

ЎЗБЕКИСТОН РЕСПУБЛИКАСИ ОЛИЙ ВА ЎРТА МАХСУС
ТАЪЛИМ ВАЗИРЛИГИ

ТЕРМИЗ ДАВЛАТ УНИВЕРСИТЕТИ

ФИЗИКА МАТЕМАТИКА ФАКУЛЬТЕТИ

Амалий математика ва информатика кафедраси

Менглиев Ш.А.

“Delphi дастурлаш тили”
фанидан маърузалар матнлари

Термиз – 20__й.

Ушбу маруза матни учун кўрсатма илмий-услубий кенгашнинг _____ йил
« ____ » _____ бўлиб ўтган ____ -сонли мажлисида кўриб чиқилди ва чоп
этишга тавсия этилди.

“Delphi дастурлаш тили” фанидан маруза матни / ТерДУ ____ б. Термиз 20_ й.

Такризчилар:

ф.м.-ф.н. Нормуродов Ч. Амалий математика ва информатика кафедраси
кафедра мудири

и.ф.н О. Садатов Амалий математика ва информатика кафедраси доценти

Тузувчи: Амалий математика ва информатика кафедраси ўқитувчиси Менглиев
Ш.А.

Кириш

Ҳозирги вақтга келиб компьютер оламида кўплаб дастурлаш тиллари мавжуд. Улар Бейсик, Паскал, Си, С++, Delphi ва бошқа дастурлаш тилларидир. Паскал тили 1969 йил Н. Вирт томонидан яратилган бўлиб, кейинчалик американинг Borland фирмаси томонидан қайта ишланиб Turbo Pascal деб номланди. Turbo Pascal ни қайта ишлаш натижасида объектли дастурлаш йўлга қўйилди ва у Object Pascal деб атала бошланди. Ҳисоблаш техникаси ва технологиясининг ривожланиши натижасида Borland фирмаси томонидан янги Delphi дастурлаш тили яратилди.

Delphi дастурлаш тили Windows учун мўлжалланган бўлиб, унинг биринчи версияси Windows 3.1 операцион система қобиғида ишлаган. Windows 95 операцион система яратилганидан сўнг, 16-разрядли Delphi 2, кейинроқ 32-разрядли Delphi 3 версияси яратилди. Windows 98 операцион системаси учун Delphi нинг тўртинчи версияси ва ҳозирги кунда Delphi 5, Delphi 6, Delphi 7, Delphi 8 ва ҳақлар пайдо бўлди.

Delphi дастурлаш тили – бу дастурларни қайта ишлаш муҳити бўлиб, Windows операцион системасида ишлайди. Унда объектли дастурлаш тили бўлган Object Pascal муҳассамлашган.

Delphi визуал проектлар, турли ҳолат процедураларини қайта ишлаш ва дастурларни қайта ишлашда вақтдан ютиш ва бошқаларни ўз ичига олади.

DELPHI дастурий востаси-бу Windows учун мўлжалланган дастурлаш муҳити бўлиб, 1995 йилда BORLAND компанияси гуруҳи дастур тузувчилари Чак (Chuck) ва Денни (Danny) томонидан яратилган.

Бу тил ўзининг кенг қамровли имкониятларига эгаллиги билан бирга, бошқа дастурлаш тилларидан ўзининг баъзи бир хусусиятлари билан ажралиб туради.

Borland Delphi нинг пайдо бўлиши дастурлашни ривожлантириш тарихида ёрқин кўриниш бўлди. Delphi нинг дунёга келишига қуйидагилар сабаб бўлди;

- Windows учун дастулаш ва компоненталар технологияси.
- Масалаларни ечиш учун объектга йўналтирилган усул.
- Компоненталар технологиясига асосланган иловаларни тез яратишнинг визуал муҳитлари.
- Интерпритациядан эмас, компиляциядан фойдаланиш. Бу шундан иборатки, интерпретатор билан ишлашга қараганда компилятор билан ишлаш тезлиги ўн мартага устунликка эга бўлади.
- Универсал усуллар ёрдамида маълумотлар базаси билан ишлаш имкониятларининг мавжудлиги.

Borland Delphi юқорида баён этилган тенденцияларни жорий этиш мақсадида яратилган. Аммо, унинг энг асосий элементи Паскаль тили бўлиб ҳисобланади. Ҳозирги кунда Delphi тилининг бир неча хил вариантлари мавжуд (Delphi 2, Delphi 3... , Delphi 10 ва ҳ.к.).

Delphi дастурлаш тили учун Паскаль дастурлаш тили асос қилиб олинган. Бу икки тилнинг қандай ўхшашлик ва фаркли томонлари бор?

Биламизки, Паскалда бирор катгароқ дастур, масалан амалий дастурлар мажмуини тузмоқчи бўлсак, албатта биз модулли дастурлашдан фойдаланамиз. Яъни қўйилган масалани кичикроқ бўлақларга бўлиб оламиз ва шулар билан ишлаймиз. Натижани олиш учун эса, бу бўлақларни бирлаштирувчи битта асосий дастур қилинади. Айнан шу нарса Delphi дастурлаш тилининг асоси ҳисобланади. Агар биз Delphi да бирор форма ҳосил қилсак, у ҳолда Delphi бизга ўзи автоматик тарзда шу формага мос бўлган модулларни ва бу модулларни бошқарувчи дастурни тузиб беради. Бошқарувчи дастурни Delphi да, агар унга фойдаланувчи ном бермаса, Project1.dpr деб номлайди. Бунда .dpr файл кенгайтмаси. Автоматик тарзда ташкил қилинган модулга фойдаланувчи ном бермаса, у ҳолда Delphi уни Unit1.pas деб но-

млайди. Фойдаланувчи эса .pas кенгайтмасини сақлаган холда хохлаган номини бериши мумкин.

Delphi'ni ishga tuшириш

Delphi ni ikki usulda ishga tuшириш mumkin:

1. Пуск (Start) кнопкаси босилади, Программы сатри танланади ва Borland Delphi 4 сатридан Delphi 4 ойнасига кириб устига сичқончани чап тугмасини босиш билан ;
2. Ишчи столга ўрнатилган ёрликни устига сичқонча кўрсаткичини ўрнатиб, чап тугмасини икки марта босиш билан (Ёрликни фойдалувчини ўзи яратиб олиши керак).

Delphi da бошлангич амаллар ва проеактлар.

Delphi ni ishga tuширгандан кейин уни экран кўриниши хосил бўлиб, у унчалик оддий эмас. Экранда бешта ойна хосил бўлади:

- бош ойна-Delphi Project 1;
- форма ойнаси -Form1,
- объект хоссаларини тахрирлаш ойнаси- Object Inspector;
- объектлар руйхатини кўриш ойнаси – Object tree View;
- дастур кодларини тахрирлаш ойнаси– Unit1.pas

Бош ойна экраннинг юқори қисмида жойлашган бўлиб, унинг биринчи қаторида сарлавха, яъни проектнинг номи жойлашган. Иккинчи қаторда буйруқлар менюси горизонтал кўринишда жойлашган. Кейинги қаторнинг чап тарафида ускуналар панели ва ўнг тарафида компоненталар палитраси жойлашган.

Object Inspector ойнаси ёрдамида объектлар хусусиятларини ўзгартириш мумкин: формалар, буйруқ тугмалари, киритиш майдонлари ва ҳоказоларни. Масалан, Width ва Height хоссалари форма ўлчамини, Top ва Left эса форманинг экрандаги ҳолати, Caption-сарлавха матнини аниқлайди. Delphi da дастурлаш иккита ўзаро таъсир этувчи бир бири билан боғлиқ жараён асосида ташкил қилинади:

-дастурни визуал лойихалаш жараёни;

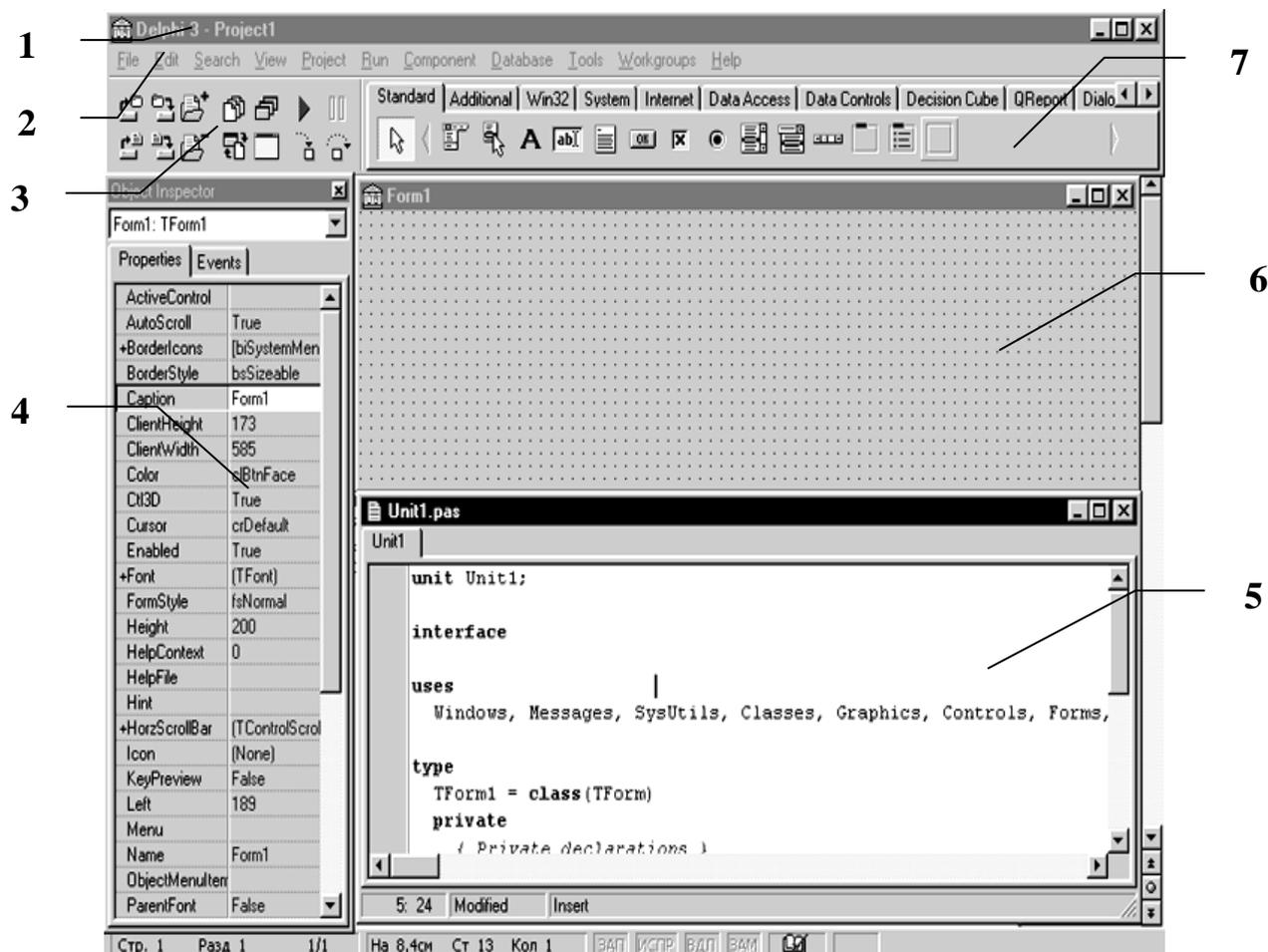
-дастур кодларини киритиш жараёни.

Delphi da дастурлаш форма ойнасини ташкил этишдан бошланади. Оддий дастур иловасини яратиш кетма-кет File-New-Application буйруғини бериш билан бошланади.

Тузилган дастурни ЭХМ хотирасида сақлаш учун асосий менюнинг File бандидан фойдаланилади. Дастурни компиляция қилиш учун асосий менюнинг RUN бандидан

фойдаланилади. Агар дастурда хатоликлар бўлса, компиляция қилиш жараёнида бу хатоликлар кўрсатиб борилади. Агар хатолик бўлмаса, дастур ишга тушади.

Delphi дастури ойнасининг умумий кўриниши куйидагича. (1-расм).



1-расм

- 1- Асосий ойна; 2 – меню сатри; 3 – пиктограмма меню сатри;
- 4 – объектлар инспектори ойнаси; 5 – дастур матн ойнаси;
- 6- форм дизайни; 7 – компоненталар менюси

Меню сатрининг кўриниши



Delphi асосий меню бандлари вазифалари Microsoft office стандарт дастурларининг меню бандлари вазифаларидан деярли фарқланмайди ва улардан фойдаланиш тартиби ҳам худди шундай сакланади.

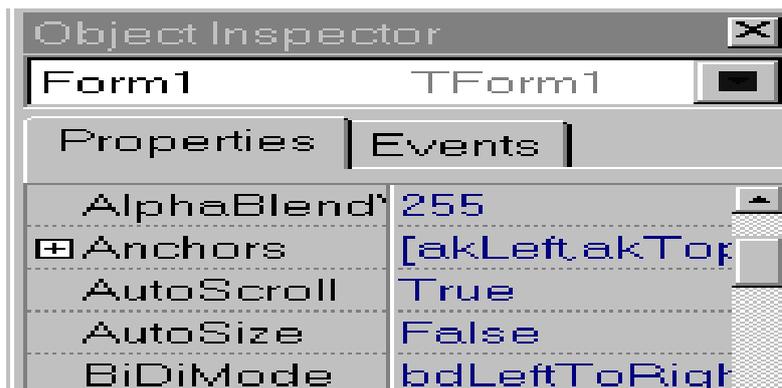
Асбоблар панели ва компоненталар рўйхати



Объектлар дарахти



Объектлар инспектори



Дастур формаси



Объектлар инспекторидан фойдаланиш

Windows учун тузилган программанинг ички тузилиши консоль программаларнинг ички тузилишидан фарқланади. Дастур бажарилишида калит сўзлар бўлган `begin` ва `end` орасига олинган операторларни `windows` бошқача шаклда бошқаради, яни уларни ҳодиса деб қарайди. Ҳар бир ҳодисанинг ўзига мос хоссалари мавжуд. Дастурлашда бу хоссалардан керагини танлаб ишлатилади. Танлаш объектлар инспекторида бажарилади. Объектлар инспектори икки қисмдан иборат `Properties` яни ходисалар ва `Events` хоссалар. Delphi нинг характерли томони шундан иборатки, агар биз бирор компонентадан фойдалансак, объектлар инспектори унга мос ходисаларни ажратиб кўрсатиб туради. Масалан биз бирор янги форма ҳосил қилган бўлсак, объектлар инспекторида `caption`да `Form1` ёзуви туради. Агар биз хоҳласак `Form1`нинг ўрнига янги ўзимизнинг программага мос номни беришимиз мумкин. Ҳар бир компонентга объектлар инспекторида унга тегишли ходиса ва шу ходисага мос хоссалар бўлади.

Delphi дастурлаш тилининг яна бир характерли томони шундан иборатки, агар биз дастурда бирор тугмага қандайдир вазифани юкламоқчи бўлсак, шу тугманинг устида сичқончани икки марта чертилса, шу тугмага мос келувчи процедуранинг ичига автоматик равишда киритиб қўяди. Бундан ташқари дастур тузиш жараёнида формадан модулнинг ичига ва модулдан формага ўтишга эҳтиёж бўлади, бунинг учун асбоблар панелидан `Toggle Form-Unit` тугмаси танланади ёки `F12` тугмасини босиш билан амалга оширилади.

Формалар билан ишлаш

Delphiда тузиладиган дастурлар бирор форма асосида тузилади. Delphi да ҳар бир янги ҳосил қилинган формага унга мос бўлган модул автоматик ташкил қилиниб турилади. Бу эса дастурчи учун жуда қулай имконият, яни унинг ишини тезлаштиради.

Бу форманинг шаклини танлаш, унда компоненталарни жойлаштириш фойдаланувчи ихтиёрида бўлади. Формага бирор компонентани қўймоқчи бўлсак, шу компонентанинг устида сичқончанинг чап тугмаси икки марта босилади. Бу компонента форманинг ўртасига жойлашади. Биз уни хоҳлаган жойга суриб кўчиришимиз мумкин.

DELPHI да кодлар муҳити

Delphi кодлар муҳити автоматик тарзда `Object Pascal` дастурлаш тилидаги калит сўзларни (`begin`, `end`, `procedure`, `const`, `var`) қалин ҳарфлар билан белгилайди.

Маълумот ёзилган `сатр` (дастур изохи) ни белгилаш учун фигурали кавслардан фойдаланилади. Кавс очилса ундан кейин турган кодлар кўриниши ўзгаради. Керакли жойда кавс беркитилса кўриниши ўзгарган кодлар фақат кавс оралиғидагина қолади ва дастур ишлаш жараёнида шу оралиқ ишлатилмайди.

Delphi кодлар муҳитининг имкониятларидан яна бири шуки, у ерга бирор функцияни масалан: «`StrToFloat`» ни ёзиб, кавс очсак `сатр` остида кичик ойна ҳосил бўлади. Бу ойнада кавс ичидаги ўзгарувчи типи кўрсатилган бўлади, ёки бирор операторни масалан: `Label1` ни ёзиб нукта қўйилса `сатр` остида нуктадан кейинги ёзиш мумкин бўлган операторлар рўйхати чиқади ва улардан кераклигини танлаб қўйишимиз мумкин.

Кодлар ойнасида бирор оператор устига курсорни олиб бориб `Ctrl+F1` тугмалари тенг босилса шу оператор ҳақидаги ёрдам ойнаси ҳосил бўлади. У ердан керакли ахборотни олиш мумкин. Агар курсорни бўш жойга олиб келиб `F1` босилса умумий ёрдам файллари чиқади.

Кодлар ойнасида тахрирлаш оддий `матн` муҳаррирлари каби амалга оширилади. Яъни белгиланган (блокка олинган) код нусхасини олиш, қирқиб олиш ва керакли жойга қўйиш мумкин. Ундан ташқари кодлар ичидан керакли белгини излаб топиш ва алмаштириш, `Delete` тугмаси ёрдамида курсордан кейин турган белгини, `Backspace` ёрдамида эса курсордан олдин

турган белги ёки белгиларни ўчириш мумкин. Ctrl+→, Ctrl+← клавишлари ёрдамида бир сўз кейинга ва олдинга, PgDn, PgUp клавишлари ёрдамида эса бир экран пастга ва юкорига ўтилади.

Проект

Delphi проекти – бу компилятор томонидан, дастур яратгандан сўнг, яратилган дастурга тегишли бўлган файллар тўпламидир. Проект, бир ёки бир нечта проект файлларини ва модулларни ўз ичига олади (Unit модули).

Проект файли *.dpr кенгайтмасига эга бўлиб, проектнинг умумий ҳолатини ўзида сақлайди. Проект модули файли эса *.pas кенгайтмали бўлиб, ишчи файлини яратишда компиляторга керак бўлувчи процедура, функция матнлари, типларни тавсифи ва бошқа маълумотларни ўзида сақлайди.

Delphi мухити.

Delphi дастурлаш тилини ишга туширганимизда унинг ишчи экран кўринишини кўрамиз. Дастур мухитидан буйруқлар менюси, буйруқ тугмачалари, компонентлар палитраси, объект инспектор, форма ва уни ортида тахрирлагич ойнаклари жой олган бўлиб деярли экранни тўлдириб туради.

Буйруқ менюси.

Delphi нинг меню сатридан қуйидагилар жой олган.

File, Edit, Search, View, Project, Run, Component, Database, Tools, Help.

Буларнинг барчасида ост менюлар мавжуддир. File нинг ост менюсида бир неча буйруқлар бўлиб улар ёрдамида янги проект очиш, янги форма очиш ва уларни сақлаш мумкин шу билан биргаликда очилган проектни ёпиш, Delphi дан чиқиш ва шуларга ўхшаш файллар билан ишлаш имкониятлари бор.

Edit менюси ост менюларидан фойдаланиб кодларни тахрирлаш, умуман кодлар устида турли хил амалларни бажариш мумкин.

View ёрдамида эса Delphi ишчи мухити кўринишининг ўзгариши мумкин.

Run менюси ёрдамида дастурни ишга туширишни турли йўллари амалга оширилади.

Database менюсида маълумот баъзасини ташкил қилиш мумкин.

Help менюси эса Delphi ва унда дастурлаш ҳақидаги барча маълумотларни олиш имкониятини яратади.

Буйруқлар тугмачаси

Буйруқлар тугмачаси ёрдамида янги формалар яратиш, мавжуд файлни очиш, Дастурни сақлаш, янги форма яратиш ва шунга ўхшаш амаллар тез бажарилади.

Компонентлар палитраси

Бу ерда стандарт ёки дастурчилар томонидан яратилган компонентлар мавжуд бўлиб, улардан тез ва сифатли дастурлар яратишда фойдаланилади.

