Кодлар муҳаррири Delphi нинг калит сўзларини (procedure, var, begin, end, if ва бошқ.) қорайтириб ёзади. Бу дастур матни ифодали қилиб, дастур структурасини тушунишни осонлаштиради.

Изоҳлар курсив (қийшайтрилган) шрифтда ифодаланади.

Дастурни тайёрлаш жараёнида кодлар муҳарриридан форма ойнасига ва аксинча ўтишга тўғри келади. Бу ишни **View** қуроллар панелидаги **Toggle Form ⁄ Unit** буйруқли тугмаси ёки **<F12>** клавишаси ёрдамида амалга ошириш мумкин.

**25-расм.**  **View** қуроллар панели

**Эслатмалар системаси.** Дастур матнини киритиш жараёнида кодлар муҳаррири процедура ва функцияларнинг параметрлари, объектларнинг ҳусусиятлари ва методлари ҳақидаги эслатмаларни экранга чиқариб боради.

Масалан, кодлар муҳаррири ойнасида ***MessageDlg*** (экранга ахборотларни чиқарувчи функция) ҳамда очилган қавс белгиси ёзилса, экранда ***MessageDlg*** функциясининг параметрлари ҳақидаги эслатмалар ойнаси пайдо бўлади. (26-расм). Параметрлардан бири қорайтириб кўрсатилади. Бунда муҳаррир фойдаланувчига қайси параметрни киритиш кераклиги эслатиб қўяди. Бу параметр ёзилиб, вергул қўйилганидан кейин навбатдаги параметр ҳақидаги эслатма пайдо бўлади. Бу жараён барча параметрлар кўрсатилмагунча давом этади.

Объектлар учун кодлар муҳаррири ҳусусиятлар ва методлар рўйхатини экранга чиқаради. Дастурчи объект (компонента) нинг

**26-расм. К**одлар муҳаррири автоматик тарзда объект (компонента) нинг ҳусусиятлари ва методлари рўйхатини чиқаради.

номини ёзиб, нуқта белгисини қўйиши билан экранда шу объектнинг ҳусусиятлари ва методлари рўйхатидан иборат эслатмалар ойнаси пайдо бўлади. (27-расм).

**27-расм. Edit** компонентасининг **э**слатмалар ойнаси

Рўйхатдаги зарур бўлган элементга сичқонча курсори ёки клавиатурадан керак бўлган ҳусусият ёки методнинг дастлабки бир нечта харфларини ёзиш орқали ўтиш мумкин. Рўйхатдан танланган элемент клавиатурадан <ENTER> ёки сичқончанинг чап тугмаси босилса, дастур матнига қўшиб қўйилади.

Эслатмалар системаси дастур матнини тайёрлаш жараёнини анча енгиллаштиради. Бундан ташқари, агар эслатма пайдо бўлмаса, дастурчи ҳатоликка йўл қўйганлигини билдиради. Масалан, процедура ёки функциянинг номи нотўғри ёзилган бўлиши мумкин.

**Коднинг навигатори.** Кодлар муҳаррири ойнаси икки қисмдан иборат. Ўнг томонга дастур матни ёзилади. Чап томон эса код навигатори (Code Explorer) деб аталади. (31-расм) У дастур матни бўйлаб навигацияни (йўл топиш) осонлаштиради. Структураси тайёрланаётган лойихага боғлиқ бўлган иерархик рўйхатда лойиха формалари, уларнинг компоненталари, ходисаларни қайта ишлаш процедуралари, глобал ўзгарувчилар ва константалар ўз аксини топган. Рўйхатдан керакли элементни топиб, дастур матнидаги код фрагментига осонгина ўтиб олиш мумкин.

**28-расм.** Code Explorer ойнаси дастур матни бўйича навигацияни енгиллаштиради.

Код навигатори ойнасини одатдаги усуллар билан ёпиш мумкин. Экранга код навигатори ойнасини чақириш учун **View** менюсидан  **Code Explorer** тугмасини танлаш етарли.

