**Делпхи дастурлаш мухити**

**1-БОБ. БОШЛАНҒИЧ МАЪЛУМОТЛАР**

[**Delphi ни ўрнатиш**](#m1)

[**Ишни бошлаш**](#m2)

[**Биринчи лойиха**](#m3)

[**Форма**](#m4)

[**Компоненталар**](#m5)

[**Ходиса ва ходисаларни қайта ишлаш процедураси**](#m6)

[**Кодлар муҳаррири**](#m7)

[**Лойиха структураси**](#m8)

[**Компиляция**](#m9)

[**Бажариш вақтидаги ҳатоликлар**](#m10)

[**Иловани якуний созлаш**](#m11)

[**Иловаларни бошқа компьютерга ўтказиш**](#m12)

Бу ерда Delphi дастурини ўрнатиш қисқа очиб берилади. Спорчининг масофани босиб ўтиш тезлигини хисоблаш мисолида визуал лойихалаш ва ходисаларни дастурлаш технологияси намойиш қилинади. Асосий тушунча ва терминлар аниқланади.

**1.1. Delphi ни ўрнатиш**

Borland Delphi 7 Studio пакетининг тўртта версияси мавжуд: Personal, Professional, Enterprise ва Architect. Бу пакетларнинг ҳар бири турли мақсадларга қаратилган юқори самарали дастурлар ишлаб чиқиш учун стандарт воситалар тўпламига эга. Пакетларнинг имкониятлари Personal дан Architect га қараб кенгайиб боради. Масалан, Enterprise комплекти масофадаги маълумотлар базаси (мисол учун, InterBase) билан ишлай олади, Personal да эса бундай имкон йўқ.

Ушбу китобдаги материаллар Delphi нинг бирор аниқ бир пакетига боғланмаган. Намуна тариқасида олинган барча масалалар Personal пакети доирасида амалга оширилиши мумкин.

Delphi 7 ни компьютерга ўрнатиш барча зарур файллар ва ўрнатиш программаси (Delphi Setup Launcher) каби материаллар сақланаётган ўрнатувчи диск ёрдамида амалга оширилади. Бу дискни CD-диск юритувчи қурилмага қўйилган заҳоти, ўрнатувчи дастур автоматик тарзида ишга тушади.

**1-расм.**  Delphi 7 ўрнатишни бошлаш

Натижада экранда ўрнатувчи CD-ROM ёрдамида компьютерга ўрнатилиши мумкин бўлган дастурий маҳсулотларнинг рўйхатини ўз ичига олган **Delphi 7 Setup Launcher ойнаси** (1-расм) пайдо бўлади.

Бу рўйхатда аввало Delphi 7, қолаверса InterBase 6.5 маълумотлар базаси сервери, InstallShield Express - ўрнатувчи CD-ROM ларни яратиш утилитлари кабиларни кўриш мукмин.

Delphi ни ўрнатишни бошлаш учун **Delphi** 7 сатри чертилади. Delphi ни ўрнатиш жараёни оддий. Серия номери (Serial Number) ва калит (Authorization Key) киритилганидан сўнг, экранда дастлаб лицензион келишув ойнаси, кейин **Setup Type** (2-расм) ойнаси пайдо бўлади. Бу ойнада ўрнатишнинг мумкин бўлган вариантлари тавсия қилинади: **Typical** (Оддий), **Compact** (қисқартилиган) ёки **Custom** (фойдаланувчи танлаб ўрнатадиган).

2-расм. **Setup Type** диалог ойнасидан ўрнатиш вариантини танлаш

*Оддий вариант* ўрнатишда CD-ROM дан қаттиқ дискка Delphi нинг барча компоненталарини тўлиқ кўчирилади. Бунинг учун дискда камида 475 Мбайт (Enterprise пакети учун) атрофида бўш жой бўлиши талаб қилинади. Агар компьютерда етарлича бўш жой бўлса, шу вариантни ўрнатиш тавсия қилинади.

*Қисқартириб ўрнатишда* эса Delphi нинг энг муҳим компоненталари кўчириб олинади. Бу вариант қаттиқ дискдан энг кам бўш жой талаб қилади. Қисқартириб ўрнатилганда, Delphi нинг айрим имкониятларидан фойдаланиб бўлмайди. Чунки, бу ҳолда қаттиқ дискка ёрдамчи маълумотномалар системасининг файллари, айрим компонента ва утилитлар, намуналар кўчирилмайди.

*Фойдаланувчи танлаб ўрнатадиган вариант* дастурчига Delphi нинг энг керакли восита ва компоненталарини танлаб ўрнатишга имкон беради. Одатда, бу вариантни тажрибали дастурчилар қўллашади. Delphi ни тўлиқ ўрнатиш учун қаттиқ дискда етарлича бўш бўлмаган ҳолда бу вариантдан фойдаланиш мумкин.

