) do
        if (s[i]='.') or (s[i]='?') or (s[i]='!') or
            (s[i]=' ')or (s[i]=',') or(s[i]='-') then k:=k+1;
    label1.Caption:=IntToStr(k)+' '+' сөз бар';
end;
```



9-есеп. Бес сөз енгізіп, оны алфавит ретімен орналастыратын бағдарлама.

```

procedure TForm1.Button1Click(Sender: TObject);
Var a:array [1..5] of string;
i,j:integer;
l:string;
Begin
Label1.Caption:='5 сөз енгізіңіз';
For i:=1 to 5 do
a[i]:=inputBox('', '',i);
For i:=1 to 5 do
For j:=i to 5 do
If a[i]>a[j] Then
Begin
l:=a[j];
a[j]:=a[i];
a[i]:=l;
end;
for i:=1 to 5 do
ListBox1.Items.Add(a[i]);
End;

```

10-есеп. Берілген «Компьютерлендіру» сөзін буынға бөліп көшіру.

Компоненттер	Қасиеттері	Мәндері
Form1	Caption	Көшіру функциясы
	Color	c1Yellow
Label1	Caption	бос
	Font	MS Sans Serif; қарайтылған; өлшемі – 12; қызыл
Label2; Label3; Label4; Label5; Label6; Label7	Caption	бос
	Font	MS Sans Serif; қарайтылған; өлшемі – 12; көк
Button1	Caption	есептеу

```

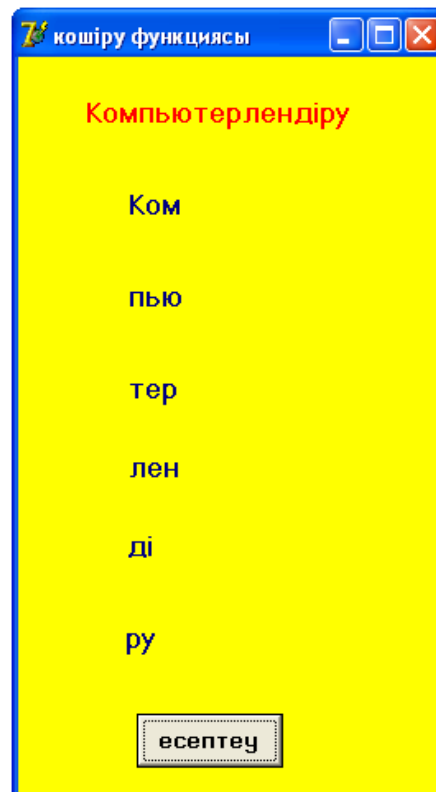
procedure TForm1.Button1Click(Sender: TObject);
Var inf:string [20];
Word1,word2,word3,word4,word5,word6:string[10];
Begin
inf:='Компьютерлендіру';
Label1.Caption:=inf;
Word1:=Copy(inf,1,3);
Label2.Caption:=(Word1);
Word2:=Copy(inf,4,3);

```

```

Label3.Caption:=word2;
Word3:=Copy(inf,7,3);
Label4.Caption:=word3;
Word4:=Copy(inf,10,3);
Label5.Caption:=word4;
Word5:=Copy(inf,13,2);
Label6.Caption:=word5;
Word6:=Copy(inf,15,2);
Label7.Caption:=word6;
End;
end.

```



11-есеп. Пернетақтадан енгізілген оналтылық санау жүйесіндегі санды ондық санау жүйесіне аударатын бағдарлама.

Оналтылық санау жүйесіндегі санды – жолдық айнымалы мәнін Edit1 компонентінен оқимыз. Ондық санау жүйесіндегі санды Memo1 компонентіне шығарамыз.

Формаға Edit1, Memo1, Button1 компоненттерін қоямыз. Есеп әдісін Button1 компонентінің OnClick оқиғасында ұйымдастырамыз.

```

procedure TForm1.Button1Click(Sender: TObject);
var r,dec,k,i,s1:integer;
sh16:string;
sn: set of char;
begin
sn:=['0'..'9'];
{оналтылық санау жүйесіндегі санды енгіз}
sh16:=edit1.text;
dec:=0; r:=1;
for i:=length(sh16) downto 1 do
begin
if (sh16[i] in sn) then begin
val(sh16[i],s1,k); dec:=dec+r*s1; end
else
if (sh16[i]>='A') and (sh16[i]<='F') then
begin s1:=ord(sh16[i]);
dec:=dec+r*(s1-55); end;

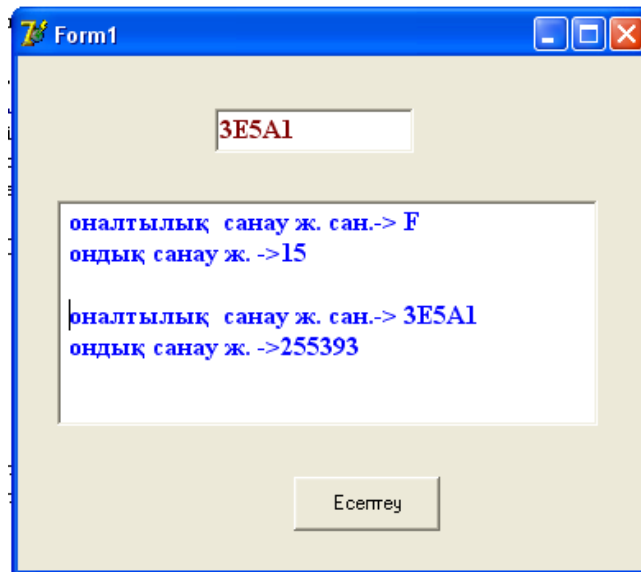
```

```

r:=r*16;
end;
Memo1.Lines.Add('оналтылық санау ж. сан.-> '+sh16);
Memo1.Lines.Add('ондық санау ж. ->'+inttostr(dec));
end;

end.

```



12-есеп. Ұзындығы әр түрлі сөздер тізбегі берілген. Тізбектегі оң жағынан да, сол жағынан да бірдей оқылатын сөздерді анықтаңыздар. Мысалы, «шалаш», «апа» т.б.

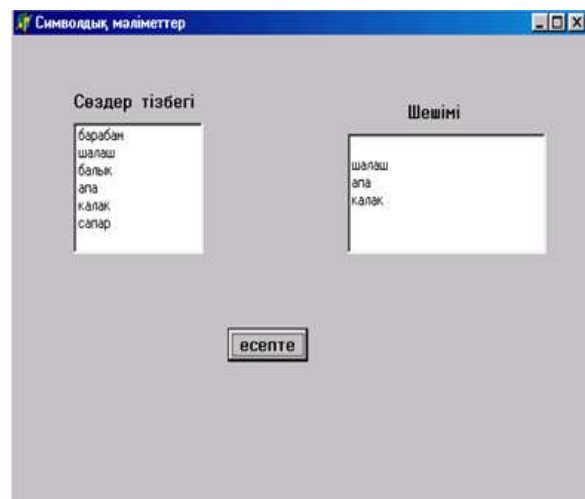
Бастапқы сөздер тізбегін Listbox1 компонентінен оқимыз. Есеп шешімін Memo1 компонентіне шығарамыз.

Формаға Listbox1, Memo1, Button1 компоненттерін қоямыз. Есеп әдісін Button1 компонентінің OnClick оқиғасында ұйымдастырамыз.

```

procedure TForm2.Button1Click(Sender: TObject);
var sz,st:string;
i,k,l:integer;
begin
k:=Listbox1.Items.Count-1;
for i:=1 to k-1 do
begin
sz:= Listbox1.Items[i];
st:="";
for l:=length(sz) downto 1 do
st:=st+sz[l];
if sz=st then Memo1.Lines.Add(sz);
end;
end;
end;

```



Тапсырмалар:

1. Символдар жолы берілген. А әріпімен аяқталатын сөзді экранға шығаратын бағдарлама құру.
2. Символдар жолы берілген. Жолдағы бос орындар санын анықтаңдар және оларды үтірмен ауыстыратын бағдарлама құру.
3. Жолдағы үтір (,) белгісін леп белгісіне (!) ауыстыратын бағдарлама құру.
4. Сөздер тізбегі берілген. Эcranға тек 6 символдан тұратын сөздерді шығаратын бағдарлама құру.
5. Символдар жолы берілген. Жолда “лар” жалғауы қанша рет кездесетінін анықтайтын бағдарлама құру.
6. Құрамында А әріпі екіден кем кездеспейтін сөзді анықтайтын бағдарлама құру.
7. Берілген сөйлемде “а” әрпінен басталып “а” әрпімен аяқталатын сөздер санының қанша екенін анықтайтын бағдарлама құру.
8. Берілген сөйлемдегі барлық қайталанатын сөздерді алып тастап қайта шығаратын бағдарлама құру.

Бақылау сұрақтары:

1. Жол неден тұрады? Жолдық типті сипаттаудың қандай жолдары бар?
2. Бір жолға қанша символ жазуға болады?
3. Жолдық айнымалы мен тұрақтыларға қандай амалдар қолданылады?
4. Қандай жолдық стандартты функциялар бар? Олардың міндеттері қандай?
5. Қандай жолдық процедуралар бар? Олардың міндеттері қандай?
6. Символдарды қалай сипаттауға болады?
8. Бүтін сандарды жолдық типке ауыстыруға бола ма?
9. Нақты типті жолдық типке қалай ауыстыруға болады?

Зертханалық жұмыс №9. Бір өлшемді массивтерді өңдеу алгоритмдері

Жұмыстың мақсаты: Массив элементтерін енгізуді және баспаға шығаруды ұйымдастыру, массивтегі іздеу және элементтерді сұрыптау әдістерін Delphi ортасында бағдарламалау дағдыларын қалыптастыру және өзіндік тапсырмаларды орындау.

Бағдарламалық дидактикалық қамтылуы: Дербес компьютерлер, Windows XP операциялық жүйесі, Borland Delphi 7 бағдарламасы, «Бағдарламалау» пәнінен оқу-әдістемелік кешен.

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 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. Массивті сипаттау құрылымының жалпы түрі:

<Массив аты>: **array** [индекстік типтер тізбегі] **of** <элемент типі>;

Мысалы:

```

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

↙
 Массив
 атауы

 {
 5 элементті
 массив

 {
 Нақты тип

мұндағы, array, of – қызметші сөздер.

1 – массивтің төменгі индексі. 5- массивтің жоғарғы индексі.

Массивтің жоғарғы индексін тұрақтылар бөлімінде анықтау:

```

const
    N=18; // фамилиялар саны
    S=25 // фамилияның ұзындығы
var
    t: array [1..N] of string [S];
    
```

Массивтерді дұрыс жазуға мысалдар келтірейік:

```

VAR
    MAS:ARRAY[1..N] OF REAL;
    Жыл: ARRAY[қаңтар..желтоқсан] OF INTEGER;
    TYPE
    Mat=array[0..5] of array[Char] of Byte;
    N: array [1..20] of string[15]
    
```

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

Массив атауы [индексі]:= нәтиже

Мысалы:

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

ә) В массивінің бесінші элементіне Улукманова сөзін меншіктеу үшін - B[5]:='Улукманова';

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;
```

Жадыны бөлу мен динамикалық массивтің әрбір өлшемі бойынша индекстердің шекарасын көрсету бағдарламаның орындалуы барысында SetLength функциясының көмегімен массивті тағайындау жолымен жүзеге асырылады.

Жадыны босату үшін атауларға Nil мәнін меншіктеу жеткілікті (басқа тәсілмен Finalize процедурасын пайдалану болып табылады).

Массивті енгізу деп бағдарламаны орындау кезінде пайдаланушыдан массив элементтерінің мәнін алу процесін айтамыз.

Delphi ортасында экранға массив элементтерін енгізу және шығаруды компоненттер палитрасының Additional бетінде орналасқан StringGrid компонентінің көмегімен ұйымдастырған ыңғайлы. Бұл компонент ақпаратты кесте түрінде бейнелеу үшін пайдаланылады.

StringGrid компонентінің қасиеттері:

Name – компонент аты.

ColCount – кесте бағандарының саны.

RowCount – кесте жолдарының саны.

Cells – сәйкес кестедегі екі өлшемді массив. Col бағаны мен Row жол номерінің қиылысуында орналасқан кесте ұяшығында элемент былай анықталады:

Cells[col,row]

FixedCols – кестенің арнайы ерекшеленген сол жақ бағандар саны.

FixedRows – кестенің арнайы ерекшеленген жоғарғы жолдар саны.

Options.goEditing – кесте ұяшықтарындағы мәліметтерді өңдеуге True – рұқсат, False – рұқсат жоқ.

Options.goTab – курсорды кестенің келесі ұяшығына жылжыту үшін Tab пернесін (True) қолдану немесе (False) қолданбау.

Options.GoAllWayShowEditor – өңдеу режимінде компонентті табу белгісі. Егер қасиеттің мәні False онда ұяшықта курсор пайда болу үшін тексті тере бастау керек, <F2> пернесін басу керек немесе тышқанмен шерту керек.

DefaultColWidth – кесте бағандарының ені.

DefaultRowHeight – кесте жолдарының биіктігі.

GridLineWidth – кесте ұяшықтарын жиектейтін сызық ені.

Left – кесте алаңының сол жақ жиегінен форманың сол жақ жиегіне дейінгі қашықтық.

Top – кесте алаңының жоғарғы жиегінен форманың жоғарғы жиегіне дейінгі қашықтық.

Height – кесте алаңының биіктігі.

Width – кесте алаңының ені.

Font – кесте ұяшығындағы мәліметті бейнелеу үшін қолданылатын қаріп.

ParentFont – форма қаріпінің сипаттамаларының белгісі.

Мемо компонентін пайдаланып массивті енгізу

Мемо компоненті көптеген жолдан тұратын мәтінді енгізу мүмкіндігін береді. Сондықтан оны символдық массивті енгізу үшін қолдану ыңғайлы. Ол Standart бетінде орналасқан.

Мемо компонентін пайдаланып массивті енгізгенде, массивтің әрбір элементінің мәні жеке жолда жазылады және әрбір элементтен соң <Enter> пернесін басу керек.

Мемо компонентінің қасиеттері:

Name – компонент аты.

Text – Мемо алаңында тұратын мәтін. Бір бүтін ретінде қарастырылады.

Lines – Мемо алаңындағы мәтін. Жолдардың (қосындысы) жиынтығы ретінде қарастырылады. Жолға оның нөмірі бойынша хабарласады.

Lines.Count – Мемо алаңындағы мәтіннің жолдарының саны.

Left – алаңның сол жақ жиегінен форманың сол жақ жиегіне дейінгі қашықтық.

Top – алаңның жоғарғы жиегінен форманың жоғарғы жиегіне дейінгі қашықтық.

Height – алаң биіктігі.

Width – алаң ені.

Font – енгізілетін мәтіннің қаріпі.

1-есеп. $A(N)$ сандық массивінің тақ нөмірлі элементтерін таңдау арқылы сұрыптау алгоритмін қолдана отырып, кему ретімен орналастыратын бағдарлама.

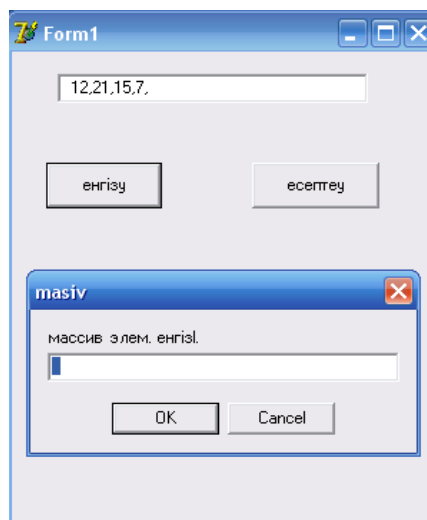
$A(N)$ сандық массивінің элементтерін InputBox диалогты терезесінен оқимыз және Edit1 компонентіне шығарамыз.

Тақ нөмірлі элементтері кему ретімен орналасқан сандық $A(N)$ массивінің элементтерін Button2 компонентіне шығарамыз.

Ең үлкен элемент анықталады да бірінші тақ орындағы элементпен орын ауыстырылады. Одан кейін қалған элементтердің ең үлкені анықталады да екінші тақ орындағы элементпен орын ауыстырылады. Процесс барлық тақ нөмірлі элементтер ретімен орналасқанша жалғасады. Формаға Edit1, Button1, Button2 компоненттерін қоямыз.

Есеп әдісін Button1 компонентінің OnClick оқиғасында ұйымдастырамыз.

```
procedure TForm1.Button1Click(Sender: TObject);
var a:array[1..15] of integer;
    i,j,max,m,n:integer;
s:string;
begin
n:=6;
for i:=1 to n do
begin
s:=inputbox('masiv','массив элем. енгізіл.',' ');
a[i]:=strtoint(s);
edit1.Text:= edit1.Text+ s+' ';
end;
i:=1;
while i< n do
begin
max:=a[i]; m:=i;
j:=i+2;
while j<=n do
begin
if a[j]>max then begin max:=a[j]; m:=j; end;
inc(j,2);
end;
a[m]:= a[i];
a[i]:=max; inc(i,2);
end;
for i:=1 to n do
button2.caption:=button2.caption + inttostr(a[i])+',';
end;
end.
```

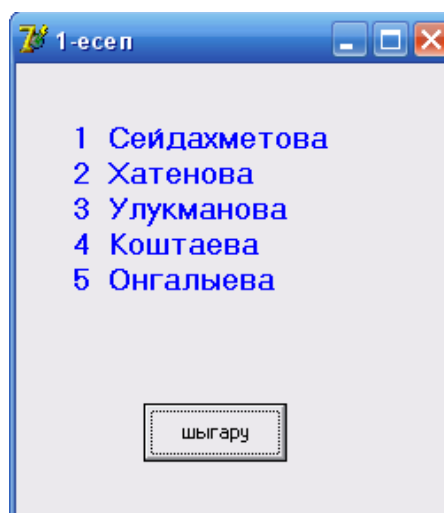


2-есеп. Нөмірленген фамилиялар тізімін жауапқа шығару.

Бағдарламаға қажетті компоненттер: Label1, Button1.

Компонент	Қасиеті	Меншіктелетін мән
Form1	Caption	1-есеп
Label1	Caption	бос
	Font	Times New Roman; өлшемі – 12; қарайтылған; түсі – көк.
Button1	Caption	шығару

```
procedure TForm1.Button1Click(Sender: TObject);
const N = 5;
t:array[1..N] of string[15]=('Сейдахметова','Хатенова','Улукманова',
'Коштаева','Онгальева');
var
  S:string;
  i:integer;
begin
  for i:=1 to N do
    s:=s + IntToStr(i)+ ' ' + t[i]+#13;
  Label1.Caption:=S;
end;
end.
```



3-есеп. Массив элементтерінің арифметикалық орташасын есептейтін бағдарлама.

Бағдарламаға қажетті компоненттер: StringGrid1; Button1; Label1.

StringGrid1 компонентінің қасиеттерінің мәндері:

ColCount = 5

FixedCols = 0

RowCount = 1

DefaultRowHeight = 24

Height = 24

DefaultColWidth = 64

Width = 328

Options.goEditing = True

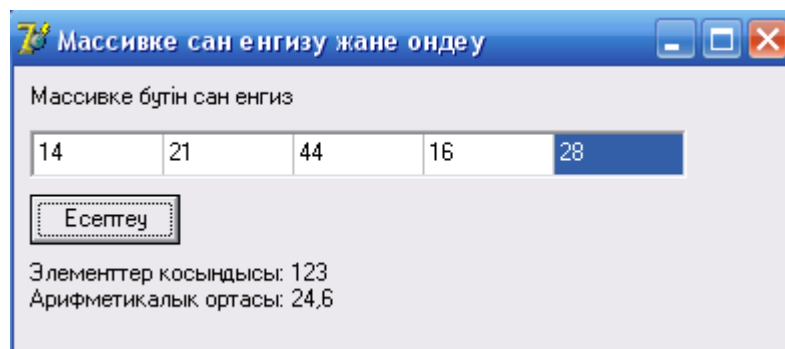
Options. goAlwaysShowEditing = True

Options. goTabs = True

```

procedure TForm1.Button1Click(Sender: TObject);
var
  a : array[1..5] of integer; // массив
  summ: integer;           // элементтер қосындысы
  sr: real;                // орташа арифметикалық
  i: integer;              // индекс
begin
  for i:= 1 to 5 do
    if Length(StringGrid1.Cells[i-1,0]) <> 0
      then a[i] := StrToInt(StringGrid1.Cells[i-1,0])
      else a[i] := 0;
  summ := 0;
  for i :=1 to 5 do
    summ := summ + a[i];
  sr := summ / 5;
  // нәтижені шығару
  Label1.Caption := 'Элементтер қосындысы: ' + IntToStr(summ) + '#13+'
    'Арифметикалық ортасы: ' + FloatToStr(sr);
end;
end.

```



4-есеп. Нөмірленген автокөлік маркаларының тізімін шығаратын бағдарлама.

Бағдарламаға қажетті компоненттер: Button1; Label1; Label2.

```

const
  NT = 5;
var
  team:array[1..NT] of string[10] = ('Toyota','Honda','Mazda','Suzuki','Nisan');

procedure TForm1.Button1Click(Sender: TObject);

var
  st:string;
  i:integer; begin

```

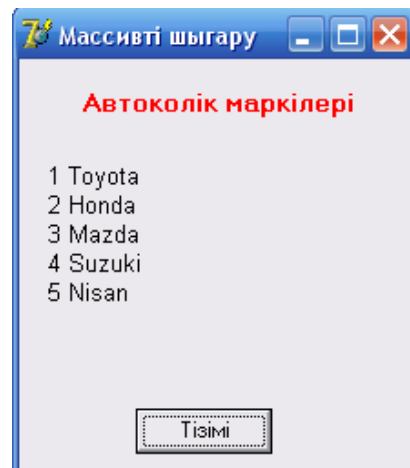
```

for i:=1 to NT do
  st:=st+IntToStr(i)+' '+team[i]+#13;

Label1.Caption := st;
end;

end.

```



5-есеп. Мемо компонентін пайдаланып, символдық массивті енгізетін бағдарлама.

Бағдарламаға қажетті компоненттер: Мемо; Button1; Label1.