**Коднинг шаблонлари.** Дастур матнини киритиш жараёнида ***код шаблонлари***(Code Templates) дан фойдаланиш мақсадга мувофиқ ҳисобланади. Коднинг шаблони – бу дастурнинг Delphi да ёзилган буйруқларининг умумий кўринишидир. Масалан, ***case***буйруғинингшаблони қўйидагича:

***case of :;***

***:;***

***else ;***

***end;***

Кодлар муҳаррири дастурчига шаблонларнинг катта тўпламини таклиф қилади: массив, класс, функция ва процедураларни эълон қилиш, танлаш ва тармоқланиш буйруқлари ва х.к. Айрим буйруқлар учун шаблонларнинг бир нечта вариантлари мавжуд.

Дастур матнини ёзиш жараёнида код шаблонларидан фойдаланиб, уларни дастур матнига қўшиш учун **Ctrl** + **J** тугмалар комбинациясини босиш керак. Экранда пайдо бўлган рўйхатдан керакли шаблонни танлаш мумкин. (29-расм) Шаблонни одатдаги усуллар билан танланади: рўйхатни айлантирилади ёки унинг дастлабки бир нечта харфларини ёзиб кўрсатилади. Рўйхатдан

**29-расм.** Код шаблонлари рўйхати Ctrl + J тугмалари ёрдамида чақирилади

керакли элемент топилганидан сўнг, <Enter> тугмаси босилади ҳамда белгиланган шаблон дастур матнига қўшиб қўйилади.

Дастурчи агар зарурат бўлса, ўзининг код шаблонларини яратиш ва улардан худди стандарт шаблонлар каби фойдаланиши мумкин. Бунинг учун у **Tools** менюсидан **Editor Options** буйруғини танлаши, сўнгра **Source Options** пунктидаги **Edit Code Templates** тугмасини босиб, экранда пайдо бўлган **Code Templates** диалог ойнасидаги A**dd** тугмасини чертиб, очилган ойнадан **Add Code Template** пунктини танлаши лозим. (30-расм) Шундан кейин пайдо бўлган ойнада шаблон номи **(Shortcut Name)** ҳамда унинг қисқа **(Description)** характеристикасини кўрсатиши лозим. Сўнгра **ОК**

**30-расм.** Ойнада шаблон номи ва унинг характеристикаси кўрсатилади.

тугмасини чертиб, **Code Templates** диалог ойнасининг **Code** ойнасигашаблонни киритади (31-расм).

**31-расм.** Дастурчи яратган шаблонга намуна

**1.5. Лойиха структураси**

Delphi да яратилган лойиха дастурий бирлик - модуллар тўпламидан иборат. Модулларнинг бири асосий бўлиб, дастур шу модулдан бошлаб бажарилади.

Асосий модул кенгайтмаси .dpr бўлган файл ҳисобланади. Лойиханинг асосий модул матнини кўриш учун **Project менюсидан View Source** буйруғирини танлаш лозим.

3-листингда югуриш тезлиги дастурининг асосий модули келтирилмоқда.

**3-листинг. Югуриш тезлиги** дастурининг асосий модули матни.

**program yugu1;**

**uses**

**Forms,**

**yug in 'yug.pas' {Form1};**

**{$R \*.res}**

**begin**

**Application.Initialize;**

**Application.CreateForm(TForm1, Form1);**

**Application.Run;**

**End.**

Асосий модул ***program*** сўзи билан бошланади. Ундан кейин номи лойиха номи билан бир ҳил бўлган дастур номи келади. Лойиха номи лойихани сақлаш вақтида берилади ва бу ном компилятор томонидан яратиладиган бажарилувчи дастур номини аниқлайди. Сўнгра, ***uses*** сўзидан кейин фойдаланилган модуллар рўйхати берилади.

{$R \*.RES} сатри — бу компиляторга ресурслар файлининг ишга туширишга кўрсатмадир. Ресурслар файли илованинг ресурсларини ўз ичига олади: пиктограммалар, курсорлар, битли тасвирлар ва х.к. Юлдузча нишони ресурслар файлининг номи ҳам лойиха файли билан бир ҳил, аммо .Res кенгайтмали эканлигини англатади.