Object Inspector ойнаси

Object Inspector ойнаси қуйидаги объектларнинг ҳолатини ўзгартиради: формалар, буйруқлар тугмачаси, кодлар майдони ва бошқалар.

Дастур формаси

Дастур тузишда ишлатиладиган барча компонентлар дастур формасига жойланади ва ана шу ердан уларга ўзгартириш киритилиши мумкин. Дастур ишга туширилгандан сўнг, барча амаллар дастур формаси ёрдамида бажарилади.

Дастур коди

Дастур коди форма орқасига яширинган бўлиб, у ерга дастур матнлари киритилади. У ойнага F12 ёки Ctrl+F12 тугмалари ёрдамида ўтиш мумкин. Delphi дастурлаш мухитида ишлаш жараёнида қуйидаги кенгайтмали файллар ишлатилади:

-лойиха файли, кенгайтмаси .dpr;

-паскал модули файли, кенгайтмаси .Pas;

-компонентлар жойлашган файл, кенгайтмаси .dcu;

-формалар жойлашган файл, кенгайтмаси .dfm;

-маълумотлар базаси файли, кенгайтмаси.dbf .

Компиляция этапида тайёрланган дастур матни Object Pascal тилига ўтказилади. Компановка босқичида эса керакли қўшимча ёрдамчи дастурлар ва остдастурлар унга бирлаштирилади. F9 тугмасини босиш билан Save Unit As диалог ойнаси пайдо бўлади ва у фойдаланувчидан Unit.pas модули учун файл номини ва у жойлашадиган папка номини кўрсатишингизни сўрайди. Агар жой кўрсатилмаса Delphi тизими автоматик равишда дастурни Bin папкасига жойлаштиради. Дастур компиляция қилиниш пайтида Delphi системаси .Pas; .dfm; .dcu кенгайтмали модуллар тузади. .Pas кенгайтмали файл кодларни ёзиш ойнасига киритилган дастур матнини, .dfm форма ойнаси ташкил этувчиларини, .dcu кенгайтмали файл эса .Pas ва .dfm кенгайтмали файлларнинг биргаликдаги машина кодига ўтказилган вариантини сақлайди. .dcu кенгайтмали файл компилятор томонидан ташкил қилинади ва ягона ишчи (бажарилувчи) .exe кенгайтмали файл ташкил қилишга база яратади.

Label, Edit, Memo матн компонентлари ва Button тугмачаси.

Label белгиси. Бу белги тушунтиришлар, номлар, мавзулар ва бошқа матнли маълумотларни экранга жойлаштириш учун ишлатилади. Белги учун Caption асосий хоссалардан бири бўлиб, унда экранга чиқариладиган матн жойлашади. Матнни экранга жойлаш учун Delphi нинг Standart палитрасидан «A» пиктограммаси белгиланиб форма устига келинади ва сичқонча тугмасини босган ҳолда матнга жойлаштирилиши лозим бўлган жой ажратилади. Натижада label1 матн майдони ҳосил қилинади ва Caption хоссасига кирилиб матн терилади. Матнга ишлов бериш учун (масалан, катталаштириш ёки кичиклаштириш; курсив ёки қалин қилиш ва бошқалар) керакли хосса танланиб улар ўзгартирилади. Кирилган матнни катталаштириш ёки кичиклаштириш учун олдин матн майдони ажратилиб, кейин Font хоссасига кирилиди ва мулоқот дарчасидан shrift, унинг ўлчами ва ранги танланиб ok тугмаси босилади.

Label компонентаси нафақат маълумотларни экранга жойлаштириш учун хизмат қилади, балки дастур натижаларини чиқаришда ҳам ишлатиш мумкин. Бунинг учун дастурда Label 5.caption:= 'дастур натижаси'; буйруғи берилиши керак. Мисол: Label 5.caption:= 'ечим=' +s; бу ерда s:string ўзгарувчиси.

Edit киритиш қатори.

Edit киритиш қатори матнни бир қатордан киритиш ва уни тахрирлаш учун ишлатилади. Матн киритиш қаторини экранга жойлаш учун Delphi нинг Standart палитрасидан «ab» пиктограммаси белгиланиб, форма устига келинади ва сичқонча тугмасини босган ҳолда матн киритилиши лозим бўлган жой ажратилади. Натижада Edit 1 матн киритиш майдони ҳосил қилинади. Матнни киритиш дастур ишчи ҳолатга ўтганда бажарилади. Матн қаторига киритилган маълумот фақат матн, яъни String (қатор) бўлиб ҳисобланади. Edit киритиш қаторига киритилган маълумот дастурда ўқиб ва уни рақамга ўтказиш учун кўп ҳолларда val функциясидан фойдаланилади. Бу функция Turbo Pascal да қуйидагича ёзилади.

val(edit 1.text,a,cod)- бу ерда a:real;- ўзгарувчиси бўлиб, edit 1.text майдонидаги маълумотни рақам қилиб ўзлаштиради. cod: integer; деб эълон қилинади.

Memo матн чиқариш қатори.

Memo матнларни бир неча қатор қилиб чиқариш учун ишлатилади. Memo матн чиқариш қаторини экранга жойлаш учун Delphi нинг Standart палитрасидан «ab» пиктограммаси ёнидаги Memo тугмачаси белгиланиб форма устига келинади ва сичқонча тугмасини босган ҳолда матн чиқарилиши лозим бўлган жой ажратилади. Натижада Memo 1 матн чиқариш майдони ҳосил қилинади. Бу чиқариш майдони дастурда натижаларни чиқаришда қўл келади. Натижани чиқаришда у дастур ичида қуйидагича ишлатилади.

Memo 1.lines.add('ечим=' +s);

Memo майдонини тозалаш эса натижани чиқаришдан олдин модулда Memo 1.clear; буйруғини бериш билан амалга оширилади.



Memo компонентаси содда матн муҳарири бўлиб, у кўп сатрли маълумотларни киритиш ва чиқариш учун ишлатилади.

Бу компоненталар мантлар устида амаллар бажариш учун ишлатилади.

Password Char хоссаси. Ушбу хосса матнни киритишдан химоя учун фойдаланилади. Парол киритиш керак пайтида фойдаланилади. Ушбу хоссадан фойдаланувчи махфийликнинг қўшимча даражасини таъминлаш учун ва фойдаланувчи исмини яшириш учун фойдаланиш мумкин.

Read Only хоссаси. Ушбу хосса объектнинг қийматини тахрирлаш имкониятини бошқаради. Фойдаланувчи унинг қийматини **False** ёки **True** қилиб қўйиши мумкин. Агарда хоссанинг қиймати **True** бўлса фойдаланувчи ушбу объектдан фойдаланишга, яъни ихтиёрлий матнни белгилаш ва нусхасини буферга олиш учун бирмунча рухсат беради.

Max Length хоссаси. Ушбу хоссадан учала объектга киритиладиган символларни чеклаш учун фойдаланилади.

Mask Edit компонентасининг умумий хоссаларидан ташқари қўшимча хоссаларга ҳам эгадир. Ушбу хоссадан бу объект орқали клавиатурадан киритиладиган символларни филтерлаш учун фойдаланилади. Маска қандай символ қабул қилиниши ёки қилинмаслигини аниқлайди.



-List Box компонентаси.

Ушбу компонента фойдаланувчига танлаш учун элементлар рўйхатини кўрсатади. Айрим пайти фойдаланувчи ўз дастурида жуда кўп элементларнинг рўйхатини кўрсатиш лозим бўлиб қолади. Бундай ҳолатлар учун рўйхатлар дарчасида стандарт кўринишдаги сургич пайдо бўлади ва ёрдам беради. Рўйхатлар панелига элементларни қўшиш ёки олиб ташлаш мумкин. Ушбу компонентанинг ички кўринишини ўзгартириш учун мўлжалланган турли хил хоссалари мавжуд.

Рўйхатлар дарчасини формада жойлаш учун кўп жой зарур бўлади. Комбинациялашган панел компонентаси **Combo Box** ни формада жойни тежаш учун фойдаланиш қулайроқдир.

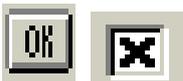


-Combo Box компонентаси.

Combo Box компонентаси **Combination Boxes** сўзидан қисқартирилган бўлиб **List Box** компонентасига ўхшашдир. Лекин комбинациялашган дарча кам жойни талаб қилади ва бир вақтнинг ўзида битта элементни кўрсатади.

Демак **List Box** компонентаси билан **Combo Box** компонентасининг фарқи бир вақтнинг ўзида қанча элемент кўрсата олишда экан. Жойдан тежаш мақсадида **Combo Box** дан фойдаланиш қулайроқ бўлса, кўпроқ элементни кўрсатиш учун **List Box** дан фойдаланиш қулайдир.

Button ва Check Box компоненталари.



Button (тугма) ва **Check Box** компоненталарининг кўпгина Windows муҳитида учратиш мумкин.

Button (тугма) компонентаси.

Button тугмачаси босилиши натижасида киритилиши лозим бўлган жараёнлар бажарилишга туширилади. Button тугмачасини экранга жойлаш учун Delphi нинг Standart палитрасидан «ok» пиктограммаси белгиланиб форма устига келинади ва сичқонча тугмачасини босган ҳолда тугмача қўйилиши лозим бўлган жой ажратилади.

Натижада Button 1 тугмачаси ҳосил қилинади. Тугмача номини ўзгартириш Caption хоссасига кирилиб ўзгартирилади. Дастурдаги ҳисоблаш жараёнлари, киритиш ва чиқариш

операциялари ҳосил қилинган тугмачани икки марта тез тез босиш билан “события” ни қайта ишлаш дарчасига ўтилиб, у ердан модул ичига керакли операторларни ёзиш билан амалга оширилади.

Бу компонента одатда диалог дарчаларида фойдаланилади. Кўпгина Windows дастурларида таниш бўлган **Ok** ва **Cancel** жуфтлигини кўргансиз. Аммо сизга тугма компонента-сидан фойдаланиш учун диалог дарчасини яратиш шарт эмас. Тугма компонентасини сиз дастурингизнинг исталган жойида жойлаштиришингиз мумкин.



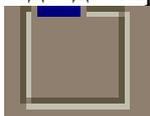
-BitBtn компонентаси.

График тугма компонента оддий тугма компонентасига жуда ҳам ўхшашдир. **Delphi** тугмаларнинг стандарт мажмуасини таклиф қилади. Буларни дастурда яратиш учун танлаб олиш мумкин.

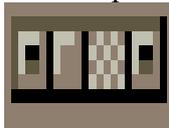
Chex Box (белгилаш тугмаси) компонентаси ушбу тугма одатда дастурда «**Ха**» ёки «**Йўқ**» жавобини олиш керак бўлган вазиятларда фойдаланилади.

Гурухлашган компоненталар

Delphi дастурлаш тилида компоненталарни гурухлаштириб фойдаланиш мумкин. Одатда бир-бирига алоқадор компоненталар битта жойида гурухлаштирилади.



-Croup Boxes гурухлашган рамка компонентаси. Гурухлашган рамка фақатгина бир тоифадаги компоненталарни ташкил топган бўлиши шарт эмас.



Scroll Bors компонентаси. Ушбу компонента турли ойналарни экранда кўринмай турган қисмларини кўриш ва бошқа мақсадларда ишлатилади.

Delphi: дастурлаш тилида **Scroll Bors** жуда яхши ташкил қилинган ва тули хил хосса-лари мавжуд.

Position жойлашиш хоссаси оний моментда қаерда жойлашишини белгилаб беради. Бу хоссанинг қиймати фақат бутун сонларни қабул қилади.

График компоненталар

График компоненталар формада фигура ва объектларни кўрсатиш ҳамда маълумотларни акслантириш учун фойдаланилади. Бу компоненталарни баъзилари жуда ҳам оддий баъзилари жуда ҳам мураккаб. Уларни ҳар бири аниқ вазифа учун мўлжалланган.



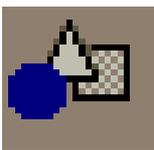
Image компонентаси

Бу компонентани формада расмларни жойлаштириш учун фойдаланилади. Расмни номи эса **Picture** хоссасида келтирилади. Бу расмларнинг кенгайтмаси **.BMP, ICO, WMF** бўлиши мумкин. Агар фойдаланувчи **Image** компонентасининг ўлчамини ўзгартиришни хоҳласа **Auto Size** хоссасига **True** қиймати бериши лозим. **Stretch** хоссасига **True** берилган бўлса, белгиланган соҳага жойлаштирилади.



Paint Box компонентаси. (Чизиш дарчаси)

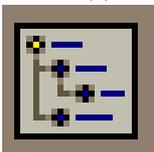
Ушбу компонента форманинг чекланган қисмида чизиш имконини беради. Чизиш дарчаси ўзининг чегараларини кўрсата олмайди. Ушбу қийинчиликни йўқотиш учун унинг ичига гурухлашган панел **Croup Boxes** қўйиш ва уни ўлчамини **Paint Box** компонента ўлчами билан бир хил қилиш лозим. Ана шундагина фойдаланувчига чизиш учун рамка пайдо бўлади.



Shapes (фигуралар) компонентаси

Фигуралар компонентаси ўз табиати билан жуда ҳам оддий. Фойдаланувчи содда геометрик фигуралардан дастурни бойитишда фойдаланишлари мумкин.

Фигураларнинг ва контурларнинг ранги мос равишда **Pen** ва **Brush** хоссалари ёрдамида ўрнатилади.

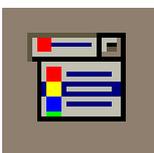


Outlines (схемалар) компонентаси

Схемалар иехархик тарзда ташкил қилинган маълумотларни акслантириш учун фойдаланилади. Мисол учун каталоглар дарахти.

Схема **Lines** ва **Items** хоссалари орқали мурожат қилиш мумкин бўлган қаторлардан ташкил топади.

Outlines Tile (схемали) хоссаси исталган конфигурациясини олиш учун фойдаланилади.



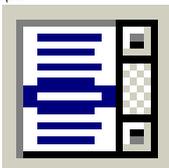
Color Crids компонентаси (ранглар тури)

Бу компонента чизиш ва фон рангларини бошқариш учун интерфейсни ифодалайди. Бу компонента ёрдамида чизиш ва фон рангини танлашда фойдаланилади.

Foreground (чизиш рангли) ва **Background** (фон ранги) хоссаларда белгиланган рангдан ташкил топган бўлади.

Файл ва каталогга мурожат қилиш компоненталари

Ушбу компоненталар файл ва каталоглар билан турли амалларни бажаришда фойдаланилади.



File List Box (файллар рўйхати панели) компонент файллар рўйхати панели

компонентаси каталогга мурожат интерфейсини ифодалайди.

Mask хоссаси филтер ёки маскани ўрнатиш учун фойдаланилади.

Фойдаланувчи томонидан белгиланган файл номи **File Name** хоссаси орқали мурожат қилиши мумкин.

Delphi нинг консол иловасини яратиш

Delphi консолли иловаларни турли усулларда яратиш мумкин. Улардан энг осон усули қуйидагича.

1. Delphi мухити ишга туширилади.

Пуск ⇒ Программы ⇒ Borland Delphi

2. Бош менюдан File пункти очилиб, у ердан New, кейин эса Other буйруқлари берилади.

File ⇒ New ⇒ Other

3. Форма ва лойихаларни сақлаш учун очилган махсус ойнадан “Console Application” пиктограммаси танланади ва ok тугмаси босилади.

4. Натижада экранда begin ва end ичига олинган лойиха ойнаси (.dpr кенгайтмали ном билан) очилади.

{TODO –oUser-cConsole Main:Insert code here} изохи ўрнига файлнинг дастур матни киритилади. Тузилган дастурни ишга туширишдан олдин уни сақлаш керак бўлади. Уни сақлаш учун file ⇒ Save All буйругини бериш лозим. Хар бир лойиха алохида янги папкада сақланиши тавсия этилади. Лойиха файлини сақлашда алохида кўрсатилмаган ҳолатида ProjectN.dpr номли файл номини тавсия этади. Лойихани сақлаб бўлгандан кейин уни бажаришга ўтилади.

Дастур матни 1 да киритилган килограммни неча фунт эканлигини кўрсатувчи дастур кўрсатилган. Унда бирор буюмнинг оғирлиги фойдаланувчи томонидан килограммда киритилади. Натижа эса компьютер томонидан фунтда экранга чоп этилади.

```
{ $APPTYPE CONSOLE }
Program Project2;
Var
k,f:Real;
Begin
Writeln('Buyum ogirlugini kilogrammda kiriting');
Writeln('va <Enter> tugmasini bosing');
Write('→');
Readln(k);
f:=k*0.4095;
Writeln(k:10:4,' kilogramm=', f:10:4,' funt');
Readln;
End.
```

Юқоридаги дастурда *{ \$APPTYPE CONSOLE }* қатори мавжуд бўлиб, у изоҳ кўринишида ёзилган. Лекин у, дастурнинг консолли илова эканлигини билдиради. Бундай дастурни тузишда албатта *{ \$APPTYPE CONSOLE }* қатори ёзилиши шарт. Дастурни ишга тушириш учун Run менюсидан Run буйруғи танланади ёки F9 клавиши босилади.

Танлаш тугмаларини ўрнатиш

RadioGroup гурухи танлаш тугмалари иловалар яратишда бир неча вариантлардан бирини танлаш учун ишлатилади. Бу компонента Standart компонентлар палитрасида жойлашган. Унинг асосий хоссаси items бўлиб, у тугмалар номлари руйхатини ўзида сақлайди. Тугмалар номлари руйхатини киритишдан олдин RadioGroup тугмаси учун формадан жой ажратилади ва кейин items хоссаси кўрсатилиб, ундан уч нуҳтали тугмача босилади, натижада stringlist Editor ойнаси очилади. Бу ойнадан танлаш тугмалари номларининг хар қайсиси янги қатордан киритилади ва кейин “ok” тугмаси босилади. Формага RadioGroup гурухли танлаш тугмаси жойлаштирилганда и RadioGroup 1 ном билан ёзилади. Бу номни бошқа мос номга алмаштириш Caption хоссасига кириб амалга оширилади.

CheckBox компонентаси руйхатдан бир нечтасини танлаш имконини беради. CheckBox компонентаси Additional палитрасида жойлашган. RadioGroup боғлиқ переключателларни, CheckBox эса боғлиқ бўлмаган переключателларни бирлаштиради. Бунда ёқувчи уч хил ҳолатда бўлиши мумкин:

- Ёқилган –тўғри белгиси;
- Ўчирилган–бўш белгиси;
- Нейтрал ҳолат –кўкиш рангда тўғри белгиси;

CheckBox нинг асосий хоссалари:

AllowOryer учинчи нейтрал ҳолат вариантини ишлатишни тақиқлайди.

Items танлаш тугмалари номлари рўйхатини сақлайди.

Listbox ва Combobox компонентлари

Listbox компоненти рўйхат ва массив кўринишидаги маълумотларни экранда акс эттиришда ишлатилади. Маълумотларни киритишда эса Edit компонентасидан фойдаланилади. Listbox компонентаси Standart компоненталар палитрасида жойлашган. Бу тугмачани босиб формадан рўйхат учун жой ажратилади ва хоссалари аниқланади. Унинг айрим хоссалари:

Items – рўйхат элементини беради.

Sorter- рўйхат элементларини алфавит бўйича автоматик равишда тартиблайди.

Clear –барча рўйхат элементларини ўчиради.

Combobox компоненти рўйхат ва массив кўринишидаги маълумотларни экранга киритиш учун ишлатилади. У Listbox ва Edit компоненталарининг биргаликдаги ишини бир ўзи бажаради. Ташқи кўринишидан бу компонент оддий Edit киритиш қаторини эслатади. Унинг ўнг қисмида пастга белгиси бўлиб, киритилаётган маълумотларни кўриб бориш мумкин. Бу компонента Standart компоненталар палитрасида жойлашган. Бу тугмачани босиб формадан рўйхат учун жой ажратилади ва кейин хоссалари аниқланади. Унинг айрим асосий хоссалари:

DropDownCount – рўйхатдаги экранга чиқадиган маълумотлар сонини аниқлайди. Бу хоссанинг бошланғич қиймати 8 га тенг бўлади. Агар экранга чиқариладиган маълумотлар сонини 10 га бўлсин десак, унда унинг қийматини 10 га ўзгартириш керак бўлади. Агар киритилган маълумотлар ундан ортиқ бўлса, у холда пастга ва юқорига силжитиш тугмачаси автоматик равишда пайдо бўлади;

Style – рўйхатдаги маълумотнинг кўринишини тасвирлайди;

Text- рўйхатдаги киритилган маълумот матн эканини билдиради.