Ўрнатиш вариантини танлагандан сўнг, **Next** тугмасини босинг. Агар **Custom** варианти танланган бўлса, **Custom Setup**  ойнаси очилади. Ундан ўрнатилмайдиган компоненталарни аниқланади. Компонентани ўрнатишни таъқиқлаш учун компонента номидан чапдаги диск тасвирини чертиш ва очилган меню пунктларидан **Do Not Install** ни танлаш лозим.

**З-расм. К**омпоненталар ўрнатиладиган каталоглар рўйхати

Агар ўрнатишнинг **Typical** варианти танланган бўлса, у ҳолда Next тугмасини чертиш натижасида **Destination Folder** ойнаси очилади. Унда Delphi пакети ва унинг компоненталари ўрнатиладиган каталоглар рўйхати таклиф қилинади.

Навбатдаги **Next** тугмасини чертиш **Save Installation Database** ойнасини очади.Унда фойдаланувчига Delphi ни қаттиқ дискка ўрнатиш жараёни ҳақидаги ахборотни сақлаб қўйиш тавсия қилинади. Бу ахборот Delphi ни ўрнатувчи дисксиз ўчиришда (деинсталляция қилишда) керак бўлиши мумкин. Шу билан ўрнатишга тайёргарлик босқичи тугайди. Экранда **Ready To Install the Program** ойнаси пайдо бўлади. **Install** тугмасини чертиш ўрнатиш жараёнини фаоллаштиради ва ўрнатиш бошланади. Ўрнатиш жараёни тугаганидан сўнг, бу ҳақида махсус ахборот экранга чиқарилади (4-расм). **Finish** тугмасини чертиш бу ойнани ёпади.

**4-расм.** Ўрнатиш жараёни тугади

Энди Delphi ни ишга тушириш мумкин. Аммо, бундан олдин ишчи каталог ва лойихалар каталогини кўрсатиб қўйиш лозим. Бунинг учун сичқонча кўрсаткичи Delphi ни ишга тушириш буйруғига келтирилади: **Пуск ⁄ Программў ⁄ Borland Delphi 7 ⁄ Delphi**7. Сўнгра сичқончанинг ўнг томонини босиб, пайдо бўлган контекст менюдан **Свойства** тугмасини танланади. Очилган **Свойства: Delphi**7 ойнасидаги **Рабочая папка** ойнасида Delphi лойихалари учун мўлжалланган папка номи кўрсатилади. (5-расм).

менюга

**1.2. Ишни бошлаш**

Delphi бошқа иловалар каби одди усулда ишга туширилади, яъни **Borland Delphi**7 менюсидан **Delphi**7 буйруғи танланади.

Delphi ишга тушганидан сўнг, экран 7-расмдаги кўринишни

**5-расм.** Лойихалар папкасини кўрсатиш

 **6-расм.** Delphi ни ишга тушириш.

олади. Бу ойнада бир вақтнинг ўзида 5 та диалог ойна пайдо бўлади:

* **Delphi7** нинг бош ойнаси**;**
* **Form1 -** бошланғич форма ойна ;
* **Object Inspector -** объектлар ҳусусиятини таҳрирлаш ойнаси;
* **Object TreeView –** объектлар рўйхатини кўриш ойнаси;
* **Unitl.pas –** кодларни таҳрирлаш ойнаси.

Кодларни таҳрирлаш ойнаси деярли тўлалигича бошланғич форма ойнаси билан тўсиб қўйилган.

**7-расм.**  Delphi ишга тушганидан кейин экраннинг кўриниши

Бош ойнада (8-расм) буйруқлар менюси, қуроллар панели, компоненталар палитраси жойлаштирилган.

Бошланғич форма ойнаси **(Form1)** яратиладиган лойиха учун олдиндан тайёрланган ишланмадан иборат.

Дастурий таъминотни системали ва амалий гуруҳларга бўлинади. Системали дастурий таъминот – бу операцион системадан иборат. Қолган барча дастурларни амалий дастурий таъминот деб аталади. Амалий дастурларни қисқа қилиб ***иловалар*** деб аташ қабул қилинган.

**8-расм**. Бош ойна

**Object Inspector** (9-расм) — объектларнинг ҳусусиятларини таҳрирлаш ойнаси объектларнинг ҳусусиятларини ўзгартириш учун мўлжалланган. ***Объект***деганда диалог ойналари, бошқариш элементлари (киритиш ва чиқариш майдонлари, буйруқли тугмалар, ўчиргичлар ва х.к.) назарда тутилади. **О*бъектнинг ҳусусияти***эсаобъектнинг кўриниши, ҳолати, ҳулқи каби характеристикаларини билдиради. Масалан, ***Width*** ва ***Height*** ҳусусиятлари форманинг ўлчамларини (баландлиги ва кенглиги) белгиласа, ***Тор*** ва ***Left*** форманинг экрандаги ҳолатини аниқлайди, ***Caption*** ҳусусияти сарлавҳа матнини кўрсатади.