Ресурслар файли "матнли" файл эмас , шунинг учун уни матнлар муҳаррири ёрдамида кўриб бўлмайди. Ресурслар файли билан ишлаш учун махсус **Resource Workshop** каби дастурлардан фойдаланилади. Шунингдек, Delphi таркибига кирган **Image Editor** утилитини ҳам қўллаш мумкин. Уни **Tools** менюсида жойлашган.

Асосий модулнинг бажариладиган қисми ***begin*** ва ***end*** сўзлари орасида берилади. Бу қисм иловани инициализация қилади ва экранга бошланғич ойнани чиқаради.

Асосий модулдан ташқари, ҳар бир дастур ўз ичига ҳеч бўлмаганда битта форма модулини олади. Унда илова бошланғич формаси ҳамда унга керак бўлган процедуралар рўйхати сақланади. Delphi да ҳар бир формага ўзининг модули мос келади. .

4-листингда югуриш тезлигини хисоблаш дастури модулининг матни келтирилган.

**4-листинг. Югуриш тезлиги** дастурининг модули.

**unit yug;**

**interface**

**uses indows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms, Dialogs, StdCtrls;**

**type**

**TForm1 = class(TForm)**

**Edit1: TEdit;**

**Edit2: TEdit;**

**Label1: TLabel;**

**Label2: TLabel;**

**Label3: TLabel;**

**Label4: TLabel;**

**Button1: TButton;**

**Button2: TButton;**

**procedure Button1Click(Sender: TObject);**

**private**

**{ Private declarations }**

**public**

**{ Public declarations }**

**end;**

**var**

**Form1: TForm1;**

**implementation**

**{$R \*.dfm}**

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

**var**

masofa : integer; ⁄⁄ *масофа, метрда*

t: real; ⁄⁄ *вақт ҳақиқий сон кўринишида*

min : integer; ⁄⁄ *вақт, минутлар*

sek : integer; ⁄⁄ *вақт, секундлар*

*v:* real; ⁄⁄ *тезлик*

**begin**

⁄⁄ *киритиш майдонидан бошланғич маълумотларни олиш*

masofa := StrToint(Edit1.Text); t := StrToFloat(Edit2.Text);

⁄⁄ *дастлабки алмаштиришлар*

min := Trunc(t); ⁄⁄ *минутлар — t сонининг бутун қисми*

sek := Trunc(t\*100) mod 100;

⁄⁄ *секундлар — t сонининг каср қисми*

⁄⁄ *хисоблаш*

v := (masofa ⁄ 1000) ⁄ ((min\*60 + sek) ⁄ 3600);

⁄⁄ *натижани чиқариш*

label4.Caption := 'Масофа: '+ Edit1.Text + ' м' + #13 + 'Вақт: '

+ IntToStr(min) + ' мин ' + IntToStr(sek) + ' сек ' + #13 +

'Тезлик: ' + FloatToStrF(v,ffFixed,4,2) + ' км⁄соат';

**end;**

**end.**

⁄⁄ **Якунлаш** тугмаси босилганда

**procedure** TForm1.Button2Click(Sender: TObject)

**begin**

Form1.Close;

**end;**

**end.**

Модуль ***unit*** сўзи билан бошланади. Ундан кейин модул номи кўрсатилади. Бу ном матни 3-листингда берилган илованинг асосий модулида эслатиб ўтилади.

Модуль интерфейс, реализации, инициализация каби бўлимлардан иборат бўлади.

Интерфейс бўлими (***Interface*** сўзи билан бошланади) компиляторга модулнинг қайси қисми дастурнинг бошқа модуллар учун зарур бўлиши ҳақида ахборот беради. Бу бўлимда (***uses*** сўзидан кейин) шу модулда фойдаланиладиган модуллар кутубхоналари рўйхати санаб ўтилади. Шунингдек, бу ерда Delphi да яратилган форма ***type*** сўзидан кейин кўрсатилади.

Реализация бўлими ***Implementation*** билан бошланади ва яратилган форма учун зарур бўлган локал ўзгарувчилар, процедуралар ва функциялар эълон қилинади. Бўлим {$R \*.DFM} директиваси билан бошланади. У компиляторга бажариладиган файлни яратишда формадаги маълумотлардан фойдаланиш ҳақида кўрсатма беради. Форма номи модул номи билан бир ҳил, аммо кенгайтмаси ***.dfm*** бўлган файлда сақланади. Бу файл Delphi муҳитида форманинг ташқи кўриниши асосида генерация қилинади.