StringGrid жадвал компонентаси

StringGrid жадвал компонентаси икки ўлчовли маълумотларни, масалан, матрица элементларининг қийматини экранда жадвал кринишида тасвирлаш, улар қийматини киритиш ва таҳрирлаш учун ишлатилади. Жадвалнинг қатор ва устун номерлари нолдан бошланади. Жадвалнинг устун ва қаторлар сонини кераклича ўзгартириш мумкин. Бу унинг хоссаси ёрдамида аниқланади. Жадвалнинг шар бир кесишган устун ва сатри ячейка дейилиб, унга киритилган маълумот символ қатори қилиб аниқланади. Масалан, (3,5) ячейка тўртинчи устун ва олтинчи қаторда жойлашган.

StringGrid жадвал компонентасининг асосий хоссалари:

ColCount - жадвалдаги устунлар сонини аниқлайди;

RowCount - жадвалдаги сатрлар сонини аниқлайди;

FixedCols - фиксирланган устунлар сонини аниқлайди;

FixedRows - фиксирланган сатрлар сонини аниқлайди;

Options -жадвал ҳолатини аниқлайди (аниқлаш унинг параметрларига асосан бажарилади, масалан Goediting параметр true қийматига эга бўлса ячейкани таҳрирлаш мумкин, акс ҳолда мумкин эмас. Бу параметрларни аниқлаш учун Options хоссасига ўтиб, у икки марта тез-тез босилади);

ColWidths - жадвалдаги ҳар бир устун кенглигини аниқлайди;

DefaultColWidth - жадвалнинг бошланғич устунлар кенглигини аниқлайди;

DefaultRowHeight - жадвал сатрининг бошланғич баландлигини аниқлайди;

FixedColor - фиксирланган ячейка рангини аниқлайди;

RowHeiohts - жадвал сатри баландлигини аниқлайди;

Cells – символ қаторли икки ўлчамли массивни аниқлайди.

Символлар ва қаторлар

Белгили тип *Char* хизматчи сўзи билан эълон қилиниб, бу типнинг қийматлари 1 байт жой эгаллайди. Тилнинг барча белгилари бу типнинг қийматлар соҳасига тегишлидир. Белгили қийматни уни қўштирноқ ичига олиб ёки # белгисидан кейин керакли белгининг ASCII коддини ёзиб аниқлаш мумкин. Мисол: "А" , ёки # 60.

Қатор - бу белгиларнинг оддий кетма-кетлигидир: 'Ab21#9!cd', 'dasturchi Saidkarim Gulomov'. Қатор бўш ёки битта белгили бўлиши ҳам мумкин. Қаторли ўзгарувчи узунлиги 255 гача бўлган белгили қийматларни қабул қилиши мумкин. Умуман олганда, ҳар бир қаторли ўзгарувчига хотирадан 256 байт жой ажратилади. Хотирани тежаш учун қаторнинг типини қуйидагича кўрсатиш мақсадга мувофиқдир: *String*[N], N - қатордаги белгилар сони. Бу ҳолда белгили ўзгарувчи учун N байт жой ажратилади. Белгилар ва қаторлар устида бир қанча амаллар бажариш мумкин, яъни қатордан керакли бўлакни кесиш олиш, қаторларни бир-бирига қўшиш ва натижада янги қаторлар ҳосил қилиш. Сатрли ва символли белгилар устида турли амаллар бажариш мумкин.

Ёзилиши	Вазифаси
Function Length(S):Integer	S сатрли ўзгарувчидаги белгилар сонини аниқлайди
Function Copy(S; Index, Count: Integer): string;	S сатрли ўзгарувчидаги Index - белги Count та белгидан нусха олиш
Function Concat(s1 [, s2,..., sn]: string): string;	S1 дан sn тагача бўлган сатрли ўзгарувчиларни битта сатрли ўзгарувчига бирлаштириш
function Pos(Substr: string; S: string): Integer;	Substr сатри S сатридан изланади. Агарда изланган сатр топилмаса натижа нолга тенг бўлади
procedure Delete(var S: string; Index, Count:Integer);	S сатрдаги Index – белгидан Count та белгини ўчириб ташлайди
procedure Insert(Source: string; var S: string; Index: Integer);	S сатрига Index – белгидан бошлаб Source сатрини жойлаштиради

Массивлар. Массивлар устида амаллар.

Программалашда энг кўп қўлланиладиган программа объектларининг бири бўлган массивлар билан танишиб чиқамиз. Массив - бу бир хил типли, чекли қийматларнинг тартибланган тўпламидир. Массивларга мисол сифатида математика курсидан маълум бўлган векторлар, матрицалар ва тензорларни кўрсатиш мумкин. Програмада ишлатилувчи барча массивларга ўзига хос исм бериш керак. Массивнинг ҳар бир ҳадига мурожаат эса унинг номи ва ўрта қавс ичига олиб ёзилган тартиб ҳади орқали амалга оширилади. Массивнинг зарур ҳадига мурожаат қуйидагича амалга оширилади: **<массив номи>[<индекс>]** бу ерда <индекс> массив ҳадининг жойлашган жойини англатувчи тартиб қиймати.

Умуман олганда, <индекс> ўрнида ифода қатнашиши ҳам мумкин. Индексни ифодаловчи ифоданинг типини индекс типи деб аталади. Индекс типининг қийматлар тўплами албатта номерланган тўплам бўлиши, шу билан бир қаторда массив ҳадлари сонини аниқлаш ва уларнинг тартибини белгилаши керак. Массивларни эълон қилишда индекс типи билан бир қаторда массив ҳадларининг типини ҳам кўрсатилиши керак. Бир ўлчамли массивни эълон қилиш қуйидагича амалга оширилади: `array [<индекс типини>] of <ҳад типини>;` Кўпинча <индекс типини> сифатида чекланма типлардан фойдаланилади, чунки бу типга тегишли тўплам тартибланган ва қатъий номерлангандир. Мисол учун 100 та ҳақиқий сонли ҳадлардан иборат массив қуйидагича эълон қилинади: `array [1..100] of real;`

Массивларни эълон қилиш ҳақида тўлиқроқ маълумот бериш учун турли типдаги индексларга оид мисолларни эътиборингизга ҳавола қиламиз:

```
array [1000..5000] of integer;
array [-754..-1] of byte;
array [0..100] of real;
array [0..10] of boolean;
```

```
array [10..25] of char;
```

Кўп ўлчамли массивлар

Бир ўлчамли массивларнинг ҳадлари скаляр миқдорлар бўлган эди. Умумий ҳолда эса массив ҳадлари ўз навбатида яна массивлар бўлиши мумкин, агар бу массивлар скаляр миқдорлар бўлса натижада икки ўлчамли массивлар ҳосил бўлади. Икки ўлчамли массивларга мисол сифатида математика курсидаги матрицаларни келтириш мумкин. Агар бир ўлчамли массивнинг ҳадлари ўз навбатида матрицалар бўлса натижада уч ўлчовли массивлар ҳосил қилинади ва ҳ.к. **Икки ўлчамли массив типини кўрсатиш қуйидагича бажарилади:**

```
array [<индекс типи>] of array [<индекс типи>] of <скаляр тип>;
```

Икки ўлчамли массив индексларининг типлари турли хил ҳам бўлиши мумкин. Бу ҳолни қуйидаги мисол устида кўриб чиқайлик:

```
const n = 24;  
type hafkun = (dush, sesh, chor, pay, jum, shan, yaksh);  
Ishkun = dush..jum;  
detson = array [1..n] of char;  
var A: array [boolean] of array [1..n] of char;  
B: detson;  
C: array [1..365] of detson;
```

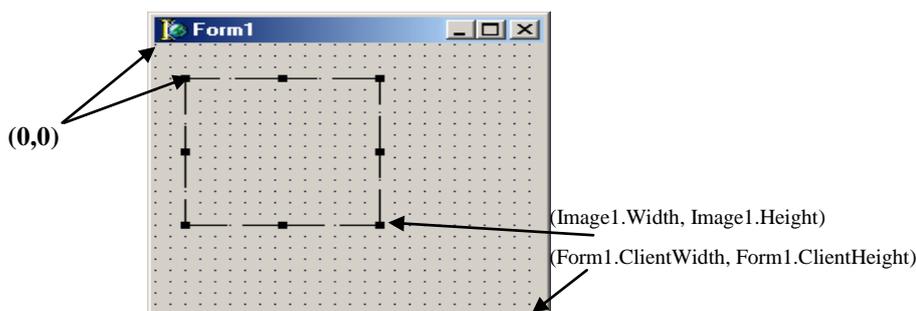
Delphi нинг график имкониятлари

Delphi дастурчига турли хилдаги схемалар, чизмалар ва иллюстрациялар билан ишлаш имкониятларини беради. Дастур графикани объект (форма ёки компонент Image) сиртида ҳосил қилади. Объект сирти Canvas хусусиятига мос келади. График элемент (тўғри чизик, айлана, тўғри тўртбурчак ва ҳ.к.) ларни объект юзасида ҳосил қилиш учун Canvas дан фойдаланилади. Масалан, `Form1.Canvas.Rectangle(10,10,50,50)` инструкцияси дастур ойнасида тўғри тўртбурчак ҳосил қилади.

Чизма ҳосил бўлувчи сирт.

Юқорида айтиб ўтилганидек, графикани ҳосил қилувчи сирт (юза) Canvas хусусиятига тўғри келади. Ўз навбатида Canvas хусусияти TCanvas типидagi объектдир. Бу тип услублари график примитивларни (нуқта, чизик, айлана ва ҳ.к.) ҳосил бўлишини таъминлайди, хусусияти эса ҳосил бўлувчи графикани характеристикаларини: ранги, чизик қалинлиги ва тури; бўялувчи худудни ранги ва кўринишини; ҳарфни характеристикаларини беради. Canvas «сирт», «чизиш учун юза» сифатида таржима қилинади. Чизиш юзаси алоҳида нуқта – пикселлардан ташкил топади. Пикселни жойлашуви горизонтал (X) ва вертикал (Y) координаталар билан характерланади.

Чап юқоридаги нуқта координатаси (0,0). Координаталар юқоридан пастга ва чапдан ўнгга қараб ўсиб боради.



1-расм. Чизиш юзаси нуқта координаталари.

Чизиш юзаси ўлчамларини иллюстрация (Image) худуди учун Height ва Width, форма учун эса ClientHeight ва ClientWidth лар аниқлаш мумкин.

Қалам ва мўйқалам.

Одатда расом сурат чизиш учун қалам ва мўйқаламдан фойдаланади. Delphi нинг график имкониятлари ҳам қалам ва мўйқаламдан фойдаланиш имкониятларини яратади. Қаламдан чизиқ ва контур чизишда, мўйқаламдан эса контур билан чегараланган юзани бўяш учун фойдаланилади.

Қалам ва мўйқалам графикани чизиш юзасида ҳосил қилишда мос равишда Pen(қалам) ва Brush(мўйқалам) хусусиятларига хосдир. Шу билан бирга улар TPen ва TBrush типларига тегишлидир.

Қалам.

Қаламдан нукта, чизиқ, геометрик шакллар: тўғри тўртбурчак, айлана, эллипс ва х.к. ларни чизишда курул сифатида фойдаланилади. TPen объект хусусияти 1-жадвалда келтирилган.

1-жадвал

Хусусият	Вазифаси
Color	Чизиқ (контур) ранги
Width	Чизиқ қалинлиги
Style	Чизиқ кўриниши
Mode	Тасвирлаш режими

Color хусусияти чизувчи қалам рангини белгилайди. Қуйидаги 2-жадвалда PenColor хусусиятлари келтирилган:

2-жадвал

Константа	Ранг	Константа	Ранг
<i>clBlack</i>	қора	<i>clSilver</i>	кумушранг
<i>clMaroon</i>	каштанранг	<i>clRed</i>	қизил
<i>clGreen</i>	яшил	<i>clLime</i>	салатранг
<i>clOlive</i>	оливковий	<i>clBlue</i>	кўк
<i>clNavy</i>	тўқ кўк	<i>clFuchsia</i>	Fuchsia
<i>clPurple</i>	атиргулранг	<i>clAqua</i>	ёруј кўк
<i>clTeal</i>	Teal	<i>clWhite</i>	оқ
<i>clGray</i>	кулранг		

Width хусусияти чизувчи қалам қалинлигини (пикселда) белгилайди.

Масалан, *Canvas.Pen.Width:=2* чизиқ қалинлиги 2 пикселга тенг бўлади.

Style хусусияти чизилувчи чизиқнинг турини белгилайди. Style компонентлари 3-жадвалда келтирилган.

Константа	Чизиқ кўриниши
<i>psSolid</i>	Тўғри чизиқ
<i>psDash</i>	Узун штрихли пунктир чизиқ
<i>psDot</i>	Қисқа штрихли пунктир чизиқ
<i>psDashDot</i>	Узун-қисқа штрихли пунктир чизиқ
<i>PsDashDotDot</i>	Бир узун ва икки қисқа штрихли пунктир чизиқ
<i>PsClear</i>	Кўринмас чизиқ

Мўйқалам

Мўйқалам (Canvas.Brush)дан ёпиқ соҳаларни тўлдириш учун фойдаланилади, масалан, геометрик шаклларни бўйаш ва х.к. Мўйқалам объект сифатида қуйидаги икки хусусиятни ўз ичига олади:

Color – бўялувчи соҳа ранги

Style – тўлдирувчи соҳа типи

Масалан, контурнинг ички соҳаси бўялиши ёки штрихланиши мумкин.

Color хусусияти сифатида Tcolor нинг барча ўзгармасларидан фойдаланиш мумкин. Style хусусиятлари 4-жадвалда келтирилган.

4-жадвал

Константа	Бўялувчи соҳа типи
bsSolid	тўлиқ
bsClear	Бўялмайди
bsHorizontal	горизонтал штрихлаш
bsVertical	вертикал штрихлаш
bsFDiagonal	олдинга эгилган диагонал штрихлаш
bsBDiagonal	орқага эгилган диагонал штрихлаш
bsCross	горизонтал-вертикал сеткали штрихлаш
bsDiagCross	диагонал сеткали штрихлаш

Матн ҳосил қилиш

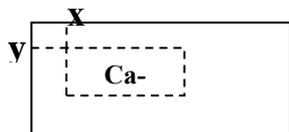
График объект сиртида матнни ҳосил қилиш учун TextOut дан фойдаланилади. TextOut нинг ёзилиш формати қуйидагича:

Объект.Canvas.TextOut(x,y,Text);

Бу ерда Объект – матн ҳосил бўлувчи объект номи;

x,y – матн бошланувчи координата

Text – ҳосил бўлувчи белги катталиқдаги матн ёки сатрли ўзгарувчи.



3-расм. Матн ҳосил бўлувчи соҳа координатаси

ҳосил бўлувчи матн белгилари Canvas объектига мувофиқ келувчи Font хусусияти орқали ифодаланади. Font хусусияти TFont объектига тегишли бўлиб, 5-жадвалда белги характеристикалари ва қўлланилувчи услублари келтирилган.

5-жадвал

Хусусият	Аниқланиши
Name	Фойдаланилувчи шрифт. Қиймат сифатида шрифт номи ёзилади, масалан, Arial Cyr
Size	пунктларда ифодаланилувчи шрифт ўлчами. Пункт-полиграфияда қўлланилувчи ўлчов бирлиги бўлиб, у тахминан 1/72 дюйм ¹ га тенг
Style	белгини ёзиш усули, қуйидагича бўлиши мумкин: оддий, калин, курсив, остига чизилган, устига чизилган. Булар қуйидаги константалар ёрдамида амалга оширилади: <i>fsBold</i> (калин), <i>fsItalic</i> (курсив), <i>fsUnderline</i> (остига чизилган), <i>fsStrikeOut</i> (устига чизилган). style бир нечта усулларни комбинация қилиши мумкин. Масалан, калин курсив ҳолатини ифодалаш:

	<i>Объект.Canvas.Font:=[fsBold, fsItalic]</i>
<i>Color</i>	Белги ранги. Қиймат сифатида <i>TColor</i> константаларидан фойдаланиш мумкин.

Қуйидаги дастур қисми TextOut функциясини қўллаш учун мисол бўла олади:
with Form1.Canvas do

```
begin
    Brush.Color:=Form1.Color;
    Font.Size:=14;
    Font.Style:=[fsItalic, fsBold];
    TextOut(10,10,'Salom, Delphi!');
End;
```

Матн эканда ҳосил бўлгандан сўнг кўрсаткич унинг ўнг юқори бурчагига силжийди. Баъзида матндан сўнг бирор маълумотни чиқариш керак бўлиб қолади. Агар матн узунлиги номаълум бўлса кўрсаткич турган координатани аниқлаш мушкул. Масалан «сўм» сўзини рақамдан кейин ҳосил қилиш керак бўлсин. Бундай ҳолатларда кўрсаткич турган координатдан бошлаб давом этиш учун PenPos дан фойдаланишга тўғри келади:

```
TextOut(10,10,SumPr); // SumPr – String типли катталик
TextOut(PenPos.X, PenPos.Y, ' сум');
```

Тўғри чизиқ.

Delphi да тўғри чизиқ ҳосил қилиш учун LineTo дан фойдаланилади. Унинг ёзилиш формати қуйидагича:

Компонент.Canvas/LineTo(x,y)

LineTo тўғри чизиқни қалам (кўрсаткич) турган координатадан бошлаб x,y – нуқтагача чизади. Шунинг учун чизиқнинг бошланғич нуқтасини керакли жойга ўрнатиб олиш лозим бўлади. Бунда биз MoveTo га мурожаат қиламиз:

Компонент.Canvas.MoveTo(X0,Y0)

Чизиқнинг кўриниши (ранги, қалинлиги ва тури) Pen объекти билан ифодаланади.

Айлана ва эллипс.

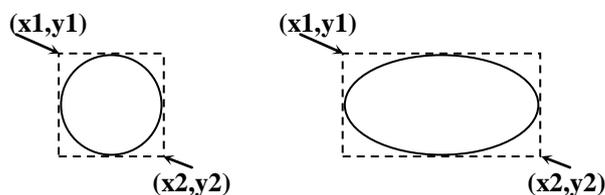
Ellipse услуби эллипс ва айлана чизиш учун қўланилади. Ellipse нинг ёзилиш формати қуйидагича:

Объект.Canvas.Ellipse(x1,y1,x2,y2)

бу ерда, объект – чизма ҳосил бўлувчи объект номи;

x1,y1,x2,y2 – ҳосил бўлувчи айлана ёки эллипсга ташқи чизилган тўғри тўртбурчакнинг мос равишда юқори чап ва қуйи ўнг нуқталарини координаталари (4-расм).

Чизиқнинг кўриниши (ранги, қалинлиги ва тури) Pen объекти билан ифодаланади.



4-расм.

Ёй.

Ёй ҳосил қилиш учун Arc услубидан фойдаланилади. Унинг ёзилиш формати қуйидагича:

Объект.Canvas.Arc(x1,y1,x2,y2,x3,y3,x4,y4)

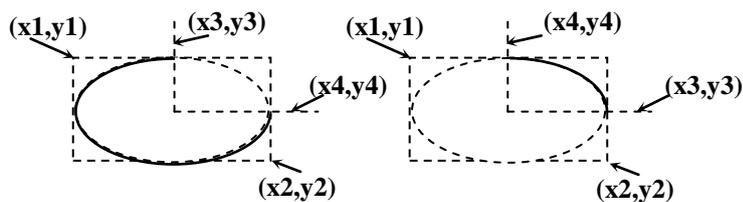
бу ерда, объект – ёй чизилувчи объект номи;

x1,y1,x2,y2 – ҳосил бўлувчи ёйни давом эттириб ҳосил қилинувчи эллипс (айлана)га ташқи чизилган тўғри тўртбурчакнинг мос координаталари;

x3,y3 – ёйнинг бошланғич нуқтаси;

x_4, y_4 – ёйнинг тугаш нуқтаси.

Шуни айтиб ўтиш лозимки, ёй соат стрелкаси йўналишига қарама-қарши йўналишда чизилади (5-расм).



5-расм.

Чизиқнинг кўриниши (ранги, қалинлиги ва тури) Pen объекти билан ифодаланади.

Тўғри тўртбурчак.

Тўғри тўртбурчак ҳосил қилишда Rectangle услубидан фойдаланилади. Унинг ёзилиш формати қуйидагича:

Объект. *Canvas.Rectangle(x1,y1,x2,y2)*

Бу ерда

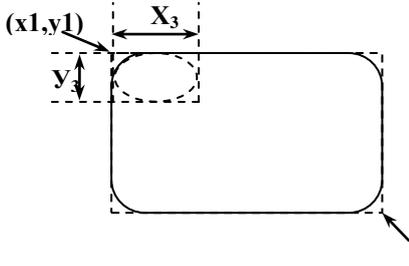
объект – тасвир ҳосил бўлувчи объект номи;

x_1, y_1, x_2, y_2 – тўғри тўртбурчакнинг мос равишда юқори чап ва қуйи ўнг бурчак координаталари.