9-расм. **Properties** пунктида объект ҳусусиятлари кўрсатилган.

Кодларни таҳрирлаш ойнасида (10-расм), (уни форма ойнасини бир четга суриб, очиш мумкин) дастур матни ёзилади. Янги лойиха устида иш бошланганда, кодларни таҳрирлаш ойнасида Delphi да яратилган тайёр дастур шаблони жойлашган бўлади.

менюга

**Биринчи лойиха**

 Delphi нинг имкониятлари ва визуал дастурлаш технологиясини намойиш қилиш учун спортчи белгиланган масофани қандай тезлик билан босиб ўтишини ҳисобловчи лойиха яратамиз. Дастурнинг диалог ойнаси ва ишлаши 11-расмда келтирилган.

**10-расм.** Кодларни таҳрирлаш ойнаси

Янги дастур устида иш бошлаш учун Delphi ни ишга туширинг. Агар сиз бу муҳитда бошқа лойиха билан ишлаётган бўлсангиз, **File** (Файл) менюсидан **New** ⁄ **Application** (Создать ⁄ Приложение) буйруғини танланг.

**11-расм.** Югуриш тезлигини ҳисоблаш ойнаси

менюга

**Форма**

Яратилаётган илова, яъни янги лойиха устида ишлаш диалог ойнасини, яъни бошланғич формани қуришдан бошланади.

Бошланғич форма **Form1** формасинингҳусусиятларини ўзгартириш ҳамда унга эхтиёжга қараб керакли компоненталарни (киритиш ва чиқариш майдонлар, буйруқли тугмалар) ни ўрнатиш орқали яратилади.

Форманинг ҳусусиятлари (1-жадвал) унинг ташқи кўриниши, ўлчамлари, сарлавҳа матни ҳошиясининг кўринишини белгилайди. Форма ва унинг компоненталари ҳусусиятлари ва қийматларини ўзгартириш учун **Object Inspector** ойнасиданфойдаланилади. Бу ойнанинг юқори қисмида объектнинг номи ҳамда ҳусусиятларининг жорий вақтдаги қийматлари кўрсатилади. **Properties** (ҳусусияти) қистирмасининг чап колонкасида объектнинг ҳусусиятлари, ўнг томонда эса уларнинг қийматлари келтирилади.

Форманинг (***mform*** объектининг) ҳусусиятлари 1-жадвал

|  |  |
| --- | --- |
| Ҳусусият | Мазмуни |
| Name |  Форманинг номи. Дастурда форманинг номи  формани бошқариш ва форма компоненталарига  мурожаат қилиш учун фойдаланилади.  |
|  Caption | Сарлавҳа матни |
| Width | Форманинг кенглиги |
| Height | Форманинг баландлиги |
| Top | Форманинг юқори чегарасидан экраннинг юқори чегарасигача бўлган масофа |
|  Left | Форманинг чап чегарасидан экраннинг чап чегарасигача бўлган масофа |
| BorderStyle |  Чегаранинг кўриниши. Чегара оддий (bsSizeable), ингичка (bs Single) бўлиши ёки умуман бўлмаслиги (bsNone) мумкин. Агар ойнанинг чегараси оддий бўлса, уни фойдаланувчи сичқончадан фойдаланиб, ўзгартириш мумкин. Ингичка чегарали ойна ўлчамларини ўзгартириб бўлмайди. Агар чегара бўлмаса, у ҳолда экранга сарлавҳасиз ойна чиқарилиши мумкин. Бундай ойнанинг ҳолати ва ўлчамларини дастурнинг иши мобайнида ўзгартириш мумкин эмас.  |
| BorderIcons | Ойнани бошқариш тугмалари. Ҳусусиятиниг қиймати дастурнинг иши давомида фойдаланувчилар қайси тугмалардан фойдаланиш мумкинлигини кўрсатади. Ҳусусиятнинг қиймати biSystemMenu, biMinimaze, biMaximaze ва biHelp ҳусусиятларининг қийматларини аниқлаш орқали берилади. biSystemMenu ҳусусияти ихчамлаш ва система тугмаларига, biMinimaze— ихчамлаш тугмасига, biMaximaze — кенгайтириш тугмасига, biHelp — маълумотномаларни чиқариш тугмаси билан ишлашга рухсат беради.  |
| Icon | Диалог ойнаси сарлавҳасидаги система менюсини чақиришни англатувчи нишон. |
| Color | Фон ранги. Рангни унинг номини кўрсатиб ёки операцион системанинг ранглари гаммасига боғлаб қўйиш орқали белгилаш мумкин. Иккинчи ҳолда ранглар жорий ранглар схемаси бўйича аниқланади ва операцион системанинг ранглар схемаси ўзгарганда ўзгаради.  |
| Font |  Шрифт. Форма сиртида "кўрсатилмаганда ҳам" фойдаланиладиган жорий шрифт. Форманинг ҳусусияти ўзгартирилганда форма сиртида жойлашган компоненталарнинг Font ҳусусияти автоматик тарзда ўзгаради, яъни компоненталар формадан Font ҳусусиятини мерос қилиб олади.  |