($R \*.DFM} директивасидан кейин форма ва унинг компоненталари ходисаларни қайта ишлаш процедуралари келтирилади. Бу ерга дастурчи бошқа процедура ва функцияларни ҳам киритиши мумкин. Инициализация бўлими модулдаги ўзгарувчиларни инициализация қилади. Инициализация бўлими реализация (барча процедура ва функцияларни ифодалаш) бўлимидан кейин ***begin*** ва ***end*** ларорасида жойлашади. Агар инициализация бўлими ўз ичига хеч қандай кўрсатмани олмаса, (худди келтирилган мисолдаги каби), у ҳолда ***begin*** сўзи кўрсатилмайди. Шуни таъкидлаш керакки, модулнинг каттагина хажмдаги буйруқларини Delphi нинг ўзи яратади. Масалан, Delphi, дастурчининг форма яратиш бўйича хатти-харакатларини таҳлил қилиб, формадаги объектлар ҳақидаги маълумотларни (***type*** сўзидан кейин) генерация қилади.

**Лойихани сақлаш.** Лойиха – бу компилятор бажариладиган файлни (ЕХЕ-файли) яратиши учун зарур бўлган файллар тўпламидан иборат. Энг оддий мисолда лойиха лойиха ҳақидаги маълумотлардан иборат файл (DOF-файли), асосий модул файли (DPR-файли), ресурслар файли (RES-файли), форма ҳақидаги маълумотлардан иборат файл (DFM-файли), илованинг асосий кодлари жойлашган форма модулининг файли, шунингдек форманинг компоненталари учун ходисаларни қайта ишлаш процедура ва функциялари (PAS-файллар) ҳамда конфигурация файли (CFG-файли) дан иборат бўлади.

Лойихани сақлаш учун **File** менюсидан **Save Project As** буйруғини танлаш лозим. Агар лойиха бирор марта ҳам сақланмаган бўлса, у ҳолда Delphi дастлаб модулни (кодлар муҳаррири ойнасидаги маълумотларни ) сақлаб қўйишни таклиф этади. Шунинг учун экранда **Save Unit1 As** ойнаси пайдо бўлади. Бу ойнада (32-расм) лойиха файллари учун ажратилган папка ва модул номини кўрсатиш лозим. **Сохранить** тугмаси босилганидан кейин экранда навбатдаги ойна (33-расм) пайдо бўлади. Унда лойиха файли номи кўрсатилади.

**32-расм.** Форма модулини сақлаш

Модул файли (pas-файл) ва лойиха файлининг (dpr-файл) номлари ҳар ҳил бўлишига эътибор беринг. Бажариладиган файл (ЕХЕ-файл) номи лойиха файлининг номи билан бир ҳил. Шунинг учун лойиха файлига ном танлаганда бажариладиган файлнинг номини ҳам ҳисобга олиш зарур. Модулга эса номни бошқача, масалан, лойиха файли номига тартиб номерларини қўшиш орқали танлаш мумкин.

**Эслатма:** Лойиха – бу файллар тўпламидан иборат бўлгани учун, ҳар бир лойихани алоҳида папкада сақлаш тавсия қилинади.

**33-расм.** Лойихани сақлаш

менюга

**1.6. Компиляция**

Компиляция — бу бошланғич дастурни бажариладиган файлга айлантириш жараёнидир. Компиляция жараёни икки босқичдан иборат. 1-босқичда дастур матннинг ҳатосиз ёзилганлиги текширилади, иккинчисида эса бажариладиган файл (ехе-файл) генерация қилинади.

Ходисаларни қайта ишлаш функцияси яратилиб, сақланганидан сўнг, **Project** менюсидан **Compile** буйруғини танлаб компиляцияни бажариш мумкин.. Компиляция жараёни ва натижаси **Compiling** диалог ойнасида (34-расм) кўрсатилади. Бу ойнага компилятор аниқлаган ҳатоликлар (Errors), огоҳлантириш (warnings) ҳамда эслатмалар (Hints) чиқарилади. Ҳатолик, эслатма ва огоҳлантиришлар кодлар муҳаррири ойнасининг қуйи қисмида берилади. (35-расм).