```

procedure TForm1.Button1Click(Sender: TObject);
const
  SIZE=5; // массив олшемі
var
  a:array[1..SIZE]of string[30]; // массив
  n: integer; // Мемо орісіне енгізілген жол саны
  i:integer; // массив элементінің индексі
  st:string;
begin
  n:=Memo1.Lines.Count;
  if n = 0 then
    begin
      ShowMessage('Бастапқы мандери берілмеген');
      Exit;
    end;

  if n > SIZE then
    begin
      ShowMessage('Енгізілген жол саны массив олшемінен улкен');
      n:=SIZE;
    end;
  for i:=1 to n do
    a[i]:=Form1.Memo1.Lines[i-1];
  if n > 0 then
    begin
      st:='Енгізілген массив:'+#13;
      for i:=1 to n do
        st:=st+IntToStr(i)+' '+ a[i]+#13;
      ShowMessage(st);
    end;
end;

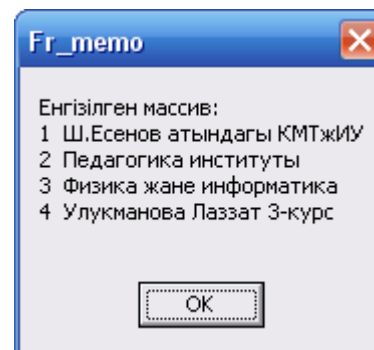
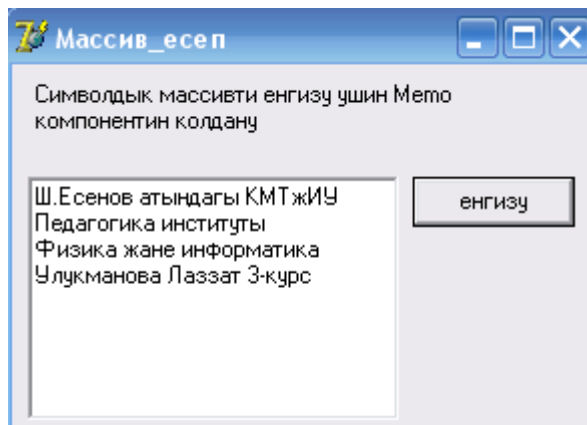
```



```

end;
end;
end.

```



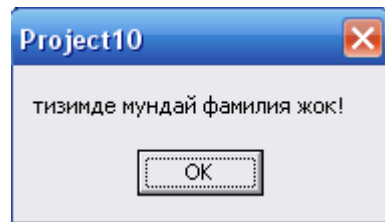
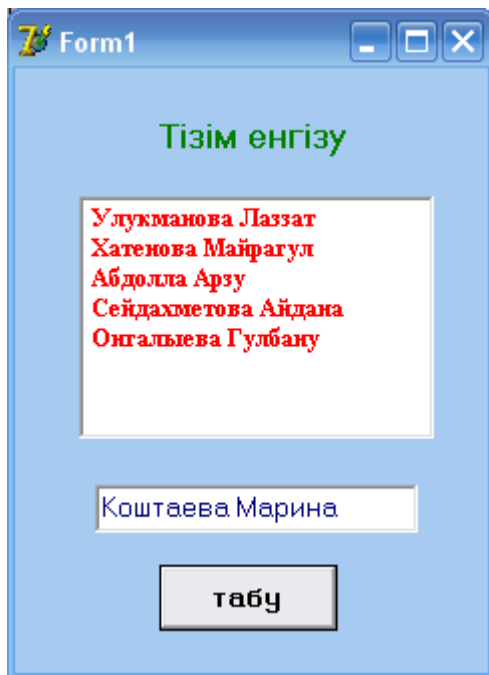
6-есеп. Мемо компоненті арқылы оқушылар тізімін енгізу. Төменде Edit1 компонентіне оқушының фамилиясы мен атын жазып, оның атын және нешінші жолда орналасқанын анықтайтын бағдарлама.

Бағдарламаға қажетті компоненттер: Label1;Memo1; Edit1; Button1;

```

procedure TForm1.Button1Click(Sender: TObject);
var
i:integer;
j:integer;
a:boolean;
begin
a:=false;
for i:=0 to Memo1.Lines.Count do
if Edit1.Text=Memo1.Lines[i] then
begin
a:=true;
for j:=1 to length(Edit1.Text) do
if Edit1.Text[j]=' ' then ShowMessage(copy(Edit1.Text,j,length(Edit1.Text))+
'+IntToStr(i{+1}));
break;
end;
if Not a then ShowMessage('тизимде мундай фамилия жок!');
end;
end.

```



7-есеп. Массивтің ең кіші элементін іздеу бағдарламасы.

Procedure TForm1.Button1Click (Sender: TObject);

const

Size = 5;

var

a: array[1..SIZE] of integer; // бүтін сандар массиві

min: integer; // Массив ең кіші элементінің номері

i: integer; // ең кішімен салыстырылатын элемент номері

begin

for i: = 1 to SIZE do

a[i]: = StrToInt(StringGrid1.Cells[i - 1,0]);

// ең кіші элементін іздеу

min: = 1; // бірінші элемент ең кіші болсын

for i: = 2 to SIZE do

if a[i] < a[min] then min: = i;

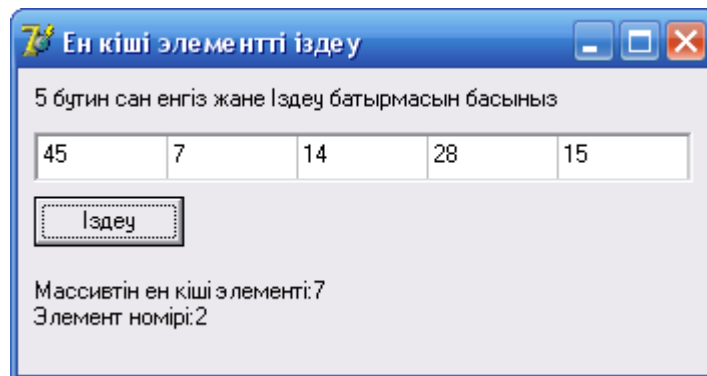
// жауапты шығару

label2.caption: = 'Массивтің ең кіші элементі: '+

IntToStr(a[min]) + # 13+

'Элемент номері: ' + IntToStr(min);

end;



8-есеп. Бір өлшемді массивтің жұп элементтерінің қосындысын табу.

Формаға Label1, Label2, StringGrid1, Button1 компоненттерін орналастырыңыздар.

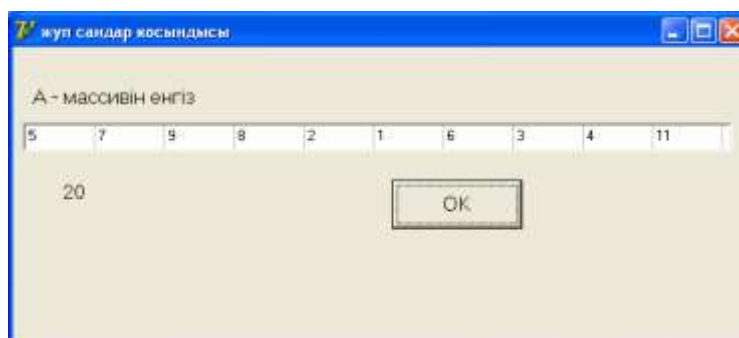
ОК батырмасына екі рет шертіп, бағдарлама коды терезесінде процедура денесін жазыңыздар.

```

procedure TForm1.Button1Click(Sender: TObject);
var
a:array[1..10] of integer;
  i:integer;
  b:integer;
  max:integer;
begin
//массивті енгізу
  for i:=1 to 10 do
    a[i]:=StrToInt(StringGrid1.Cells[i-1,0]);
  b:=0;
  for i:=1 to 10 do
    if a[i] mod 2 = 0 then b:=b+a[i];
    label2.Caption:=inttostr(b);
  end;
end;

```

Бағдарламаны орындау, формаға массив элементтерін енгізу. ОК батырмасын басу, сонда енгізілген массив элементтерінің жұп элементтерінің қосындысы шығады.



9-есеп. Бір өлшемді массив берілген. Массивтің жұп нөмірлі элементтерінің ішінен теріс элементтерінің санын анықтайтын бағдарлама.

Формаға Label1, Label2, StringGrid1, Button1 компоненттерін орналастырыңыздар.

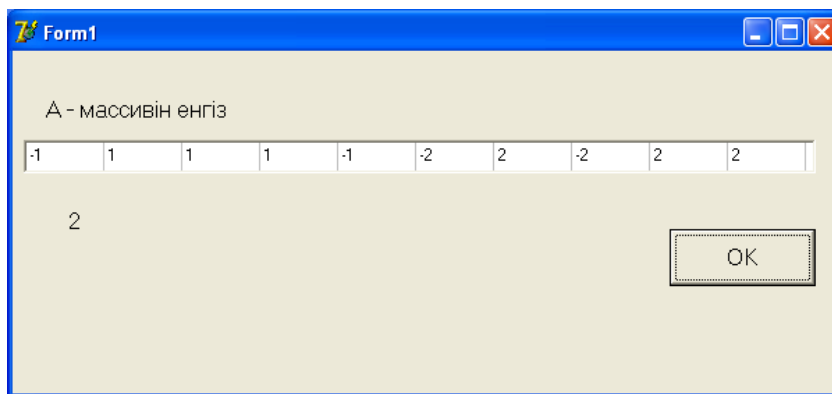
OK батырмасына екі рет шертіп, бағдарлама коды терезесінде процедура денесін жазыңыздар.

```

procedure TForm1.Button1Click(Sender: TObject);
var a:array[1..10] of integer;
    i,k:integer;
begin
    // массивті енгізу
    for i:=1 to 10 do
        a[i]:=StrToInt(StringGrid1.Cells[i-1,0]);
        k:=0;
        for i:=1 to 10 do
            if (i mod 2=0) and (a[i]<0) then k:=k+1;
            label2.Caption:=inttostr(k);
        end;
end;

```

Бағдарламаны орындау, формаға массив элементтерін енгізу. OK батырмасын басу.



10-есеп. Бүтін сандардан тұратын массив элементтерін өсу реті бойынша сұрыптау бағдарламасы.

Бағдарламаға қажетті компоненттер: StringGrid1; Label1; Label2; Button1;

Компонент	Қасиеті	Меншіктелетін мән
Form1	Caption	Form1
	Color (түсі)	clGradientActiveCaption
Label1	Caption	Массив элементтерін енгізу
	Font	Times New Roman;

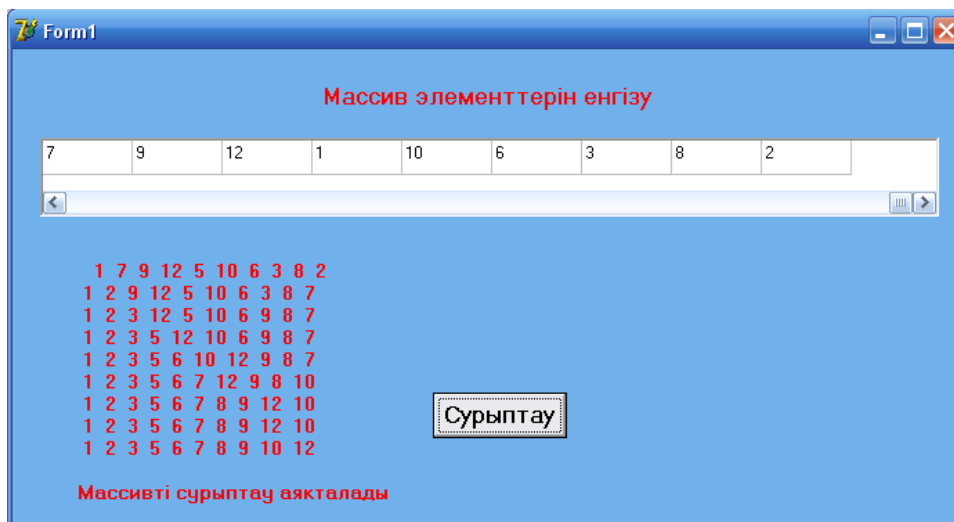
		өлшемі – 12; қарайтылған; түсі – қызыл.
Label2	Caption	бос
Button1	Caption	сұрыптау
StringGrid1	ColCount	10
	FixedCols	0
	RowCount	1
	DefaultRowHeight	24
	Height	24
	DefaultColWidth	64
	Width	650
	Options.goEditing	True
	Options. goAlwaysShowEditing	True
	Options. goTabs	True

```

Procedure TForm1.Button1Click(Sender:TObject);
Const S=10;
var   A:array[1..S] of integer;
min: integer; {ең кіші элемент номері}
j: integer; {ең кішімен салыстырылатын элемент номері}
buf: integer; {массив элементтерін салыстыру кезінде қолданылатын
буфер айнымалы}
i, k: integer; {цикл параметрі}
begin
// массивті енгізу
for i:=1 to S do
A[i] := StrToInt (StringGrid1. Cells[i - 1,0]);
Label2. Caption: = ' ';
for i:=1 to S - 1 do
begin
min: =i;
for j: = i + 1 to S do
if A[j] < A[min] then min: = j;
buf: = A[i];
A[i]: = A[min];
A[min]: = buf;
// массивті шығару
for k:=1 to S do
Label2. Caption: = Label2. Caption + ' ' + IntToStr (A[k]);
Label2. Caption: = Label2. Caption + #13;
end;
Label2.Caption: = label2.Caption + # 13 + ' Массивті сұрыптау аяқталады';
end;

```

end.



11-есеп. Бағдарламаның орындалу барысында массив өлшемі өзгеретін болсын: алдымен 10 санның квадраты есептеліп экранға шығарылсын; одан кейін 20 санның квадраты есептеліп шығарылатын бағдарлама.

Ол үшін бір ListBox, екі Button компонентін орналастырайық. Бірінші батырма басылғанда 10 санның квадраты, екінші батырма басылғанда 20 санның квадраты шығарылатын оқиға өңдеуші процедураларының мәтіні төмендегідей:

```
unit Unit1;
interface
uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls;
type
  TForm1 = class(TForm)
    ListBox1: TListBox;
    Button1: TButton;
    Button2: TButton;
    procedure Button1Click(Sender: TObject);
    procedure Button2Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;
var
  Form1: TForm1;
implementation
{$R *.dfm}
```

```

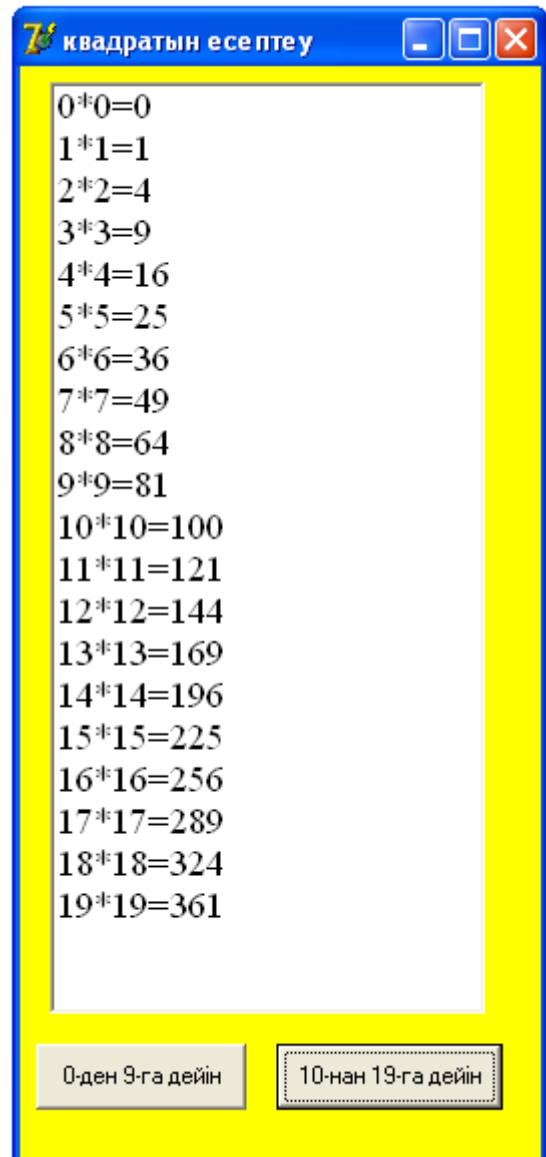
procedure TForm1.Button1Click(Sender: TObject);
Var
R: array of integer;
I: Integer;
Begin
ListBox1.Items.Clear;
SetLength (r, 10);
For i:=0 to High(r) do
Begin
R[i]:=i*i;
ListBox1.Items.Add(IntToStr(i)+'*' + IntToStr(i)+'=' + IntToStr(r[i]));
End;
End;

```

```

procedure TForm1.Button2Click(Sender: TObject);
Type
TDynArr= array of integer;
Var
R: TDynArr;
I: Integer;
Begin
ListBox1.Items.Clear;
SetLength(r,10);
For i:=0 to High (r) do
R[i]:=i*i;
SetLength(r,20);
For i:=10 to High(r) do
R[i]:=i*i;
For i:=0 to High(r) do
R[i]:=i*i;
For i:=0 to High(r) do
ListBox1.Items.Add('+IntToStr(i)+
'+IntToStr(i)+ '=' +IntToStr(r[i]));
End;
end.

```



Тапсырмалар:

1. Бір өлшемді C[20] массивінің бірінші элементін соңғы элементімен, ал екінші элементін соңғы элементтің алдындағы элементпен, т.с.с. қосындар. Нәтижені D массивіне жазып, оның ең үлкенін табындар.

2. Екі бір өлшемді массив берілген. Олардың жұп нөмірлі элементтерінің қосындысын табындар және қосындылардың арасынан ең үлкенін анықтаңдар.

3. A[10] сандар тізбегін екі массивке бөліңдер, олардың біріншісіне жұп нөмірлі элементтерді, ал екіншісіне тақ нөмірлі элементтерді жазындар.

4. A[10], B[10] массивтері берілген. Есептеу керек:

$$(A_1 + B_n) * (A_2 + B_{n-1}) * (A_3 + B_{n-2}) \dots (A_n + B_1)$$

5. A[10] массиві берілген. Массивтегі 2-ден кіші элементтерді 0-мен алмастырындар, экранға олардың санын және реттік нөмірін шығарындар.

6. A[15] массивінің оң элементтерінің көбейтіндісін табындар.

Бақылау сұрақтары:

1. Массив дегеніміз не және оның жазылу форматы қандай?

2. Массивтің айнымалы және тип бөлігіндегі бейнеленуі.

3. Бағдарламалауда массивтерді қолдану арқылы қандай жетістіктерге қол жеткізуге болады?

4. Массив типіне қандай типтерді қолдануға болады?

5. Массивтерді сипаттаудың қандай ерекшеліктері бар?

6. Массивке тұрақтыларды қолдануға бола ма?

7. SetLength функциясының қызметі қандай?

Зертханалық жұмыс №10. Екі өлшемді массивті өңдеу алгоритмдері

Жұмыстың мақсаты: Екі өлшемді массив элементтерін енгізуді және баспаға шығаруды ұйымдастыру, екі өлшемді массив элементтерін өңдеу әдістерін Delphi ортасында бағдарламалау дағдыларын қалыптастыру және өзіндік тапсырмаларды орындау.

Бағдарламалық дидактикалық қамтылуы: Дербес компьютерлер, Windows XP операциялық жүйесі, Borland Delphi 7 бағдарламасы, «Бағдарламалау» пәнінен оқу-әдістемелік кешен.

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;

Type бөлімінде екі өлшемді массивті сипаттау:

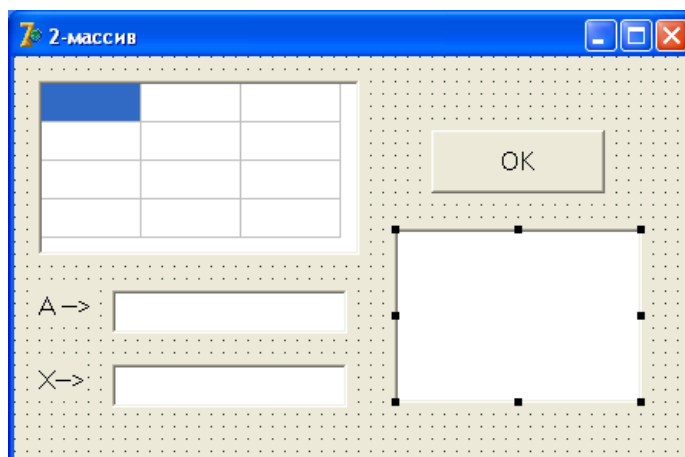
Type

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

1-есеп. Екі өлшемді массивтен a-дан үлкен, x-тен кіші болатын санды және оның қандай жолда орналасқанын табатын бағдарлама.

Формаға Label1, Label2, Edit1, Edit2, StringGrid1, Button1, Memo1 компоненттерін орналастыру. StringGrid1 компонентінің қасиеттерінің мәндері кестеде көрсетілген.

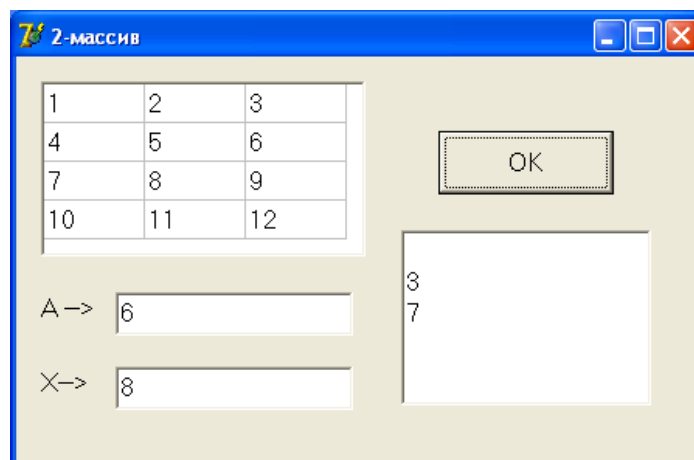
Қасиеттері	Меншіктелетін мән
ColCount	3
RowCount	4
FixedCols	0
DefaultRowHeight	24
Height	113
DefaultColWidth	64
Width	209
Options.goEditing	True
Options.AlwaysShowEditing	True
Options.goTabs	True



«OK» батырмасына екі рет шертіп, бағдарлама код терезесінде процедура денесін жазыңыздар.

```
procedure TForm1.Button1Click(Sender: TObject);
var b:array[0..2,0..3] of integer;
i,j:integer;
k,t:integer;
a,x:integer;
begin
  for i:=0 to 2 do
    for j:=0 to 3 do
      b[i,j]:=strtoint(StringGrid1.Cells[i,j]);
a:=strtoint(Edit1.text);
x:=strtoint(Edit2.text);
k:=0;
  for i:=0 to 2 do
    for j:=0 to 3 do
      begin
        if (b[i,j]>a) and (b[i,j]<x) then begin
          k:=j+1; t:=b[i,j];
        end;
      end;
  memo1.lines.add(inttostr(k));
  memo1.lines.add(inttostr(t));
end;
```

Бағдарламаны орындау, формаға массив элементтерін енгізу. А-ның мәнін (6), Х-тің мәнін (8) енгізу, «OK» батырмасын басу сонда Memo1 алаңына нәтиже шығады.



2-есеп. $A(N, N)$ екі өлшемді сандық массивінің ең үлкен элементін анықтайтын бағдарлама.

$A(N, N)$ сандық массивінің элементтерін StringGrid1 кестесінен оқимыз.

$A(N, N)$ сандық массивінің ең үлкен элементін Button2 компонентіне шығарамыз.

Формаға StringGrid1, Button1, Button2 компоненттерін қоямыз. Есеп әдісін Button1 компонентінің OnClick оқиғасында ұйымдастырамыз.

StringGrid компонентінің қасиеттерінің мәні кестеде көрсетілген:

ColCount	Кестедегі бағандар саны.
RowCount	Кестедегі жолдар саны.
DefaultColWidth	Кесте бағанының ені.
Cells (ACol, ARow)	Ұяшық мәнін анықтайды
Options/ go Editing	True енгізілген. Бұл жағдайда кестені редакциялауға болады.

```
procedure TForm1.Button1Click(Sender: TObject);
```

```
var a:array[1..3,1..3] of integer;
```

```
    i,j,max:integer;
```

```
begin
```

```
for i:=1 to 3 do
```

```
for j:=1 to 3 do
```

```
a[i,j]:=strtoint(StringGrid1.Cells[i,j]);
```

```
max:=a[1,1];
```

```
for i:=1 to 3 do
```

```
for j:=1 to 3 do
```

```
if max<a[i,j] then max:=a[i,j];
```

```
button2.Caption:='max='+inttostr(max);
```

```
end;
```

3-есеп. $A(N, N)$ екі өлшемді сандық массивінің әр жол элементтерін алмастыру арқылы сұрыптау алгоритмін қолдана отырып өсу ретімен орналастыратын бағдарлама.

$A(N, N)$ сандық массивінің элементтерін StringGrid1 кестесінен оқимыз.

$A(N, N)$ массивінің әр жолы өсу ретімен реттелген элементтерін StringGrid2 компонентіне шығарамыз.

Формаға StringGrid1, StringGrid2, Button1 компоненттерін қоямыз. Есеп әдісін Button1 компонентінің OnClick оқиғасында ұйымдастырамыз.

```
procedure TForm1.Button1Click(Sender: TObject);
```

```
var a:array[1..3,1..3] of integer;
```

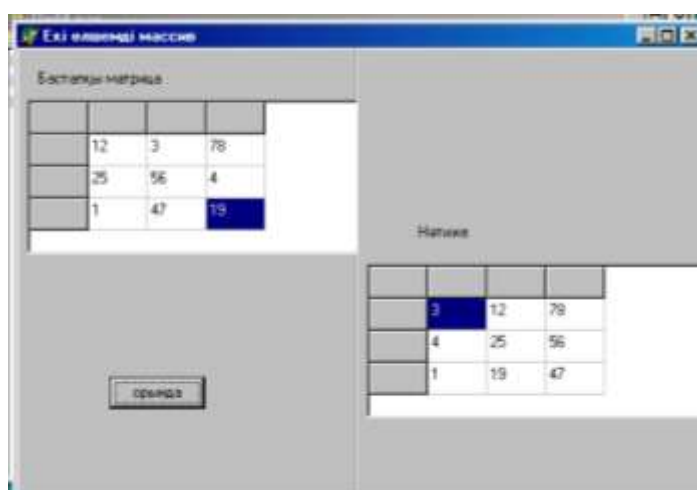
```
    i,j,k,m:integer;
```

```
begin
```

```

for i:=1 to 3 do
for j:=1 to 3 do
a[i,j]:=strtoint(StringGrid1.Cells[j,i]);
for i:=1 to 3 do
begin
for j:=1 to 2 do
for k:=j+1 to 3 do
if a[i,j]>a[i,k] then begin
m:=a[i,j];a[i,j]:=a[i,k]; a[i,k]:=m;end;
end;
for i:=1 to 3 do
for j:=1 to 3 do
StringGrid2.Cells[j,i]:=inttostr(a[i,j]);
end;

```



4-есеп. 2008 ж. Пекинде өткен олимпиада нәтижелері өрнектелген кесте берілген.

Мемлекет	Алтын	Күміс	Қола	Барлығы	Балл
Австралия	3	5	2		
Беларусия	2	3	4		
Великобритания	2	4	6		
Германия	6	7	9		
Италия	5	4	6		
Китай	10	8	6		
Корея	5	3	4		
Куба	6	5	3		
Казахстан	6	7	4		
Россия	8	6	5		
АҚШ	9	7	4		
Франция	4	6	3		
Япония	5	3	6		

Бағдарлама әр мемлекет бойынша медальдардың жалпы санын және оларға мына ереже бойынша балл беру керек. Әрбір алтын медаль үшін – 7, күміс медаль үшін – 6, қола медаль үшін – 5 балл командаға беріледі.

StringGrid1 компонентінің қасиеттерінің мәні

```
Name = Tabl  
ColCount = 6  
RowCount = 14  
FixedCols = 0  
FixedRows = 1  
Options.goEditing = True  
DefaultColWidth = 65  
DefaultRowHeight = 14  
GridLineWidth = 1
```

Кестенің бірінші жолын бағандардың тақырыбын жазу үшін қолданамыз. Яғни бірінші жолға мынадай мәліметтер енгіземіз: Мемлекет, Алтын, Күміс, Қола, Барлығы, Балл. Ал Мемлекет бағанына мына елдердің аттарын жазайық: Австралия, Белорусия, Великобритания, Германия, Италия, Қытай, Корей, Куба, Қазақстан, Россия, АҚШ, Франция, Япония.