Форма яратишда биринчи навбатда caption (сарлавҳа) ҳусусиятининг қийматини ўзгартириш лозим. Бизнинг мисолимизда "Form1" матнинг "югуриш тезлиги" билан алмаштириш керак. Бунинг учун **Object Inspector** ойнасида сичқонча тугмасини **Caption** сатрида чертамиз. Натижада ҳусусиятнинг жорий қиймати ажратиб кўрсатилади ва шу сатрда курсор пайдо бўлади. Шундан кейин "Югуриш тезлиги" матнини киритиш мумкин. (12-расм).

Худди шу усул билан форманинг кенглиги ва баландлигини аниқловчи **Height** ва **width** ҳусусиятларни ўзгартириш мумкин. Форманинг ўлчамлари, унинг ҳолати ҳамда бошқа бошқарув элементларининг ўлчамлари ва уларнинг форма сиртидаги ҳолатлари пикселларда (экрандаги нуқталар) берилади. **Height** ва **width** ҳусусиятларини мос равишда 250 ва 330 қилиб белгиланг.

**12-расм.** Ҳусусият қийматини қийматни киритиш орқали ўзгартириш

Форма — бу оддий ойнадир. Шунинг учун унинг ўлчамларини бошқа ойналар каби сичқонча ёрдамида ўзгартириш мумкин. Бунда **Height** ва **Width** ҳусусиятларининг қийматлари автоматик тарзда ўзгаради.

Диалог ойнасининг экрандаги ҳолати формани ташкил қилишдаги ҳолатига мос келади. Бу ҳолатни **Тор** (экраннинг юқори чегарасидан чекиниш) ва **Left** (экраннинг чап чегарасидан чекиниш) ҳусусиятларининг қийматлари белгилаб беради. Бу қийматларни ҳам сичқонча ёрдамида ўзгартириш мумкин.

Айрим ҳусусиятларни танлашда, масалан, **Borderstyle**, ҳусусиятнинг жорий қийматини белгилашда, ўнг томонда очиладиган рўйхат таклиф қилинади. Қийматни ана шу рўйхатдан танлаш мумкин. (13-расм)

Айрим ҳусусиятлар мураккаб ҳисобланади, чунки уларнинг қийматлари бошқа ҳусусиятларнинг қийматларидан келиб чиқиб аниқланади. Бундай ҳусусиятларнинг олдида "+" нишони туради. У чертилса, аниқловчи ҳусусиятлар рўйхати таклиф қилинади. (14-расм). Масалан, BorderIcons ҳусусияти ойналарни бошқаришнинг қайси тугмалари билан дастур иши давомида ишлаш мумкинлигини белгилайди. Агар biMaximize ҳусусиятига False қиймати берилса, дастурнинг ишлаши жараёнида **Развернуть** тугмаси ойна сарлавҳасида кўринмайди.

|  |  |
| --- | --- |
|  |  |
| 13-расм. Ҳусусият қийматларини рўйхатдан танлаш | 14-расм. Мураккаб ҳусусиятларнинг очилган рўйхати |

Айрим ҳусусиятларнинг ёнида уч нуқтали тугма жойлашган. Бу ҳусусият қийматини аниқлашда янги диалог ойнасидан фойдаланиш мумкинлигини англатади. Масалан, **Font** мураккаб ҳусусиятининг қийматини белгилашда шрифт танлашнинг стандарт ойнасидан фойдаланиш мумкин.

2-жадвалда яратилаётган форманинг ўзгартириладиган ҳусусиятлари келтирилган. Ўзгармайдиган ҳусусиятлар бу жадвалга киритилмаган.

 Бошланғич форманинг ҳусусиятлари 2-жадвал.

|  |  |
| --- | --- |
|  **Ҳусусият**  |  **қиймат** |
|  Caption |  Югуриш тезлиги |
|  Height |  250 |
|  Width |  330 |
|  BorderStyle |  bsSingle |
|  BorderIcons . biMinimize |  False |
|  BorderIcons . biMaximize |  False |
|  Font. Size |  10 |

Келтирилган жадвалда айрим ҳусусиятларнинг ёнида нуқта (.) белгиси турибди. Бу ҳусусиятнинг аниқланадиган қийматини белгилашни билдиради. Бошланғич форманинг жадвалдаги ҳусусият лари ўрнатилганидан сўнг, форма 15-расмдаги кўринишни олади.