**Эслатма:** Агар компиляция вақтидаэкранда **Compiling** ойнаси кўринмаса, у ҳолда **Tools** менюсидан **Environment options** буйруғини танланг. **Preferences** пунктидаги **Show compiler progr*ess*** ўчиргичини ёқилган ҳолатга ўтказинг.

**35-расм. К**омпиляцияни бошлаш

**35-расм. К**омпилятор аниқлаган ҳатоликлар ҳақидаги ахборот

**Ҳатоликлар.** Компилятор бажариладиган файлни фақат дастур матнида бирорта ҳам синтактик ҳатолик бўлмагандагина яратади. Кўпинча, хозиргина ёзилган дастур матнида ҳатоликлар мавжуд бўлади. Дастурчи уларни бартараф қилиши лозим.

Хатолик мавжуд бўлган дастур парчасига ўтиш учун курсорни ҳатолик ҳақидаги ахборот устига келтириб, контекст менюсидан (36-расм) **Edit source** буйруғини танлаш керак.

Ҳатоликлар бирин-кетин йўқотиб борилади. Ҳар бир ҳатолик йўқотилгандан кейин, такрорий компиляция ўтказилади. Компилятор одатда ҳатолик мавжуд бўлган парчани аниқ кўрсатмаслиги мумкин. Бу ҳолда компилятор кўрсатган парчани таҳлил қилиш билан чегараланиб қолмай, парчадан олдинги сатрга ҳам эътибор қаратиш лозим.

**36-расм. Ҳатолик мавжуд бўлган парчага ўтиш**

10-жадвалда энг кўп учраши мумкин бўлган ҳатоликлар ва компиляторнинг уларга мос равишда берадиган ахборотлари санаб ўтилади.

Компиляторнинг хатоликлар ҳақидаги ахбороти 10-жадвал

|  |  |
| --- | --- |
| Ахборот | Мумкин бўлган сабаб |
| Missing operator or semicolon (оператор ёки нуқтали вергул етишмаяпти) | Буйруқдан кейин нуқтали вергул қўйилмаган |
| Undeclared identifier: '...' | '...' ўзгарувчи эълон қилинмаган |

Агар компилятор етарлича кўп ҳатоликларни аниқлаган бўлса, дастлаб энг оддий ҳатоликларни бартараф этинг ва такрорий компиляция ўтказинг. Ҳатоликлар сони анчагина камайиши керак. Чунки, кичик бир ҳатолик ортидан унга боғлиқ бўлган кўплаб ҳатоликлар келиб чиқиши мумкин.

Агар дастур матнида синтактик ҳатоликлар мавжуд бўлмаса, у ҳолда компилятор дастурнинг бажариладиган файлини яратади. Унинг номи лойиха файли номи билан бир ҳил, кенгайтмаси эса — .exe бўлади. Delphi бажариладиган файлни лойиха файли сақланган папкада сақлайди.

**Огоҳлантириш ва эслатмалар.** Дастур матнида ҳато бўлмаган ноаниқликлар мавжуд бўлса, компилятор экранга эслатма (Hints) ва огоҳлантиришлар (warnings) чиқарилади. Масалан, дастур матнида эълон қилинган, аммо фойдаланилмаган ўзгарувчилар ҳақидаги эслатма энг кўп учрайди:

Variable ... is declared but never used in ...

Ҳақиқатдан ҳам, фойдаланилмаган ўзгарувчини эълон қилишнинг нима кераги бор?11-жадвалда энг кўп учрайдиган огоҳлантиришлар келтирилган.