```
unit Unit1;  
interface  
uses  
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,  
  Grids, StdCtrls;  
type  
  TForm1 = class(TForm)  
    Tabl: TStringGrid;  
    Button1: TButton;  
    Label1: TLabel;  
    procedure FormActivate(Sender: TObject);  
    procedure Button1Click(Sender: TObject);  
  private  
    { Private declarations }  
  public  
    { Public declarations }  
  end;  
var  
  Form1: TForm1;  
implementation  
{$R *.DFM}  
  
procedure TForm1.FormActivate(Sender: TObject);  
begin
```

```

tabl.Cells[0,0]:='Мемлекет';
tabl.Cells[1,0]:='Алтын';
tabl.Cells[2,0]:='Кумис';
tabl.Cells[3,0]:='Кола';
tabl.Cells[4,0]:='Барлығы';
tabl.Cells[5,0]:='Балл';
tabl.Cells[0,1]:='Австралия';
tabl.Cells[0,2]:='Белоруссия';
tabl.Cells[0,3]:='Великобритания';
tabl.Cells[0,4]:='Германия';
tabl.Cells[0,5]:='Италия';
tabl.Cells[0,6]:='Китай';
tabl.Cells[0,7]:='Корея';
tabl.Cells[0,8]:='Куба';
tabl.Cells[0,9]:='Казахстан';
tabl.Cells[0,10]:='Россия';
tabl.Cells[0,11]:='АКШ';
tabl.Cells[0,12]:='Франция';
tabl.Cells[0,13]:='Япония';
end;

procedure TForm1.Button1Click(Sender: TObject);
var
  c,r: integer; // кесте бағандары мен жолдарының номері
  s: integer; // командадағы медальдардың жалпы саны
  p: integer; // команда балы
  m: integer; // ең көп балл орналасқан жолдың номері
  buf: array[0..5] of string; // жолдарды алмастыру буфері
  i: integer; // жол нөмірі. Сорттау кезінде қолданылады
begin
  for r:=1 to tabl.rowcount do // барлық жолды өңдеу
  begin
    s:=0;
  // медальдардың жалпы санын есептеу
    for c:=1 to 3 do
      if tabl.cells[c,r] <> "
        then s:=s+StrToInt(tabl.cells[c,r])
        else tabl.cells[c,r]:='0';
  // балл санын есептеу
    p:=7*StrToInt(tabl.cells[1,r])+ 6*StrToInt(tabl.cells[2,r])+
      5*StrToInt(tabl.cells[3,r]);
  // нәтижені шығару
    tabl.cells[4,r]:=IntToStr(s); // барлық медальдар
    tabl.cells[5,r]:=IntToStr(p); // баллдар
  end;
end;

```

```

// Кестені балдардың кему реті бойынша сорттау
// Таңдау тәсілімен сорттау
for r:=1 to tabl.rowcount-1 do
begin
m:=r; // ең үлкен элемент r жолда
for i:=r to tabl.rowcount-1 do
if StrToInt(tabl.cells[5,i])>StrToInt(tabl.cells[5,m])
then m:=i;
if r <> m then
begin // кестенің r және m жолдарын алмастыру
for c:=0 to 5 do
begin
buf[c]:=tabl.Cells[c,r];
tabl.Cells[c,r]:=tabl.Cells[c,m];
tabl.Cells[c,m]:=buf[c];
end;
end;
end;
end;
end.

```

Алдымен бағдарлама әрбір мемлекет үшін медальдардың жалпы санын есептейді және сәйкес баллдарды шығарады. Сонан соң жай таңдау тәсілін қолданып бағдарлама алған баллдарының кему реті бойынша кестені сорттайды. Сорттау кезінде кестелердің жолдарының орнын ауыстыру үшін Buf жолдық массиві қолданылады, оның индексі кесте индексі сияқты нөлден беске дейін өзгереді.

Пекин олимпиадасына қатысқан мемлекеттердің тізімін спортшылардың алған медальдары мен жинаған балдарына қарай реттеу.

Мемлекет	Алтын	Күміс	Қола	Барлығы	Балл
Қытай	10	8	6	24	148
Германия	6	7	9	22	129
АҚШ	9	7	4	20	125
Россия	8	6	5	19	117
Қазақстан	6	7	4	17	104
Италия	5	4	6	15	89
Куба	6	5	3	14	87
Япония	5	3	6	14	83

Корытынды

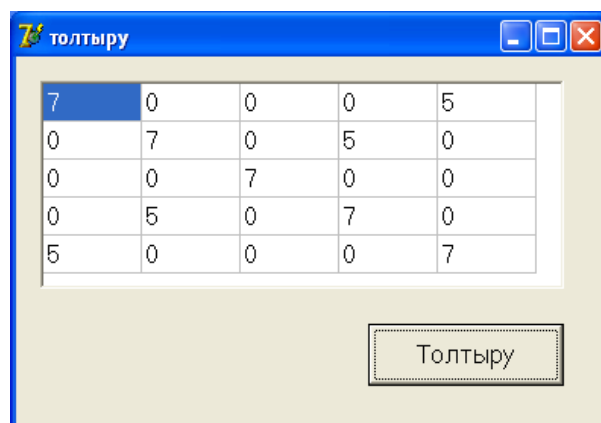
5-есеп. Квадрат массив элементтерінің бас диагоналын 7 санымен, ал кері диагоналын 5 санымен толтыру.

Формаға StringGrid1, Button1 компоненттерін орналастыру.

Толтыру батырмасын екі рет шертіп, бағдарлама коды терезесінде процедура денесін жазыңыздар.

```
procedure TForm1.Button1Click(Sender: TObject);
var i,j:integer;
begin
  for i:=0 to 4 do
  for j:=0 to 4 do
  begin
  if (i=j) then StringGrid1.Cells[i,j]:=inttostr(7)
  else begin
  if (j=4-i) then StringGrid1.Cells[j,i]:=inttostr(5)
  else StringGrid1.Cells[i,j]:=inttostr(0);
  end;
  end;
end;
```

Бағдарламаны орындап, «Толтыру» батырмасын шерту, сонда StringGrid1 алаңына нәтиже шығады.



6-есеп. 5x5 квадрат матрицасы берілген. Осы матрицаның әрбір жолының қосындысын табатын бағдарлама.

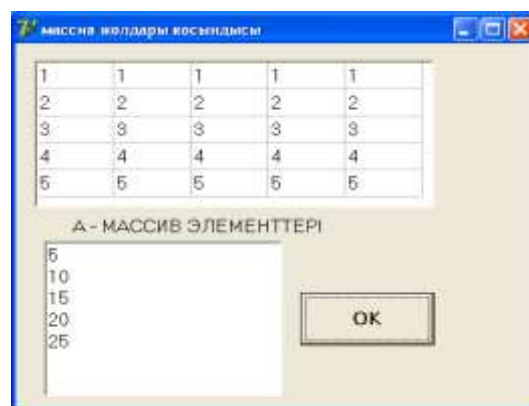
Формаға StringGrid1, Label1, Memo1, Button1 компоненттерін орналастырыңыздар.

«ОК» батырмасына екі рет шертіп, бағдарлама коды терезесінде процедура денесін жазыңыздар.


```

procedure TForm1.Button1Click(Sender: TObject);
const n=4;
var x:array[0..n,0..n] of integer;
    a:array[0..n] of integer;
    i,j:integer;
begin
  for i:=0 to n do
    for j:=0 to n do
      x[i,j]:=strtoint(StringGrid1.Cells[i,j]);
  for i:=0 to n do
    a[i]:=0;
  for i:=0 to n do
    for j:=0 to n do
      a[i]:=a[i]+x[j,i];
  for i:=0 to n do
    memo1.lines.add(inttostr(a[i]));
end;

```



Бағдарламаны орындап, «OK» батырмасын шерту, сонда Memo1 алаңына массивтің әрбір жолының қосындысы шығады.

Тапсырмалар:

1. $A[5,5]$ матрицаның бас диагонал бойындағы элементтердің көбейтіндісін және одан жоғары элементтердің 0-ден үлкендерінің қосындысын табыңдар.

2. $B[5,5]$ матрицасының тақ индексті жолдарының барлық элементтерін бас диагоналдың сәйкес элементтеріне бөліңдер, ал жұп индексті жолдарды көбейтіндер.

3. 5×5 квадраттық матрица берілген. Бас диагонал бойындағы және одан жоғары орналасқан элементтерді 0-ге алмастырыңдар.

4. $A[5,6]$ матрицасының жұп бағаналарының оң элементтерінің қосындысын және тақ бағаналарының теріс элементтерінің көбейтіндісін табыңдар, оларды сәйкесінше массивтерде сақтаңдар.

5. $A[5,5]$ матрицасы берілген, екі бір өлшемді массив құрып олардың біреуіне матрицаның жоғарғы үшбұрышының элементтерін, екіншісіне төменгі үшбұрышының элементтерін орналастырыңдар.

6. $A[5,5]$ матрицасы берілген. Бас диагоналдан төменгі элементтерді 0-ге, жоғарыдағыларын 3-ке және диагонал бойындағыларын 7-ге алмастырыңдар. $A[3,4]$ бүтін сандардан тұратын матрицаның 3-ке тең элементтерінің санын анықтаңдар. Егер ондай элемент болмаса, онда сәйкесінше хабар беріңдер.

Бақылау сұрақтары:

1. Екі өлшемді массив қандай жағдайда қолданылады және қалай ұйымдастыруға болады?

2. Екі өлшемді массив типіне қандай типтерді қолдануға болады?

3. Екі өлшемді массивтерді сипаттаудың қандай ерекшеліктері бар?
4. Екі өлшемді массив элементтерін қалай енгізуге болады?

Зертханалық жұмыс №11. Жазулар. Жиындар

Жұмыстың мақсаты: Delphi ортасында жазу, логикалық байланысқан бір типті объектілер жиынына бағдарлама құру дағдыларын қалыптастыру және өзіндік тапсырмаларды орындау.

Бағдарламалық дидактикалық қамтылуы: Дербес компьютерлер, Windows XP операциялық жүйесі, Borland Delphi 7 бағдарламасы, «Бағдарламалау» пәнінен оқу-әдістемелік кешен.

Жазулар – бұл өріс деп аталатын шектеулі компоненттер мөлшерінен тұратын мәліметтер құрылымы. Массивтерден айырмашылығы жазудың компоненттері (өрістері) әр түрлі типті болуы мүмкін. Жазудың қандай да бір компонентіне сілтеме жасау үшін өріске атау беріледі.

Жазуларды сипаттаудың форматы төмендегідей:

```
<тип аты> = record  
<өрістер тізімі>  
end;
```

мұндағы, <тип аты> - жазудың атауы; Record, end – қызметші сөздер (жазу, соңы дегенді білдіреді); <өрістер тізімі> - өрістер тізімі, ол нүктелі үтір арқылы ажыратылып жазылған жазулар бөлімдерінің тізбегі.

Жазудың әрбір бөлімі бір немесе бірнеше өріс атауларынан тұрады, олар бір-бірінен үтір арқылы ажыратылып жазылады. Атаулардан кейін қос нүкте қойылып, өрістер типі сипатталады.

Егер құрама атау пайдаланылатын болса, жазудың әр компонентіне кіруге мүмкіндік алуға болады. Яғни айнымалылардың атын көрсетіп, одан кейін нүкте қойылып, өріс аты жазылады.

Жазудың өрісіне кіруді жеңілдету үшін Width жалғастыру операторы пайдаланылады.

```
Width <айнымалы> do <оператор>;
```

мұндағы Width, do - қызметші сөздер; <айнымалы> - жазу типті айнымалы, мұнда бірінің ішіне бірі орналасқан өрістер тізімі жазылады; <оператор> - Object Pascal тілінің кез келген операторы .

Жиындар - бір-бірімен логикалық байланысқан бір типті объектілер жиыны. Жиынға кіретін элементтер саны 0 мен 256 аралығында өзгеруі мүмкін.

Жиын типтерінің жазылу форматы төмендегідей:

```
<тип аты> = set of <базалық тип>;
```

мұндағы, <тип аты> – жиын типінің атауы, set, of - қызметші сөздер, <базалық тип> - жиын элементтерінің базалық типі ретінде word, integer, LongInt типтерінен басқа кез келген реттелген тип алынады.

Жиын элементтерін беруде жиын конструкторы пайдаланылады: бір-бірінен үтір арқылы ажыратылған жиын элементтерінің тізімі, тізім квадрат жақшаның ішіне орналасады.

Жиындарға төмендегідей операцияларды қолдануға болады:

1) жиындардың қиылысуы - (*); екі жиынның қиылысу нәтижесінде алынған жиынға екі жиынның ортақ элементтері де кіреді.

2) жиындардың бірігуі (+). Екі жиынның қиылысуы бір мезгілде екі жиынға да кіретін ортақ элементтерден тұратын үшінші жиын болып табылады.

3) жиындардың айырымы. Екі жиынның айырымы екінші жиын элементтері кірмейтін, бірінші жиын элементтерінен тұратын үшінші жиын болып табылады.

Жиындарға қолданылатын амалдар

Жиындық типтердің берілгендері үшін қиылыстыру, біріктіру, айырымдық және in амалдары анықталған. Сондай-ақ, "<", ">", "<=", ">=", "=" және "<>" қатынас белгілері пайдаланылады.

"=" қатынас белгісі. Егер А және В жиындары бірдей элементтерден тұратын болса, онда бұл жиындар тең болып есептеледі. Салыстырылатын жиындардағы элементтердің орналасу ретінің теңдік шартына әсері болмайды.

А жиынының мәні	В жиынының мәні	Өрнек	Нәтиже
[1, 2, 3, 4]	[1, 2, 3, 4]	A=B	True
['a', 'b', 'c']	['c', 'a']	A=B	False
['a'..'z']	['z'..'a']	A=B	True

Тең емес қатынас белгісі ("<>"). А және В жиындарының кем дегенде бір элементі өзгеше болатын болса, онда олар тең емес болып есептеледі.

А жиынының мәні	В жиынының мәні	Өрнек	Нәтиже
[1, 2, 3]	[3, 1, 2, 4]	A<>B	True
['a'..'z']	['b'..'z']	A<>B	True
['c'..'t']	['t'..'c']	A<>B	False

Үлкен немесе тең қатынас белгісі (">="). Егер В жиынының барлық элементтері А жиынының құрамына кіретін болса, A>=B қатынасының нәтижесі ақиқат болады, ал кері жағдайда жалған болады.

А жиынының мәні	В жиынының мәні	Өрнек	Нәтиже
[1, 2, 3, 4]	[3, 2, 4]	A>=B	True
['a'..'z']	['b'..'t']	A>=B	True
['c', 'x', 't']	['t', 'c']	A>=B	True

Кіші неме тең "<=" қатынас белгісі. Егер А жиынының барлық элементтері В жиынының құрамына кіретін болса, онда A<=B өрнегінің нәтижесі ақиқат болады, кері жағдайда жалған болады.

А жиынының мәні	В жиынының мәні	Өрнек	Нәтиже
[1, 2, 3]	[2, 3, 4]	A<=B	True
['d'..'h']	['b'..'t']	A<=B	True
['c', 'x', 't']	['t', 'c']	A<=B	True

In операциясы. Бұл операция қандай да бір мәннің көрсетілген жиында жататындығын тексеру үшін пайдаланылады. Әдетте шартты операторда пайдаланылады.

A жиынының мәні	Өрнек	Нәтиже
2	if A in [2, 3, 4] then ...	True
'v'	if A in ['a'..'n'] then	False
X1	if A in [x0, x1, x2, x3]	True

In операциясы бірқатар басқа да амалдарды алмастыра отырып, күрделі шартты тексерулерді тиімді, көрнекі сипаттауға мүмкіндік береді. Мысалы,

If (a=1) or (a=2) or (a=3) or (a=4) or (a=5) then ... өрнегін қысқа түрде былай сипаттауға болады:

If A in [1..6] then...

Бұл операцияларға қосымша төмендегідей екі процедураны пайдалануға болады:

INCLUDE – жиынға жаңа элемент қосады. Процедураның шақырылуы:

INCLUDE(S,I)

мұндағы, S – TSETBASE базалық тип элементтерінен тұратын жиын; I - жиынға қосуға қажетті TSETBASE элементінің типі.

EXCLUDE – жиыннан элементті шығарады. Процедураның шақырылуы:

EXCLUDE(S,I). Шақыру параметрлері INCLUDE процедурасындағыдай

1-есеп. 1-ден 20-ға дейінгі бүтін сандар берілген. Осы сандардың ішінен 6-ға, 3-ке, 2-ге және 2 немесе 3-ке қалдықсыз бөлінетін сандарды анықтайтын бағдарлама.

Формаға нәтижені шығару үшін Memo1 компонентін және процедура денесін жазуға Button1 компоненттерін орналастырайық.

Компонент	Қасиеті	Меншіктелетін мән
Form1	Caption	Қалдықсыз бөлінетін сандар
	Color	clLime
Memo1	Lines	бос
	Font	Қаріп - MS Sans Serif; өлшемі – 12; қою көк; қарайтылған
	Color	clYellow
Button1	Caption	есепте

unit Unit1;

interface

uses

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,

```

Dialogs, StdCtrls;
type
  TForm1 = class(TForm)
    Button1: TButton;
    Memo1: TMemo;
    procedure Button1Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;
var
  Form1: TForm1;
implementation
{$R *.dfm}

procedure TForm1.Button1Click(Sender: TObject);
  Type num= set of 1..20;
  Var n2, n3, n6, n23: num;
      K: integer;
  Begin
  N2:=[];
  N3:=[];
  For k:=1 to 20 do
    Begin
  If k mod 2=0 then n2:=n2+[k];
  If k mod 3=0 then n3:=n3+[k];
  End;
  N6:=n2*n3;
  N23:=n2+n3;
  Memo1.Lines.Add('6-га
  калдықсыз бөлінетін сандар');
  For k:=1 to 20 do
  If k in n6 then
  memo1.Lines.Add(inttostr(k));
  Memo1.Lines.Add('2-ге
  немесе 3-ке калдықсыз бөлінетін сандар');
  For k:=1 to 20 do
  If k in n23 then memo1.Lines.Add(inttostr(k))
  End;
  end.

```



2-есеп. Топтағы 9 оқушының оқу үлгерімін орта балының кему реті бойынша сұрыптап шығаратын бағдарлама. Әрбір жазу оқушының атынан және үш пәннен алған бағасынан тұрады.

Жазулармен жұмыс істегенде экранға ақпаратты енгізу мен шығаруды StringGrid компоненті арқылы ұйымдастырған ыңғайлы. Ол үшін формаға StringGrid және Button компонентін орналастырамыз. Бағандар мен жолдарға сәйкес жолдарды енгізу үшін StringGrid компонентінің белгіленген аймағы пайдаланылады. Сондықтан объектілер инспекторы терезесіндегі FixedCols және FixedRows қасиеттеріне 1 мәнін тағайындаймыз. Есептің шартына сәйкес баған және жолдар санын көрсететін қасиеттерге: ColCount=6 және RowCount=10 мәндерін тағайындаймыз. StringGrid компонентіндегі жолдарды толық көру үшін вертикаль орағыш жолағын орнатуға болады, ScrollBars қасиетіне ssVertical мәнін тағайындаймыз. StringGrid компонентіне пернетақтадан ақпарат енгізіп, жөндеу мүмкіндігін алу үшін +Options қасиетіндегі goEditing жолына true мәнін беру қажет. Осы мәндер толық тағайындалып болғаннан кейін бағдарлама коды терезесінде FormCreate және ButtonClick оқиға өңдеуші процедураларын құрамыз.

```
unit Unit1;
interface
uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, Grids;
type
  TForm1 = class(TForm)
    StringGrid1: TStringGrid;
    Button1: TButton;
    procedure FormCreate(Sender: TObject);
    procedure Button1Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;
var
  Form1: TForm1;
implementation
{$R *.dfm}
  Type
  Zap=record
  Fio:string[20];
  informatika, bagdarlamalau, malimet: integer;
  Ortball:Real;
End;
```

```
Var mzap: array[1..9] of zap;
```

```
procedure TForm1.FormCreate(Sender: TObject);
```

```
Var i:Integer;
```

```
Begin with StringGrid1 do
```

```
Begin
```

```
Cells[0,0]:='№';
```

```
Cells[1,0]:='Фамилия';
```

```
Cells[2,0]:='Информатика';
```

```
Cells[3,0]:='Багдарламалау';
```

```
Cells [4,0]:='Малиметтер коры';
```

```
Cells[5,0]:='Орта балы';
```

```
For i:=1 to 9 do
```

```
Cells[0,i]:=intTostr(i);
```

```
Cells[1,1]:='Арзу'; Cells[2,1]:='5'; Cells[3,1]:='5'; Cells[4,1]:='5';
```

```
Cells[1,2]:='Улукманова'; Cells[2,2]:='4'; Cells[3,2]:='4' ;Cells[4,2]:='4';
```

```
Cells[1,3]:='Сейдахметова';Cells[2,4]:='3'; Cells[3,4]:='4'; Cells[4,3]:='4';
```

```
Cells[1,4]:='Хатенова';Cells[2,4]:='4'; Cells[3,4]:='3'; Cells[4,4]:='3';
```

```
Cells[1,5]:='Абдолла';Cells[2,5]:='5'; Cells[3,5]:='5'; Cells[4,5]:='4';
```

```
Cells[1,6]:='Кенжебаева';Cells[2,6]:='4'; Cells[3,6]:='4'; Cells[4,6]:='5';
```

```
Cells[1,7]:='Онгалыева';Cells[2,7]:='2'; Cells[3,7]:='3'; Cells[4,7]:='3';
```

```
Cells[1,8]:='Мусабаева';Cells[2,8]:='5'; Cells[3,8]:='4'; Cells[4,8]:='3';
```

```
Cells[1,9]:='Жангалиева';Cells[2,9]:='5'; Cells[3,9]:='5'; Cells[4,9]:='3';
```