**15-расм.** Бошланғич форманинг кўриниши.

менюга

**Компоненталар**

Югуриш тезлигининг дастури фойдаланувчидан бошланғич маълумотлар – масофа ҳамда шу масофани югуриб босиб ўтиш вақтини олиши лозим. Бундай ҳолларда одатда бошланғич маълумотлар киритиш майдонларига клавиатура ёрдамида киритилади. Шунинг учун формага киритиш майдони **Edit** компонеталарини жойлаштириш лозим.

Энг кўп фойдаланиладиган компоненталар **Standard** қуроллар панелида жойлаштирилган(16-расм).

Формага компонентани қўшиш учун компоненталар палитрасидан шу компонента пиктограммаси устига сичқончани келтириб, чертилади. Сўнгра курсорни компонентанинг чап юқори бурчаги формада жойлашиши керак бўлган нуқтага ўрнатилади ва чап тугмани яна бир марта чертилади. Натижада формада стандарт ўлчамли компонента пайдо бўлади.

**16-расм.**  **Standard** қуроллар панелининг компоненталари рўйхати

Компонентанинг ўлчамларини уларни формага қўшиш жараёнида белгиланиши мумкин. Бунинг учун компонента сичқонча ёрдамида танланганидан сўнг, курсорни форманинг компонента чап юқори бурчаги туриши керак бўлган нуқтасига келтириб, чап тугмаси босилади ва уни қўйиб юбормаган ҳолда курсорни компонентанинг қуйи ўнг бурчаги туриши керак бўлган нуқтага олиб келинади. Шундан сўнг сичқончанинг чап тугмасини қўйиб юбориш мумкин. Формада кўрсатилган ўлчамдаги компонента пайдо бўлади.

Delphi формага қўшилаётган хар бир компонентага ном беради. Бу ном компонента номи ва унинг тартиб номеридан иборат бўлади. Масалан, формага иккита Edit компонентаси қўшилган бўлса, уларнинг номлари мос равишда Edit1 ва Edit2 бўлади. Дастурчи Name ҳусусияти қийматини ўзгартириб, бу номларни бошқасига алмаштириши мумкин. Одатда содда дастурларда компоненталарнинг номлари алмаштирилмайди.

17-расмда форманинг иккита Edit компоненталари, яъни киритиш майдонларини қўшилганидан кейинги ҳолати келтирилган. Компоненталарнинг бири ажратилган. Ажратилган компонентанинг ҳусусиятлари **Object Inspector** ойнасида тасвирланган. Бошқа компонента ҳусусиятларини кўриш учун сичқончанинг чап тугмасини шу компонента устида чертиш лозим. Шунингдек, компонента номини **Object TreeView** ойнасидан ёки **Object Inspector** ойнасининг юқори қисмидаги объектларнинг очиладиган рўйхатидан ҳам танлаш мумкин.

**17-расм.** Иккита **Edit** компоненталари қўшилган форма

3-жадвалда киритиш майдони - Edit компонентасининг ҳусусиятлари келтирилган.

Киритиш майдони Edit компонентаси ҳусусиятлари 3-жадвал

|  |  |
| --- | --- |
| **Ҳусусияти** | **Мазмуни** |
|  Name |  Компонентанинг номи. Дастурда ундан компонента ва унинг ҳусусиятлари билан ишлаш учун фойдаланиш мумкин. Масалан, таҳрирлаш майдонига киритилган матн билан ишлаш |
|  Text |  Киритиш ва таҳрирлаш майдонидаги матн  |
|  Left |  Компонента чап чегарасидан форманинг чап  чегарасигача бўлган масофа |
|  Top |  Компонента юқори чегарасидан форманинг юқори  чегарасигача бўлган масофа |
|  Height |  Майдоннинг баландлиги |
|  Width |  Майдоннинг кенглиги |
|  Font |  Киритилаётган матн учун шрифт |
|  ParentFont | Компонента томонидан форма шрифти аломатларини мерос қилиб олиш белгиси. Агар ҳусусиятнинг қиймати **True** бўлса, у ҳолда форманинг **Font** ҳусусияти ўзгарганда компонентанинг **Font** ҳусусияти ҳам автоматик тарзда ўзгаради. |

Delphi компонента ўлчамларини сичқонча ёрдамида ўзгартиришга ҳам имкон беради.