Компиляторнинг огоҳлантиришлари 11-жадвал

|  |  |
| --- | --- |
| Огоҳлантириш | **Мумкин бўлган сабаби** |
| Variable... is declared but never used in ... | Ўзгарувчидан фойдаланилмаган |
| Variable . . . might not have been initialized. | Ўзгарувчига бошланғич қиймат берувчи буйруқ етишмайди. (инициализация қилинмаган ўзгарувчидан фойдаланилмоқда |

**Дастурни ишга тушириш.** Delphi муҳитидан туриб ҳам дастурни ишга тушириш мумкин. Бунинг учун **Run** менюсидан **Run** буйруғини танлаш ёки **Debug** қуроллар панелидаги махсус тугмани лозим (37-расм).

**37-расм.** Дастурни ишга тушириш

менюга

**1.7. Бажариш вақтидаги ҳатоликлар**

Дастурнинг ишга туширилганда **бажариш вақтидаги ҳатолиги** (run-time errors) ёки **чиқариш ҳатолиги** (exceptions) деб аталадиган ҳатоликлар юзага келиши мумкин. Бунга кўпинча нотўғри бошланғич маълумотлар сабаб бўлади. Масалан. Югуриш тезлигини ҳисоблаш дастури учун **Вақт** майдонига 3.20 матни, (яъни бутун ва каср қисмини ажратишда вергул ўрнига нуқта қўйилган бўлса) киритилган бўлса, у ҳолда **Хисоблаш** тугмаси босилганда экранда ҳатолик ҳақида ахборот пайдо бўлади (дастур Windows муҳитидан туриб ишга туширилган): (38-расм).

**38-расм.** Бажариш вақтидаги ҳатолиги

Ҳатоликнинг юзага келишининг асосий сабаби дастур матнида соннинг бутун ва каср қисми нуқта билан, киритиш ойнасида эса одатда вергул (Windows нинг айрим версиялари нуқта билан ажратишга рухсат беради) билан ажратилишидадир.

Агар Windows ни бутун ва каср қисмни вергул билан ажратишга созланган бўлса-ю, фойдаланувчи диалог ойнасида масалан, 3.20 сатрини киритган бўлса, у ҳолда

***t: = StrToFloat(Edit2.Text)***

буйруғини бажаришда мослик йўқолади, чунки ***strToFloat*** функциясининг қиймати ҳақиқий соннинг ифодаси бўлмай қолади .

Агар дастур Delphi муҳитидан ишга туширилиб, мослик йўқолган бўлса, дастурнинг иши тўхтайди ва экранда хатолик ва унинг характери ҳақидаги ахборот пайдо бўлади. Масалан, 39-расмдаги ахборотда фойдаланувчи киритган соннинг ҳақиқий сон эмаслиги ҳақида маълумот берилмоқда.

**39-расм.** Ҳатолик ҳақидаги ахборотга мисол

Дастурчи ОК тугмасини босиб, ўз ишини давом эттириши (бунинг учун **Run** менюсидан **Step Over** буйруғини танлайди**) ёки** дастурнинг бажарилишини тўхтатиши (**Run** менюсидан **Program Reset** буйруғи танланади) мумкин.

Дастурни ишлаб чиқишда дастурчи фойдаланувчиларнинг ҳатоликка олиб борувчи барча хатти-харакатларини ҳисобга олиши ва дастурни улардан химоя қилишни таъминлаши зарур.

5-листингдаги югуриш тезлигини хисоблаш дастурида фойдаланувчининг тўғри бўлмаган айрим хатти-харакатларидан дастурни химоя қилиш амалга оширилган. Ҳусусан, масофа (Edit1) майдонига фақат рақамлар киритилиши таъминланган.

**Ўзгаришлар киритиш. Югуриш тезлиги** дастурини бир неча марта ишга туширилганидан сўнг, унинг матнига ўзгартириш киритишга ҳохиш пайдо бўлиши мумкин. Масалан, дастурни шундай ўзгартириш керакки, масофани киритиб, <Enter> тугмаси босилганда, курсор **Вақт** майдонига ўтсин. Ёки **Масофа** ва **Вақт** майдонларига фойдаланувчи фақат рақамларни кирита олсин.

Дастур матнига ўзгартириш киритиш учун, дастлаб Delphi ни ишга тушириб, ўзгартириладиган лойиха очилади. Буни **File** менюсидан **Open Project** буйруғини танлаш орқали амалга ошириш мумкин. **Reopen** буйруғи танланса, дастурчи ишлаган охирги лойихалар рўйхати очилади.