RoundRec услуби ҳам тўғри тўртбурчак чизади, фақат Rectangle дан фарқи шундаки, унинг бурчаклари юмалоқ (силлиқ) шаклда бўлади. Ёзилиш формати:

Объект. *Canvas.RoundRec(x1,y1,x2,y2)*

Бу ерда



объект – тасвир ҳосил бўлувчи объект номи;

x_1, y_1, x_2, y_2 – тўғри тўртбурчакнинг мос равишда юқори чап ва қуйи ўнг бурчак координаталари;

x_3, y_3 – юмалоқ ҳосил қилишда қўлланилувчи эллипс ўлчамлари (6-расм).

Кўпбурчак.

(x_1, y_1) Polygon дан фойдаланиб кўпбурчак чизиш мумкин.

Polygon TPoint типли массивни параметр сифатида қабул қилади. ҳар бир массив элементи ўзида кўпбурчакнинг битта бурчаги координатаси (x, y) ни сақлайди. Polygon эса шу нуқталарни кетма-кет тўғри чизиқлар билан тугаштириб чиқади. Чизиқнинг кўриниши (ранги, қалинлиги ва тури) Pen объекти билан ифодаланади.

Қуйида учбурчак чизиш учун дастур қисми келтирилган:

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

```
var
```

```
    pol: array[1..3] of TPoint; //учбурчак нуқталари координатаси
```

```
    {TPoint=record
```

```
        X:longint;
```

```
        Y:longint
```

```
    End;}
```

```
Begin
```

```
    Pol[1].x:=10;
```

```
    Pol[1].y:=50;
```

```
    Pol[1].x:=40;
```

```
    Pol[1].y:=10;
```

```
    Pol[1].x:=70;
```

```
    Pol[1].y:=50;
```

```
    Form1.Canvas.Polygon(pol);
```

End;

Сектор.

Эллипс ёки айлана секторини ҳосил қилишда Pie услубидан фойдаланилади. Pie нинг умумий ёзилиш формати:

Объект. Canvas.Pie(x1,y1,x2,y2,x3,y3,x4,y4)

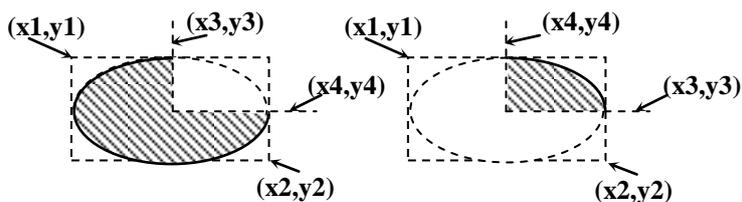
бу ерда

объект – ёй чизилувчи объект номи;

x1,y1,x2,y2 – ҳосил бўлувчи секторни давом эттириб ҳосил қилинувчи эллипс (айлана)га ташқи чизилган тўғри тўртбурчакнинг мос координаталари;

x3,y3 – секторнинг бошланғич нуқтаси;

x4,y4 – секторнинг тугаш нуқтаси.



7-расм.

Мультипликация.

Мультипликация дейилганда одатда ҳаракатланувчи ёки ўзгарувчи расмни тушунилади. Оддий ҳолатларда расм ҳаракатланиши ёки ўзгариши мумкин. 2-осил қилинган расм (чизик, айлана, ёй ва х.к.)ларни силжитиш жуда оддий: аввал расм ҳосил қилинади, бир оздан сўнг уни тозаланади ва яна янгитдан аввалги жойидан бошқа ерда ҳосил қилинади. Бундай алмаштириш бир маромда давом эттирилса, натижада тасвир экран бўйлаб ҳаракатланаётган ўхшайди.

Қуйидаги кичик дастур ёрдамида айланани дастур ойнасининг чап чегарасидан ўнг чегарасига қараб ҳаракатлантиришимиз мумкин. 8-расмда форма кўриниши келтирилган.



8-расм.

Дастур матни

```
unit Unit1;
```

```
interface
```

```
uses
```

```
Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms, Dialogs, 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,dx:byte;
implementation
{$R *.dfm}
procedure Ris;
begin
  {айланани куринамас килиш}
  form1.Canvas.Pen.Color:=form1.Color;
  form1.Canvas.Ellipse(x,y,x+10,y+10);
  x:=x+dx;
  {айланани янги жойда ҳосил килиш}
  form1.Canvas.Pen.Color:=clBlack;
  form1.Canvas.Ellipse(x,y,x+10,y+10);
end;
procedure TForm1.Timer1Timer(Sender: TObject);
begin
  ris;
end;
procedure TForm1.FormActivate(Sender: TObject);
begin
  x:=0;
  y:=10;
  dx:=5;
  timer1.Interval:=50;
  form1.Canvas.Brush.Color:=form1.Color;
end;
end.

```

Асосий ишни айланани ўчириб янги жойда ҳосил қилувчи Ris процедураси бажаради. Айланани ўчиришни унинг рангини



9-расм.

Timer

форма рангига ўзгартириш йўли билан амалга оширилади. Форма ёки дастурда эътибор берган бўлсангиз визуал бўлмаган компонент Timer (таймер)дан фойдаландик. У ёрдамида ҳаракатни вақт бўйича амалга оширилиши таъминланган. Timer компоненти компонентлар палитрасининг System бўлимида жойлашган (9-расм). Timer хусусиятлари 10-жадвалда келтирилган.

10-жадвал

Константа	Бўялувчи соҳа типи
<i>Name</i>	Компонент номи
<i>Interval</i>	Миллисекундларда берилувчи OnTimer генерацияси
<i>Enabled</i>	Ишга рухсат бериш. Қиймат true бўлса рухсат берилади, false бўлса берилмайди.

Timer компоненти OnTimer ходисасини ишга туширади. OnTimer вақтли ходисаси миллисекундларда ўзгаради ва Interval хусусиятларига мос келади. Enabled хусусияти эса дастурда таймерни «ишга тушириш» ёки «тўхтатиш» имкониятини яратади. Агар Enabled True бўлса OnTimer ишламайди.

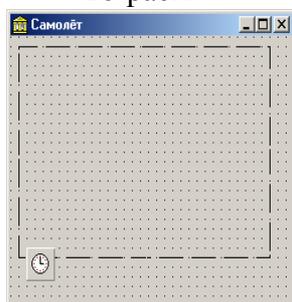
Битли тасвирлардан фойдаланиш.

Юқоридаги мисолда тасвирни ўзимиз ҳосил қилиб олдик. Энди эса қандай қилиб бир мураккаб тасвирни бошқасини фонида ҳаракатланишини кўриб ўтамиз. Масалан, шаҳар тасвири фонида самолётни юргизишни олайлик. Суратни силжитиш эффекти суратни бир неча жойда вақти-вақти билан қайтадан чизиш усули билан ташкил қилиниши мумкин. Бундан олдин, тасвирни янги нуқтада чиқаришдан аввал уни аввалгиси ўчирилади. Суратни ўчириш тўлиқ фонни бошқатдан ёки фақат ўша қисмини чизиш йўли билан амалга оширилиши мумкин. Биз кўриб ўтадиган дастурда иккинчи йўлдан фойдаланамиз. Тасвир Image компонентининг Canvas хусусиятида Draw услуби билан чиқарилади, тозалаш эса фоннинг керакли қисмини нусхасини олиш йўли (CopyRect услуби) билан амалга оширилади.

Дастур форма кўриниши 10-расмда келтирилган.

Image компоненти фонни чиқариш учун, Timer компоненти эса ҳаракатни ҳосил қилиш учун фойдаланилади.

10-расм



Дастур матни

```
unit anim_;
interface
uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs;
type
  TForm1 = class(TForm)
    Image1: TImage;
    Timer1: TTimer;
    Procedure FormActivate(Sender:TObject);
    Procedure Timer1Timer(Sender:TObject);
    Procedure FormClose(Sender:Tobject; var Action:TCloseAction);
  private
    { Private declarations }
  public
    { Public declarations }
  end;
var
  Form1: TForm1;
  Back, bitmap, Buf:Tbitmap; //фон, тасвир, буфер
  BackRct, bitmapRct, BufRct:Trect; //фон, тасвир, буфер сохаси
  x,y:integer; // тасвирнинг юқори чап бурчак координатаси
  w,h:integer; // тасвир ўлчами
implementation
{$R *.DFM}
Procedure TFormActivate(Sender:TObject);
```

```

Begin
  Back:=Tbitmap.Create; //фон
  Bitmap:=Tbitmap.Create; //тасвир
  Buf:=Tbitmap.Create; //буфер
  // юклаш ва фонни хосил қилиш
  Back.LoadFromFile('factory.bmp');
  Form1.Image1.Canvas.Draw(0,0,Back);
  // харакатланувчи тасвирни юклаш
  bitmap.LoadFromFile('aplane.bmp');
  bitmap.Transparent:=true;
  bitmap.TransparentColor:=bitmap.Canvas.pixels[1,1];
  // фон соҳаси нухасини сақлаш учун буфер ташкил этиш
  w:=bitmap.Width;
  h:=bitmap.Height;
  Buf.Width:=w;
  Buf.Height:=h;
  Buf.Palette:=Back.Palette;
  Buf.Canvas.CopyMode:=cmSrcCopy;
  BufRct:=Bounds(0,0,W,H);
  X:=-40;
  Y:=20; //сақланувчи фон соҳасини аниқлаймиз
  BackRct:=Bounds(x,y,w,h); // ва уни сақлаймиз
  Buf.Canvas.CopyRect(BufRct,Back.Canvas,backRct);
End;
Procedure TForm1.Timer1Timer(Sender:TObject);
Begin // фонни тиклаймиз (буфердан) расмни йўқ қиламиз
  Form1.Image1.canvas.Draw(x,y,Buf);
  X:=x+2;
  If x>form1.Image1.width then x:=-40; //сақланувчи фон соҳасини аниқлаймиз
  BackRct:=Bounds(x,y,w,h); // унинг нухасини сақлаймиз
  Buf.Canvas.CopyRect(BufRct,Back.Canvas,BackRct); //расмни чиқарамиз
  Form1.Image1.canvas.Draw(x,y,bitmap);
End;
Procedure TForm1.FormClose(Sender:TObject; var Action: TcloseAction);
Begin
  Back.Free;
  Bitmap.Free;
  Buf.Free;
End;
end.

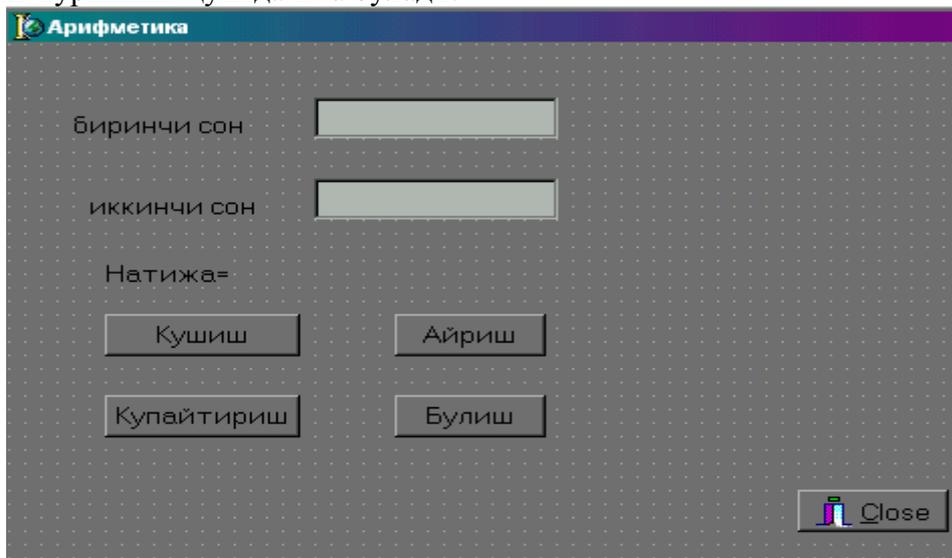
```

Чизиқли алгоритмларга доир дастурлар тузиш.

1.1-масала. Delphi да чизиқли дастурга мисол сифатида арифметик амалларни ҳисоблаш дастурини кўриб чиқамиз.

Бу дастурни тузиш учун янги формада киритиладиган иккита сон учун алоҳида «ойна» очамиз. Бунинг учун компоненталар рўйхатидан Edit танланади. Edit да киритиладиган маълумотлар матнли маълумот бўлади, шунинг учун биз киритган матн шаклидаги сонларни ҳақиқий сонга айлантириш керак. Бунинг учун Delphi да strtofloat(edit1.text) функцияси

ишлатилади. Бунда `strtfloat` функция номи `edit1.text` эса аргументи. Биз ҳосил қилган форманинг кўриниши куйидагича бўлади:



Бу формадаги биринчи сон, иккинчи сон, Натижа ёзувлари компоненталар рўйхатидан Label ни танлаш билан ҳосил қилинган. Одатда формада Label ишлатилса, унинг ишлатилиш сонига қараб Label1, Label2 ва ҳаказо шаклда чиқади, буни ўзгартириш учун Caption да Label1, Label2 ва ҳақозоларни ўчириб ўрнига дастурга мос матн киритилади.

Формада қўшиш, айриш, қўпайтириш, бўлиш амалларига стандарт тугмалардан Button компонентаси танланган. Бунда ҳам Caption да Button лар ўрнига мос равишда қўшиш, айриш, қўпайтириш, бўлиш деб ўзгартирилган.

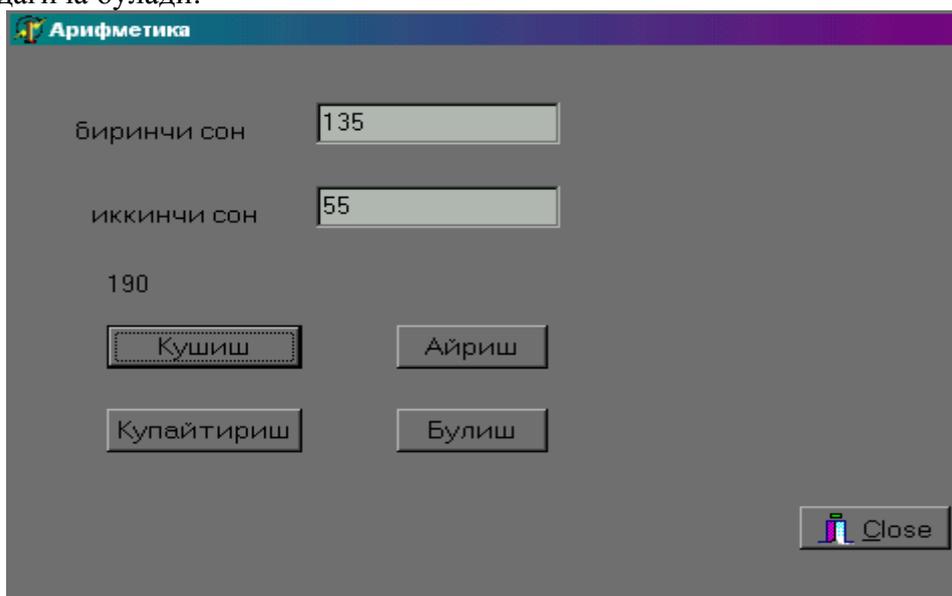
Delphi да тузилган дастурдан нормал ҳолда чиқиб кетиш учун стандарт х тугмасидан ташқари яна бошқа тугма ишлатилади. Бунинг учун компоненталар меню сатридан Additional банди танланади ва унинг ичидан BitBtn компонентаси танланади. Агар Caption да &Close деб ўзгартирила, Kind да Vkclosе деб ёзиб қўйсак, у ҳолда ёпиш тугмасида тагига чизилган C_lосе ёзуви ҳосил бўлади.

Амалларни бажариб, натижани олиш учун мос тугмаларнинг устида сичқончанинг чап тугмасини икки марта босилса, у ҳолда шу тугмага мос бўлган процедуранинг ичига автоматик кирилади. Бунда ҳар бир тугмага мос бўлган процедураларнинг кўриниши куйидагича бўлади:

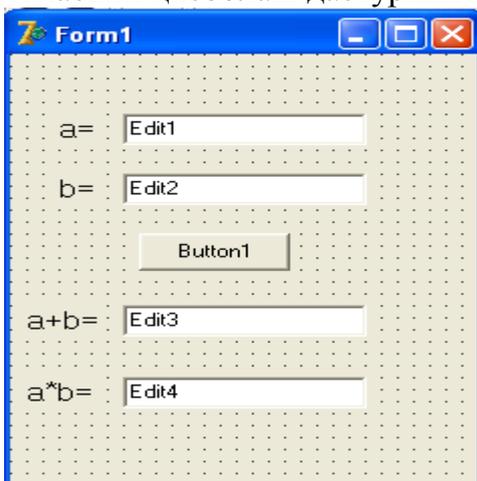
```
procedure TForm1.Button1Click(Sender: TObject);
begin
label3.Caption:=floattostr(strtfloat(edit1.Text)+ strtfloat(edit2.Text));
end;
procedure TForm1.Label3Click(Sender: TObject);
begin
label3.Caption:=floattostr(strtfloat(edit1.Text)- strtfloat(edit2.Text));
end;
procedure TForm1.Button3Click(Sender: TObject);
begin
label3.Caption:=floattostr(strtfloat(edit1.Text)* strtfloat(edit2.Text));
end;
procedure TForm1.Button4Click(Sender: TObject);
begin
label3.Caption:=floattostr(strtfloat(edit1.Text)/strtfloat(edit2.Text));
end;
procedure TForm1.Button2Click(Sender: TObject);
begin
label3.Caption:=floattostr(strtfloat(edit1.Text)- strtfloat(edit2.Text));
```

end;

Бу дастур бажарилгандан кейинги экранда ҳосил бўладиган натижанинг кўриниши қуйидагича бўлади:



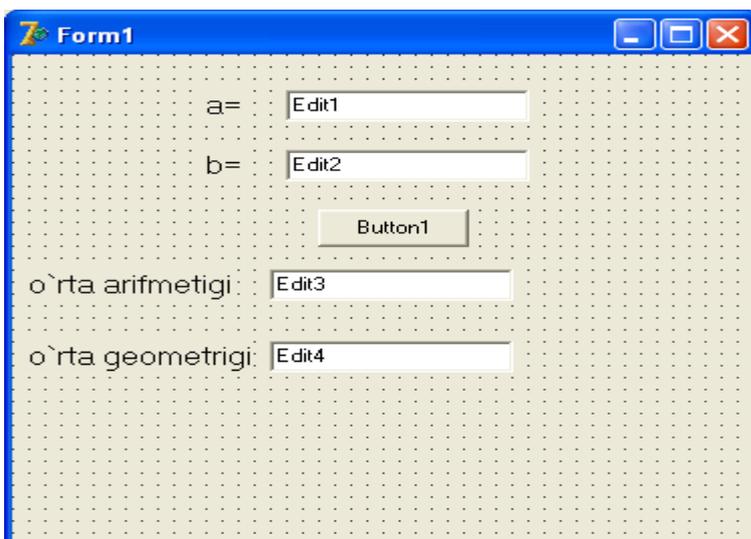
1.2-масала. А ва Б иккита ҳақиқий сонлар берилган. Уларнинг йиғиндиси ва кўпайтмасини ҳисоблаш дастурини тузинг.



Дастур матни

```
procedure TForm1.Button1Click(Sender: TObject);
var a,b,s,p:real;
begin
a:=StrToFloat(Edit1.Text);
b:=StrToFloat(Edit2.Text);
s:=a+b;
p:=a*b;
Edit3.Text:=FloatToStr(s);
Edit4.Text:=FloatToStr(p);
end;
end.
```

1.3-масала. Иккита мусбат сон берилган, бу сонларнинг ўрта арифметик ва ўрта геометрик \Rightarrow ийматларини ҳисоблаш дастурини тузинг.



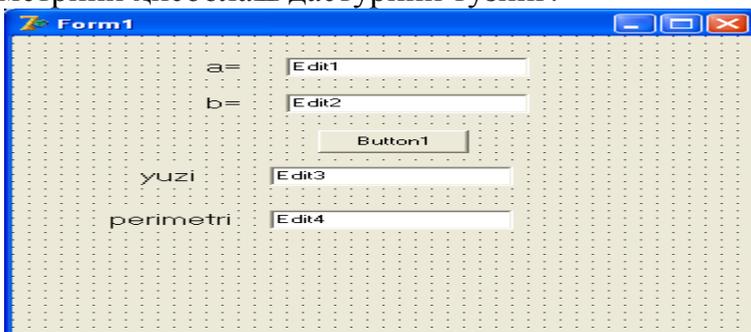
Дастур матни

```

procedure TForm1.Button1Click(Sender: TObject);
var a,b,s,p:real;
begin
a:=StrToFloat(Edit1.Text);
b:=StrToFloat(Edit2.Text);
s:=(a+b)/2;
p:=sqrt(a*b);
Edit3.Text:=FloatToStr(s);
Edit4.Text:=FloatToStr(p);
end;
end.