Компонента ҳолатини ўзгартириш учун сичқонча курсорни унинг тасвири устига келтириб, чап тугмаси босилади. Уни қўйиб юбормаган ҳолда компонентани форманинг керакли жойига олиб борилади. Компонентанинг сурилиб бориши билан (18-расм) компонентанинг чап юқори бурчагининг координаталари кўрсатиб борилади (Left ва Тор нинг қийматлари).

Компонента ўлчамини ўзгартириш учун стрелкани компонента чегарасини кўрсатувчи маркерлардан бирига ўрнатиб, чап тугмасини босиб турган ҳолда компонента чегарасини эхтиёжга қараб сурилади. Сўнгра чап тугмани қўйиб юбориш мумкин. Ўлчамни ўзгартириш вақтида компонентанинг Height ва Width ҳусусиятларининг жорий қиймати кўрсатиб борилади.

**18-расм.** Left ва Тор ҳусусиятларининг жорий қийматлари

4-жадвалда Edit1 ва Edit2 киритиш майдонларининг ҳусусиятлари келтирилган. Edit1 - компонентаси масофа, Edit2 –эса шу масофани босиб ўтиш вақтини киритиш учун мўлжалланган. Ҳар икки компонентанинг ҳусусиятининг қиймати бўш сатр эканлигига эътибор беринг.

Киритиш майдонларидан ташқари, дастур ойнасида дастур ва киритиш майдонининг нимага мўлжалланганлиги ҳақида қисқа ахборот бўлиши лозим. Бундай ахборотларни формага қўйиш учун матнларни чиқариш майдонидан фойдаланилади. **Матнларни** **чиқариш майдони** – бу **Label** компонентасидир. **Label** компонентаси

Edit компоненталарининг ҳусусиятлари 4-жадвал

|  |  |
| --- | --- |
| **Ҳусусияти**  | **Компонента** |
| **Edit1** | **Edit2** |
| **Text** |  |  |
| **Тор** | 56 | 88 |
| **Left** | 128 | 128 |
| **Height** | 21 | 21 |
| **Width** | 121 | 121 |

**Standard** қуроллар панелида жойлашган.(19-расм). Label компонентасини формага киритиш майдони каби қўйиш мумкин.

**19-расм.** Label компонентаси — матнларни чиқариш майдони

Тайёрланаётган илова формасига тўртта Label компонента-сини қўшиш лозим. 1-майдон ахборотнома учун, 2- ва 3 - майдонлар киритиш майдонларининг мақсадини кўрсатиш учун, 4-майдон эса ҳисоб натижасини (тезликни) чиқариш учун мўлжалланган.

Label компонентасининг ҳусусиятлари 5-жадвал

|  |  |
| --- | --- |
| **Ҳусусияти**  | **Мазмуни** |
|  Name | Компонентанинг номи бўлиб, компонента ва унинг ҳусусиятларига мурожаат қилиш учун хизмат қилади.  |
|  Caption | Чиқариладиган матн |
|  Font | Матннинг шрифи |
|  ParentFont | Форма белгиларини компонента томонидан мерос қилиб олиниши. Агар ҳусусиятнинг қиймати True бўлса, матн форма учун белгиланган шрифтда чиқарилади.  |
|  AutoSize | Майдоннинг ўлчами унга ёзилган белгилар сонига қараб белгиланади.  |
|  Left | Чиқариш майдонининг форманинг чап чегарасидан чекиниш масофаси  |
|  Top | Чиқариш майдонининг форманинг юқори чегарасидан чекиниш масофаси |
|  Height | Чиқариш майдонининг баландлиги |
|  Width | Чиқариш майдонининг кенглиги |
|  Wordwrap | Сўзлар жорий сатрга сиғмаганда, уларни автоматик тарзда навбатдаги сатрга ўтказиш аломати.  |

**Autosize** ва **Wordwrap** ҳусусиятларига алоҳида эътибор беринг. Бу ҳусусиятлардан чиқариш майдонидаги маълумотлар бир нечта сатрга мўлжалланган ҳолларда фойдаланиш мумкин. Формага **Label** компонентасини қўшилганда **Autosize** ҳусусиятининг қиймати True, яъни, ўлчам caption ҳусусиятининг қийматининг ўзгаришига қараб аниқланади. Агар чиқариш майдонидаги матн бир нечта сартни эгаллаши талаб қилинса, формага **Label** компонентаси қўшилганидан кейин, **Autosize** - га **False**, **wordwrap** - га **True** қийматларини бериш лозим. Сўнгра **Width** ва **Height** ҳусусиятлари қийматларини аниқлаб қўйиш керак. Шундан кейингина caption ҳусусиятига матнни киритиш мумкин.

Формага тўртта киритиш майдони (Label компоненталари) жойлаштирилиб, уларнинг ҳусусият қийматлари кўрсатилганидан сўнг форма 20-расмдаги кўринишни олади.