5-листингдаги Югуриш тезлиги дастури матнига Edit1 ва Edit2 компоненталари учун  **OnKeyPress** ходисаларни қайта ишлаш процедураси қўшилган.

Дастур матнига ходисаларни қайта ишлаш процедурасини қўшиш учун **Object Inspector** ойнасидан ходисаларни қайта ишлаш процедураси яратиладиган компонента танланади. Сўнгра **Events** бўлимидан ходисани танлаб, процедура номи майдонида сичқонча икки марта чертилади. Delphi ходисаларни қайта ишлаш процедураси шаблонини яратади. Шундан кейин процедура буйруқларини киритиш мумкин.

**5-листинг**. **Югуриш тезлиги** дастурининг модули ўзгаришлар киритилганидан кейин қуйидагича бўлади.

**interface**

**uses**

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms, Dialogs, StdCtrls;

**type**

TForml = class(TForm) Editl: TEdit;

Edit2: TEdit; Label1: TLabel;

Label2: TLabel;

Label3: TLabel;

Label4: TLabel;

Buttonl: TButton;

Button2: TButton;

**procedure** Button1Click(Sender: TObject);

**procedure** Button2Click(Sender: TObject);

**procedure** EditlKeyPress(Sender: TObject; var Key: Char);

private

*{ Private declarations }* public

*{ Public declarations }* end;

**var**

Form1: TForm1;

**implementation**

{$R \*.dfm)

⁄⁄ *Хисоблаш тугмаси босилганидан кейин*

**procedure** TForm1.ButtonlClick(Sender: TObject);

**var**

masofa : integer ⁄⁄ *масофа, метрларда*

t: real; ⁄⁄ *вақт ҳақиқий сон кўринишида*

min : integer; ⁄⁄ вақт*, минутлар*

sek : integer; ⁄⁄ вақт, *секундлар*

v: real; ⁄⁄ тезлик

**begin**

⁄⁄ *киритиш майдонидан бошланғич маълумотларни олиш*

masofa := StrToint(Edit1.Text); t := StrToFloat(Edit2.Text);

⁄⁄ *дастлабки алмаштиришлар*

min := Trunc(t); ⁄⁄ *минутлар — t сонининг бутун қисми*

sek := Trunc(t\*100) mod 100;

⁄⁄ *секундлар — t сонининг каср қисми*

⁄⁄ *хисоблаш*

v := (masofa ⁄ 1000) ⁄ ((min\*60 + sek) ⁄ 3600);

⁄⁄ *натижани чиқариш*

label4.Caption := 'Масофа: '+ Edit1.Text + ' м' + #13 + 'Вақт: '

+ IntToStr(min) + ' мин ' + IntToStr(sek) + ' сек ' + #13 +

'Тезлик: ' + FloatToStrF(v,ffFixed,4,2) + ' км⁄соат';

**end;**

⁄⁄ Я*кунлаш тугмаси босилганида*

**procedure** TForml.Button2Click(Sender: TObject);

**begin**

Form1.Close;

end;

⁄⁄ *Масофа майдонида тугма босилганда*

**procedure** TForm1.Edit1KeyPress(Sender: TObject; var Key: Char);

**begin**

⁄⁄ *Key —босилган тугмага мос келувчи белги*

*⁄⁄ Агар мумкин бўлмаган белги киритилган бўлса, процедура уни*

*⁄⁄ коди 0 бўлган белги билан алмаштиради. Натижада*

*⁄⁄ фойдаланувчида дастур айрим тугмаларни босилишига эътибор*

*⁄⁄ бермас экан деган тасаввур пайдо бўлади*

**case** Key **of**

'0'..'9': ; ⁄⁄ рақамлар

#8 : ; ⁄⁄ ўчириш клавишаси *<Back Space>*

#13 : Edit2.SetFocus ; ⁄⁄ *<Enter> клавишаси*

*⁄⁄ қолган белгиларни киритиш таъқиқланади*

**else** Key :=Chr(0); ⁄⁄ *белгини кўрсатмаслик*

**end;**

**end;**

**end.**