```
For i:=1 to 9 do
```

```
With mzap[i] do
```

```
Begin
```

```
Fio:=Cells[1,i];
```

```
Informatika:=StrToInt(Cells[2,i]);
```

```
Bagdarlamalau:=StrToInt(Cells[3,i]);
```

```
Malimet:=StrToInt(Cells[4,i]);
```

```
Ortball:=( Informatika + Bagdarlamalau + Malimet)/3;
```

```
Cells[5,i]:=FloatToStrF(ortball,ffFixed,5,2);
```

```
End;
```

```
end;
```

```
end;
```

```
procedure TForm1.Button1Click(Sender: TObject);
```

```
Var I,j:integer; vper:zap;
```

```
Begin
```

```
For i:=1 to 9 do
```

```
With StringGrid1,mzap[i] do
```

```
begin
```

```
Fio:=Cells[1,i];
```

```
Informatika:=StrToInt(Cells[2,i]);
```

```
Bagdarlamalau:=StrToInt(Cells[3,i]);
```

```

Malimet:=StrToInt(Cells[4,i]);
Ortball:=( Informatika + Bagdarlamalau + Malimet)/3;
Cells[5,i]:=FloatToStrF(ortball,ffFixed,5,2);
End;
For i:= 2 to 9 do
For j:=9 downto 1 do
If mzap[j-1].ortball<mzap[j].ortball then
Begin
Vper:=mzap[j-1];
Mzap[j-1]:=mzap[j];
Mzap[j]:=vper;
End;
For i:= 1 to 9 do
With StringGrid1,mzap[i] do
begin
Fio:=Cells[1,i];
Informatika:=StrToInt(Cells[2,i]);
Bagdarlamalau:=StrToInt(Cells[3,i]);
Malimet:=StrToInt(Cells[4,i]);
Ortball:=( Informatika + Bagdarlamalau + Malimet)/3;
Cells[5,i]:=FloatToStrF(ortball,ffFixed,5,2);
End;
end;
end.

```

№	Фамилия	Информатика	Бағдарлама	Малиметтер	Орта балы
1	Арзу	5	5	5	5,00
2	Члукманова	4	4	4	4,00
3	Сейдахмето	3	4	4	3,67
4	Хатенова	4	3	3	3,33
5	Абдолла	5	5	4	4,67
6	Кенжебаева	4	4	5	4,33
7	Онгальева	2	3	3	2,67
8	Мусабаева	5	4	3	4,00
9	Жангалиева	5	5	3	4,33

Тапсырмалар:

1. Қалааралық автобекеттегі автобустар қозғалысының кестесін сақтайтын бағдарлама құрындар: рейс нөмірі, маршрут (бастапқы және соңғы пунктері), жүру мезгілі.

2. Топта 26 білімгер бар. Білімгерлердің аты-жөні және 7 сабақтан тапсырған емтихан бағасы бойынша әрбір білімгердің орта балын және ең жақсы білімгердің орта балын есептеп экранға шығаратын бағдарлама құрындар.

3. Мекеменің автомобильдері жөніндегі мәліметтер (марка, түсі, шыққан жылы, сапасы, құны) сақтап, жасалған сұрау бойынша мәліметтерді беретін бағдарлама құрындар.

4. “Білімгер” деп аталатын жазу типті мәлімет берілген. Төмендегідей ақпаратты өңдейтін бағдарлама құрындар.

а) Оқу жылы бойынша орташа оқу үлгерімін есептеу;

ә) Үлгермейтін білімгерлердің тізімін шығару;

б) Тізімді алфавиттік ретпен реттеу.

5. “Білімгер” деп аталатын жазу типті мәлімет берілген. Төмендегідей ақпаратты өңдейтін бағдарлама құрындар.

а) Бірінші семестрде қанша білімгердің балы 24-тен жоғары;

ә) Екінші семестрде қанша студенттің балы 22-ден төмен;

б) Тізімді баллдың кемуі бойынша реттеу.

Бақылау сұрақтары:

1. Жазулар дегеніміз не және олар қалай сипатталады?

2. Width_do операторының қызметі қандай?

3. Жиын дегеніміз не?

4. Жиынға кіретін элементтер санының өзгеру аралығы қандай?

5. Жиындарға қандай операциялар қолданылады және қандай қатынас белгілері қолданылады?

Зертханалық жұмыс №12. Процедуралар мен функциялар

Жұмыстың мақсаты: Студенттерді процедуралар мен функциялар және параметрлермен таныстырып, олармен есептер шығарып үйрену дағдыларын қалыптастыру.

Бағдарламалық дидактикалық қамтылуы: Дербес компьютерлер, Windows XP операциялық жүйесі, Borland Delphi 7 бағдарламасы, «Бағдарламалау» пәнінен оқу-әдістемелік кешен.

Процедуралар мен функциялар салыстырмалы түрде берілген атауы бойынша шақырылып орындалатын бағдарламаның бөлігі болып табылады. Бағдарлама мәтініндегі процедура (функция) атауының кездесуі оның шақырылуы деп аталады. Функцияның процедурадан айырмашылығы функция денесін құрайтын операторлардың орындалу нәтижесінен алынған мән біреу ғана болады және ол функция атауына меншіктеледі. Функция мен процедураны жалпы "қосалқы бағдарлама" деп атайды.

Қосалқы бағдарлама орындалмас бұрын негізгі бағдарламаның бөлімінде сипатталуы тиіс.

Қосалқы бағдарламаны сипаттау дегеніміз оның тақырыбы мен денесін көрсету. Тақырыбында бағдарламаның аты мен формальды параметрлер тізімі хабарланады. Ал, функция үшін, мұнымен қатар оның атауына меншіктелетін мәннің типі көрсетіледі. Қосалқы бағдарламаның тақырыбынан кейін оның бейнелеу бөлімі мен орындалатын операторлар бөлімі жазылады. Қосалқы бағдарламаның бейнелеу бөлімінде төменгі деңгейдегі қосалқы бағдарлама кездесуі мүмкін.

Кез келген қосалқы бағдарлама өзін-өзі шақыра алады, шақырудың бұл тәсілі рекурсия деп аталады.

Процедураның тақырыбының жазылу түрі төмендегідей:

Procedure <аты> (<формальды параметрлер тізімі>);

Функцияның тақырыбы

Function <аты> (<формальды параметрлер тізімі>) : <тип>

Мұндағы <аты> - қосалқы бағдарламаның аты; <формальды параметрлер тізімі> - формальды параметрлер тізімі; <тип> - функция нәтижесінің алатын типі. Қосалқы бағдарламаның тақырыбынан кейін ASSEMBLER, EXTERNAL, FAR, FORWARD, INLINE, INTERRUPT, NEAR тәрізді стандартты директиваларының бірі келуі тиіс.

Бұл директивалар өзі жазылып отырған қосалқы бағдарлама үшін орындалады, бірақ өзі орналасқан қосалқы бағдарламадан кейінгі қосалқы бағдарлама үшін орындалмайды.

ASSEMBLER - бұл директива орындалғанда процедураға кірер және шығар алдындағы орындалатын стандартты машиналық нұсқаулардың тізбегінің жұмысын (өзгертеді) тоқтатады. Бұл жағдайда қосалқы бағдарламаның денесі орнатылған Ассемблер командаларының көмегімен орындалады.

EXTERNAL - бұл директиваның көмегімен сыртқы қосалқы бағдарлама хабарланады.

FAR - компилятор шақырудың шеткері моделіне есептелген қосалқы бағдарлама кодын құруы тиіс.

NEAR - директивасы бойынша компилятор жадының жақындағы моделіне есептелген кодын құруды міндеттейді.

FORWARD - директивасы компиляторға бағдарлама мәтінінен ары қарай қосалқы бағдарлама кездесетінін хабарлайды, бірақ ағымдағы бағдарламалық модуль шеңберінде.

INLINE - директивасы қосалқы бағдарламаның денесі орнатылған машиналық нұсқаулар көмегімен орындалатынын көрсетеді.

INTERRUPT - үзу, тоқтатуды өңдеу процедураларын құруда пайдаланылады.

Қосалқы бағдарламада формальды параметрлер тізімнің болуы міндетті емес және болмауы да мүмкін.

Object Pascal тілінде қосалқы бағдарламаны шақырған кезде формальді параметрлердің типі мен мөлшері шынайы параметрлердің типі мен мөлшеріне сәйкес келуі тиіс. Пайдаланып отырған шынайы параметрлердің мәні қосалқы бағдарламаны шақырғанда оның қандай

ретпен орналасуына тәуелді. Программист қосалқы бағдарламаны шақырғанда шынайы параметрлердің дұрыс орналасу ретін қадағалауы тиіс.

1-есеп. Сатып алатын заттың бағасын, санын енгізіп, төленетін ақшаны есептейтін бағдарлама.

```
unit sat_1;
interface
uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls;
type
  TForm1 = class(TForm)
    Edit1: TEdit;
    Edit2: TEdit;
    Label1: TLabel;
    Label2: TLabel;
    Button1: TButton;
    Label3: TLabel;
    procedure Button1Click(Sender: TObject);
    procedure Edit2KeyPress(Sender: TObject; var Key: Char);
    procedure Edit1KeyPress(Sender: TObject; var Key: Char);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form1: TForm1;
implementation
{$R *.dfm}

// қосалқы бағдарлама
procedure Summa;
var
  cena: real;    // бағасы
  kol: integer; // саны
  s: real;       // төленетін ақша
  mes: string[255];
begin
  cena := StrToFloat(Form1.Edit1.Text);
  kol := StrToInt(Form1.Edit2.Text);
  s := cena * kol;
  if s > 500 then
```

```

begin
    s := s * 0.9;
    mes := ' 10 % жеңілдік бар ' + #13;
end;
mes := mes+ 'сатып алынған заттың төленетін ақшасы: '
    + FloatToStrF(s,ffFixed,4,2) +' тенге.';
Form1.Label3.Caption := mes;
end;

procedure TForm1.Button1Click(Sender: TObject);
begin
    Summa;
end;

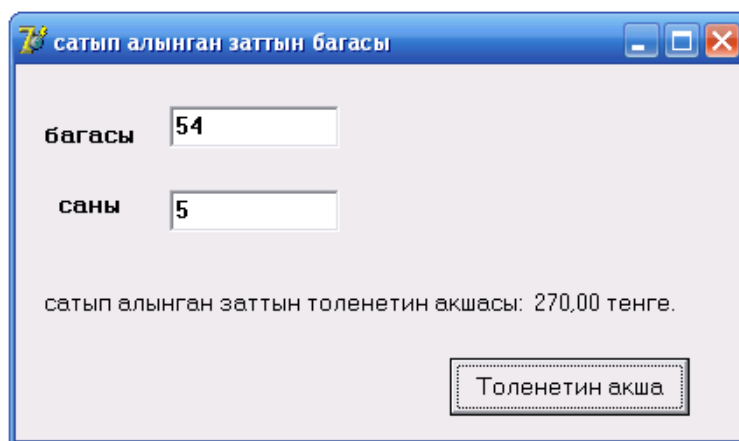
procedure TForm1.Edit2KeyPress(Sender: TObject; var Key: Char);
begin
    case Key of
        '0' .. '9',#8: ;
        #13:    Summa;
        else Key := Chr(0);
    end;
end;

procedure TForm1.Edit1KeyPress(Sender: TObject; var Key: Char);
begin
    case Key of
        '0' .. '9', #8 :    ;
        #13: Form1.Edit2.SetFocus; // <Enter>

        ',', ';':
            begin
                if Key = ','
                    then Key:=',';
                if Pos(',',Edit1.Text) <> 0
                    then Key:= Chr(0);
            end;
        else
            Key := Chr(0);
    end;
end;
end;

end.

```



2-есеп. Үш бүтін санның көбейтіндісін есептейтін процедура жазып, оны негізгі бағдарламада шақырып, нәтижесін экранға шығару.

Формаға нәтижені шығару үшін Memo1 компонентін және процедура денесін жазуға Button компоненттерін орналастырайық.

Компонент	Қасиеті	Меншіктелетін мән
Form1	Caption	Үш бүтін санның көбейтіндісі
	Color	clLime
Memo1	Lines	бос
	Font	Қаріп - MS Sans Serif; өлшемі – 18; қызыл; қарайтылған
Button1	Caption	есептеу
Button2	Caption	шығу

Есептеу батырмасын екі рет басып, бағдарлама коды терезесіне өту. Сонда OnClick оқиғасы процедурасының дайындамасы шығады.

```

unit Unit1;
interface
uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls;
type
  TForm1 = class(TForm)
    Memo1: TMemo;
    Button1: TButton;
    Button2: TButton;
  procedure Button1Click(Sender: TObject);
  procedure Button2Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

```

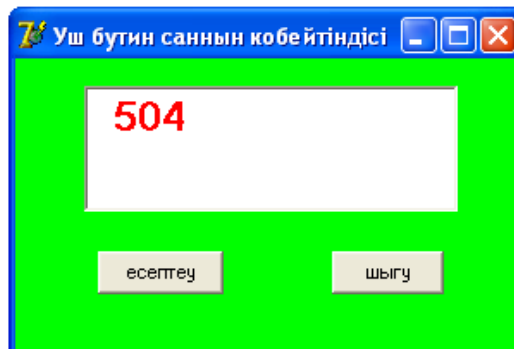
```

end;
var
  Form1: TForm1;
implementation
{$R *.dfm}

procedure TForm1.Button1Click(Sender: TObject);
var
  m:integer;
procedure kob(a,b,c:integer; var nat: integer);
begin
  nat:=a*b*c;
end;
begin
  kob(7,6,12,m);
  memo1.Lines.Add(Inttostr(m));
end;

procedure TForm1.Button2Click(Sender: TObject);
begin
  form1.close
end;
end.

```



3-есеп. $Z = \frac{P^5 + P^{-5}}{2P^m}$ өрнегін есептейтін бағдарлама

```

unit Unit1;
interface
uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls;
type
  TForm1 = class(TForm)
    Memo1: TMemo;
    Button1: TButton;
    Edit1: TEdit;
    Edit2: TEdit;
    procedure Button1Click(Sender: TObject);
  private

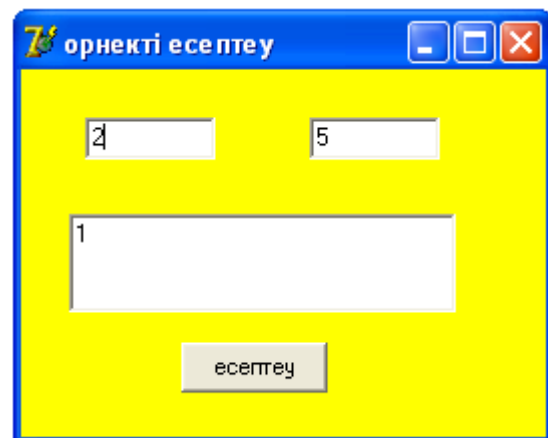
```

```

    { Private declarations }
public
    { Public declarations }
end;
var
    Form1: TForm1;
implementation
{$R *.dfm}

procedure TForm1.Button1Click(Sender: TObject);
var M:integer;
P,Z,R:real;
Function Step(N:integer; x:real):real;
var i:integer;
y:real;
begin
Y:=1;
for i:=1 to N do
y:=y*y;
Step:=y;
end;
begin
M:=StrToInt(Edit1.text);
P:=StrToInt(Edit2.text);
Z:=Step(5,P)+Step(5,1/p);
if M=0 then R:=1 else if M>0 then R:=Step(M,P)
else R:=Step(-M,1/P);
Z:=Z/2*R;
Memo1.Lines.Add(floattostr(Z));
end;
end.

```



4-есеп. Студенттердің тегі мен атынан тұратын тізім құратын бағдарлама.

Бағдарламаға қажетті компоненттер: Label1; Label2; Label3; Edit1; Edit2; Button1; Button2.

```

type
TPStudent=^TStudent;
TStudent = record
    f_name:string[20];
    l_name: string[20];
    next: TPStudent;
end;

```

```

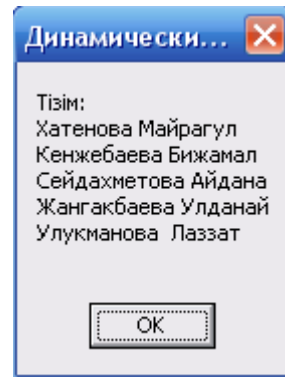
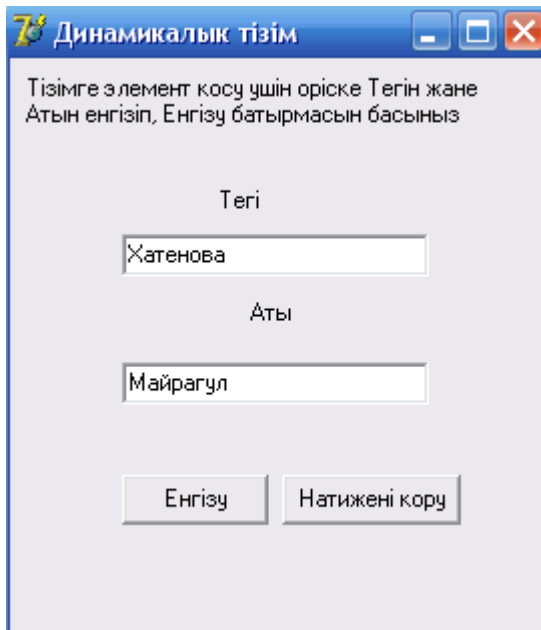
var
  head: TPStudent;

procedure TForm1.Button1Click(Sender: TObject);
var
  curr: TPStudent;
begin
  new(curr);
  curr^.f_name := Edit1.Text;
  curr^.l_name := Edit2.Text;
  curr^.next := head;
  head := curr;
  Edit1.text:="";
  Edit2.text:="";
end;

procedure TForm1.Button2Click(Sender: TObject);
var
  curr: TPStudent;
  n:integer;
  st:string;
begin
  n := 0;
  st := "";
  curr := head;
  while curr <> NIL do
    begin
      n := n + 1;
      st := st + curr^.f_name + ' ' + curr^.l_name + #13;
      curr := curr^.next;
    end;
  if n <> 0
  then ShowMessage('Tizim:' + #13 + st)
  else ShowMessage("");
end;

end.

```

5-есеп. Object Pascal-да шаманың кез келген дәрежесін есептеу стандартты функциясы қарастырылмаған. Бірақ осы есепті стандартты \ln және \exp функциялары арқылы шешуге болады, яғни $x^y = \exp(y * \ln(x))$, мұнда x, y - кез келген нақты сандар.

Кез келген санның дәрежесін есептеуді процедураның құрамындағы ішкі программаны Power функциясы түрінде анықтайық.

```

unit Unit1;
interface
uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, Buttons;
type
  TForm1 = class(TForm)
    Edit1: TEdit;
    Edit2: TEdit;
    Memo1: TMemo;
    Button2: TButton;
    Label1: TLabel;
    Label2: TLabel;
    Label3: TLabel;
    Label4: TLabel;
    BitBtn1: TBitBtn;
  procedure Button2Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

```

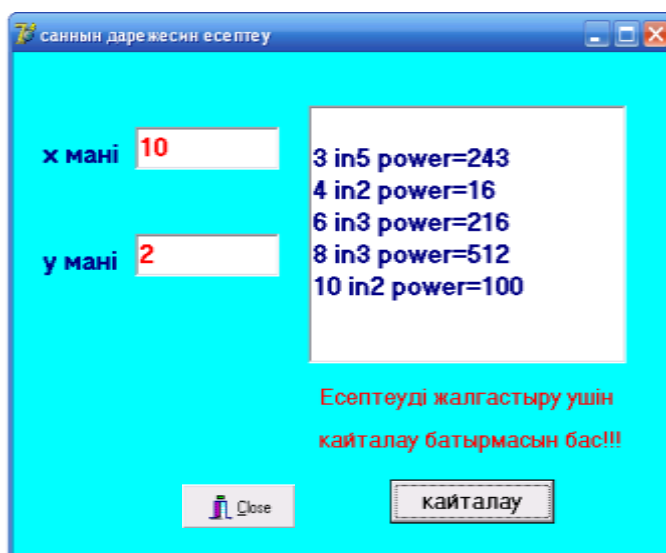
```

var
  Form1: TForm1;
implementation
{$R *.dfm}

procedure TForm1.Button2Click(Sender: TObject);
var x,y,s: real;      //негізгі процедураның айнымалылары
Function Power(a,b: real): real;      // бейстандартты функция
Begin  { Power}
  If a> 0 then      // егер а мәні нөлден үлкен болса,
    Power := exp(b*ln(a))      //онда бірінші жолымен есептеледі
  Else if a<0 then      // егер а мәні нөлден кіші болса,
    Power := exp(b*ln(abs(a))) // онда екінші жолымен есептеледі
  Else if b=0 then
    Power :=1
  Else Power:=0;
End;      { Power }
Begin { main}
  Button2.Show;
  x:=StrToInt(edit1.text);      // x мәнін енгізу
  y:=StrToInt(edit2.text);      // y мәнін енгізу
  s:=power(x,y);      // xy мәнін s-ке меншіктеу
  memo1.lines.add(FloatToStr(x)+ ' in'+
FloatToStr(y)+ ' power=' + Floattostr(s)); // нәтижесін шығару
  BitBtn1.hide;
  Button2.Enabled:=True;
end; { main}