**20-расм.** Чиқариш майдонлари қўшилганидан сўнг форманинг кўриниши

**Сaption** сатрига матн битта сатр каби киритилади. Матннинг чиқариш майдонидаги ҳолати **Autosize** ва **Wordwrap** ҳусусиятлари ҳамда матннинг шрифтига боғлиқ бўлади.

**Label1, Label2, Label3 и Label4 компоненталарининг ҳусусиятлари 6-жадвал**

|  |  |  |
| --- | --- | --- |
| **Компонента** | **Ҳусусияти**  | **Қиймати** |
| Labell |  AutoSize |  False |
|  Wordwrap |  True |
|  Caption | Ушбу дастур спортчининг югуриш тезлигини хисоблайди |
| Top | 8 |
| Left | 8 |
| Height | 33 |
| Width | 209 |
| Label2 | Top | 56 |
| Left | 8 |
| Caption | Масофа (метр) |
| Label3 | Top | 88 |
| Left | 8 |
| Caption | Вақт (мин,сек) |
| Label4 | AutoSize | False |
| Wordwrap | True |
| Top | 120 |
| Label 4  | Left | 8 |
| Height | 41 |
| Width | 273 |

Форманинг сўнгги босқичида иккита буйруқли тугма ўрнатилади: **Хисоблаш** ва **Якунлаш.** Уларнинг маъноси равшан.

Буйруқли тугма – **Button** компонентаси формага бошқа компоненталар каби қўшилади. Унинг Button нишони **Standard** қуроллар панелида жойлашган. (21-расм).

**21-расм.** Буйруқли тугма— Button компонентаси

Button компонентасининг ҳусусиятлари 7-жадвал

|  |  |
| --- | --- |
| **Ҳусусияти**  | **Мазмуни**  |
| Name | Компонента номи. Дастурда компонента ва унинг ҳусусиятларига мурожаат қилишда ишлатилади.  |
| Caption | Тугма устидаги матн |
| Enabled | Тугма билан ишлашга рухсат. Унинг қиймати True бўлса тугма билан ишлаш мумкин, акс ҳолда – йўқ.  |
| Left | Чиқариш майдонининг форманинг чап чегарасидан чекиниш масофаси  |
| Top | Чиқариш майдонининг формани юқори чегарасидан чекиниш масофаси |
| Height | Тугманинг баландлиги |
| Width | Тугманинг кенглиги |

Формага иккита буйруқли тугма ўрнатилганидан сўнг, уларнинг қийматларини 8-жадвалга мос равишда белгилаш лозим.

**Button1 ва Button2 компоненталарининг ҳусусиятлари. 8-жадвал**

|  |  |
| --- | --- |
| **Ҳусусияти**  | **Компонента** |
| **Button1** | **Button2** |
| **Caption** | **Ҳисоблаш** | **Якунлаш** |
| **Тор** | 176 | 176 |
| **Left** | 16 | 112 |
| **Height** | 25 | 25 |
| Width | 75 | 75 |

Форманинг якуний кўриниши 22- расмда келтирилган.

22-расм. Югуриш тезлиги дастурининг формаси

Форма яратилганидан сўнг, дастур матнини ёзиш мумкин. Бундан олдин Delphi да дастурлашнинг икки муҳим тушунчасини аниқлаб оламиз: **ходиса** ва **ходисаларни қайта ишлаш процедураси.**

менюга

**1.3.** **Ходиса ва ходисаларни қайта ишлаш процедураси**

Форманинг кўриниши дастурнинг қандай ишлашидан дарак бериб турибди. Фойдаланувчи киритиш майдонига бошланғич маълумотларни киритиши ҳамда **Хисоблаш** тугмасини чертиши лозим. Буйруқ тугмасини чертиш Delphi да ***ходиса*** деб аталади.

 Ходиса (Event) — бу дастурнинг иши жараёнида содир бўладиган воқеадир. Сичқонча тугмасини бир марта чертиш - ***OnClick***, икки марта чертиш ***OnDblClick*** ходисаси ҳисобланади. 9-жадвалда айрим ходисалар келтирилган.