```

1.4-масала. Томонлари a ва b га тенг бўлган тўри тўртбурчакнинг юзи ва периметрини ҳисоблаш дастурини тузинг.



Дастур матни

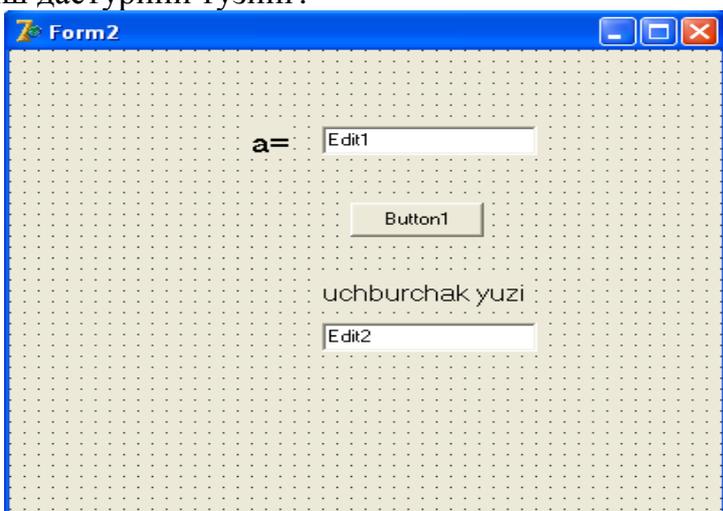
```

procedure TForm1.Button1Click(Sender: TObject);
var
a,b,s,p: real;
begin
a:=StrToFloat(Edit1.Text);
b:=StrToFloat(Edit2.Text);
s:=a*b; p:=2*(a+b);
Edit3.Text:=FloatToStr(s);
Edit4.Text:=FloatToStr(p);
end;
end.

```

End; end.

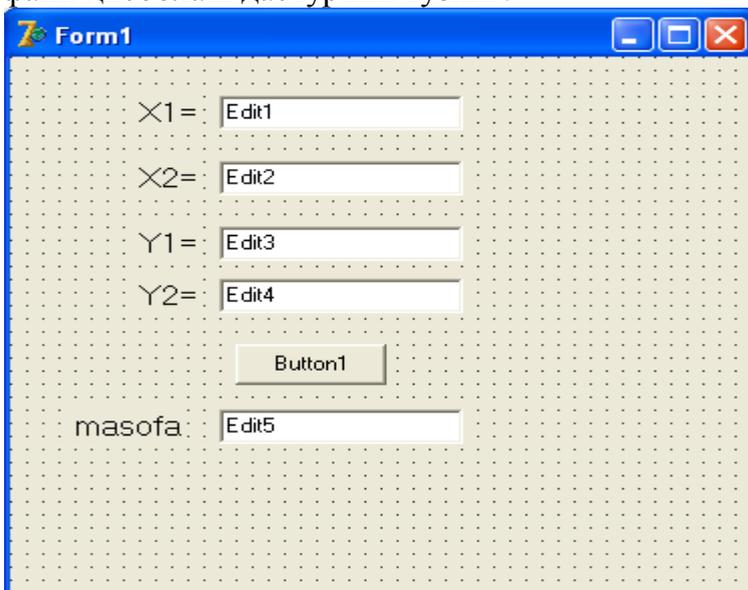
1.5-масала. Тенг томонли учбурчакнинг томони А га тенг. Учбурчакнинг юзини топиш дастурини тузинг.



Дастур матни

```
procedure TForm1.Button1Click(Sender: TObject);
var a,s:real;
begin
a:=StrToFloat(Edit1.Text);
S:=(sqrt(3))*a*a/4;
Edit2.Text:=FloatToStr(S);
end;
end.
```

1.6-масала. Координаталари x_1, y_1 ва x_2, y_2 га тенг бўлган нуқталар орасидаги масофани ҳисоблаш дастурини тузинг.



Дастур матни

```
procedure TForm1.Button1Click(Sender: TObject);
var x1,x2,y1,y2,d:real;
begin
x1:=StrToFloat(Edit1.Text);
```

```

x2:=StrToFloat(Edit2.Text);
y1:=StrToFloat(Edit3.Text);
y2:=StrToFloat(Edit4.Text);
d:=sqrt(sqr(x2-x1)+sqr(y2-y1));
Edit5.Text:=FloatToStr(d);
end;
end.

```

1.7-масала. Биринчи ҳади a_1 , айирмаси d , ҳадлари сони n га тенг арифметик прогрессиянинг ҳадларининг йиғиндисини ҳисоблаш дастурини тузинг.

Дастур матни

```

procedure TForm1.Button1Click(Sender: TObject);
var a1,d,n,S: real;
begin
a1:=StrToFloat(Edit1.Text);
d:=StrToFloat(Edit2.Text);
n:=StrToFloat(Edit3.Text);
S:=(2*a1+d*(n-1))/2*n;
Edit4.Text:=FloatToStr(S);
end;
end.

```

1.8-масала. Биринчи ҳади b , махражи q ва ҳадлари сони n га тенг геометрик прогрессиянинг ҳадларининг йиғиндисини S ни ҳисоблаш дастурини тузинг..

Дастур матни

```

procedure TForm1.Button1Click(Sender: TObject);
var B,N,Q,S:real;
begin
B:=StrToFloat(Edit1.Text);

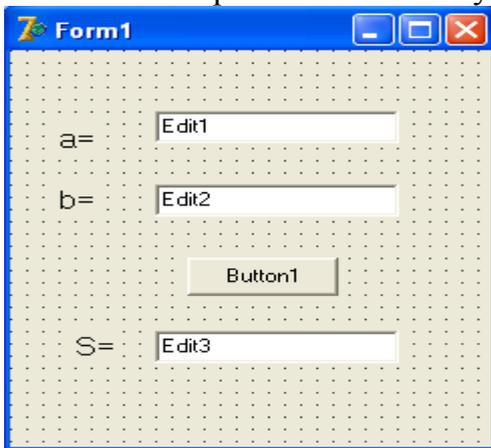
```

```

Q:=StrToFloat(Edit2.Text);
N:=StrToFloat(Edit3.Text);
S:=(B*(1-exp(N*ln(Q)))/(1-Q);
Edit4.Text:=FloatToStr(S);
end;
end.

```

1.9-масала. Берилган соннинг бутун қисмини аниқланг.



Дастур матни

```

procedure TForm1.Button1Click(Sender: TObject);
var a,b,s:real;
begin
a:=StrToFloat(Edit1.Text);
b:=StrToFloat(Edit2.Text);
s:=a div b;
Edit3.Text:=FloatToStr(s);
end;
end.

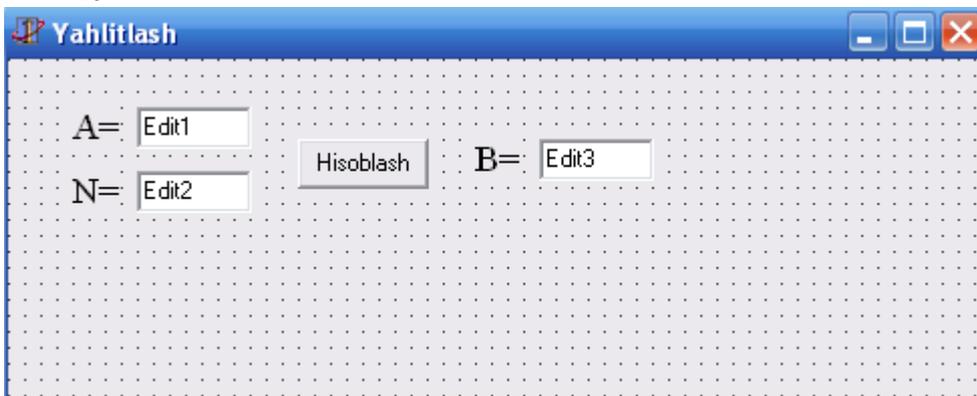
```

1.10-масала. Сонни берилган аниқликда яхлитланг.

Ечиш.

А сонни N та ўнли хоналар аниқлигида яхлитлаш учун

$$B = \text{INT} \left(\frac{A \cdot 10^N + 0.5}{10^N} \right);$$
 формуладан фойдаланамиз.



Дастур матни

```

procedure TForm1.Button1Click(Sender: TObject);
var A,N,B:real;

```

```

begin
  A:=StrToFloat(Edit1.Text);
  N:=StrToFloat(Edit2.Text);
  B:=INT(A*EXP(N*LN(10))+0.5)/EXP(N*LN(10));
  Edit3.Text:=FloatToStr(B);
end;

```

1.11-масала. Учта идишга сув солинган. Идишлардаги сувнинг температураси T₁, T₂, T₃ га тенг, ҳажми V₁, V₂, V₃ (l) га тенг. Идишлардаги сувни битта идишга қуйилса, унинг ҳажми ва температураси қандай бўлади?

Ечиш. Идишлардаги сувни битта идишга қуйилса, сувнинг ҳажми ва температураси

$V = V_1 + V_2 + V_3$; $T = \frac{V_1 * T_1 + V_2 * T_2 + V_3 * T_3}{V}$ формулалар билан аниқланади.

Дастур матни

```

procedure TForm1.Button1Click(Sender: TObject);
var T1,T2,T3,T,V1,V2,V3,V:real;
begin
  T1:=StrToFloat(Edit1.Text);
  T2:=StrToFloat(Edit2.Text);
  T3:=StrToFloat(Edit3.Text);
  V1:=StrToFloat(Edit4.Text);
  V2:=StrToFloat(Edit5.Text);
  V3:=StrToFloat(Edit6.Text);
  T:=(T1*V1+T2*V2+T3*V3)/(V1+V2+V3);
  V:=V1+V2+V3;
  Edit7.Text:=FloatToStr(T);
  Edit8.Text:=FloatToStr(V);
end;

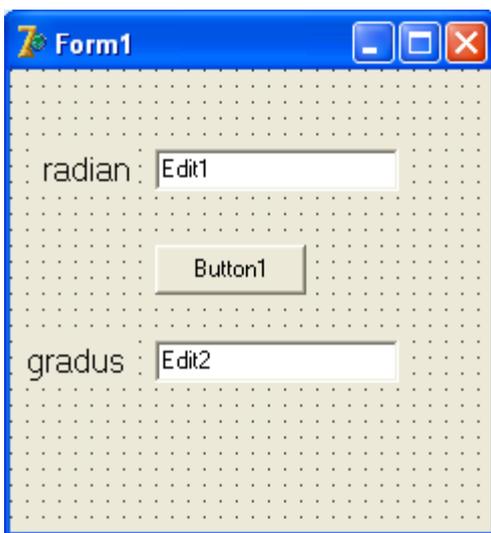
```

1.11-масала.Берилган бурчакни радиан ўлчовидан градус ўлчовига ўтказинг.

Ечиш. А градусга тенг бурчакни радиан ўлчовига

$$B = A \cdot \frac{3,14159}{180};$$

ушбу формула ёрдамида ўтказилади.



Дастур матни

```

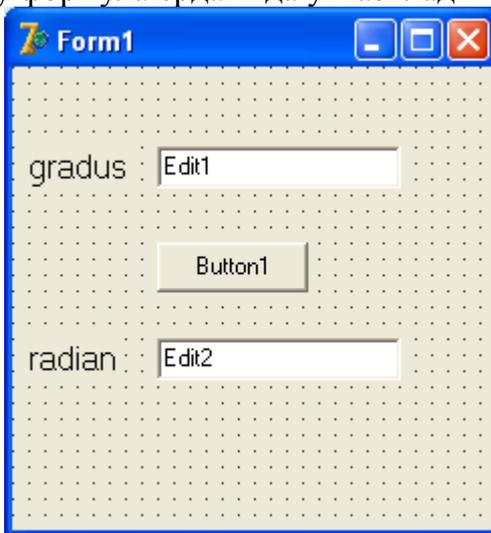
procedure TForm1.Button1Click(Sender: TObject);
var alfa,k:real;
begin
alfa:=strtofloat(Edit1.Text);
k:=alfa*pi/180;
Edit2.Text:=FloatToStr(k);
end;

```

1.12-масала. Берилган бурчакни градус ўлчовидан радиан ўлчовига ўтказинг.
 Ечиш. А радианга тенг бурчакни градус ўлчовига ўтказиш учун,

$$B = \frac{A \cdot 180}{3,14159};$$

ушбу формула ёрдамида ўтказилади.



Дастур матни

```

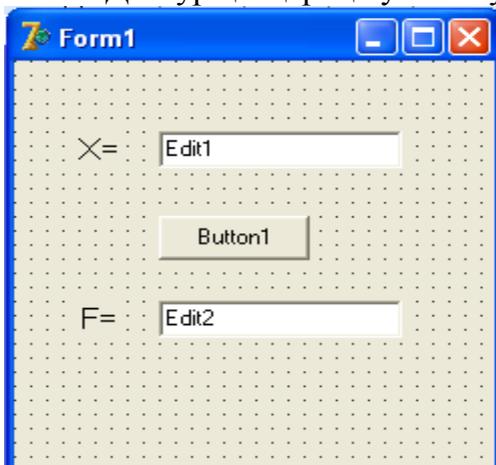
procedure TForm1.Button1Click(Sender: TObject);
var a,b:real;
begin
a:=StrToFloat(Edit1.Text);
b:=a*180/pi;

```

```
Edit2.Text:=FloatToStr(b);  
end;
```

1.13-масала. Аргумент X нинг қийматлари берилганда $F=2(x+3)+3(x+3)^2$ функциянинг қийматларини аниқланг.

Ечиш. Дастур қисқароқ бўлиши учун $y=x+3$ оралиқ ўзгарувчини киритамиз.



Дастур матни

```
procedure TForm1.Button1Click(Sender: TObject);  
var x,f:real;  
begin  
x:=StrToFloat(Edit1.Text);  
F:=2*(x+3)+3*sqr(x+3);  
Edit2.Text:=FloatToStr(F);  
end;
```

1.14-масала. Узунлиги $L(m)$ га тенг математик маятникнинг тебраниш даврини аниқланг. (ҳисоблаш формуласи $T=2\pi\sqrt{LG}$, бунда $\pi = 3.14$; $G = 9.81$ (m/s^2)).



Дастур матни

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

```

const G=9.81;
var L,T:REAL;
begin
L:=StrToFloat(Edit1.Text);
T:=2*pi*sqrt(L/G);
Edit2.Text:=FloatToStr(T);
end;

```

1.15-масала. Айлананинг узунлиги C берилган. Шу айлана билан чегараланган доиранинг юзи S ни аниқланг. (хисоблаш формуласи: $S=C^2/4\pi$).



Дастур матни

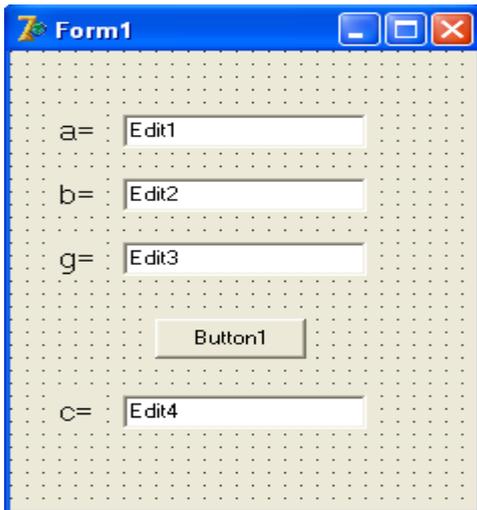
```

procedure TForm1.Button1Click(Sender: TObject);
var c,S:real;
begin
c:=StrToFloat(Edit1.Text);
S:=c*c/(4*pi);
Edit2.Text:=FloatToStr(S);
end;

```

1.16-масала. Учбурчакнинг A ва B иккита томони ва улар орасидаги бурчаги Γ (градусда) берилган. Учбурчакнинг учинчи томонини топинг.

(хисоблаш формуласи: $C=\sqrt{(A^2 + B^2 - 2AB \cdot \cos\Gamma)}$).



Дастур матни

```

procedure TForm1.Button1Click(Sender: TObject);
var a,b,g,c:real;
begin
a:=StrToFloat(Edit1.Text);
b:=StrToFloat(Edit2.Text);
g:=StrToFloat(Edit3.Text);
g:=g*pi/180;
c:=sqrt(a*a+b*b-2*a*b*cos(g));
Edit4.Text:=FloatToStr(c);
end;

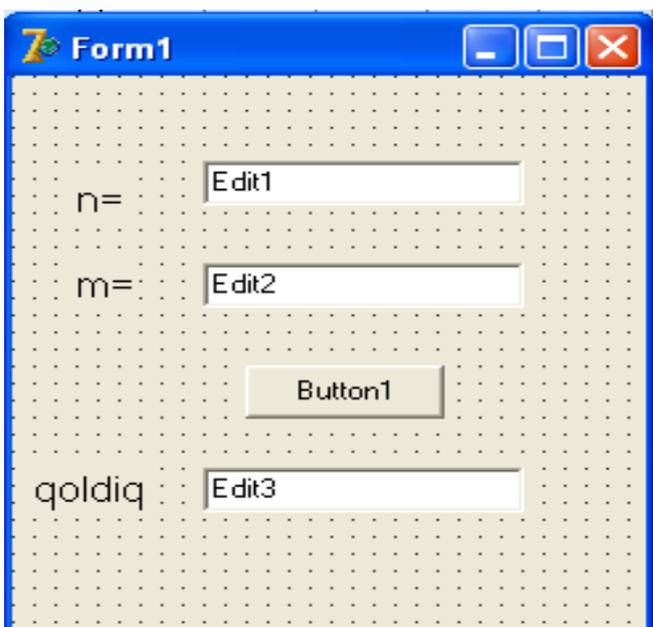
```

1.17-масала. N/M ифодани ҳисоблашда ҳосил бўладиган қолдиқни топинг.

Ечиш.

Қолдиқни Z билан белгиласак, у ҳолда

$Z = N - INT \left(\frac{N}{M} \right) \cdot M$ формула билан ҳисобланади.



Дастур матни

```

procedure TForm1.Button1Click(Sender: TObject);
var
N,M,R:integer;
begin
N:=StrToInt(Edit1.Text);
M:=StrToInt(Edit2.Text);
R:=N mod M;
Edit3.Text:=IntToStr(R);
end;

```

1.18-масала. R_1, R_2, R_3 учта қаршиликлар кетма-кет уланган занжирнинг қаршилигини аниқланг.

Ечиш.

Занжирнинг умумий қаршилиги R билан белгиласак, кетма-кет улашда формулалар $R = R_1 + R_2 + R_3$ ўринли бўлади.

Дастур матни

```

procedure TForm1.Button1Click(Sender: TObject);
var
R1,R2,R3,R:real;
begin
R1:=StrToFloat(Edit1.Text);
R2:=StrToFloat(Edit2.Text);
R3:=StrToFloat(Edit3.Text);
R:=R1+R2+R3;
Edit4.Text:=FloatToStr(R);
end;

```

1.19-масала. Массалари M_1 ва M_2 (kg) га тенг, ораларидаги масофа R (m) га тенг бўлган иккита қисмнинг ўзаро тортилиш кучи F аниқлансин. Бунда гравитацион доимийси

$G=6,672 \cdot 10^{-11}$ ($N \cdot m^2/kg^2$) деб олинсин.

Ечиш. Бутун олам тортилиш қонунига кўра

$$F = G \frac{m_1 \cdot m_2}{R^2};$$

Ернинг массаси $m_1 = 5,97 \cdot 10^{24}$

Ойнинг массаси $m_2 = 7,35 \cdot 10^{22}$

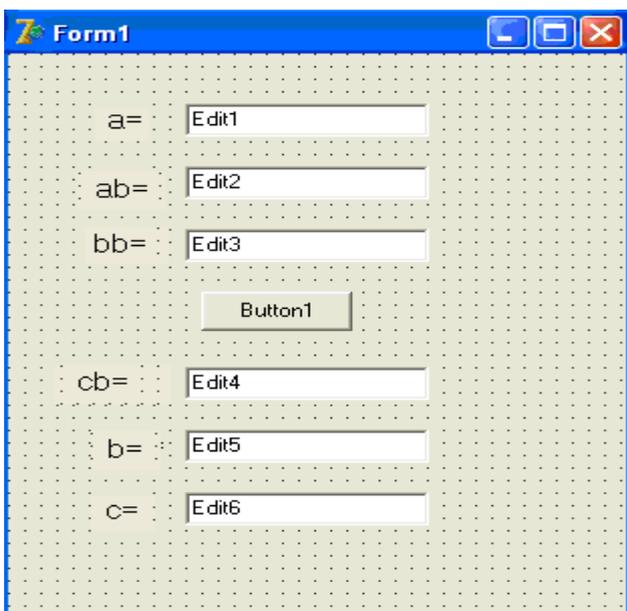
Ер билан ойнинг орасидаги масофа $R = 3,844 \cdot 10^8$

Изох. Ер билан ойнинг массалари килограммда, масофа метрда, куч Нютонда ўлчанади.