end.

```



Тапсырмалар:

1. Процедура мен функцияны пайдаланып төмендегі есепті шешіндер:

$$C_n^m = \frac{n!}{m!(n-m)!}$$

2. Процедура мен функцияны пайдаланып төмендегі есепті шешіндер:

$$F = m! + (m-k)! + k!$$

3. C(5,5) және B(6,6) матрицаларының әр жолының элементтерінің көбейтіндісін есептендер.

4. Қосалқы бағдарлама функцияны қолдана отырып функцияның мәнін есептендер:

$$Z = \frac{1}{45} \left(\sum_{j=1}^{15} \sin^2 A(j) + \sum_{j=1}^{20} \sin^2 B(j) \right) \text{ мұндағы } A(j) \text{ және } B(j) \text{ массив элементтері.}$$

5. Қосалқы бағдарлама функцияны пайдалана отырып A[3,6], B[5,10] массивтерінің оң элементтерінің қосындысын және олардың санын табыңдар.

6. Функция мәндерінің кестесін құрыңдар:

$$y = \sin x + \frac{3x}{6 + 0,3x^2}; \quad x \in [-3, 8], \quad n = 20$$

Бақылау сұрақтары:

1. Қосалқы бағдарлама деп қандай бағдарламаны айтады?

2. Қосалқы бағдарламалар қалай шақырылады?

3. Процедураның жазылу түрі. Процедура түрінде жазылған қосалқы бағдарламаның ерекшелігін көрсетіндер?

4. Процедураны параметрсіз жазуға бола ма?

5. Функцияның жазылу түрі.

6. Функция түрінде жазылған қосалқы бағдарламаның ерекшелігін көрсетіндер?

7. Функция мен процедураның шақырылу және мәндер беру ерекшелігі неде?

8. Функция түрінде жазылған қосалқы бағдарламаның процедура-бағдарламадан ерекшелігі.

9. Процедура мен функция арасында қандай ұқсастық және қандай айырмашылықтар бар?

10. Формальды параметрлер дегеніміз не?

11. Object Pascal тілінде қанша процедуралық тип бар?

12. GetArrayAverage функциясының қызметі қандай?

13. Рекурсия дегеніміз не және ол қалай ұйымдастырылады?

Зертханалық жұмыс №13. Delphi-дің графикалық және мультимедиалық мүмкіндіктері

Жұмыстың мақсаты: Delphi ортасында әр түрлі суреттер салу, қозғалысқа келтіру, дыбысы бар анимациялар шығару дағдыларын қалыптастыру

Бағдарламалық дидактикалық қамтылуы: Дербес компьютерлер, Windows XP операциялық жүйесі, Borland Delphi 7 бағдарламасы, «Бағдарламалау» пәнінен оқу-әдістемелік кешен.

Delphi ортасында графикалық құралдарды пайдалануды жеңілдететін арнайы кластар қарастырылған: контекст үшін - Tcanvas, қаріп үшін - Tfont, қалам үшін – Tpen, қылқалам үшін - Tbrush. Осы кластарға байланысты объектілер барлық көрінетін элементтер үшін автоматты түрде құрылып, бағдарламада - canvas, font, pen және brush қасиеттері арқылы жүзеге асырылады.

Tfont класының көмегімен кез келген графикалық құрылғы (экран, принтер, плоттер және т.б.) үшін қаріп объектісі құрылады.

Tpen класының көмегімен сызықтарды сызуға арналған қалам объектісі құрылады.

Tbrush класының объектілері тұйық фигуралардың ішін толтыру қызметін атқарады.

Tcanvas класы әр түрлі құралдармен - қалам, қылқалам, қаріп - сурет салынатын кенепті құруға мүмкіндік береді. Tcanvas класының объектілері барлық көрінетін компоненттер үшін автоматты түрде құрылады. Аталған класстардың бірнеше қасиеттері бар.

Canvas қасиеті. Бағдарлама графиканы объектінің бетіне (форма немесе Image компоненті) шығарады. Объектінің бетіне графикалық элементті шығару үшін осы объектінің Canvas қасиетіне сәйкес әдісті қою керек. Мысалы, Form1.Canvas.Rectangle (10,10,100,100) - бағдарлама терезесінде тіктөртбұрыш салады.

Кенеп. Графиканы бағдарлама бетіне шығаратын қасиет - Canvas қасиеті. Өз кезегінде Canvas қасиеті - бұл Canvas типті объект. Бұл типтің әдістері графикалық примитивтерді (нүктелер, сызықтар, тіктөртбұрыштар) шығаруды қамтамасыз етеді, ал қасиеті бұл примитивтердің сипаттамасын (түсі, сызықтардың ені және стилі, қаріпі, т.б.) бере алады.

Графикалық примитивтерді шығару үшін әдістерінің Canvas қасиетін сурет сала алатын абстрактілі бет ретінде қарастырылады. Бет бөлек нүктелерден – пиксельдерден тұрады. Пиксельдің орны оның горизонталь (X) және вертикаль (Y) өстері арқылы сипатталынады. Жоғарғы сол жақ пикселдің координатасы (0,0). Координаттар жоғарыдан төмен және солдан оңға қарай өседі. Беттің өлшемін Height және Width қасиеттері арқылы өзгертуге болады.

Қалам. Қалам нүктелерді, сызықтарды, геометриялық фигураларды: шеңбер, тіктөртбұрыш, эллипс, және т.б. суреттерді салу үшін қолданылады. Төменгі кестеде Tpen объектінің қасиеттері көрсетілген.

Қасиет	Анықтамасы
Color	Сызық түсі
Width	Сызық ені
Style	Сызық түрі

Color қасиеті қаламмен салынған сызықтың түсін анықтайды. Келесі кестеде Color қасиетінің мәні ретінде пайдалануға болатын белгілі тұрақтылар берілген.

Тұрақты	Түсі	Тұрақты	Түсі
clBlack	Қара	clWhite	Ақ
clGreen	Жасыл	clAqua	Көкшіл
clNavy	Көк қара	clRed	Қызыл
clPurple	Ақшыл	clBlue	Көк
clTeal	Көк жасыл	clSilver	Жылтыр сұр
clGray	Сұр	clMaroon	Каштан

Style қасиеті сызықтың стилін көрсетеді. Ол түзу, штрихталған немесе ирек болуы мүмкін. Төменгі кестеде сызықтың стилін көрсететін тұрақтылар келтірілген. Жартылай пунктир сызықтың ені 1-ден кем болуы мүмкін емес. Pen.Width қасиетіне бірден үлкен мән берсек, онда жартылай пунктир сызығы жай сызық болып шығады.

Тұрақты	Сызық түрі
PtSolid	Жай сызық
PtDash	Пунктир, ұзын штрих
PtDot	Пунктир, қысқа штрих
PtDashDot	Пунктир, ұзын және қысқа штрихпен аралас
PtDashDotDot	Пунктир, бір ұзын бір қысқа штрих аралас
PtClear	Сызық көрінбейді

Mode қасиеті нүкте түсі мен бет түсінің үйлесімділігін анықтайды. Келісім бойынша сызық Pen.Color қасиетінің мәнімен сызылады.

Бірақ, программист фон түсіне байланысты сызық түсін өзгерте алады. Төменгі кестеде Pen.Mode қасиетінің мәні ретінде қолданылатын тұрақтылар сипатталған.

Тұрақты	Сызық түсі
MrBlack	Қара, Pen.Color қасиетінің мәнінен тәуелсіз
MrWhite	Ақ, Pen.Color қасиетінің мәнінен тәуелсіз
MrCopy	Сызық түсі Pen.Color қасиетінің мәніне тәуелді
MrNotCopy	Сызық түсі Pen.Color қасиетінің мәніне инверсиялы
MrNot	Сызық түсі бет нүктелерінің түсіне инверсты

Қылқалам. Қылқалам тұйықталған бөліктерді салу әдісімен, мысалы геометриялық фигураларды салумен сипатталады. Қылқалам, объект ретінде төменгі кестеде келтірілген екі қасиетке ие.

Қасиет	Сипаттама
Color	Тұйықталған бөлікті салу түсі
Style	Бөлікті толтыру стилі

Контур ішіндегі бөлік боялған немесе штрихталған болуы мүмкін. Бөлікті штрихтаудың мынадай түрлері бар:

Тұрақты	Бөлікті толтыру типі
BsSolid	Жай
BsClear	Бөлік боялмайды
BsHorizontal	Горизонтал штрих
BsVertical	Вертикал штрих
BsFDiagonal	Сызықтары алға бағытталған диагоналды штрих
BsBDiagonal	Сызықтары төмен бағытталған диагоналды штрих
BsCross	Торлы горизонтал – вертикал бағыттағы штрих
BsDiagCross	Торлы диагоналды штрих

Мәтінді шығару. Графикалық объектінің бетіне мәтінді шығару үшін TextOut командасы қолданылады. Оның жалпы түрі:

Объект.Canvas.TextOut(x,y, мәтін)

Мұндағы, объект – мәтін шығаратын объект аты; x, y – мәтін шығаратын графикалық беттің координатасы; мәтін – символ типті айнымалы немесе тұрақты.

Мәтінді шығару үшін қолданылатын қаріп Canvas объектісіне Font қасиеті арқылы анықталады. Төменгі кестеде Tfont объектісінің қасиеттері келтірілген.

Қасиет	Сипаттамасы
Name	Қолданылатын қаріп мәні ретінде қаріптің атын көрсеткен жөн.
Size	Қаріп өлшемі
Style	Символдардың стилі. Ол әр түрлі болуы мүмкін (fsbold, fsItalic және т.б.).
Color	Символ түсі

Графикалық примитивтерді салу әдістері.

Түзу. Түзу сызықты салу әдісі LineTo командасы арқылы орындалады. Оның жалпы түрі:

Компонент. Canvas.LineTo(x,y);

LineTo әдісі түзу сызықты қаламның ағымдағы орнынан координатасы көрсетілген нүктеге дейін салады. Түзудің бастапқы нүктесін қаламды графикалық беттің керекті нүктесіне жылжыту арқылы орындауға болады. Оны MoveTo параметрі арқылы беруге болады.

Графикалық беттегі түзудің түсі Pen объектісінің қасиеті арқылы беріледі.

Сынық сызық. Polyline әдісі сынық, кесінділер сызады. Параметр ретінде бұл әдіс Troint типті массивті алады. Массивтің әр элементі жазу ретінде қарастырылады, x және y өрістері иілу нүктелерінің координаттары.

Шеңбер және эллипс. Ellips әдісі берілген параметрге байланысты эллипс немесе шеңбер сызады. Оның жалпы түрі:

Объект. Canvas.Ellips(x1,y1,x2,y2)

Мұндағы, объект - сызу орындалатын беттің объектісі; x1, y1, x2, y2 – ішінде шеңбер салынатын тіктөртбұрыштың координаттары.

Доға. Доғаны салу Arc әдісі арқылы орындалады. Оның жалпы түрі:

Объект. Canvas.Arc(x1,y1,x2,y2,x3,y3,x4,y4)

Мұндағы, x1, y1, x2, y2 – эллипсті анықтайтын параметрлер; x3, y3 – доғаның бастапқы нүктесін анықтайтын параметр; x4, y4 – доғаның соңғы нүктесін анықтайтын параметр;

Бастапқы нүкте – бұл эллипс және түзу шекараларының қиылысу нүктесі. Бұл шекара эллипс центрінен (x3, y3) нүктесіне дейінгі кесінді.

Доғаның түсі, ені, түзу стилі беттегі Pen қасиетімен анықталады.

Тіктөртбұрыш. Тіктөртбұрыш Rectangle әдісі арқылы салынады. Оның жалпы түрі:

Объект. Canvas.Rectangle (x1, y1, x2, y2)

Мұндағы, объект – объект немесе компонент аты; x1, y1 және x2, y2 - тіктөртбұрыштың жоғарғы сол жақ және төменгі оң жақ координаттары.

Кейде Объект. Canvas.Rectangle (x1, y1, x2, y2, x3, y3) болады. Бұл кезде тіктөртбұрыштың бұрыштары тік немесе доғал болып келеді.

Мұндағы, x3, y3 – эллипс түрінде келетін тіктөртбұрыш бұрышының өлшемі.

Көпбұрыш. Көпбұрыш Polygon әдісі арқылы салынады. Polygon әдісі координаталары массивте орналасқан нүктелерді түзу арқылы қосады.

Сектор. Pie әдісі арқылы эллипс немесе шеңбердің секторын салады. Оның жалпы түрі:

Объект. Canvas.Pie (x1, y1, x2, y2, x3, y3, x4, y4)

Мұндағы, x1, y1, x2, y2 – эллипстің бөлшегі болатын сектордың параметрлері; x3, y3, x4, y4 – сектордың шекаралары болатын түзулердің соңғы координата нүктелері. Сектор сағат тіліне қарсы қиылады.

Суретті шығару. Заты .bmp, .jpg, .ico болып келген графикалық файлдарды шығару үшін Additional бетіндегі Image компоненті қолданылады.

Image компонентінің қасиеттері:

Қасиеті	Анықтамасы
Picture	Компонент өрісінде бейнеленетін сурет
Width, Height	Компонент өлшемі
AutoSize	Өлшемнің автоматты түрде өзгертілу белгісі
Stretch	Компонент өлшеміне байланысты сурет өлшемін өзгерту белгісі
Visible	Компонент үстінде суреттің көріну, көрінбеу белгісі

Delphi-дің мультимедиялық мүмкіндіктері. Delphi ортасында мультимедиялық бағдарламалар жасау үшін екі компоненттің қызметін пайдалануға болады:

1) *Animate* - жай анимацияны шығаруға мүмкіндік береді.

2) *MediaPlayer* - аса қиын есептерді шешуге мүмкіндік береді. Мысалы, дыбысты, бейнероликтерді, дыбысы бар анимацияны шығару және т.б.

Animate компоненті формаға әдеттегідей орнатылады. Қосқаннан кейін оның сәйкес қасиеттерін тағайындау керек. *Animate* компонентінің қасиеттері:

Қасиет	Анықтамасы
Name	Компонент аты
FileName	Анимация орналасқан AVI файлының аты
StartFrame	Анимация басталатын кадр нөмірі
StopFrame	Анимация бітетін кадр нөмірі
Activate	Анимация кадрларын активизациялау белгісі
Color	Анимация жасалатын фон түсі
Transparent	мөлдір түс тәртібі
Repetitions	Анимацияның қайталану саны

Animate компоненті тек кана AVI-файлдарын ұйымдастыру үшін ғана қолданылады.

MediaPlayer компоненті бейнероликтерді және дыбысы бар анимацияларды орындауға мүмкіндік береді.

Компонентті формаға қосу нәтижесінде формада батырмалар тобы пайда болады. Бұл батырмалардың мәні төмендегі кестеде келтірілген:

Батырма	Белгілеу	Жұмысы
Шығару	<i>BtPlay</i>	Видео немесе дыбысты шығару
Пауза	<i>BtPause</i>	Тоқтату
Стоп	<i>BtStop</i>	Тоқтату
Келесі	<i>BtNext</i>	Келесі кадрға көшу
Алдыңғы	<i>BtPrev</i>	Алдыңғы кадрға көшу
Қадам	<i>BtStep</i>	Келесі дыбыстық файлға көшу

Артқа	VtBack	Өткен дыбыстық файлға көшу
Жазу	VtRecord	Жазу
Ашу/Жабу	VtEject	Компьютер дисководын ашу немесе жабу

Media Player компонентінің қасиеттері:

Қасиет	Анықтамасы
Name	Аты
DeviceType	Тип
FileName	Бейнеролик немесе дыбыстық фрагменті бар файл аты
AutoPen	Бағдарламаны жүктеген соң автоматты ашылу белгісі
Display	Бейнеролик орналасқан компонентті анықтайды
VisibleButtons	Компоненттердің көрінетін батырмаларын анықтайды

Дыбысты шығару. Дыбыстық фрагменттер заты .WAV болып келген файлдарда орналасқан. Delphi-де дыбысты шығару үшін де Media Player компоненті қолданылады.

1-есеп. Форма бетіне Ақтау сөзін және іші қызыл түспен боялған эллипс, іші жасыл түспен боялған төртбұрыш, іші күлгін түспен боялған үшбұрыш салатын бағдарлама.

```

unit Unit1;
interface
uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls;
type
  TForm1 = class(TForm)
    Button1: TButton;
    procedure Button1Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;
var
  Form1: TForm1;
implementation
{$R *.dfm}

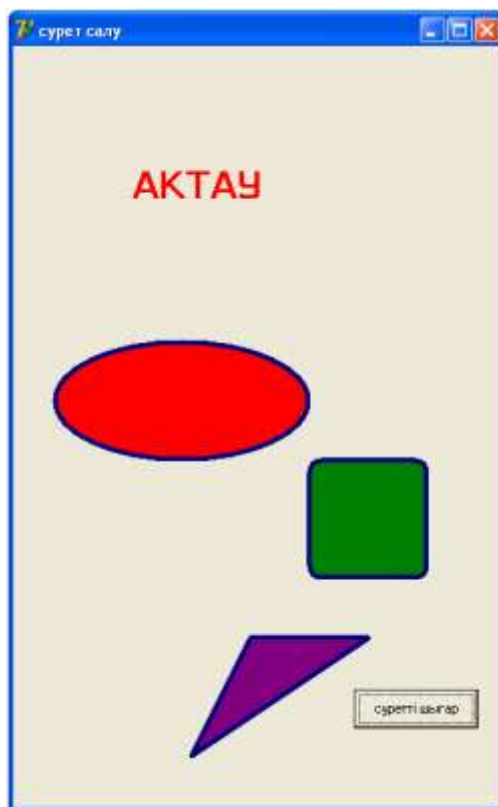
procedure TForm1.Button1Click(Sender: TObject);
begin

```

```

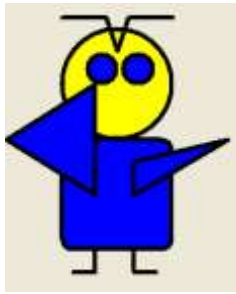
With Form1.Canvas do begin
TextOut(100,100,'АКТАУ');
Pen.Color:=clNavy;
Pen.Width:=4;
Brush.Style:=bssolid;
Brush.Color:=clRed;
Ellipse(250,250,35,350);
Brush.Style:=bssolid;
Brush.Color:=clNavy;
Brush.Style:=bssolid;
Brush.Color:=clGreen;
RoundRect(250,350,350,450,20,20);
Brush.Color:=clPurple;
Polygon([Point(150,600),Point(300,500),Point(200,500)]);
End;
End;
end.

```



TextOut әдісі мәтін шығарады. PenColor қасиеті сызықтың түсін білдіреді. Pen.Width қасиеті сызықтың қалыңдығын білдіреді. Brush.Style және Brush.Color қасиеті салынған объектінің ішін бояйды. Ал Ellipse эллипс салады. Round Rect қасиеті төртбұрыш салады. Polygon ұшбұрыш салады.

2-есеп. Форма бетіне төмендегі суретте көрсетілген графикалық бейнені шығаратын бағдарламаны құрыңыз.



Бағдарламаға қажетті компонент: Button1.

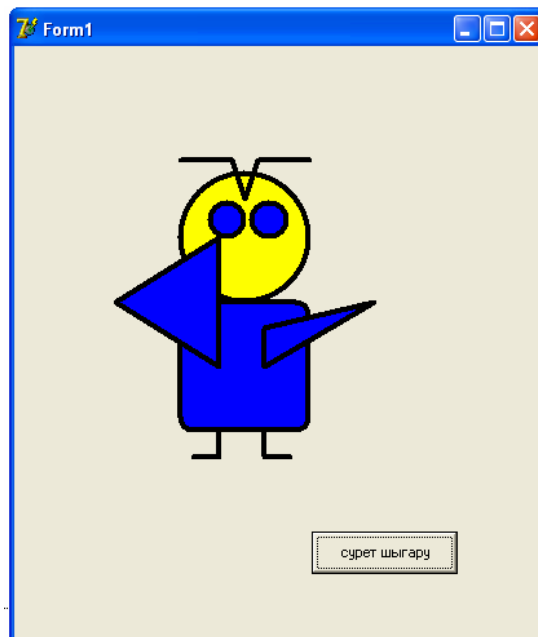
```
unit Unit1;
interface
uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls;
type
  TForm1 = class(TForm)
    Button1: TButton;
    procedure Button1Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;
var
  Form1: TForm1;
implementation
{$R *.dfm}

procedure TForm1.Button1Click(Sender: TObject);
begin
  With form1.Canvas do begin
    Pen.Color:= clBlack;
    Pen.Width:=4;
    Brush.Style:=bssolid;
    Brush.color:=clYellow;
    Ellipse(250,250,350,350);
    Brush.Style:=bsSolid;
    Brush.Color:=clBlue;
    Ellipse(273,273,300,300);
    Ellipse(333,273,305,300);
    Brush.Style:=bssolid;
```

```

RoundRect(250,350,350,450,20,20);
MoveTo(280,450);
LineTo(280,470);
MoveTo(280,470);
LineTo(260,470);
MoveTo(315,450);
LineTo(315,470);
MoveTo(315,470);
LineTo(335,470);
Polygon([point(200,350),Point(280,300),Point(280,400)]);
Polygon([Point(315,370),Point(400,350),Point(315,400)]);
MoveTo(250,240);
LineTo(290,240);
MoveTo(290,240);
LineTo(300,270);
MoveTo(300,270);
LineTo(310,240);
MoveTo(310,240);
LineTo(350,240);
End;
end;
end.

```



3-есеп. Форма бетіне олимпиадалық жалау суретін салайық. Объектіні Image компонентінің үстіне шығарамыз. Формаға Image, Button1 компоненттерін орналастырамыз. Button1 компонентінің Caption қасиетіне «сурет салу» мәнін енгіземіз. Button1 компонентінің Onclick оқиғасын құрамыз. Коды:

```

unit Unit1;
interface
uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, ExtCtrls;
type
  TForm1 = class(TForm)
    Image1: TImage;
    Button1: TButton;
    procedure Button1Click(Sender: TObject);

```

```

private
  { Private declarations }
public
  { Public declarations }
end;
var
  Form1: TForm1;
implementation
{$R *.dfm}

procedure TForm1.Button1Click(Sender: TObject);
begin
  with Image1.canvas do
  begin
    pen.Width:=3;Pen.Color:=clblack;
    brush.Color:=clcream;Rectangle(30,30,150,115);
    pen.Width:=5;brush.Style:=bsclear;
    Pen.Color:=clblue;ellipse(40,40,80,80);
    Pen.Color:=clblack;ellipse(70,40,110,80);
    Pen.Color:=clred;ellipse(100,40,140,80);
    Pen.Color:=clyellow;ellipse(55,65,95,105);
    Pen.Color:=clgreen;ellipse(85,65,125,105);
  end;
end;

end.

```



4-есеп. Форманың кез келген жерінде тышқанның оң жақ батырмасын басқанда қызыл түсті жұлдызша, сол жақ батырмасын басқанда қара түсті жұлдызша салатын бағдарлама.

```

unit Unit1;
interface
uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls;

```

```

type
  TForm1 = class(TForm)
    Label1: TLabel;
    procedure FormMouseDown(Sender: TObject; Button: TMouseButton;
      Shift: TShiftState; X, Y: Integer);
  private
    { Private declarations }
  public
    { Public declarations }
    procedure StarLine(x0,y0,r: integer);
  end;
var
  Form1: TForm1;
implementation
{$R *.dfm}

procedure TForm1.StarLine(x0,y0,r: integer);
var
  p : array[1..11] of TPoint;
  a: integer;
  i: integer;
begin
  a := 18;
  for i:=1 to 10 do
    begin
      if (i mod 2 = 0) then
        begin
          p[i].x := x0+Round(r/3*cos(a*2*pi/360));
          p[i].y:=y0-Round(r/3*sin(a*2*pi/360));
        end
      else
        begin
          p[i].x:=x0+Round(r*cos(a*2*pi/360));
          p[i].y:=y0-Round(r*sin(a*2*pi/360));
        end;
      a := a+36;
    end;

  p[11].X := p[1].X;
  p[11].Y := p[1].Y;

  Canvas.Polyline(p);
end;

```

```

procedure TForm1.FormMouseDown(Sender: TObject; Button:
TMouseButton; Shift: TShiftState; X, Y: Integer);
begin
  if Button = mbLeft
    then Canvas.Pen.Color := clBlack
    else Canvas.Pen.Color := clRed;
  StarLine(x, y, 30);
end;
end.

```



5-есеп. Облысты толтыру стилінің бағдарламасы. Ол терезеге әр түрлі стильдерді пайдаланып, 1 көк түспен боялған, 1 іші боялмаған және іші әр түрлі стильде сызылған (диагональ, вертикаль және т.б.) тіктөртбұрыш шығарады.

Бағдарламаға қажетті компонент: Image

```

unit Unit1;
interface
uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, ExtCtrls;
type
  TForm1 = class(TForm)
    Image1: TImage;
    procedure Image1Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;
var
  Form1: TForm1;
implementation
{$R *.dfm}

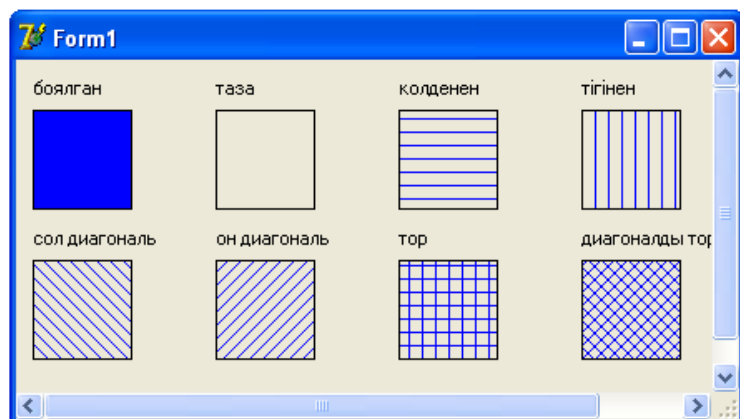
procedure TForm1.Image1Click(Sender: TObject);

```

```

const
bsname:array[1..8] of string=('боялган','таза',
'колденен','тігінен','сол диагональ','он диагональ','тор',
'диагоналды тор');
var
x,y: integer;
w,h:integer;
bs: TBrushStyle;
k: integer;
i, j: integer;
begin
w:=60; h:=60; y:=30;
for i:=1 to 2 do
begin
x:=10;
for j:=1 to 4 do
begin
k:=j+(i-1)*4;
case k of
1:bs:=bsSolid;
2:bs:=bsClear;
3:bs:=bsHorizontal;
4:bs:=bsVertical;
5:bs:=bsFDiagonal;
6:bs:=bsBDiagonal;
7:bs:=bsCross;
8:bs:=bsDiagCross
end;
Canvas.Brush.Color:=ClBlue;
Canvas.Brush.Style:=Bs;
Canvas.Rectangle(x,y,x+w,y+h);
Canvas.Brush.Style:=BsClear;
Canvas.TextOut(x,y-20,bsName[k]);
x:=x+w+50;
end;
y:=y+h+30;
end;
end;
end.

```



6-есеп. $F=2*\sin(x)*\exp(x/5)$ функциясының графигін тұрғызатын бағдарлама.

```
Function f(x:real):real;
begin
  f:=2*Sin(x)*exp(x/5);
end;

// функцияның графигін тұрғызады
procedure GrOfFunc;
var
  x1,x2:real;
  y1,y2:real;
  x:real;
  y:real;
  dx:real;
  l,b:integer;
  w,h:integer;
  mx,my:real;
  x0,y0:integer;
begin
  l:=10;
  b:=Form1.ClientHeight-20;
  h:=Form1.ClientHeight-40
  w:=Form1.Width-40;
  x1:=0;  x2:=25;
  dx:=0.01
  y1:=f(x1);
  y2:=f(x1);
  x:=x1;
  repeat
    y := f(x);
    if y < y1 then y1:=y;
    if y > y2 then y2:=y;
    x:=x+dx;
  until (x>=x2);
  my:=h/abs(y2-y1);
  mx:=w/abs(x2-x1);
  x0:=1;
  y0:=b-Abs(Round(y1*my));
  with form1.Canvas do
  begin
    MoveTo(l,b);LineTo(l,b-h);
    MoveTo(x0,y0);LineTo(x0+w,y0);
    TextOut(l+5,b-h,FloatToStrF(y2,ffGeneral,6,3));
```

```

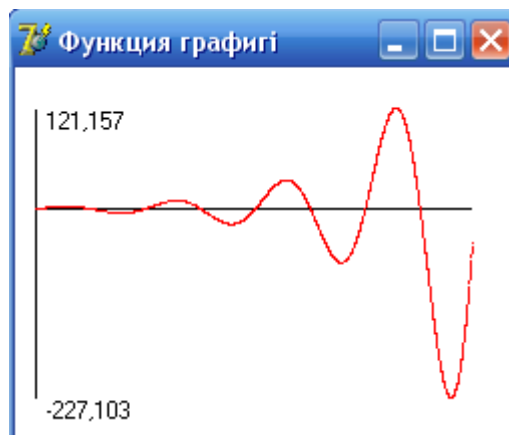
TextOut(l+5,b,FloatToStrF(y1,ffGeneral,6,3));
x:=x1;
repeat
  y:=f(x);
  Pixels[x0+Round(x*mx),y0-Round(y*my)]:=clRed;
  x:=x+dx;
until (x>=x2);
end;
end;

procedure TForm1.FormPaint(Sender: TObject);
begin
  GrOfFunc;
end;

procedure TForm1.FormResize(Sender: TObject);
begin
  form1.Canvas.FillRect(Rect(0,0,ClientWidth,ClientHeight));
  GrOfFunc;
end;

end.

```



7-есеп. X мәні [a,b] интервалында анықталатын $y=\sin(x)$ функциясының графигін тұрғызалық. Формаға Edit1, Edit2, Button1 компоненттерін қоямыз. a, b мәндерін Edit1, Edit2 компонентіне енгіземіз. Button1 компонентінің Caption қасиетіне «график тұрғызу» мәнін енгіземіз. Button1 компонентінің Onclick оқиғасын құрамыз. Коды:

```

unit Unit1;
interface
uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls;
type
  TForm1 = class(TForm)
    Button1: TButton;

```

```

    Edit1: TEdit;
    Edit2: TEdit;
    procedure Button1Click(Sender: TObject);
private
    { Private declarations }
public
    { Public declarations }
end;

var
    Form1: TForm1;
implementation

{$R *.dfm}
    function f(var x:real):real;
begin
    f:=x*x;
end;

procedure TForm1.Button1Click(Sender: TObject);
var x1,x2,y1,y2,x,y,dx,mx,my,k:real;
    t,l,b,w,h,x0,y0,lx,ly,kx:integer;
begin
    form1.Canvas.FillRect(rect(0,0,clientwidth,clientheight));
    lx:=form1.Width div 2;
    ly:=form1.Height div 2;
    form1.Canvas.textout(lx,ly,'y=x^2');
    l:=10;
    b:=form1.ClientHeight-20;
    h:= form1.ClientHeight-20;
    w:=form1.Width-40;
    x1:=strtofloat(edit1.text);x2:=strtofloat(edit2.text);
    dx:=0.01; lx:=0; ly:=10;
    y1:=f(x1);y2:=f(x2); x:=x1;kx:=10;
    repeat
    y:=f(x);
    if y<y1 then y1:=y;
    if y>y2 then y2:=y;
    x:=x+dx;
    until(x>=x2);
    my:=h/abs(y2-y1);
    mx:=w/abs(x2-x1);
    x0:=l;
    y0:=b-abs(round(y1*my));
    with form1.Canvas do

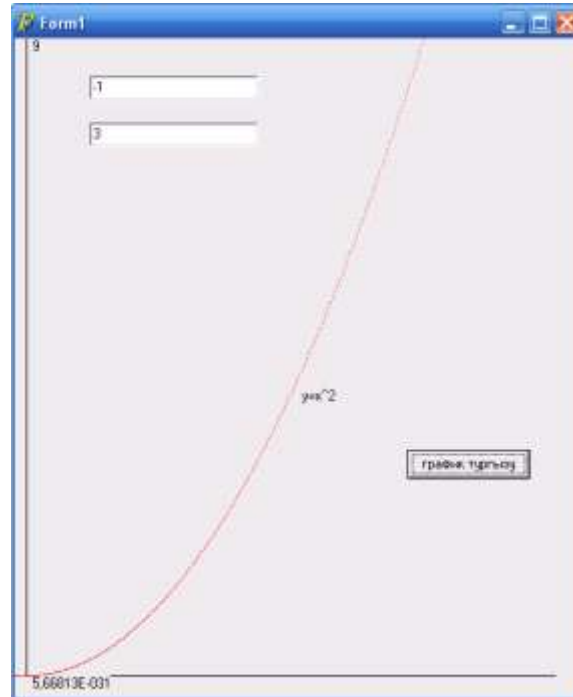
```

```

begin
moveto(1,b);lineto(1,b-h);
moveto(x0,y0); lineto(x0+w,y0);
textout(1+5,b-h,floattostrf(y2,ffgeneral,6,3));
textout(1+5,b,floattostrf(y1,ffgeneral,6,3));
x:=x1;
repeat
y:=f(x);
pixels[x0+round(x*mx),y0-round(y*my)]:=clred;
x:=x+dx;
until(x>=x2);
end;
end;

end.

```



8-есеп. Button батырмасын шерткенде, Күн сәулесінің бағыты өзгеріп отыратын бағдарлама.

Қажетті компоненттер: Image1; Button1;

```

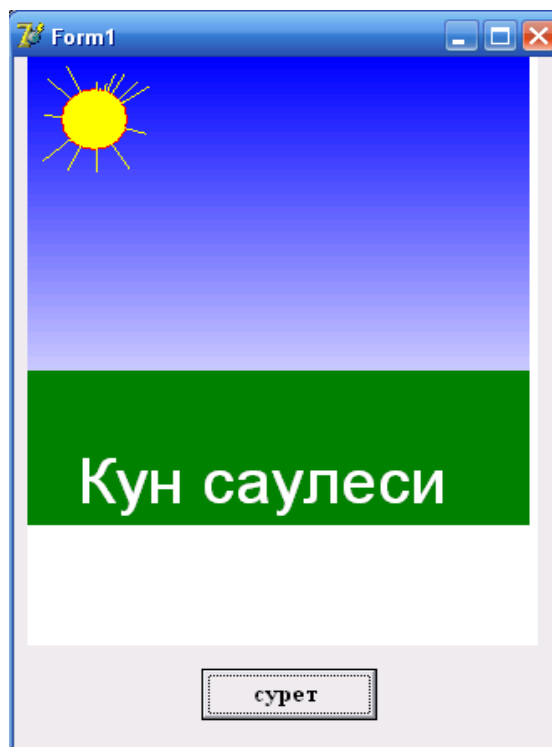
procedure TForm1.Button1Click(Sender: TObject);
var i:integer;
begin
Image1.Canvas.Pen.Color := clGreen;
Image1.Canvas.Brush.Color := clGreen;
Image1.Canvas.Rectangle(0, 200, 300, 300);
Image1.Canvas.Font.Color := clWhite;
Image1.Canvas.Font.Size := 30;
Image1.Canvas.Font.Name := 'Arial';
Image1.Canvas.TextOut(30, 250, 'Күн саулеси');
for i := 0 to 200 do
begin

```

```

Image1.Canvas.Pen.Color := RGB(I, I, 255);
Image1.Canvas.MoveTo(0, i);
Image1.Canvas.LineTo(300, i);
end;
Image1.Canvas.Pen.Color := clRed;
Image1.Canvas.Brush.Color := clYellow;
Image1.Canvas.Ellipse(20, 20, 60, 60);
Image1.Canvas.Pen.Color := clYellow;
for i := 1 to 20 do
begin
Image1.Canvas.MoveTo(40, 40);
Image1.Canvas.LineTo(5 + random(70), 5 + random(70));
end;
end;
end.

```



9-есеп. Графикалық редакторда сурет салып, оны Delphi ортасында қозғалысқа келтіретін бағдарлама.

```

unit Unit1;
interface
uses
Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
ExtCtrls, StdCtrls;
type
TForm1 = class(TForm)
Timer1: TTimer;
procedure FormActivate(Sender: TObject);

```

```

    procedure Timer1Timer(Sender: TObject);
private
    { Private declarations }
public
    { Public declarations }
end;
var
    Form1: TForm1;
implementation
{$R *.DFM}

const
    FILMFILE = 'film2.bmp';
    N_KADR=7;
var
    Film: TBitmap;
// Kadr: TBitmap;
    WKadr, HKadr: integer;
    CKadr: integer;
    RectKadr: TRect;
    Rect1 : Trect;

procedure TForm1.FormActivate(Sender: TObject);
begin
    Film := TBitmap.Create;
    Film.LoadFromFile(FILMFILE);
    WKadr := Round(Film.Width/N_Kadr);
    HKadr := Film.Height;

    Rect1 := Bounds(40,40,WKadr,HKadr);

    Ckadr:=5;

    Form1.Timer1.Interval := 150;
    Form1.Timer1.Enabled:=True;
end;

procedure DrawKadr;
begin
    RectKadr:=Bounds(WKadr*CKadr,5,WKadr,HKadr);
    Form1.Canvas.CopyRect(Rect1,Film.Canvas,RectKadr);
    CKadr := CKadr+1;
    if CKadr = N_KADR
    then CKadr:=5;

```

end;

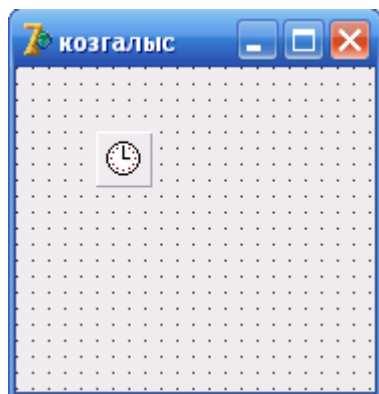
```
procedure TForm1.Timer1Timer(Sender: TObject);
```

```
begin
```

```
    DrawKadr;
```

```
end;
```

```
end.
```



10-есеп. Қозғалып келе жатқан корабльдің суретін салатын бағдарлама.

```
unit Unit1;
```

```
interface
```

```
uses
```

```
    Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,  
    StdCtrls, ExtCtrls;
```

```
type
```

```
    TForm1 = class(TForm)
```

```
        Timer1: TTimer;
```

```
        procedure Timer1Timer(Sender: TObject);
```

```
        procedure FormActivate(Sender: TObject);
```

```
    private
```

```
        { Private declarations }
```

```
    public
```

```
        { Public declarations }
```

```
end;
```

```
var
```

```
    Form1: TForm1;
```

```
    x,y: integer;
```

```
implementation
```

```
    {$R *.DFM}
```

```
procedure Titanik(x,y: integer;
```

```
color: TColor);
```

```
const
```

```
    dx = 5;
```

```
    dy = 5;
```

```

var
  buf: TColor;
begin
  with form1.canvas do
  begin
    buf:=pen.Color;
    pen.Color:=color;
    MoveTo(x,y);
    LineTo(x,y-2*dy);
    LineTo(x+10*dx,y-2*dy);
    LineTo(x+11*dx,y-3*dy);
    LineTo(x+17*dx,y-3*dy);
    LineTo(x+14*dx,y);
    LineTo(x,y);
    MoveTo(x+3*dx,y-2*dy);
    LineTo(x+4*dx,y-3*dy);
    LineTo(x+4*dx,y-4*dy);
    LineTo(x+13*dx,y-4*dy);
    LineTo(x+13*dx,y-3*dy);
    MoveTo(x+5*dx,y-3*dy);
    LineTo(x+9*dx,y-3*dy);
    Rectangle(x+8*dx,y-4*dy,x+11*dx,y-5*dy);
    Rectangle(x+7*dx,y-4*dy,x+8*dx,y-7*dy);
    Ellipse(x+11*dx,y-2*dy,x+12*dx,y-1*dy);
    Ellipse(x+13*dx,y-2*dy,x+14*dx,y-1*dy);
    MoveTo(x+10*dx,y-5*dy);
    LineTo(x+10*dx,y-10*dy);
    MoveTo(x+17*dx,y-3*dy);
    LineTo(x+10*dx,y-10*dy);
    LineTo(x,y-2*dy);
    pen.Color:=buf;
  end;
end;

procedure TForm1.Timer1Timer(Sender: TObject);
begin
  Titanik(x,y,form1.color);
  if x < Form1.ClientWidth
  then x := x+5
  else begin
    x := 0;
    y := Random(50) + 100;
  end;
  Titanik(x,y,clWhite);
end;

```

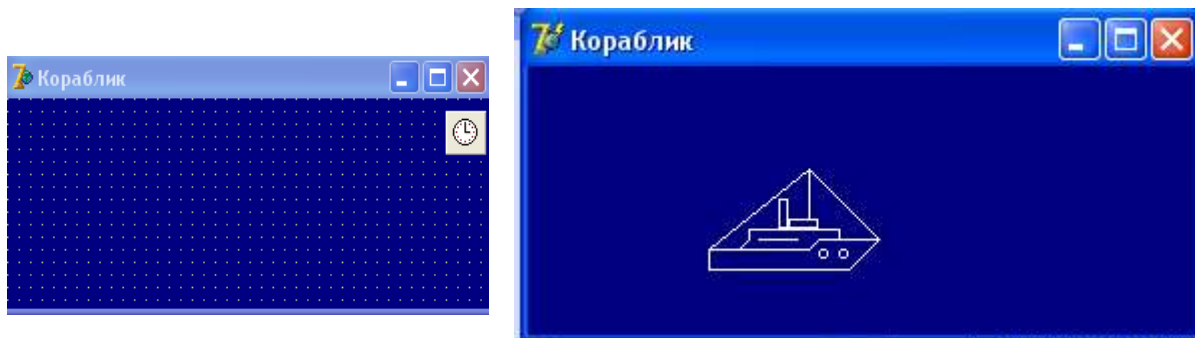


```

procedure TForm1.FormActivate(Sender: TObject);
begin
    x:=0;
    y:=100;
    Form1.Color:=clNavy;
    Timer1.Interval := 50;
end;

end.

```



11-есеп. Форма бетінде сурет (объект) жылжиды. Біздің мақсатымыз тышқанның сол жақ батырмасымен суретті шерте отырып жоғарғы мәнді ұпай жинау. Мысалы, атыс саны 10 болса, оның нешеуі тиетінін анықтауымыз керек. Delphi-ді жүктейміз. Формаға Label1, button1, Timer1 компоненттерін орналастырамыз. Label1 (Caption) тақырыбына есеп шартын жазамыз.

Button1 компонентінің Onclick оқиғасы Label1, button1 компоненттерін көрінбейтін қылады және Timer-ді қосады.

```

procedure TForm1.Button1Click(Sender: TObject);
begin
    Label1.Visible:=false;
    Button1.Visible:=False;
    Timer1.Enabled:=True;
end;

```

S_f процедурасында объект суретін саламыз:

```

procedure TForm1.S_f(x,y:integer);
begin
    with Canvas do begin
        Pen.Color:=clblack;
        Brush.Color:=clYellow;
        Ellipse(x,y,x+30,Y+30);
        Ellipse(x+9,y+10,x+11,Y+13);
        Ellipse(x+19,y+10,x+21,Y+13);
        Arc(x+4,y+4,x+26,y+26,x,y+20,x+30,y+20)
    end;
end;

```

```
end;  
end;
```

EF процедурасында суретті форма түсімен бояймыз, яғни сурет экранда көрінбейді.

```
procedure TForm1.EF(x,y:integer);  
begin  
with Canvas do begin  
Pen.Color:=clBtnFace;  
Brush.Color:=clBtnFace;  
Ellipse(x,y,x+30,Y+30);  
end;  
end;
```

Timer1 компонентінің OnTimer оқиғасы кездейсоқ сандар генераторы арқылы объектінің жаңа координатасын анықтайды.

```
procedure TForm1.Timer1Timer(Sender: TObject);  
begin  
EF(fx,fy);  
fx:=Random(ClientWidth-30);  
fy:= Random(ClientHeight-30);  
S_F(fx,fy);  
end;
```

Форманың OnCreate оқиғасында объектінің бастапқы координатасы анықталады және кездейсоқ сандар генераторы реализацияланады.

```
procedure TForm1.FormCreate(Sender: TObject);  
begin  
fx:=100;  
fy:=100;  
randomize;  
end;
```

Форманың OnMouseDown оқиғасында тышқанның сол жақ кнопкасымен суретті шерте отырып ұпай жинаймыз және 10 атыстың нәтижесін көрсетеді.

```
procedure TForm1.FormMouseDown(Sender: TObject; Button:  
TMouseButton;  
Shift: TShiftState; X, Y: Integer);  
begin  
inc(n);  
if(x>fx) and(x<fx+30) and(y>fy) and(y>fy) and(y<fy+30)  
then inc(p);  
if n=10 then begin  
Timer1.Enabled:=false;  
ShowMessage('10 атыс. тиг. '+inttostr(p)+'');  
end;
```

```
EF(fx,fy);
Label1.Visible:=True;
Button1.Visible:=True;
n:=0;p:=0;
end;
end;
```

12-есеп. Цельсий шкаласында берілген температура мәнін Фаренгейт шкаласына түрлендіретін бағдарлама. Температура мәнінің өзгерісін графикалық түрде беріңіз.

Қажетті компоненттер: Edit1; Edit2; Button1; Button2; Button3; Button4; Image1; Image2; Label1; Label2;

```
procedure TForm1.FormCreate(Sender: TObject);
begin
  Edit1.Text := "";
  Edit2.Text := "";
  Image1.Canvas.Pen.Color := clRed;
  Image1.Canvas.Brush.Color := clWhite;
  Image1.Canvas.Rectangle(10,1,23,360);
  Image2.Canvas.Pen.Color := clBlue;
  Image2.Canvas.Brush.Color := clWhite;
  Image2.Canvas.Rectangle(10,1,23,360);
end;
```

```
procedure TForm1.Button2Click(Sender: TObject);
var a, b, c : real;
begin
  a := StrToFloat(Edit1.Text);
  if (a > 0) or (a < 100) then b := 1.8 * a + 32;
  if a > 100 then b := 212;
  if a < 0 then b := 32;
  Edit2.Text := FloatToStr(b);
  Image1.Canvas.Pen.Color := clRed;
  Image1.Canvas.Brush.Color := clWhite;
  Image1.Canvas.Rectangle(10,1,23,360);
  Image1.Canvas.Pen.Color := clRed;
  Image1.Canvas.Brush.Color := clRed;
  Image1.Canvas.Rectangle(10,round(360-a),23,360);
  Image2.Canvas.Pen.Color := clBlue;
  Image2.Canvas.Brush.Color := clWhite;
  Image2.Canvas.Rectangle(10,1,23,360);
  Image2.Canvas.Pen.Color := clBlue;
  Image2.Canvas.Brush.Color := clBlue;
```

```
Image2.Canvas.Rectangle(10,round(360-b),23,360);  
end;
```

```
procedure TForm1.Button1Click(Sender: TObject);  
var a, c, d : real;  
begin  
  c := StrToFloat(Edit2.Text);  
  if (c > 32) or (c < 212) then d := (c - 32)/1.8;  
  if c < 32 then d := 0;  
  if c > 212 then d := 100;  
  Edit1.Text := FloatToStr(d);  
  Image2.Canvas.Pen.Color := clBlue;  
  Image2.Canvas.Brush.Color := clWhite;  
  Image2.Canvas.Rectangle(10,1,23,360);  
  Image2.Canvas.Pen.Color := clBlue;  
  Image2.Canvas.Brush.Color := clBlue;  
  Image2.Canvas.Rectangle(10,round(360-c),23,360);  
  Image1.Canvas.Pen.Color := clRed;  
  Image1.Canvas.Brush.Color := clWhite;  
  Image1.Canvas.Rectangle(10,1,23,360);  
  Image1.Canvas.Pen.Color := clRed;  
  Image1.Canvas.Brush.Color := clRed;  
  Image1.Canvas.Rectangle(10,round(360-d),23,360);  
end;
```

```
procedure TForm1.Button3Click(Sender: TObject);  
begin  
  Edit1.Text := " ";  
  Edit2.Text := " ";  
  Image1.Canvas.Pen.Color := clRed;  
  Image1.Canvas.Brush.Color := clWhite;  
  Image1.Canvas.Rectangle(10,1,23,360);  
  Image2.Canvas.Pen.Color := clBlue;  
  Image2.Canvas.Brush.Color := clWhite;  
  Image2.Canvas.Rectangle(10,1,23,360);  
end;
```

```
procedure TForm1.Button4Click(Sender: TObject);  
begin  
  Close;  
end;
```

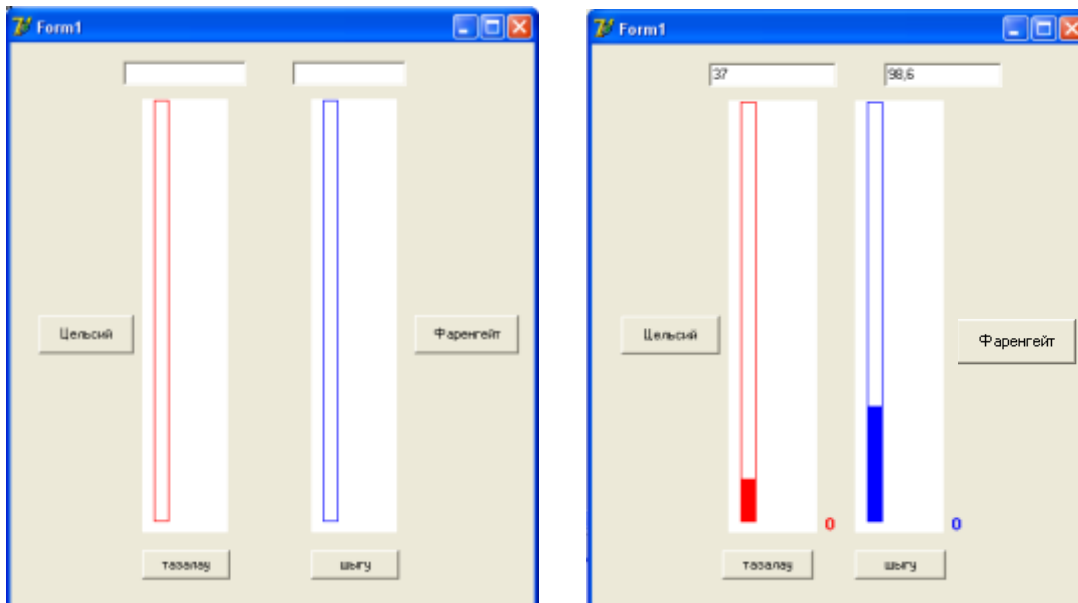
```
procedure TForm1.Edit1Enter(Sender: TObject);  
begin  
  Button2.Visible := true;
```

```

    Button1.Visible := False;
end;

procedure TForm1.Edit2Enter(Sender: TObject);
begin
    Button1.Visible := True;
    Button2.Visible := False;
end;
end.

```



Тапсырмалар:

1. Экранда үшбұрыш салындар және оны вертикаль бағытта дыбыспен сүйемелдей отырып қозғалтындар.
2. Сары түспен бір түзудің бойында орналасқан екі трапеция, жасыл түспен параллелепипед сызатын бағдарлама құрыңыздар.
3. Қызыл түспен боялған тік төртбұрыш салып, оның ішіне іші штрихталған алты бұрыш салыңыздар.
4. Әр түрлі түстегі шеңберден кездейсоқ кесіп алынған секторлар тобын салындар.
5. Формаға қозғалатын объектісі бар қандайда бір көріністің (мысалы, автокөлік қозғалысы, мультипликациялық элементтер) суретін салыңыз.

Бақылау сұрақтары:

1. Графикалық құралдарды пайдалануға арналған кластарды атаңдар?
2. Қаріп, сызық, толтыру, кенеп қандай кластың көмегімен құрылады?
3. Қаламның қандай қасиеттері бар?
4. Қылқаламның қызметі не?
5. Графикалық бетке мәтін қалай шығарылады?
6. Графикалық примитивтерді салудың қандай әдістері бар?

7. Мультимедиялық бағдарламалар қандай компоненттердің көмегімен құрылады?

8. Бейнероликтер, дыбысы бар анимациялар қандай компоненттердің көмегімен құрылады?

Зертханалық жұмыс №14. Форманың атрибуттары және стилі. Форманы құру және өшіру. Қосымшада бірнеше форманы қолдану

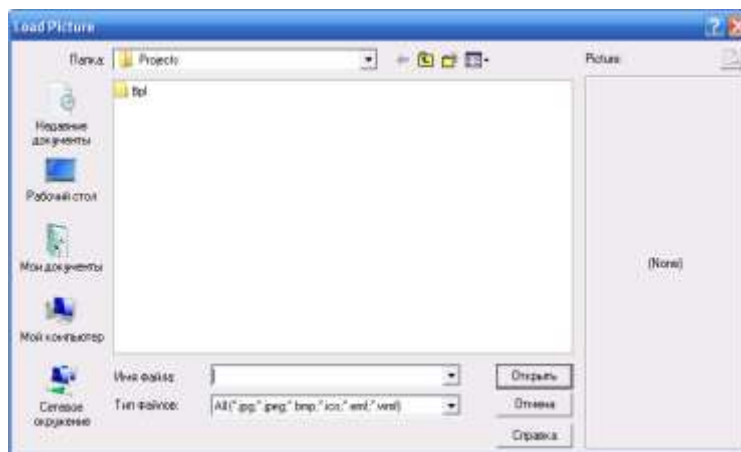
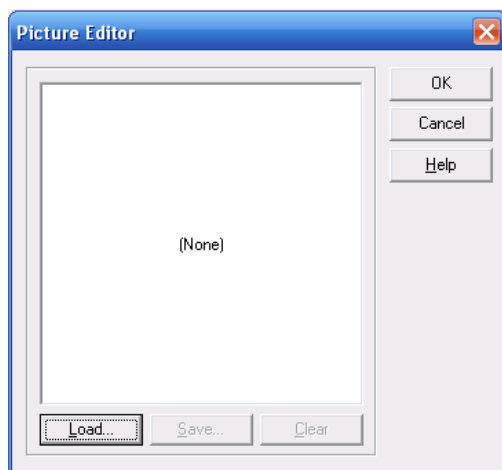
Жұмыстың мақсаты: Қосымшада формаларды құру және жою әдістерін, формалардың бір-бірімен байланысын ұйымдастыруды, формаға Show, Hide әдістерін қолдана білулерін үйрету.

Бағдарламалық дидактикалық қамтылуы: Дербес компьютерлер, Windows XP операциялық жүйесі, Borland Delphi 7 бағдарламасы, «Бағдарламалау» пәнінен оқу-әдістемелік кешен.

1-есеп. Формаға студент туралы анықтама және оның бейне суретін, тобымен түскен суретін шығаратын бағдарлама.

Бағдарламаға қажетті компоненттер: Label1, Label2, Label3, Label4, Label5, Image1, Image2, Button1, Button2.

Image компонентіне сурет орналастыру үшін, Object Inspector терезесіндегі Picture қасиетінің Events бағанындағы None өрісіндегі ... (үш нүкте) үстінен тышқан батырмасын басамыз. Нәтижеде Picture Editor диалогтық терезесі пайда болады. Осы терезедегі Load батырмасын бассақ, Load Picture диалогтық терезесі шығады. Осы терезеден Image компонентіне орналастыратын суретті таңдап алып, Открыть батырмасын басамыз.



unit Unit1;

interface

uses

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms, Dialogs, StdCtrls, ExtCtrls;

```

type
  TForm1 = class(TForm)
    Button1: TButton;
    Button2: TButton;
    Image1: TImage;
    Image2: TImage;
    Label1: TLabel;
    Label2: TLabel;
    Label3: TLabel;
    Label4: TLabel;
    Label5: TLabel;
    procedure Button1Click(Sender: TObject);
    procedure Button2Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;
var
  Form1: TForm1;
implementation
{$R *.dfm}

procedure TForm1.Button1Click(Sender: TObject);
begin
  Image1.Visible:=True;
  Image2.Visible:=False;
end;

procedure TForm1.Button2Click(Sender: TObject);
begin
  Image1.Visible:=False;
  Image2.Visible:=True;
end;

end.

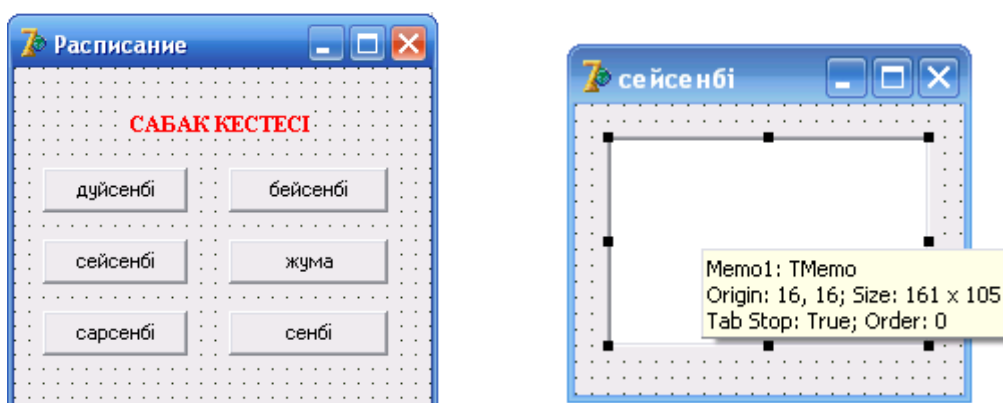
```



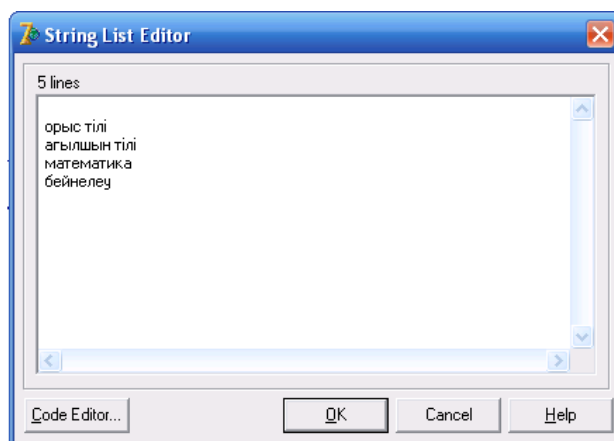
2-есеп. Аптадағы сабақ кестесін бейнелейтін бағдарлама.

Бағдарламаға қажетті компоненттер: Form1 бетіне - Label1, Button1, Button2, Button3, Button4, Button5, Button6;

Дүйсенбіден сенбіге дейінгі 6 күннің сабақ кестесін шығару үшін Form2-Form7 беттері пайдаланылады. Form2-ден Form7-ге дейінгі форма беттеріне Мемо компонентін орналастыру керек.



Объектілер инспекторы терезесінде Lines қасиетін тышқанмен белгілеп, ондағы TStrings мәнін екпінді жасайды `Lines` `TStrings` , бұдан жолдың оң жақ шетінде үш нүкте (...) пайда болады. Үш нүктенің (...) үстінен тышқан батырмасын бассақ, String List Editor сұхбат терезесін экранға шығарады. Осы терезеде сабақ кестесін пернетақтадан енгізіп, ОК батырмасын басамыз. Осылайша Form2-Form7 беттері дайындалады.