Дастур матни

```
procedure TForm1.Button1Click(Sender: TObject);
const G=6.672E-11;
var
M1,m2,R,F:real;
begin
M1:=StrToFloat(Edit1.Text);
M2:=StrToFloat(Edit2.Text);
R:=StrToFloat(Edit3.Text);
F:=G*M1*M2/(R*R);
Edit4.Text:=FloatToStr(F);
end.
```

1.20-масала. Бир томони ва унга ёпишган иккита бурчаги берилган учбурчакнинг учинчи бурчаги ва қолган икки томонини аниқланг.



Дастур матни

```

procedure TForm1.Button1Click(Sender: TObject);
var a,ab,bb,b,c,cb:real;
begin
a:=StrToFloat(Edit1.Text);
ab:=StrToFloat(Edit2.Text);
bb:=StrToFloat(Edit3.Text);
cb:=pi-ab*pi/180-bb*pi/180;
b:=a*sin(bb)/sin(ab);
c:=a*sin(cb)/sin(ab);
Edit4.Text:=FloatToStr(cb);
Edit5.Text:=FloatToStr(b);
Edit6.Text:=FloatToStr(c);
end;

```

Тармоқланувчи алгоритмларга доир дастурлар тузиш.

2.1-масала. Тармоқланувчи дастурларга доир Delphi да квадрат тенгламанинг ҳақиқий илдизларини топиш дастури тузилган. Бу дастурга мос процедураларнинг кўриниши куйидагича бўлади:

```

procedure TForm1.Label4Click(Sender: TObject);
var d,a,b,c,x1,x2:real;
begin
a:=strtofloat(edit1.Text);
b:=strtofloat(edit2.Text);
c:=strtofloat(edit3.Text);
d:=b*b-4*a*c;
if d<0 then Label4.Caption:='хақиқий ечими йук'
else begin x1:=(-b-sqrt(d))/(2*a);

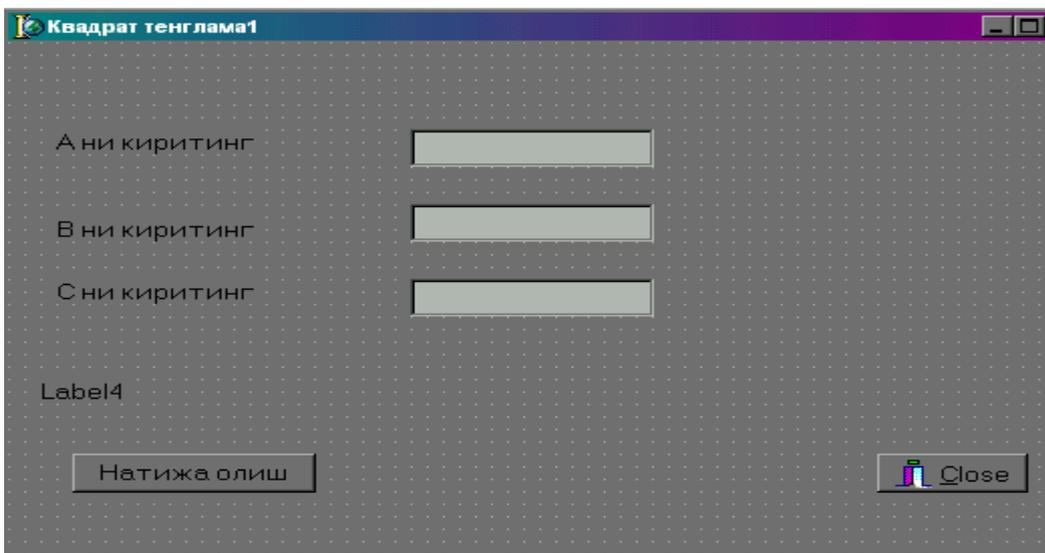
```

```

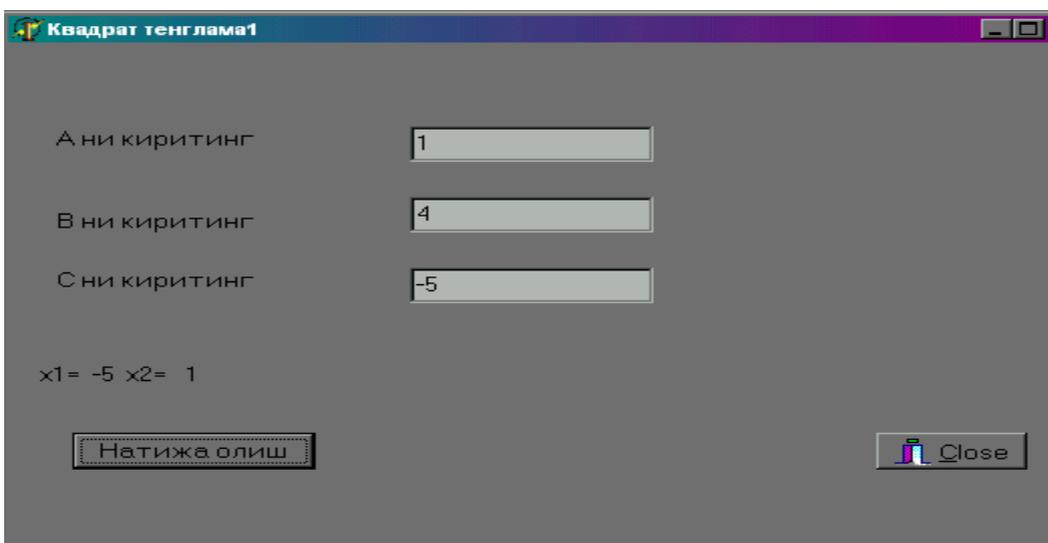
x2:=(-b+sqrt(d))/(2*a);
Label4.Caption:='x1= '+floattostr(x1)+' x2= '+floattostr(x2);
end;
end;
procedure TForm1.Button1Click(Sender: TObject);
var d,a,b,c,x1,x2:real;
begin
a:=strtofloat(edit1.Text);
b:=strtofloat(edit2.Text);
c:=strtofloat(edit3.Text);
d:=b*b-4*a*c;
if d<0 then Label4.Caption:='хакикий ечими йук'
else begin x1:=(-b-sqrt(d))/(2*a);
x2:=(-b+sqrt(d))/(2*a);
Label4.Caption:='x1= '+floattostr(x1)+' x2= '+floattostr(x2);
end;
end;

```

Бу дастурга мос келувчи форманинг кўриниши қуйидагича:

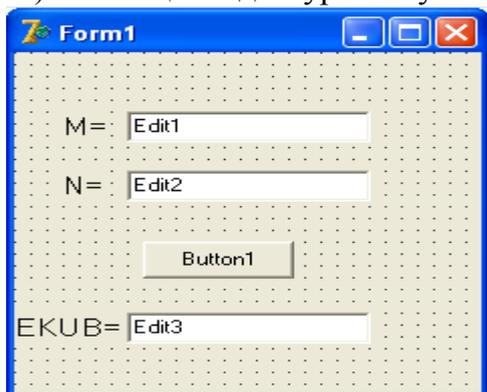


Дастур бажарилгандан кейинги натижанинг кўриниши эса қуйидагича бўлади:



2.

2.2масала. Икки бутун мусбат сон m ва n ларнинг энг катта умумий бўлувчиси (EKUB) ни аниқлаш дастурини тузинг.



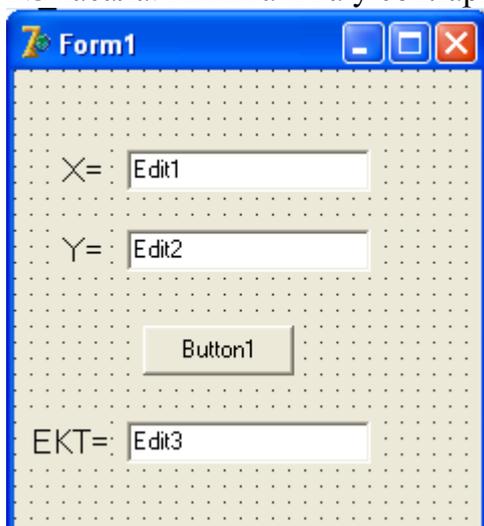
Дастур матни

```

procedure TForm1.Button1Click(Sender: TObject);
label 1,2;
var m,n,x,y:integer;
begin
m:=StrToInt(Edit1.Text);
n:=StrToInt(Edit2.Text);
x:=m;y:=n;
1: if x=y then goto 2;
if x>y then x:=x-y else y:=y-x; goto 1;
2: Edit3.Text:=IntToStr(x);
end;

```

2.3масала. Иккита x ва y сонларнинг каттасини танлаш (EKT) дастурини тузинг.



Дастур матни

```

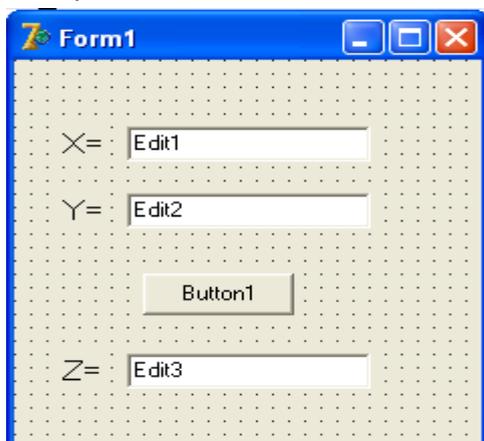
procedure TForm1.Button1Click(Sender: TObject);
var m,y,x:integer;
begin
x:=StrToFloat(Edit1.Text);
y:=StrToFloat(Edit2.Text);
if x>y then begin m:=x; Edit3.Text:=FloatToStr(m); end
else if y>x then begin m:=y; Edit3.Text:=FloatToStr(m); end

```

```
else showmessage('Bu sonlar teng!');
end;
```

2.4масала. X ва у ҳақиқий сонлар берилган. z ни ҳисоблаш дастурини тузинг:

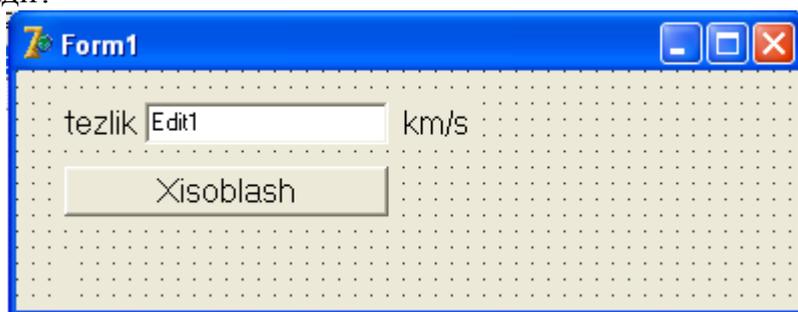
$$Z = \begin{cases} X - Y, & \text{агар } X > Y \text{ бўлса} \\ X + 1, & \text{агар } X \leq Y \text{ бўлса} \end{cases}$$



Дастур матни

```
procedure TForm1.Button1Click(Sender: TObject);
var z,y,x:integer;
begin
x:=StrToFloat(Edit1.Text);
y:=StrToFloat(Edit2.Text);
if x>y then z:=x-y;
else z:=x+1; Edit3.Text:=FloatToStr(z);
end;
```

2.5масала. Ракета v (км/соат) тезлик билан Ер экваторидаги нуқтадан Ернинг қуёш атрофидаги орбитаси бўйлаб учиради. Ракетани учуриш натижаси қандай бўлади?



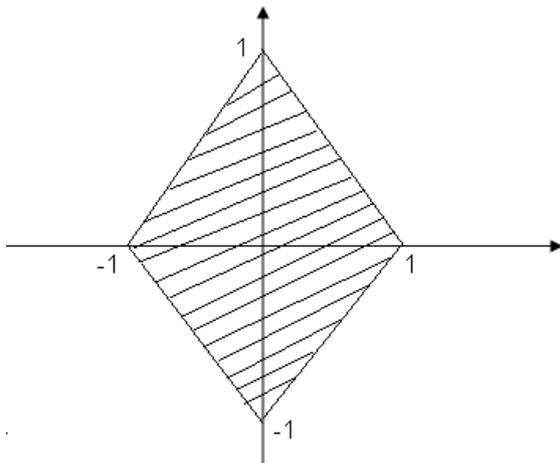
Дастур матни

```
procedure TForm1.Button1Click(Sender: TObject);
var a:real;
begin
a:=strtofloat(edit1.Text);
if a<7.9 then label3.Caption:='Raketa Yerga qaytib tushadi';
if (a>=7.9) and (a<11.2) then label3.Caption:='Raketa Yerning sun`iy yo`ldoshiga
aylanadi';
```

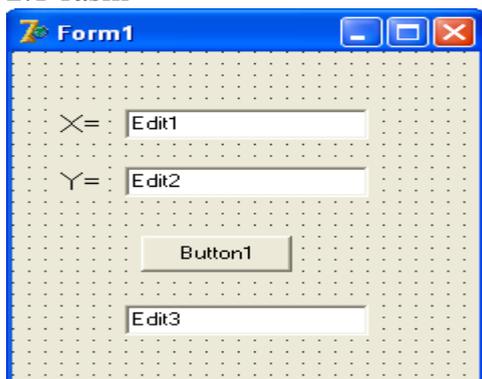
if (a>=11.2) and (a<16.7) then label3.Caption:= 'Quyoshning sun`iy yo`ldoshiga aylanadi';

if a>=16.7 then label3.Caption:='Raketa Galaktikaning sun`iy yo`ldoshiga aylanadi';
end;
end.

2.6-масала. Координаталари x ва y га тенг бўлган нуқта 2.1-расмда тасвирланган текисликдаги шаклга тегишлими?



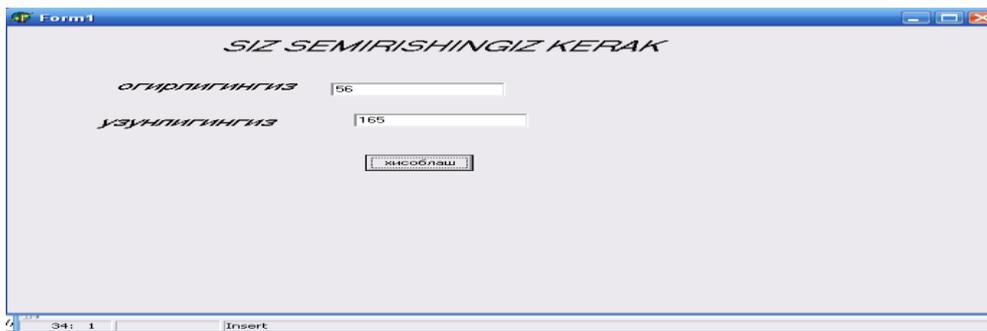
2.1-rasm



Дастур матни

```
procedure TForm1.Button1Click(Sender: TObject);  
var y,x:real;  
begin  
x:=StrToFloat(Edit1.Text);  
y:=StrToFloat(Edit2.Text);  
if (x>-1) and (x<1) and (y>-1) and (y<1) then  
Edit3.Text:='Tegishli!' else Edit3.text:='Tegishli emas!';  
end;
```

2.7-масала. Оптимал оғирликни аниқлаш дастурини ёзинг. Дастур формаси қуйидагича кўринишга эга.



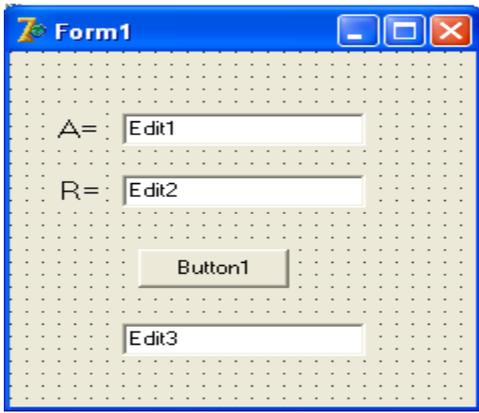
Дастур матни

```

procedure TForm1.button1click(sender: TObject);
var w,h,opt,d:real;
begin
w:=strtofloat(edit1.text);
h:=strtofloat(edit2.text);
opt:=h-100;
if w=opt then
begin
label3.caption:='optimal';
end
else
if w<opt then
begin d:=opt-w;
label3.caption:='siz semirishingiz kerak';
end
else
begin
d:=w-opt;
label3.caption:='siz ozishingiz kerak'+floattostr(d);
end;
end;
end.

```

2.8-масала. Агар квадратнинг томони A , доиранинг радиуси r га тенг бўлса, квадрат ва доиранинг юзларини солиштириб каттасини аниқланг.



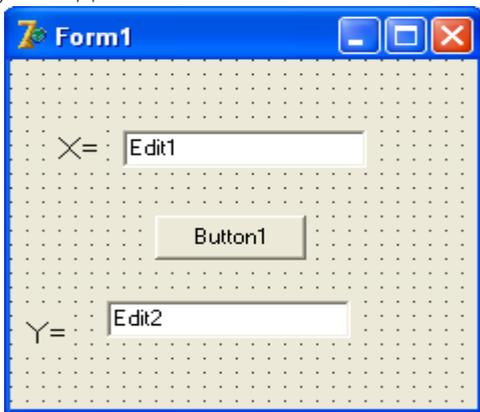
Дастур матни

```

procedure TForm1.Button1Click(Sender: TObject);
var a,r,sk,sd:real;
begin
a:=StrToFloat(Edit1.Text);
r:=StrToFloat(Edit2.Text);
sk:=a*a;sd:=pi*r*r;
if sk>sd then Edit3.Text:='Kvadratning yuzi katta!' else if sk<sd then Edit3.text:='Doiraning yuzi katta!' else Edit3.Text:='Yuzlari teng!';
end;

```

2.9-масала. Қуйидаги функция ҳисоблансин: $x > 0$ бўлганда 1 га тенг; $x = 0$ да нолга тенг; $x < 0$ да -1 га тенг.



Дастур матни

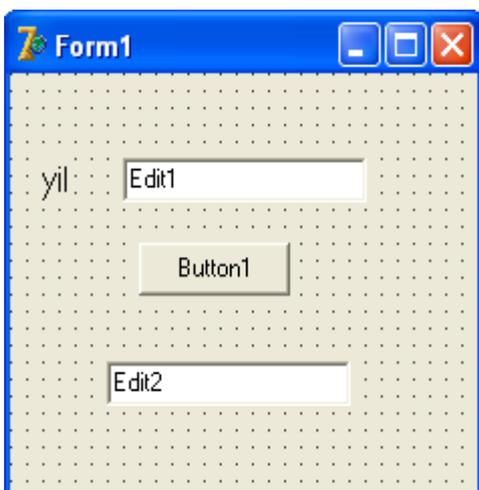
```

procedure TForm1.Button1Click(Sender: TObject);
var x,y:real;
begin
x:=StrToFloat(Edit1.Text);
if x>0 then y:=1 else if x=0 then y:=0 else y:=-1;
Edit2.Text:=FloatToStr(y);
end;

```

2.10-масала. Берилган N йил кабиса йили бўлиш-бўлмаслигини аниқланг.

Изоҳ. Агар N 100 га каррали сон бўлмаса ва унинг охири икки рақами 4 га каррали сон бўлса, у ҳолда N -йил кабиса йилидир. Агар N сони 100 каррали бўлса, у ҳолда N сони 400 га каррали бўлгандагина мазкур йил кабиса йили бўлади.



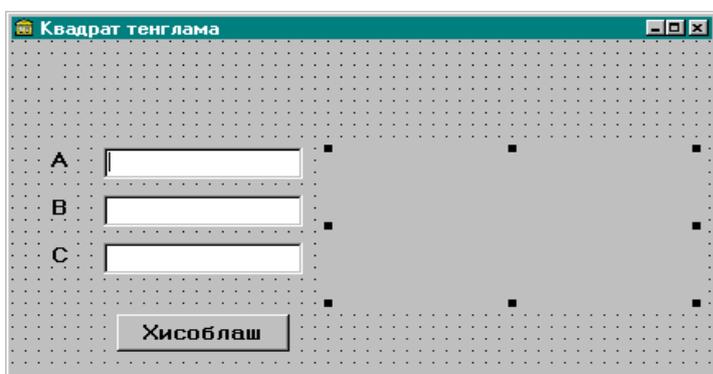
Дастур матни

```

procedure TForm1.Button1Click(Sender: TObject);
var n:integer;
begin
n:=StrToInt(Edit1.Text);
if n mod 100=0 then begin if n mod 400 =0 then Edit2.Text:='Bu yil kabisa yili!' else
Edit2.Text:='Bu yil kabisa yili emas!'; end else if n mod 4 =0 then Edit2.Text:='Bu yil
kabisa yili!'else Edit2.Text:='Bu yil kabisa yili emas!';
end;

```

2.11 Квадрат тенгламанинг илдизини топиш дастурини тузинг.