```
unit Unit1;  
interface  
uses  
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,  
  Dialogs, StdCtrls;  
type  
  TForm1 = class(TForm)  
    Button1: TButton;
```



```

    Button2: TButton;
    Button3: TButton;
    Button4: TButton;
    Button5: TButton;
    Button6: TButton;
    Label1: TLabel;
    procedure Button1Click(Sender: TObject);
    procedure Button2Click(Sender: TObject);
    procedure Button3Click(Sender: TObject);
    procedure Button4Click(Sender: TObject);
    procedure Button5Click(Sender: TObject);
    procedure Button6Click(Sender: TObject);
private
    { Private declarations }
public
    { Public declarations }
end;

var
    Form1: TForm1;

implementation

uses Unit2, Unit3, Unit4, Unit5, Unit6, Unit7;

{$R *.dfm}

procedure TForm1.Button1Click(Sender: TObject);
begin
    Form2.Show;
end;

procedure TForm1.Button2Click(Sender: TObject);
begin
    Form3.show;
end;

procedure TForm1.Button3Click(Sender: TObject);
begin
    form4.show
end;

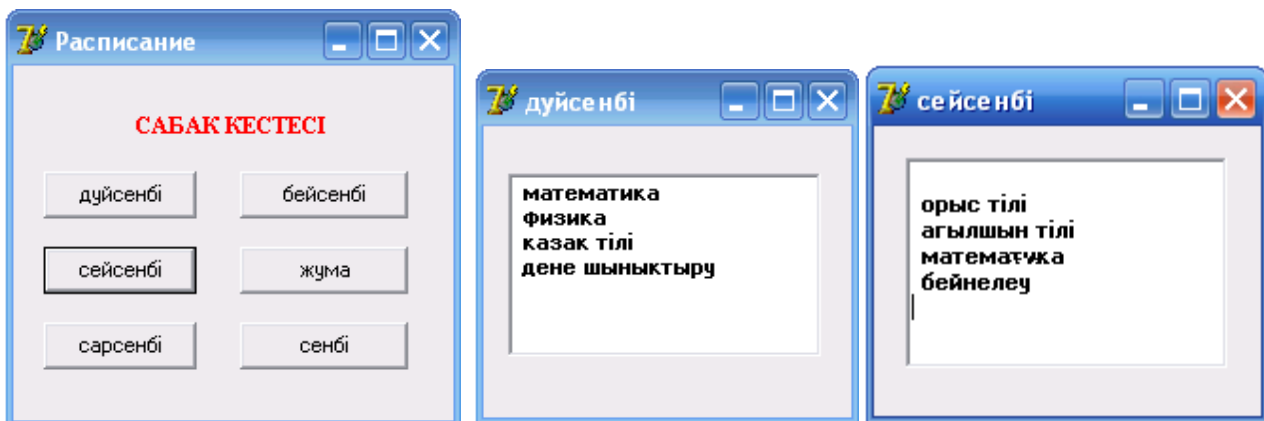
procedure TForm1.Button4Click(Sender: TObject);
begin
    Form5.show;

```

end;

```
procedure TForm1.Button5Click(Sender: TObject);  
begin  
Form6.show;  
end;
```

```
procedure TForm1.Button6Click(Sender: TObject);  
begin  
form7.show;  
end;  
end.
```



3-есеп. Delphi ортасында электронды оқу құралдарын жасау

Бағдарламаға қажетті компоненттер: Panel, Button1, Button2, Button3, Button4, WebBrowser

Әрбір батырманы шерткенде WebBrowser компонентінде сәйкес бетшелер ашылу үшін әр батырмаға төмендегідей код жазамыз:

Дәріс №1 батырмасы үшін :

```
procedure TForm1.Button1Click(Sender: TObject);  
begin  
WebBrowser1.Navigate(extractfilepath(application.ExeName)+'C:\дарис1.html')  
end;
```

Дәріс №2 батырмасы үшін :

```
procedure TForm1.Button2Click(Sender: TObject);  
begin  
WebBrowser1.Navigate(extractfilepath(application.ExeName)+'C:\дарис2.html')  
end;
```

Дәріс №3 батырмасы үшін :

```

procedure TForm1.Button3Click(Sender: TObject);
begin
WebBrowser1.Navigate(extractfilepath(application.ExeName)+'C:\дарис3.html')
end;

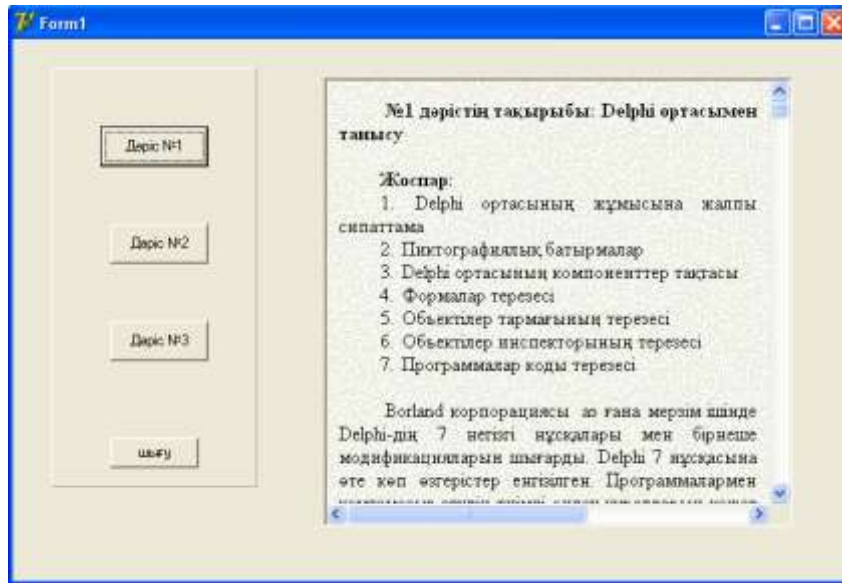
```

ШЫҒУ батырмасы үшін :

```

procedure TForm1.Button4Click(Sender: TObject);
begin
Form1.close;
end;

```



4-есеп. Қарапайым тестілеу бағдарламасының қосымшасын құру.

1-үлгі

Бағдарламаға қажетті компоненттер: Label1; Label2; Label3; Label4; Label5; RadioGroup1; RadioGroup2; RadioGroup3; RadioGroup4; RadioGroup5; Panel1; Button1; BitBtn1.

```

procedure TForm1.Button1Click(Sender: TObject);
type T1 = array[0..4, 0..4] of byte;
const otv:T1=((0,0,0,1,0),(0, 0,0,0, 1),(0,1,0,0,0),(0,0,0, 1,0),( 1,0,0,0,0));
var
Form1: TForm1;
sum :integer;
begin
sum:=otv[0,RadioGroup1.ItemIndex]+otv[1,RadioGroup2.ItemIndex]+
otv[2,RadioGroup3.ItemIndex]+otv[3,RadioGroup4.ItemIndex]+
otv[4,RadioGroup5.ItemIndex];
case sum of
0..2:begin

```

```

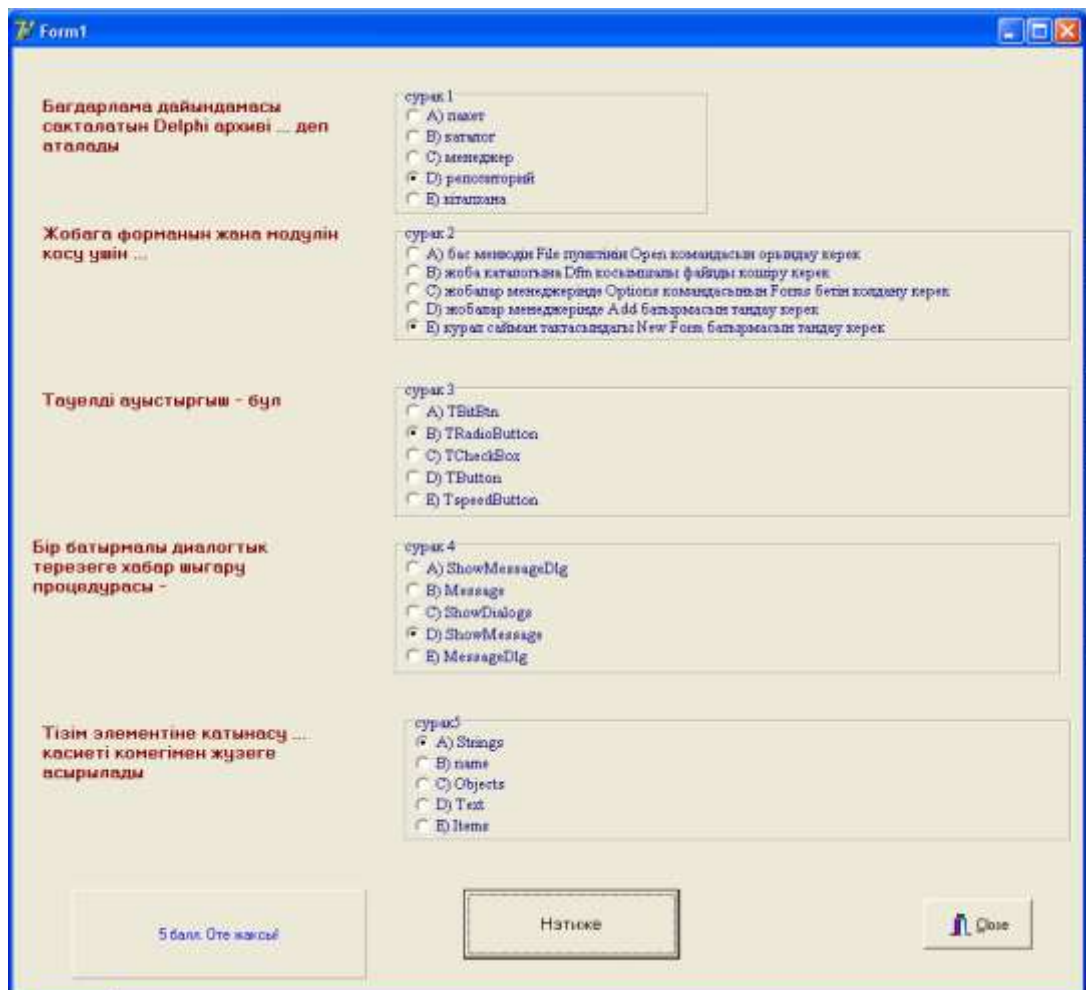
Panel1.Font.Color:=$000000;
Panel1.Caption:=IntToStr(sum)+' балл. Нашар!';
end;
3:
begin
Panel1.Font.Color:=$0000FF;
Panel1.Caption:=IntToStr(sum)+' балл. Канагаттанарлык!';
end;
4:
begin
Panel1.Font.Color:=$00FF00;
Panel1.Caption:=IntToStr(sum)+' балл. Жаксы!';
end;
5:begin
Panel1.Font.Color:=$FF0000;
Panel1.Caption:=IntToStr(sum)+' балл. Оте жаксы!';
end;
end;
end;

```

```

procedure TForm1.BitBtn1Click(Sender: TObject);
begin
close;
end;
end;

```



2-үлгі

Word бағдарламасын немесе «Блокнотты» жүктеп тест сұрақтарын дайындаймыз. Тест мазмұнының жазу үлгісі мынадай:

- 1) тест сұрағы
- 2) жауаптың 1 варианты
- 3) жауаптың 2 варианты
- 4) жауаптың 3 варианты
- 5) дұрыс жауап нөмірі

Тест мазмұнын test.txt файлына сақтаймыз. Қосымша 3 формадан тұрады: 1 – титул беті, 2 - тестілеу, 3 – нәтиже. Формалар арасында байланыс орнатамыз.

1 форманы жүктейміз. Формаға MainMenu1 компонентін қоямыз. Меню пункттері: 1) өзіндік бақылау, 2) тестілеу, 3) нәтиже.

Менюдің “тестілеу” пунктіннің коды:

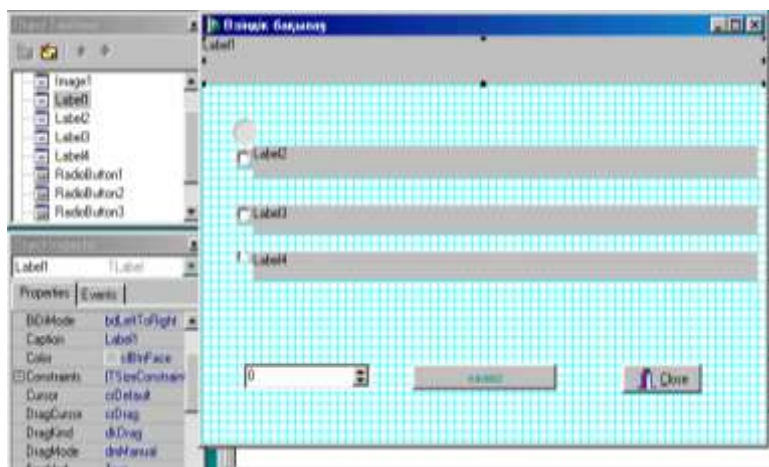
```
procedure TForm1.N2Click(Sender: TObject);  
begin  
Form2.Showmodal;  
end;
```

Менюдің “нәтиже” пунктіннің коды:

```
procedure TForm1.N3Click(Sender: TObject);  
begin  
Form3.Showmodal;  
end;
```

2-форманы жүктейміз. Формаға 4 Label компоненттерін, 3 RadioButton компоненттерін, SpinEdit1, SpeedButton, BtnButton компоненттерін қоямыз.

Қосымша интерфейсін төменде келтірілген.



SpinEdit1 компоненті санағыш ретінде қолданылады, дұрыс жауаптың санын көрсетеді. SpeedButton1 компонентінде тестілеу алгоритмі

ұйымдастырылады. Label (1-4) компоненттеріне тест сұрағы мен жауаптары оқылады.

SpeedButton1 компонентінің оқиғасының коды:

```
var
  Form2: TForm2;
  f:textfile;
  op, o : integer;
implementation

uses Unit3;

{$R *.dfm}

procedure Test;
var s: string;
begin
  readln(f,s);
  form2.label1.caption:=s;
  readln(f,s);
  form2.label2.caption:=s;
  readln(f,s);
  form2.label3.caption:=s;
  readln(f,s);
  form2.label4.caption:=s;
end;

procedure TForm2.SpeedButton1Click(Sender: TObject);
begin
  if radiobutton1.Checked then op:=1;
  if radiobutton2.Checked then op:=2;
  if radiobutton3.Checked then op:=3;
  readln(f,o);
  if op=o then spinedit1.value:=spinedit1.value+1;
  if not(eof(f)) then
    Test
  else
    Form3.show;
end;
```

Бұл процедурада ішкі Test процедурасы қолданылады. Form2-нің OnShow оқиғасының коды:

```
procedure TForm2.FormShow(Sender: TObject);
begin
```

```
assignfile(f,'test.txt');
Reset(f);
Test;
spinedit1.value:=0;
end;
```

3-форманы жүктейміз. Бұл формада нәтиже сипатталады. Формаға 2-Label, 2 - Button компоненттерін қоямыз. Label1 компонентінің Caption қасиетіне “ Сіздің ұпайыңыз” сөзін жазамыз. Label2 компонентінің Caption қасиетіне “ Сіздің бағаңыз” сөзін жазамыз. Форманың FormShow оқиғасының коды:

```
procedure TForm3.FormShow(Sender: TObject);
var b,bg:integer;
begin
button1.Caption:=inttostr(form2.spinedit1.value);
b:=form5.spinedit1.value;
if b<4 then bg:=2
    else if b<7 then bg:=3;
if (b>=7) and (b<9) then bg:=4
    else bg:=5;
button2.Caption:=inttostr(bg);
end;
```

Бағалау критерийі төменде көрсетілген:

егер балл <40% , баға - 2; егер 40%< балл < 70% , баға – 3;
егер 70%< балл <85% , баға – 4; егер балл > 85% , баға – 5.

Тапсырма:

1. «Дербес компьютер құрылғылары» тақырыбында электронды оқу құралын жасау.

Бақылау сұрақтары:

1. Форма дегеніміз не?
2. Форма конструкторының жазылуы.
3. Форма стилі форманың қандай қасиетімен анықталады?
4. Форманың белгісі форманың қандай қасиетімен анықталады?
5. Басқарудың стандарт элементтер санын қандай қасиет анықтайды?
6. Қандай жағдайда құрылған форма автоматты түрде жойыла алады?
7. Жұмыс жасап тұрған форманы ReLease әдісімен қалай жоюға болады?
8. Форманың көрінуін қандай әдіс қамтамасыз етеді?
9. Hide әдісі не үшін қажет?
10. Show әдісі не үшін қажет?

Зертханалық жұмыс №15. Мәліметтерді файлдан оқу және файлға жазу

Жұмыстың мақсаты: Delphi ортасында файлдармен жұмыс жасауға арналған есептерге бағдарлама құрып үйрену дағдыларын қалыптастыру.

Бағдарламалық дидактикалық қамтылуы: Дербес компьютерлер, Windows XP операциялық жүйесі, Borland Delphi 7 бағдарламасы, «Бағдарламалау» пәнінен оқу-әдістемелік кешен.

Файл деп дербес компьютердің сыртқы жадысының атау берілген бөлігін түсінеміз. Кез келген файлдың өзіне тән үш ерекшелігі болады. Біріншіден, файлдың атауы бар, бағдарламаға осы атауы бойынша бір мезгілде бірнеше файлмен жұмыс істеуге мүмкіндік береді. Екіншіден, ол бір типті компоненттерден тұрады. Файлдан басқа Object Pascal-дың кез келген типі компоненттер типі бола алады. Үшіншіден, файлдың ұзындығы жөнінде оны хабарлауда ешнәрсе айтылмайды, ол тек сыртқы жады құрылғысының көлемімен шектеледі. Файлдық типті төмендегі үш тәсілдің бірімен беруге болады:

<аты> = File of <тип>;

<аты> = TextFile;

<аты> = File;

мұндағы, < аты > - файлдық типтің аты; File of - қызметші сөздер (файл, содан); TextFile - стандартты мәтіндік файл типінің атауы; <тип> – файлдан басқа Object Pascal-дың кез-келген типі.

Хабарлау тәсіліне қарай, файлды үш түрге бөлуге болады: мәтіндік, типтік және типтік емес.

Ақпарат ASCII кодымен берілген файл мәтіндік файл болып табылады.

Типтік файл - бұл компоненттері - мәндері екілік кодпен берілген, файлдық типтен басқа, жай және құрылымдық типті файл.

Типтік емес файл - бұл компоненттері 0 мен 1 екілік цифрлардан құралған сегіз битті тізбек болып табылады.

Object Pascal тілінде файлдармен төмендегідей жұмыс түрін ұйымдастыруға болады:

- файл құру;
- мәліметтерді файлға жазу;
- мәліметтерді файлдан оқу;
- файлды түрлендіру.

Файл үшін логикалық және физикалық файл ұғымы енгізіледі. Логикалық файл - бұл ақпараттың қандай да бір абстракт көлемі. Физикалық файл - бұл физикалық тасымалдаушыда жазылған логикалық файл: диск, дискет, дисплей экраны, магнитті лента және т.б.

Логикалық файл мен физикалық файлды байланыстыру стандартты AssignFile апроцедурасы арқылы жүзеге асырылады:

AssignFile(<логикалық файл аты>, <физикалық файл аты>);

мұндағы, <логикалық файл аты> - бағдарламада файлдық айнымалы ретінде хабарланған файлдық айнымалы, <физикалық файл аты> - файл атынан

тұратын мәтіндік өрнек, егер қажет болса файлға кіру маршруты толық көрсетіледі,

Файлмен жұмыс істеу үшін - ашу, ал, жұмыс аяқталғаннан кейін - жабу қажет. Оқуға арналған файл RESET(FF) операторымен, жазуға арналған файл REWRITE(FF) операторымен ашылады. APPEND(FF) - мәтіндік физикалық файлға мәліметтер қосып жазуға арналған.

Мәліметтерді оқу және жазу белгілі READ/WRITE операторларының көмегімен жүзеге асырылады, тек тізімнің басында файлдық айнымалының аты көрсетіледі:

READ(FF, тізім)

WRITE(FF, тізім)

Файлдың жабылуы CLOSE(FF) операторымен орындалады.

REWRITE командасы жазуға арналған файлды ашып, файлды жазу режимінің бастапқы күйіне келтіреді, маркер нөлдік қадамға орналасады. Егер FT файлы бұрын болса, онда ол жойылады.

RESET командасы бойынша диск каталогындағы физикалық файлдың бар екендігіне көз жеткізеді, егер физикалық файл табылмаса, онда файл табылған жоқ деген хабарлама беріледі. Егер файл табылса, маркер көрсеткіші нөлдік күйге келтіріледі, файл мазмұны жойылмайды.

APPEND командасы бойынша файл көрсеткіші оның соңына орналастырылып, бұрыннан бар файлға қосымша мәліметтер тіркеліп жазылады.

Файлдармен жұмыс істеуге арналған процедуралар мен функциялар

Procedure AssignFile(var F; FileName:String); - F файлдық айнымалысын FileName атты файлмен байланыстырады;

Function ChangeFileExt(const FileName, Extension: String): String; - файлдың затын Extension параметрімен берілген файл затына өзгертеді;

Procedure ChDir(Path:String); - ағымдағы каталогты өзгертеді: Path - ескертусіз орнатылған каталогқа баратын жолды көрсететін жолдық өрнек;

Procedure CloseFile(Var F); - файлды жабады, бірақ F файлдың айнымалысымен AssignFile процедурасымен алдын-ала орнатылған байланыс сақталады. CloseFile процедурасының функциясы бағдарлама жұмысы дұрыс аяқталған жағдайда барлық ашылған файлдар автоматты түрде жабылады.

Function DateTimeToFileDate(DateTime:TDateTime):Integer; - DateTime мәнін файл құру уақытының жүйелік форматына аударады;

Function DiskFree(D:Byte):LongInt; - көрсетілген дискідегі бос орынның байттық көлемін береді: D - диск нөмірі (0 ескертусіз құрылғы; 1 - диск A; 2 - диск B және т.б.). Функция, егер жоқ дискінің нөмірі көрсетілсе 1 деген мәнді береді.

Function EOF(Var F):Boolean; - файлдың соңын көрсетеді, егер файлдық көрсеткіш файлдың соңында тұрса, онда True мәнін береді. Жазуда бұл кезекті компонент файлдың соңына қосылғанын, ал оқуда файлдың жойылғанын көрсетеді.

Procedure Erase (Var F): - F файлын өшіреді. Процедураны орындар алдында файлды жабу қажет.

Function FileAge(const FileName: String):integer; - FileName файлы үшін оның соңы жаңартылған (қайта жазылған) уақытын береді (жүйелік форматта); егер мұндай файл болмаса, онда -1-ге тең.

Function ExcludeTrailingBackslash(const S:String): String; - S жолын тұйықтайтын "/" символын өшіреді (егер бұл символ жолды тұйықтамаса, онда S-ті өзгеріссіз қалдырады);

Function ExpandFileName(const FileName: String): String; - файл атын ағымдағы каталогпен толықтырады;

Function ExpandUncFileName(const FileName: String); String: - файл атын ағымдағы желілік каталогпен толықтырады;

Function ExtractFileDir(const FileName: String): String: - файлдың толық атауынан оған кіру маршрутын шығарады (соңғы "/" символынсыз);

Function ExtractFileExt(const FileName: String): String; - файлдың толық атауынан оның затын нүктесімен шығарады;

Function ExtractFileName(const FileName: String): String: String; - файлдың толық атауынан оның атын затымен шығарады;

Function ExtractFilePath(const FileName: String): String; - файлдың толық атауынан оған кіру маршрутын "/" символымен қоса шығарады;

Function ExtractRelativePath(const BaseName, DestName: String): String; - файлдың толық атауынан DestName-ге қатысты маршрут атын шығарады (аралық каталогтар: "/" символымен алмастырылады).

Function FileDateTimeToDateTime (DateTime: integer): TDateTime; - жүйелік форматы FileDate файлының құрылу уақытын дата-уақыт форматына түрлендіреді.

Function FileExists(const FileName: String):Boolean; - егер FileName атты файл бар болса, онда True мәнін береді;

Procedure FindClose(Var F:TSearchRec); - FindFirst/FindNext функциясымен файлды іздеуге бөлінген жадыны босатады;

Function FindNext(Var F:TSearchRec): Integer; - F файлының каталогтағы келесі файлдың атын береді;

Procedure Flush(var F); - дискідегі файлдың соңғы өзгертулерін сақтай отырып, файлдың ішкі буферін тазартады;

Procedure GETDir(D:Byte; var S:String); - ағымдағы каталогты береді: D- құрылғы нөмірі; 1 - A дискісі; 2 - B дискісі және т.б. S - String типті айнымалы, ол көрсетілген дискідегі ағымдағы каталогқа баратын жолды көрсетеді;

Function IncludeTrailingBackslash(const S:String):String; - «/» символымен файлға кіретін толық маршрутты береді;

Function IOResult: Integer; - соңғы енгізу- шығару операциясының шартты белгісін береді;

Function IsPathDelimiter(const S:String; Index: Integer):Boolean; - егер S жолындағы Index символында "/" белгісі болса, онда true мәнін береді;

Function MatchesMask(const FileName, Mask: String):Boolean; - егер FileName аты Mask топтық атауына сәйкес келсе, true мәнін береді;

Procedure MKDir(Dir: String); - көрсетілген дискіде жаңа каталог құрады: Dir каталогты іздеу маршруты. Бұрын құрылған каталогтар атауы қайталанбауы тиіс;

Procedure Rename(var F; NewName:String); - F файлының атын өзгертеді; NewName - файлдың жаңа атынан тұратын жолдық өрнек. Процедураны орындамас бұрын файлды жабу қажет;

Procedure Reset(Var F:File; [RecSize:Word]); - бұрын құрылған файлды ашады. RecSize-дың тек типтік емес файлдар үшін ғана мағынасы бар және мәліметтер блогының өлшемін береді;

Procedure Rewrite(Var F:File; [RecSize:Word]); - жаңа файл құрады. ResSize-дың тек типтік емес файлдар үшін ғана мағынасы бар және мәліметтер блогының өлшемін береді.

Procedure Rmdir(Dir:String); - Dir каталогын өшіреді. Өшірілетін каталог бос болуы тиіс.

Файлдарды бейнелеу үш тәсілмен жүзеге асырылады. Алдымен файл

Function FileCreate(FileName: String): Integer; функциясының көмегімен құрылып,

Function FileOpen(const Filename:String; Mode:LongWord): integer; функциясының көмегімен ашылады.

Mode параметрі файлға кіру режимін анықтайды және төмендегі мәндердің біреуін ғана қабылдайды: fmOpenRead – тек оқу үшін, fmOpenWrite – тек жазу үшін, fmOpenReadWrite – жазу және оқу.

Екінші кезеңде жадыда бейнелеу объектісі құрылады. Ол үшін төмендегідей функция пайдаланылады:

Function CreateFileMapping(hFile:Thandle; IpFileMapping Attributes: Psecurity Attributes; flProtect, dwMaximumSizeHigh, dwMaximumSizeLow: Dword; lpName: PChar) :Thandle;

Мұндағы hFile – файл дескрипторы; IpFileMapping Attributes – құрылатын объектінің ұрпақ объект құра алатындығы жөніндегі құрылымды анықтайтын көрсеткіш; flProtect – файлды бейнелеу терезесінде қолданылатын қорғау типін анықтайды; dwMaximumSizeHigh, dwMaximumSizeLow – сәйкес үлкен және кіші 32 разрядты файл өлшемі, егер ұзындығы 4 ГБайтка дейінгі файлды бейнелеу қажет болса, онда dwMaximumSizeHigh – нөл мәні меншіктеледі, ал dwMaximumSizeLow – файл ұзындығы, егер екі параметрде нөлге тең болса, онда бейнелеу терезесінің өлшемі файл өлшеміне тең; lpName – бейнеленетін объектінің атауы.

FlProtect параметрі – файлды көру терезесіне қолданылатын қорғау типін береді. Ол операциясының көмегімен FlProtect параметріне төмендегідей атрибуттарды жалғауға болады.:

Sec_Commit – бейнелеуге арналған физикалық жадыны бөліп көрсетеді; SEC_IMAGE – файл образынан атрибуттарды бейнелеу жөнінде ақпарат алынады; SEC_NoCaSHE – бейнеленетін мәліметтерге оперативті жадыдан орын бөлінбей, тікелей дискіге жазылады; SEC_RESERVE – физикалық жадыдан орын бөлінбей, бірден бөлім беттері резервтеледі.

Үшінші кезеңде көру терезесі құрылады, яғни мәліметтерді бағдарламаның адрестік кеңістігінде бейнелеу:

```
function MapViewOfFile (hFileMappingObject: Thande; dwDesires Access;
DWord; dwFileOffSetHigh, dwFileiffSetLow, dwNumberBytes- ToMap:DWord):
Pointer.
```

Мұнда hFileMappingObject – объектіні бейнелеу дескрипторы, dwDesires Access – мәліметтерге кіру тәсілін анықтайды және төмендегі мәндердің біреуін қабылдауы мүмкін: File_Map_Write – оқуға және жазуға мүмкіндік береді, бұдан CreateFillMapping функциясында PageReadONLY немесе PageReadWrite атрибуты пайдаланылуы мүмкін; File_Map_All_Access атрибутының жұмысы File_Map_Write жұмысымен бірдей; File_Map_Copy – мәліметтерді оқуға және жазуға болады, бірақ жаңартылған мәліметтер жадының жеке қорғалған бөлігінде сақталады; CreateFileMapping функциясында Page Write атрибуты пайдалануы тиіс; dwFileOffSetHigh, dwFileiffSetLow – файл басынан ығысатын сәйкес үлкен және кіші разрядтарды анықтайды және сол жерден бейнелеу орындалады; dwNumberBytesToMap – бейнелеу терезесінің ұзындығын анықтайды.

Бейнеленген мәліметтерді пайдаланғаннан кейін терезені бейнелеу ресурстарын

```
Function UnMapViewOfFile(lpBaseAdress:Pointer):Bool;
```

функциясының көмегімен босату керек, оны шақыратын жалғыз параметр бірінші бейнеленген байттың адресінен тұруы тиіс, яғни ол MapViewOfFile функциямен шақырылатын адрес. Бейнелеу объектісін және файлдың өзін жабу Function CloseHandle (hObject:THandle) функциясының көмегімен орындалады.

1-есеп. Мәліметтерді файлға жазатын бағдарлама.

```
procedure TForm1.Button1Click(Sender: TObject);
```

```
var
```

```
  f: TextFile;    // файл
```

```
  fName: String[80]; // файл аты
```

```
  i: integer;
```

```
begin
```

```
  fName := Edit1.Text;
```

```
  AssignFile(f, fName);
```

```
  Rewrite(f); // жазу үшін файлды ашу
```

```
  // файлға жазу
```

```
  for i:=0 to Memo1.Lines.Count do // жол нолден нөмірленеді
```

```
    writeln(f, Memo1.Lines[i]);
```

```
  CloseFile(f); // файл жабу
```

```
  MessageDlg('Мәліметтер файлға жазылды ',mtInformation,[mbOk],0);
```

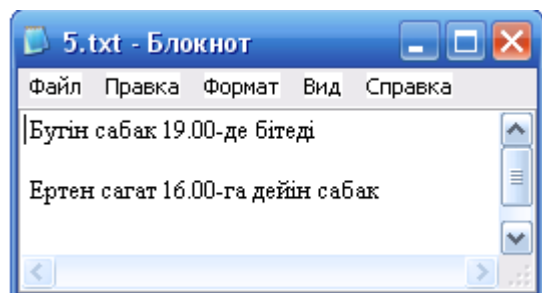
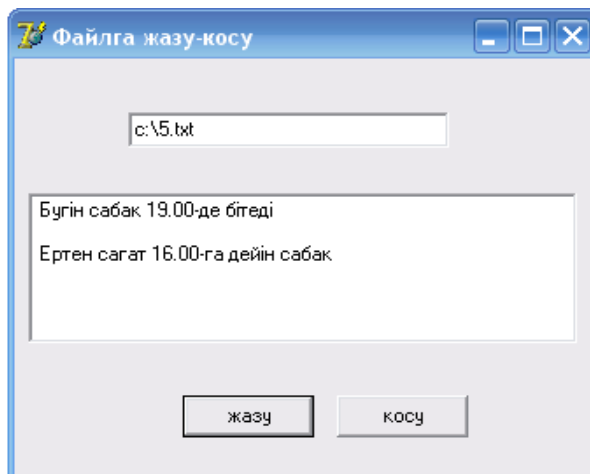
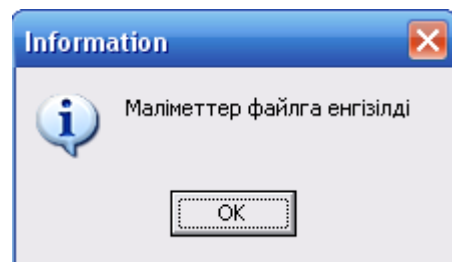
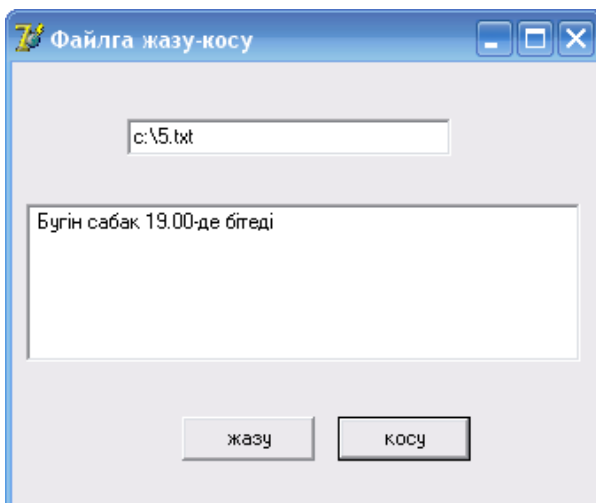
```

end;

procedure TForm1.Button2Click(Sender: TObject);
var
  f: TextFile;    // файл
  fName: String[80]; // файл аты
  i: integer;
begin
  fName := Edit1.Text;
  AssignFile(f, fName);
  Append(f); // косу үшін ашу

  // файлға жазу
  for i:=0 to Memo1.Lines.Count do // жол нолден нөмірленеді
    writeln(f, Memo1.Lines[i]);
  CloseFile(f); // файлды жабу
  MessageDlg('Мәліметтер файлға енгізілді', mtInformation, [mbOk], 0);
end;
end.

```



2-есеп. Файлдан оқитын бағдарлама.

```
procedure TForm1.Button1Click(Sender: TObject);
var
  f: TextFile;
  fName: String[200];
  buf: String[200];

begin
  fName := Edit1.Text;
  AssignFile(f, fName);

  {$I-}
  Reset(f);
  {$I+}
  if IOResult <> 0 then
    begin
      MessageDlg(' Файлға ену кате ' + fName,
                 mtError,[mbOk],0);
      exit;
    end;

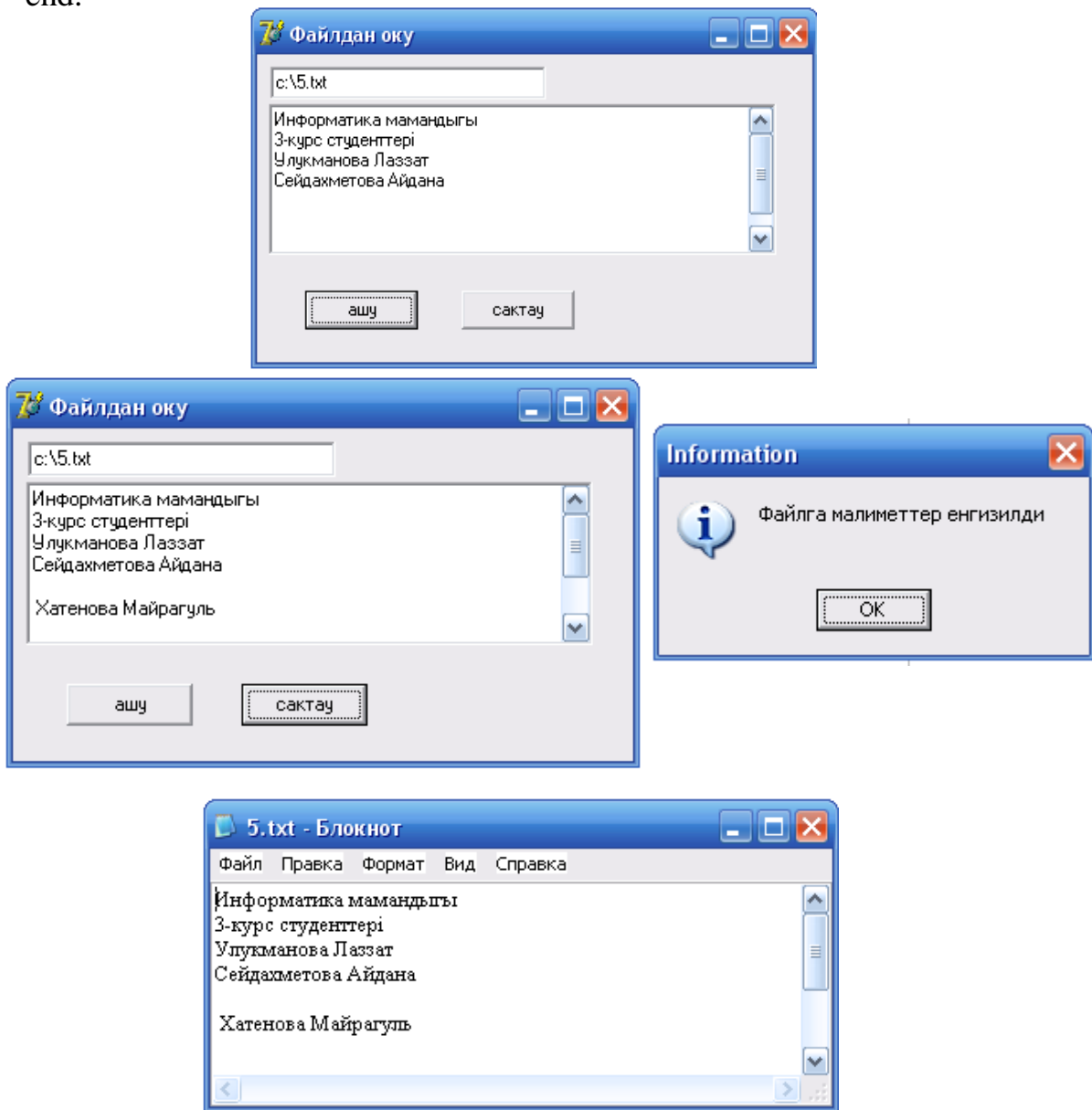
  while not EOF(f) do
    begin
      readln(f, buf);
      Memo1.Lines.Add(buf);
    end;

  CloseFile(f);
end;
```

```
procedure TForm1.Button2Click(Sender: TObject);
var
  f: TextFile;
  fName: String[200];
  i: integer;
begin
  fName := Edit1.Text;
  AssignFile(f, fName);
  Rewrite(f);
  for i:=0 to Memo1.Lines.Count do
    writeln(f, Memo1.Lines[i]);
```

```
CloseFile(f);
```

```
MessageDlg('Файлга малиметтер енгізілді',mtInformation,[mbOk],0);  
end;  
end.
```



3-есеп. Компьютердің жинақтауыш қаты магниттік дискісіндегі, иілгіш магниттік дискілердегі және оптикалық дискілердегі мәліметтерден қажетті файлды бумадан іздейтін бағдарлама.

//көрсетілген каталог немесе ішкі каталогтен файлды іздеу үшін Find рекурсивті процедурасы қолданылады

```
unit FindFile_  
interface  
uses  
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
```

```

Dialogs, StdCtrls,
FileCtrl;
type
  TForm1 = class(TForm)
    Edit1: TEdit;
    Edit2: TEdit;
    Memo1: TMemo;
    Button1: TButton;
    Button2: TButton;
    Label1: TLabel;
    Label2: TLabel;
    Label3: TLabel;
    Label4: TLabel;
    procedure Button1Click(Sender: TObject);
    procedure Button2Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;
var
  Form1: TForm1;
implementation
{$R *.dfm}

var
  FileName: string;
  cDir: string;
  n: integer;

// ағымдағы каталогтен файлды іздеу
procedure Find;
var
  SearchRec: TSearchRec;
begin
  GetDir(0,cDir);
  if cDir[length(cDir)] <> '\' then cDir := cDir+'\'';
  if FindFirst(FileName, faArchive,SearchRec) = 0 then
    repeat
      if (SearchRec.Attr and faAnyFile) = SearchRec.Attr then
        begin
          Form1.Memo1.Lines.Add(cDir + SearchRec.Name);
          n := n + 1;
        end;
    until FindNext(SearchRec) <> 0;
end;

```



```

// ағымдағы каталогтың ішкі каталогтерін өңдеу
if FindFirst('*', faDirectory, SearchRec) = 0 then
  repeat
    if (SearchRec.Attr and faDirectory) = SearchRec.Attr then
      begin
        if SearchRec.Name[1] <> '.' then
          begin
            ChDir(SearchRec.Name);
            Find;
            ChDir('..');
          end;
        end;
      until FindNext(SearchRec) <> 0;
end;

```

```

// пайдаланушының таңдалынған каталогын шақыру
function GetPath(mes: string):string;
var
  Root: string;
  pwRoot : PWideChar;
  Dir: string;
begin
  Root := "";
  GetMem(pwRoot, (Length(Root)+1) * 2);
  pwRoot := StringToWideChar(Root,pwRoot,MAX_PATH*2);
  if SelectDirectory(mes, pwRoot, Dir)
  then
    if length(Dir) = 2
    then GetPath := Dir+'\'
    else GetPath := Dir
  else
    GetPath := "";
end;

```

```

// Іздеу батырмасы
procedure TForm1.Button1Click(Sender: TObject);
begin
  Memo1.Clear;
  Label4.Caption := "";
  FileName := Edit1.Text;
  cDir := Edit2.Text;
  n:=0;
  ChDir(cDir);
  Find;

```

```

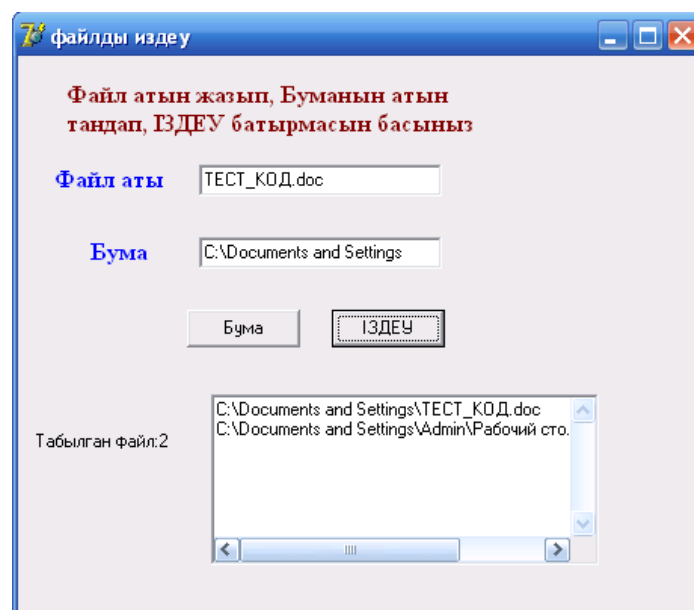
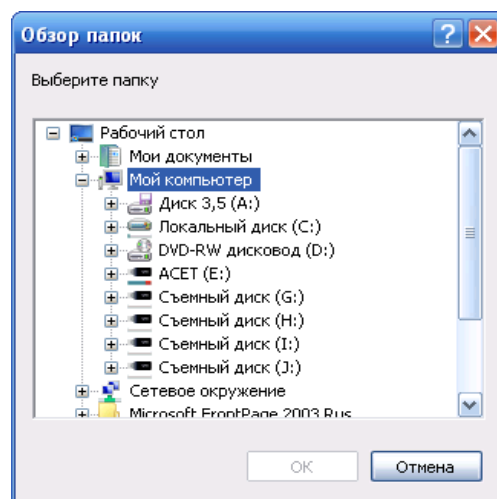
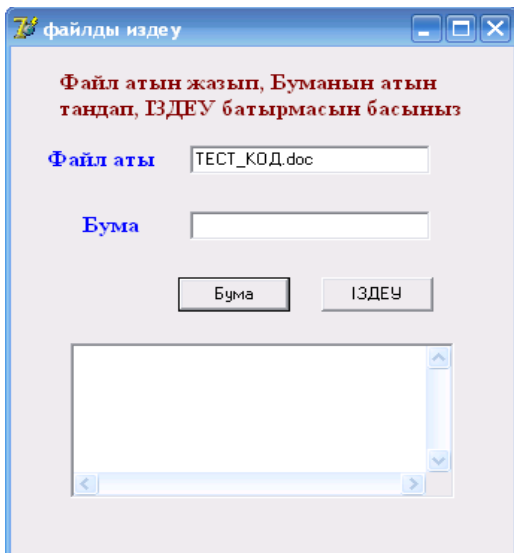
if n = 0 then
    ShowMessage('Іздеу критерийі бойынша канагаттанарлык файл
жок.')
```

else Label4.Caption := 'Табылган файл:' + IntToStr(n);

```

end;

// Бума батырмасы
procedure TForm1.Button2Click(Sender: TObject);
var
    Path: string;
begin
    Path := GetPath('Бума танданыз');
    if Path <> ''
        then Edit2.Text := Path;
end;
end.
```



Тапсырмалар:

1. Файлдан мәтінді оқып, сөйлемдердегі ең қысқа сөздерді тауып, оларды іріктеп экранға шығаратын бағдарлама құру.
2. Файлдан мәтінді оқып, экранға тек сандық деректері бар сөйлемдерді іріктеп шығару бағдарламасын құру.
3. А және В файлдары берілген. Реттерін сақтай отырып С файлына бірінші А файлының, ал содан соң В файлының компоненттерін жазыңдар.
4. Файлдан мәтінді оқып, тек үтір кездесетін сөйлемдерді экранға іріктеп шығаратын бағдарлама құру.
5. Файлдан мәтінді оқып, берілген мәтіндегі тыныс белгісі ең көп кездесетін сөйлемді экранға шығаратын бағдарлама құру.
6. А символдық файл берілген. Файлдағы әрбір леп белгісін нүктеге ауыстырып, А файлының компоненттерін В файлына жазатын бағдарлама құру.

Бақылау сұрақтары:

1. Файл дегеніміз не?
2. Файл типінің қанша түрі бар?
3. Тізбекті кіру мүмкіндігі бар файл дегенді қалай түсіндіресіз?
4. Типтік файл дегеніміз не?
5. Типтік емес файл дегеніміз не?
6. Файлдармен орындалатын жұмыс түрлері қандай?
7. Логикалық және физикалық файл дегеніміз не, олар қалай байланыстырылады?
8. Файлды ашудың түрлері қандай және олар қандай операторлармен орындалады?
9. Файлдармен жұмыс істеуге арналған қандай процедуралар мен функциялар бар?
10. Файлдарды бейнелеу тәсілдерін атаңдар және ол қалай жүзеге асырылады?

Әдебиеттер

1. Иванова Г.С., Ничушкина Т.Н., Пугачев Е.К. Объектно-ориентированное программирование: Учеб. для вузов. – М.: Изд-во МГТУ им. Н.Э.Баумана, 2001. – 320 с.
2. Архангельский А.Я. Программирование в Delphi 7. – М.: ЗАО «Издательство БИНОМ», 2005. – 1120 с.
3. Фаронов В. Delphi 7: учебный курс. - СПб.: «Питер», 2006. – 512 с.
4. Фаронов В.В. DELPHI. Программирование на языке высокого уровня. - СПб.:Питер, 2004. – 640 с.
5. Культин Н.Б. Delphi в задачах и примерах. – СПб.: БХВ-Петербург, 2006. – 288 с.
6. Халыкова К.З., Тұрғанбаева А.Р., Бостанов Б.Ғ. Программалау тілдерін оқыту. Delphi ортасы. Оқу құралы. - Алматы, 2005. – 217 б.
7. Орлик С.В. Секреты Delphi на примерах. - М., 1996. – 136 с.
8. Прищепов М.А., Севернёва Е.В., Шакирин А.И. Програмирование на языках Basic, Pascal и Object Pascal в среде Delphi. – Минск, 2006.
9. Юркин А.Г. Задачник по программированию. – СПб.: Питер, 2002.
10. Суворов К.Ф., Черемных М.Н. Справочник DELPHI. - СПб.: БХВ-Петербург, 2004.
11. Нурбекова Ж.К. Практикум решения задач на Delphi 5.0 – Павлодар, 2001. – 105 с.

М А З М Ұ Н Ы

Алғы сөз	3
1. Delphi ортасымен танысу	4
2. Сызықтық алгоритмдерді бағдарламалау	19
3. Тармақталу алгоритмдерін бағдарламалау	31
4. Таңдау командасы	49
5. Қайталанатын алгоритмдерге бағдарлама құру (For қайталану операторы)	54
6. Қайталанатын алгоритмдерге бағдарлама құру (WHILE, REPEAT қайталану операторлары)	61
7. Мәлімет типтері	70
8. Жолдық және символдық типтер	79
9. Бір өлшемді массивтерді өңдеу алгоритмдері	89
10. Екі өлшемді массивті өңдеу алгоритмдері	104
11. Жазулар. Жиындар	114
12. Процедуралар мен функциялар	121
13. Delphi-дің графикалық және мультимедиалық мүмкіндіктері	132
14. Форманың атрибуттары және стилі. Форманы құру және өшіру. Қосымшада бірнеше форманы қолдану	158
15. Мәліметтерді файлдан оқу және файлға жазу	168
Әдебиеттер	180