Дастур матни

```

procedure TForm1.Button1Click(Sender: TObject);
Var
a,b,c:Real; { Тенглама коэффициентлари }
d:Real; { Дискриминант }
x1,x2:Real; { Тенглама илдизлари }
begin
{ Керакли маълумотларни киритиш }
a:=StrToFloat(Edit1.Text);
b:=StrToFloat(Edit2.Text);
c:=StrToFloat(Edit3.Text);
d:=b*b-4*a*c; { Дискриминантни хисоблаш }

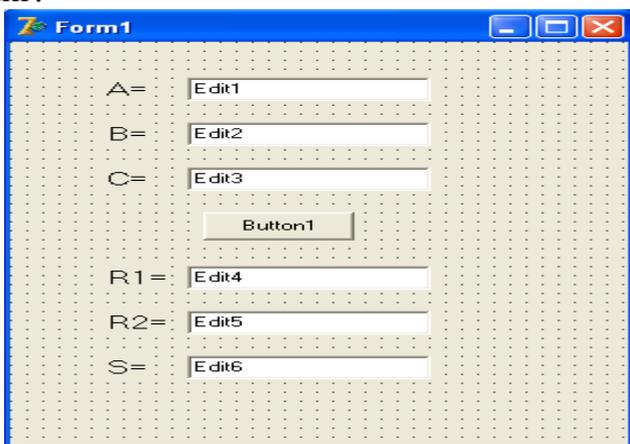
```

```

If d<0 Then
Begin
  Label5.Caption:='Дискриминант нолдан кичик'+#13+ 'Тенгламанинг илдизи йук.'
End
Else
Begin
  x1:=(-b-Sqrt(d))/(2*a);   { Илдизларни ҳисоблаш }
  x2:=(-b+Sqrt(d))/(2*a);   { x1, x2 натижани чоп этиш }
  Label5.Caption:='Тенглама илдизлари'
  +#13+'x1='+FloatToStr(x1)+#13+'x2='+FloatToStr(x2);
End;
end;
procedure TForm1.FormActivate(Sender: TObject);
begin Label1.Caption:='Тенглама коэффициентини киритинг'  +#13+'ва Ҳисоблаш туг-
масини босинг';
end;
end.

```

2.12-масала. a , b , c сонлар мос равишда учта кесманинг узунликларини ифодалайди. Агар кесмалар учбурчакнинг томонларини ифодаласа, учбурчакнинг юзи s , учбурчакка ташқи ва ички чизилган айланаларнинг радиуслари r_1 ва r_2 ларни топинг.



Дастур матни

```

procedure TForm1.Button1Click(Sender: TObject);
var a,b,c,r1,r2,s,p:real;
begin
a:=StrToFloat(Edit1.Text);
b:=StrToFloat(Edit2.Text);
c:=StrToFloat(Edit3.Text);
if ((a+b)>c) and ((a+c)>b) and ((b+c)>a) then begin
p:=(a+b+c)/2;
s:=sqrt(p*(p-a)*(p-b)*(p-c));
r2:=s/p;
r1:=a*b*c/(4*s);
Edit4.Text:=FloatToStr(r1);
Edit5.Text:=FloatToStr(r2);

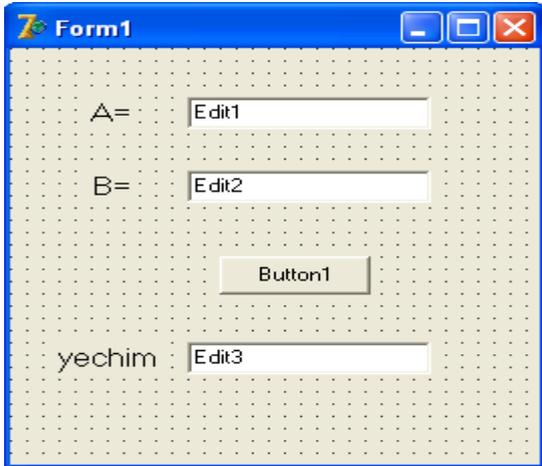
```

```

Edit6.Text:=FloatToStr(s); end else showmessage('Kiritilgan sonlar uchburchak
tomonlarini ifodalamaydi!');
end;

```

2.13-масала. $Ax+B=0$ тенгламани ечинг.



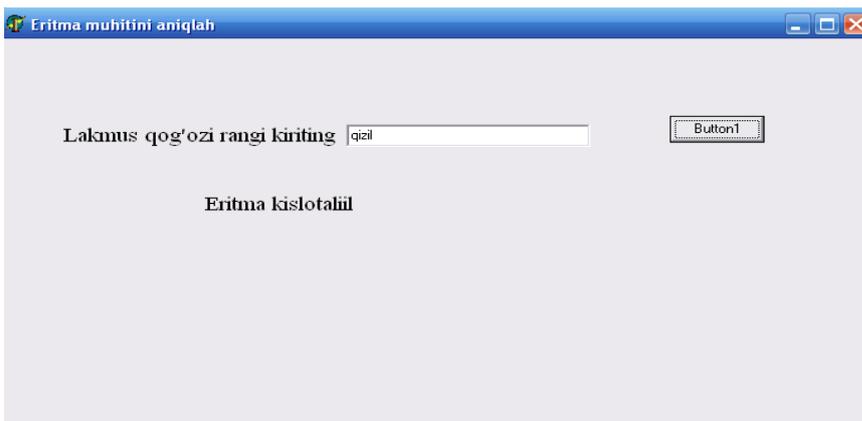
Дастур матни

```

procedure TForm1.Button1Click(Sender: TObject);
var a,b,x:real;
begin
a:=StrToFloat(Edit1.Text);
b:=StrToFloat(Edit2.Text);
if a=0 then if b<>0 then showmessage('Bu tenglamaning yechimi yo`q!')
else Edit3.Text:='Bu tenglamaning yechimlari cheksiz ko`p!' else begin x:=-b/a; Ed-
it3.Text:=FloatToStr(x);end;
end;

```

2.14-масала. Лакмус қоғозидан фойдаланиб эритма муҳитини аниқланг.



Дастур матни

```

procedure TForm1.Button1Click(Sender: TObject);
var a:string[10];
begin
a:=Edit1.text;
if a='qizil' then
Label3.Caption:='Eritma kislotaliil' else

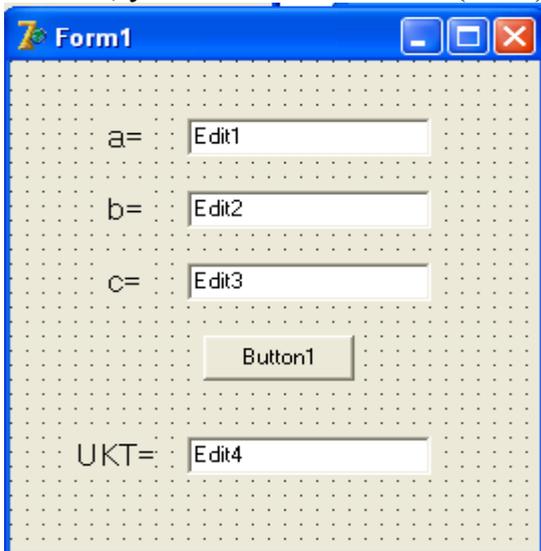
```

```

if a='ko`k' then
Label3.Caption:='Eritma ishqorli'
else
Label3.Caption:='Eritma neytral';
end;
end.

```

2.15-масала. Иккита соннинг каттаси (ИКТ) ни топиш ёрдамчи алгоритмидан фойдаланиб, укта соннинг каттаси (УКТ) ни топинг.



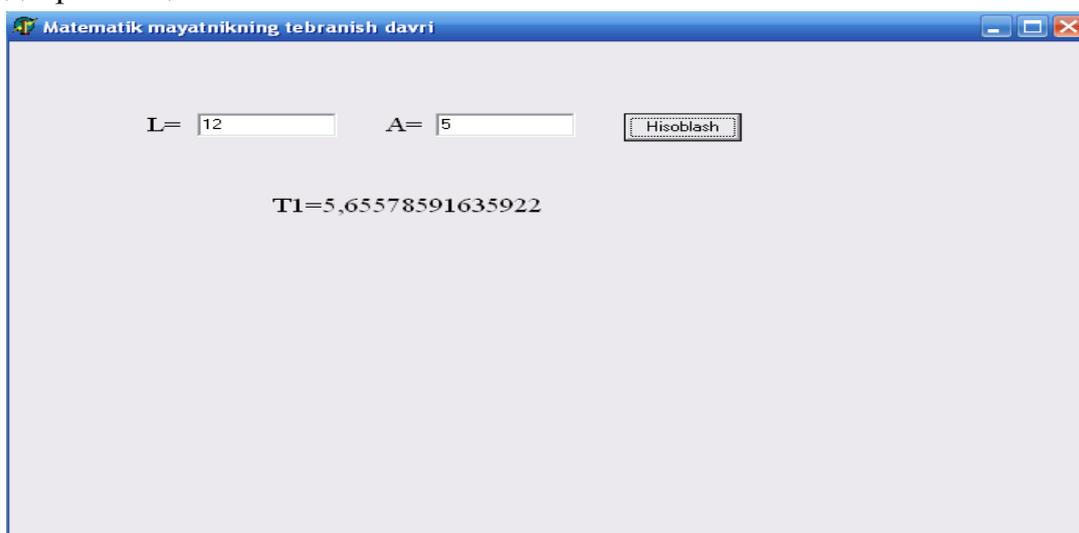
Дастур матни

```

procedure TForm1.Button1Click(Sender: TObject);
var a,b,c,k:real;
function ИКТ(x,y:real):real;
begin if x>y then ИКТ:=x else ИКТ:=y;
end;
begin
a:=StrToFloat(Edit1.Text);
b:=StrToFloat(Edit2.Text);
c:=StrToFloat(Edit3.Text);
if ИКТ(a,b)=a then k:=ИКТ(a,c) else k:=ИКТ(b,c);
Edit4.Text:=FloatToStr(k);
end;

```

2.16. Узунлиги l га тенг математик маятникнинг осилган нуқтаси қўзғалмас ёки юқорига ё пастга тезланиш билан ҳаракатланган ҳолларда унинг тебраниш даври аниқлансин.



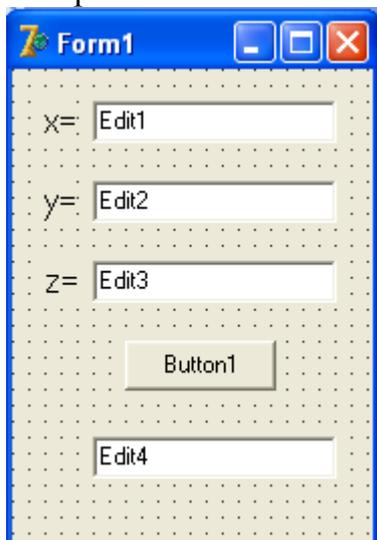
Дастур матни

```

procedure TForm1.Button1Click(Sender: TObject);
const G=9.81;
var L,A:real;
begin
L:=StrToFloat(Edit1.Text);
A:=StrToFloat(Edit2.Text);
if A=0 then Label3.Caption:='T='+FloatToStr(2*pi*sqrt(L/G))
else if A=G then Label3.Caption:='mayatnik vazinsiz holatda bo`ladi'
else if A<G then Label3.Caption:='T1='+FloatToStr(2*pi*sqrt(L/(G+A)))
else Label3.Caption:='T2='+FloatToStr(2*pi*sqrt(L/(A-G)));
end;
end.

```

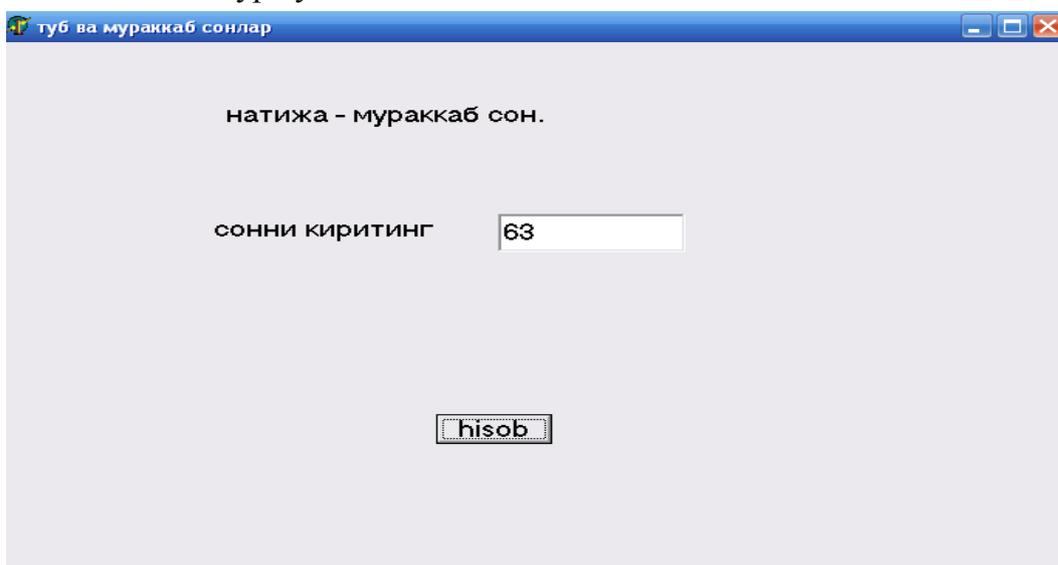
2.17-масала. Учта x , y , z хақиқий сонлар берилган. Бу сонлардан қайси бири $[1,5]$ интервалга тегишли эканлигини аниқланг.



Дастур матни

```
procedure TForm1.Button1Click(Sender: TObject);
var x,y,z:real;
begin
x:=StrToFloat(Edit1.Text);
y:=StrToFloat(Edit2.Text);
z:=StrToFloat(Edit3.Text);
if (x>1) and (x<5) then Edit4.Text:= Edit4.Text +'x,';
if (y>1) and (y<5) then Edit4.Text:= Edit4.Text +'y,';
if (z>1) and (z<5) then Edit4.Text:= Edit4.Text +'z';
end;
```

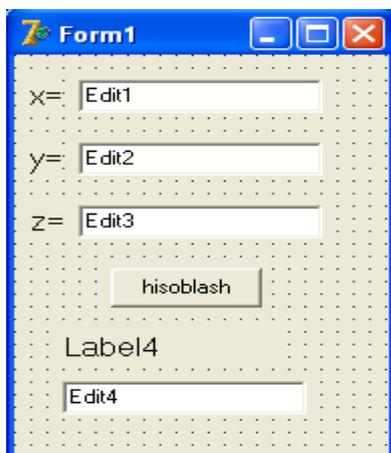
2.18. Клавиатура орқали киритилган сон туб сон ёки мураккаб сон эканлигини аникловчи дастур тузинг.



Дастур матни

```
procedure TForm1.Button1Click(Sender: TObject);
var
n: integer; // текширилаётган сон
d: integer; // бўлувчи
r: integer; // n ни d га бўлишдан қолган қолдик
begin
n:=StrToInt(Edit1.text);
d := 2; // дастлаб иккига бўламиз
repeat
r := n mod d;
if r <> 0 // n сони d га қолдиксиз бўлинмади
then d := d + 1;
until r = 0; // n га қолдиксиз бўлинадиган сонни топгунча давом этиш
label1.caption:=Edit1.text;
if d = n
then label1.caption:=label1.caption + ' - туб сон.'
else label1.caption:=label1.caption + ' - мураккаб сон.';
end;
```

2.19-масала. Учта x , y , z мусбат сонлар берилган. Томонлари x, y, z га тенг учбурчак мавжудми? Агар мавжуд бўлса бу учбурчакнинг юзини топинг.



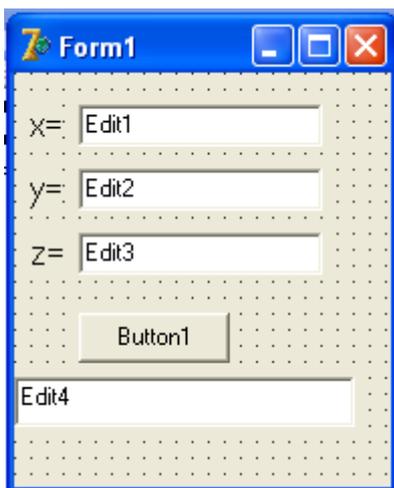
Дастур матни

```

procedure TForm1.Button1Click(Sender: TObject);
var x,y,z,s,p:real;
begin
x:=StrToFloat(Edit1.Text);
y:=StrToFloat(Edit2.Text);
z:=StrToFloat(Edit3.Text);
if (x+y>z) and (x+z>y) and (z+y>x) then begin label4.caption:='Bunday uchburchak
mavjud!';
p:=(x+y+z)/2;
s:=sqrt(p*(p-x)(p-y)(p-z));
Edit4.Text:=FloatToStr(s);end else label4.caption:='Bunday uchburchak mavjud
emas!';
end;

```

2.20-масала. Координаталари берилган $M(X, Y)$ нуқтанинг радиуси R га тенг ва маркази координаталар бошида бўлган доирага тегишли бўлишини аниқланг.



Дастур матни

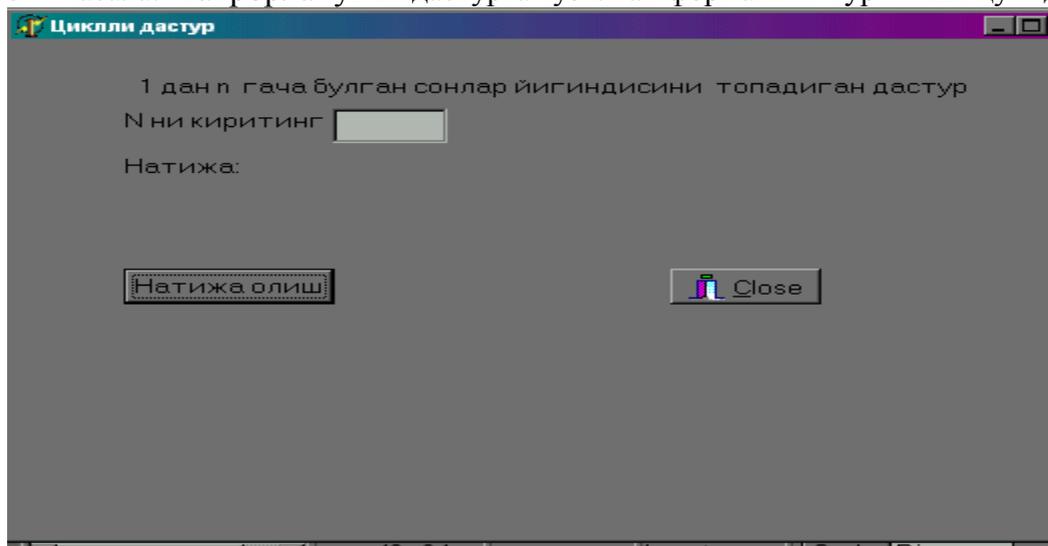
```

procedure TForm1.Button1Click(Sender: TObject);
var x,y,r:real;
begin
x:=StrToFloat(Edit1.Text);
y:=StrToFloat(Edit2.Text);
r:=StrToFloat(Edit3.Text);
if x*x+y*y>r*r then
Edit4.Text:='M('+ FloatToStr(x)+','+ FloatToStr(y)')+ nuqta radiusi
'+FloatToStr(r)+ 'bo'lgan doiraga tegishli emas!'else Edit4.Text:='M('+ FloatTo-
Str(x)+','+ FloatToStr(y)')+ nuqta radiusi '+FloatToStr(r)+'bo'lgan doiraga tegishli
!';
end;

```

Такрорланувчи алгоритмларга доир дастурлар тузиш.

3.1-масала. Такрорланувчи дастурга тузилган форманинг кўриниши қуйидагича:



Такрорланувчи дастурга Delphiда тузилган программадаги мос процедураларнинг кўриниши қуйидагича:

```

procedure TForm1.Label2Click(Sender: TObject);

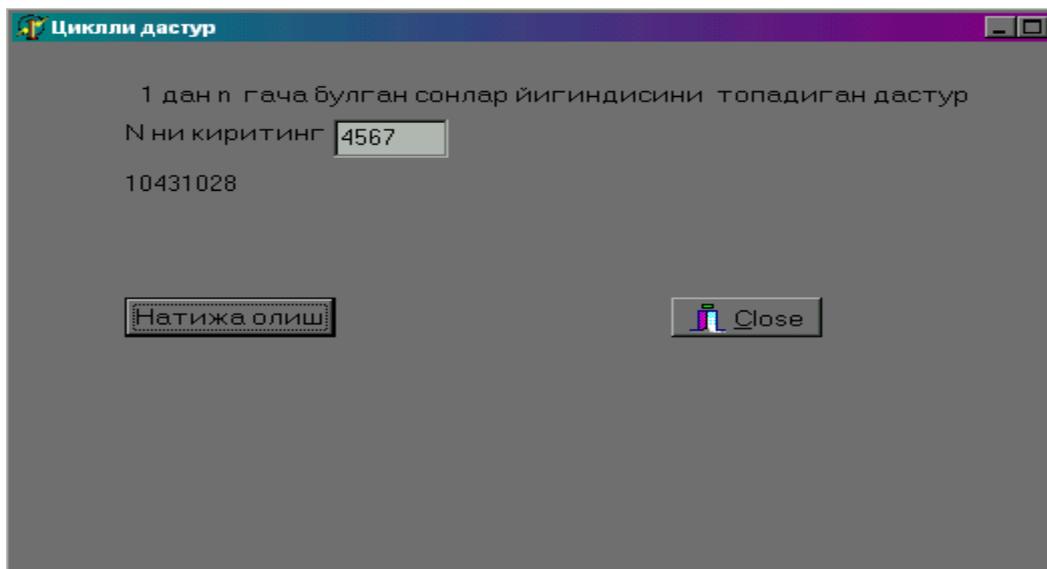
```

```

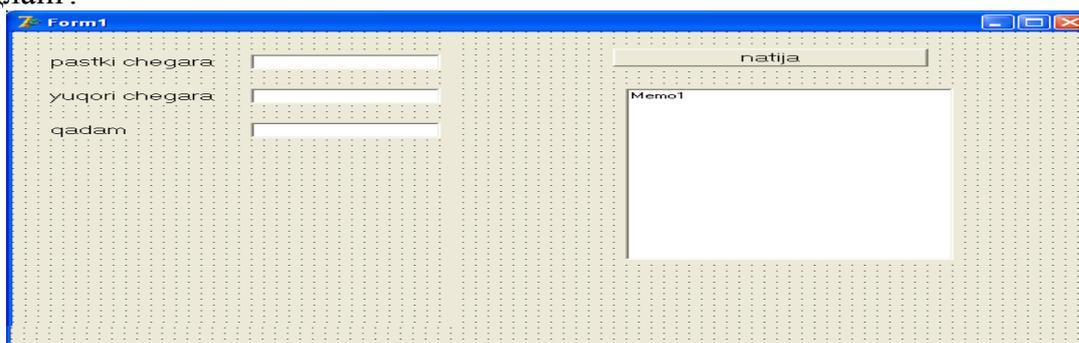
var i,s,a:integer;
begin  a:=strToInt(edit1.text);
s:=0;
for i:=1 to a do s:=s+i;
label2.Caption:=floattostr(s);
end;
procedure TForm1.Button1Click(Sender: TObject);
var i,s,a:integer;
begin  a:=strtoint(edit1.text);
s:=0; for i:=1 to a do s:=s+i;
label2.Caption:=floattostr(s);
end;

```

Дастур бажарилгандан кейин экранда куйидаги шаклда натижа ҳосил бўлади:



3.2- масала. $Y=X^2$ ни $[0,1]$ интервалда 0,1 қадам билан қийматлар жадвалини аниқланг.



Дастур матни

```

procedure TForm1.Button1Click(Sender: TObject);
var a,b:byte;
n,y,x:real;s1,s2:string;
begin
a:=strtoint(edit1.Text);
b:=strtoint(edit2.Text);
n:=strtofloat(edit3.Text); x:=a;
memo1.Clear;

```

```

repeat y:=sqr(x);
  str(x:2,s1);str(y:5,s2);
memo1.lines.Add('x='+s1+' y='+s2);
x:=x+n;
until x>=b;
end; end.

```

3.3 Мисол: $s = \sum_{i=1}^n \frac{1}{i}$ йиғиндининг n та ҳадини топиш дастурини тузиш.

Дастур матни

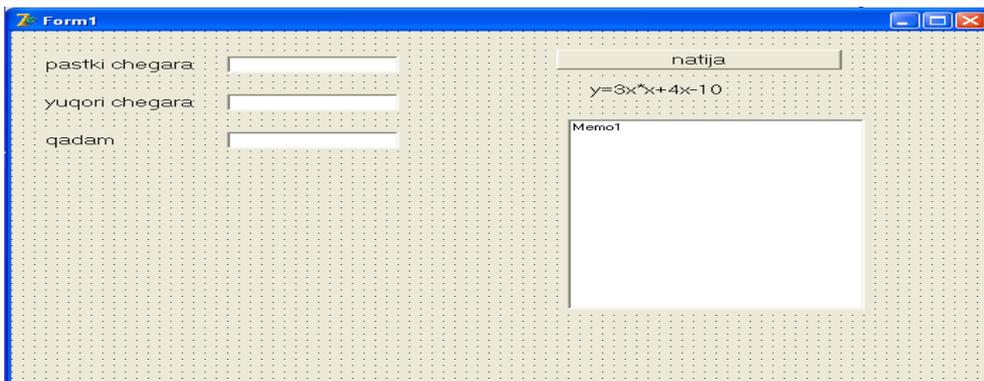
```

procedure TForm1.Button1Click(Sender: TObject);
Var
S:String;
i,n:Integer;
Summ:Real;
begin
S:=InputBox('Киритиш ойнаси','N ни киритинг','');
N:=StrToInt(S);
Summ:=0;
For i:=1 to n do Summ:=Summ+(1/i);
Label2.Caption:='Сумма=' +FloatToStr(Summ);
end;
end.

```

3.4-масала. $Y=3x^2 + 4x - 10$ функциянинг x ўзгарувчи 1). 0,1,2,3,4,5;

2). 0,3,6,9,12 га тенг қийматларини қабул қилганда у нинг қийматини ҳисобланг.



Дастур матни

```

procedure TForm1.Button1Click(Sender: TObject);
var a,b:byte;
n,y,x:real;s1,s2:string;
begin
a:=strtoint(edit1.Text);
b:=strtoint(edit2.Text);
n:=strtofloat(edit3.Text);
x:=a; memo1.Clear;
repeat

```

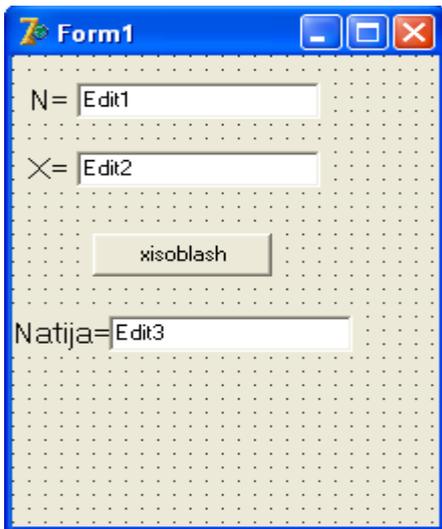
```

y:=3*sqr(x)+4*x-10; str(x:1,s1);str(y:5,s2);
memo1.lines.Add('x='+s1+' y='+s2);
x:=x+n;
until x>b;
end; end.

```

3.5- масала. N натурал сон ва x ҳақиқий сонлар берилган. Қуйидаги йиғиндини ҳисобланг.

$$\sin X + \sin^2 X + \dots + \sin^N X .$$



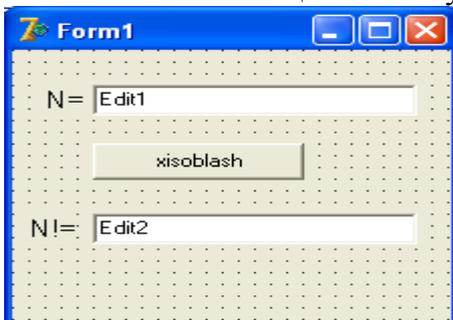
Дастур матни

```

procedure TForm1.Button1Click(Sender: TObject);
var n,i:byte; x,s:real;
begin
n:=strtoint(edit1.text);
x:=strtofloat(edit2.text);
s:=0; x:=sin(x);
for i:=1 to n do
s:=s+exp(i*ln(x));
edit3.text:=floattostr(s);
end; end.

```

3.6- масала. n! аниқлансин. Бунда n натурал сон.



Дастур матни

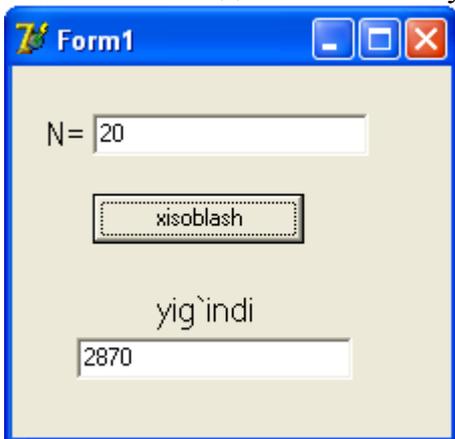
```

procedure TForm1.Button1Click(Sender: TObject);
var n,i:integer; f:int64;
begin
n:=strtoint(edit1.Text);

```

```
f:=1;
for i:=1 to n do
f:=f*i;
edit2.Text:=inttostr(f);
end; end.
```

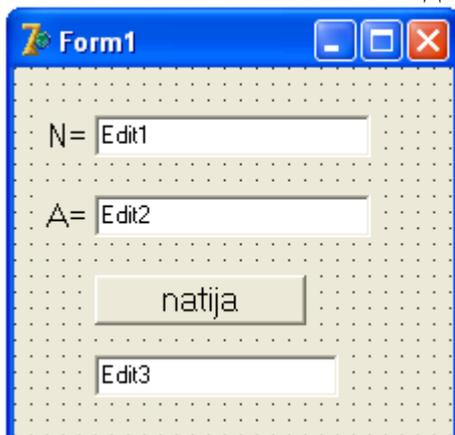
3.7- масала. 1 дан 20 гача натурал сонлар квадратлари йиғиндисини топинг.



Дастур матни

```
procedure TForm1.Button1Click(Sender: TObject);
var n,i:integer; f:int64;
begin
n:=strtoint(edit1.Text);
f:=0;
for i:=1 to n do
f:=f+sqr(i);
edit2.Text:=inttostr(f);
end; end.
```

3.8- масала. A сонининг N - даражасини такрорлаш буйру\и ёрдамида ҳисобланг.

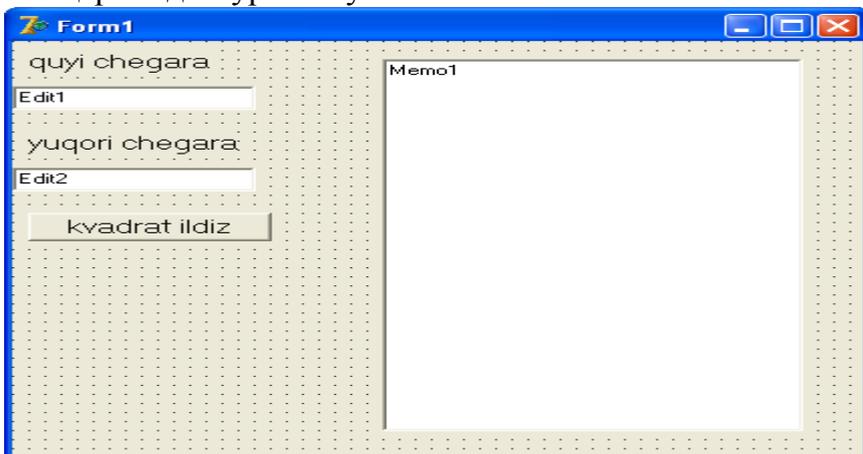


Дастур матни

```
procedure TForm1.Button1Click(Sender: TObject);
var n,i,a,f:integer;
begin
n:=strtoint(edit1.Text);
a:=strtoint(edit2.Text);
```

```
f:=1;
for i:=1 to n do f:=f*a;
edit3.Text:=inttostr(f);
end; end.
```

3.9-масала. 1 дан 10 гача бўлган сонлардан цикл қадами 1 га тенг ҳолда квадрат илдиз чиқариш дастурини тузинг.



Дастур матни

```
procedure TForm1.Button1Click(Sender: TObject);
var a,b,i:byte;
s:real; s1,s2:string;
begin
a:=strtoint(edit1.Text);
b:=strtoint(edit2.Text);
memo1.Clear;
for i:=a to b do begin
s:=sqrt(i);
str(i,s2); str(s,s1);
memo1.Lines.Add('x='+s2+' ildiz='+s1);
end;
end; end.
```

3.10-масала. 1 дан 9 гача бўлган сонларни кўпайтириш жадвалини экранга чиқаринг.



Дастур матни

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

```

var a,b,i,j,s:integer;s1,s2,s3:string;
begin
a:=strtoint(edit1.Text);
b:=strtoint(edit2.Text); memo1.Clear;
for i:=a to b do
for j:=1 to 10 do begin
s:=i*j;
str(s,s1);str(i,s2);str(j,s3);
memo1.Lines.Add(s2+'*'+s3+'='+s1);
end; end;
end.

```

3.11-масала. Мураккаб циклга доир дастур тузиш. $s = \sum_{i=1}^{10} \prod_{j=1}^5 \frac{i+j}{\sqrt{i*j}}$;



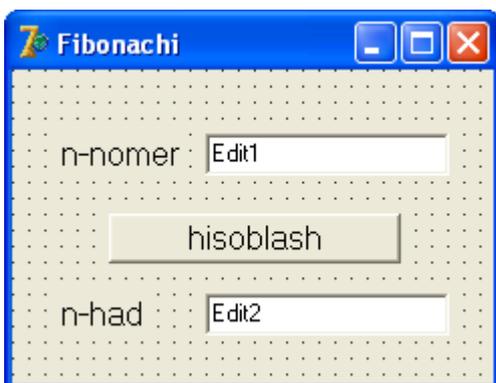
Дастур матни

```

procedure TForm1.Button1Click(Sender: TObject);
Var I,j:integer; s,p:real;
Begin
s:=0;
For i:=1 to 10 do
Begin
p:=1;
For j:=1 to 5 do p:=p*(i+j)/sqrt(i*j);
s:=s+p;
End;
Edit3.text:=floattostr(s);
End;
end.

```

3.12-масала. L номерли Фибоначчи сонини экранга чиқаринг.



Дастур матни

```

procedure TForm1.Button1Click(Sender: TObject);
var n,v,w,r,i:integer;
begin
n:=strtoint(edit1.Text);
w:=0;v:=1;i:=1;
while i<n do begin
r:=v+w; w:=v; v:=r;
i:=i+1;end;
edit2.Text:=inttostr(v);
end;
end.

```

3.13-масала. $y = e^x = 1 + x + \frac{x^2}{2!} + \dots + \frac{x^N}{N!} + \dots$

функциянинг қийматини 0,001 аниқликда ҳисобланг.



Дастур матни

```

procedure TForm1.Button1Click(Sender: TObject);
var x,y,p,s,eps:real; n:integer;
begin
eps:=strtfloat(edit1.Text);
x:=strtfloat(edit2.Text);
s:=1;p:=1;n:=1;
repeat
p:=p*x/n; s:=s+p;
n:=n+1;
until abs(p)<=eps;
edit3.Text:=floattostr(s);
end; end.

```

3.14-масала. N натурал сон ва A ҳақиқий сон берилган. Қуйидаги кўпайтмани ҳисобланг:

$$A(A+1)(A+2) \dots (A+N)$$

Дастур матни

```
procedure TForm1.Button1Click(Sender: TObject);
var a,p:real;i,n:integer;
begin
n:=strtoint(edit1.Text);
a:=strtofloat(edit2.Text);
p:=1;
for i:=0 to n do
p:=p*(a+i);
edit3.Text:=floattostr(p);
end; end.
```

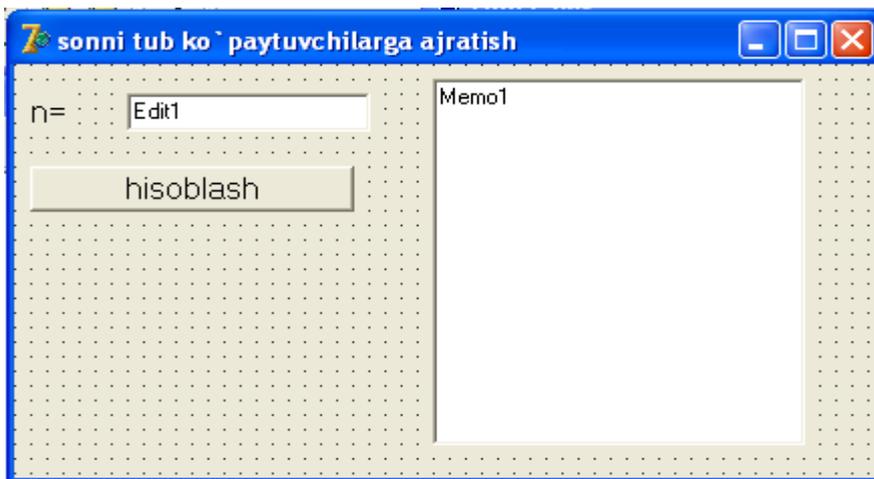
3.15-масала. Даражага кўтариш амалини бажармай, ушбу йиғиндини ҳисобланг:

$$s = \sum_{n=1}^{10} (-1)^n n^2$$

Дастур матни

```
procedure TForm1.Button1Click(Sender: TObject);
var s:real; n:integer;
begin
s:=0;
for n:=1 to 10 do
if odd(n)=true then s:=s-sqr(n) else s:=s+sqr(n);
edit1.Text:=floattostr(s);
end; end.
```

3.16-масала. Натурал сонни туб кўпайтувчиларга ажратинг.



Дастур матни

```
procedure TForm1.Button1Click(Sender: TObject);
var n,i:integer; s:string;
begin
n:=strtoint(edit1.Text);
memo1.Clear;
for i:=1 to n do
if n mod i=0 then begin
str(i,s);
memo1.Lines.Add(s);
end;end;end.
```

Фойдаланилган адабиётлар

1. Абрамов В.Г., Трифонов Н.П., Трифонова Г.Н. Введение в язык Паскаль.-М.:Наука, 1988.-320с.
2. Абрамов С.А.,Гнезделова Капустина Е.Н.и др. Задачи по программированию. - М.: Наука, 1988.
3. Вирт Н. Алгоритмы + структуры данных программа.-М.:Мир,1985.-405с.
4. Культин Н.Б. Программирование в Turbo Pascal 7.0 и Delphi. СПб.: БХВ-Петербург, 2001.-416с.
5. Кэнту М. Delphi 5 для профессионалов.- СПб: Питер, 2001. -944 с.
6. Немнюгин С.А. Turbo pascal, учебник. Изд. Питер., 2001, -496 с.
7. Ставровский А.Б. Турбо Паскаль. 7.0 и Delphi. 2-е изд. 2001, -416с.
8. Файсман А. Профессиональное программирование на Турбо-Паскаль. Ташкент 1992.
9. Шумаков П.В.Delphi3и разработка приложений баз данных.-М.:«НОЛИДЖ»,1998.-704
10. Пилшиков В.Н. Упражнения по языку Паскаль-М.: МГУ, 1986.
11. Фаронов В.В. Програмирование в персональном ЭВМ в среде Турбо-Паскаль.-М.:МГТУ,1990. -443с.
12. Информатика фанидан амалий ва тажриба машғулотлари. Ш. Н. Носирова, Ш. Р. Ўринов. Навоий-2000
13. Ҳисоблаш математикаси ва программалаш. Абдуқодиров А. А., Тошкент-Ўқитувчи 1992.
14. Численные методы для ПЭВМ на языках Бейсик, Фортран, Паскаль. Томск-МП. «Раceso» 1992
15. Ҳисоблаш усулларидаи машқлар ва лаборатория ишлари. А. Абдухамидов, С. Худойназаров. Тошкент. -Ўқитувчи. , 1994.
16. IBM PC шахсий компьютерларида ишлаш. С. И. Раҳмонқулова Тошкент-Шарқ. , 1996.
17. Паскаль алгоритмик тили. Ш. Н. Носирова, З. У. Турсинбаева. Навоий-1999.
18. Алгоритмический язык Паскаль. Ш. Н. Носирова, З. У. Турсинбаева. Г. А. Қулахмедова. Навоий-1999.