 **Ходиса 9-жадвал**

|  |  |
| --- | --- |
| **Ходиса** | **Содир бўлади** |
| OnClick | Сичқонча тугмаси чертилганда  |
| OnDblClick | Счт икки марта чертилганда |
| OnMouseDown | Сичқонча тугмаси босиб турилганда  |
| OnMouseUp | Сичқонча тугмаси қўйиб юборилганда |
| OnMouseMove | Сичқонча сурилганда |
| OnKeyPress | Клавиатура тугмаси босиб турилганда |
| OnKeyDown | Клавиатура тугмаси босилганда OnKeyDown ва OnKeyPress ходисалари- то клавиатуранинг ушлаб турилган тугмаси қўйиб юборилмагунча навбати билан такрорланиб турувчи ходисалардир. (Бу вақтда OnKeyUp ходисаси рўй беради) |
| OnKeyUp  | Клавиатура тугмаси қўйиб юборилганда |
| OnCreate | Объект (форма, бошқариш элементлари) яратилганда. Бу ходисаларни қайта ишлаш процедураси ўзгарувчиларни эълон қилиш ва тайёргарлик кўришда фойдаланилади.  |
| OnPaint | Дастур иш бошлаган вақтда экранда ойна пайдо бўлганда. Бошқа ҳолларда эса ойнанинг бир қисми бошқа ойна билан тўсиб турилганда. |
| OnEnter | Бошқарув элементи томонидан фокус олинганида  |
| OnExit | Бошқарув элементи фокусни йўқотганда  |

Ходиса рўй берганда дастур қандайдир реакция билдириши керак. Delphi ходисага ходисаларни қайта ишлаш процедураси орқали жавоб беради. Шундай қилиб, фойдаланувчининг бирор хатти-харакатига (ходисага) дастур бирор вазифани бажариб жавоб бериши учун дастурчи шу ходисага мос қайта ишлаш процедурасини ёзиши лозим. Шуни эътиборга олиш керакки, ходисаларни қайта ишлаш процедурасининг каттагина қисми компонента зиммасига юкланган. Шунинг учун дастурчи ходисага жавоб реакцияси стандарт бўлмаган ёки аниқланмаган ҳоллардагина ходисаларни қайта ишлаш процедурасини ёзиши керак. Масалан, масала шарти бўйича **Edit** майдонига киритиладиган белгилар учун чекловлар бўлмаса, **OnKeyPress** ходисаларни қайта ишлаш процедурасини ёзишнинг кераги йўқ, чунки дастурнинг иши давомида бу вазифа стандарт равишда (дастурчига кўрсатилмаган ҳолда) бажарилади.

**OnClick** ходисасининг қайта ишлаш процедурасини **Хисоблаш** буйруқли тугмаси мисолида яратамиз.

Ходисаларни қайта ишлаш процедураси яратишни бошлаш учун дастлаб **Object Inspector** ойнасида ходисаларни қайта ишлаш процедураси ёзиладиган компонентани танлаймиз. Шу ойнанинг ўзида **Events** (Ходиса) пунктини танлаймиз.

**Events** нинг чап устунида (23-расм) белгиланган объект учун содир бўлиши мумкин бўлган ходисалар рўйхати келтирилган. Агар ходиса учун ходисаларни қайта ишлаш процедураси аниқланган (ёзилган) бўлса, у ҳолда ўнг томондаги устунда шу процедуранинг номи пайдо бўлади.

Ходисаларни қайта ишлаш функциясини ташкил қилиш учун мос ходисаларни қайта ишлаш процедурасининг керакли номи майдонида сичқонча тугмаси икки марта чертилади. Натижада кодларнинг таҳрирлаш ойнаси очилади. Бу ойнага ходисаларни қайта ишлаш процедурасининг шаблони автоматик тарзда қўшилади, **Object Inspector** ойнасида эса ходисанинг номи билан бир қаторда уни қайта ишлаш функцияси номи пайдо бўлади. (24-расм).

Delphi ходисаларни қайта ишлаш функциясига икки қисмдан иборат ном беради. Номнинг биринчи қисми ходисаларни қайта ишлаш функцияси ёзилаётган компонентани ўз ичига олган формани таниш учун, иккинчи қисми эса объект ва ходисани белгилаш учун хизмат қилади. Бизнинг мисолимизда форманинг номи — Form1, буйруқли тугма номи — Button1, ходисанинг номи эса - Click.

**23-расм. Events** пунктида компонента (бизнинг мисолимизда - буйруқли тугма) қабул қила оладиган ходисалар рўйхати келтирилган.

**24-расм**. Delphi ходисаларни қайта ишлаш процедурасининг шаблони

Кодларни таҳрирлаш ойнасида ***begin*** ва ***end*** сўзлари орасида ходисаларни қайта ишлаш процедураси буйруқларини ёзиш мумкин.

1-листингда **Хисоблаш** буйруқли тугмаси учун **onclick** ходисаларни қайта ишлаш функциясининг матни келтирилган. Унинг умумий кодларни таҳрирлаш ойнасидаги кўринишига мос келади: калит сўзлар қорайтириб, изоҳлар курсив шрифтда, буйруқлар эса сатр бошидан чекинтириб (шундай одат мавжуд) ёзилган.

**1-листинг. Хисоблаш** тугмаси учун ***Onclick*** ходисасини қайта ишлаш процедурасининг матни.

⁄⁄ *хисоблаш тугмаси* **босилганда**

**procedure** TForm1.ButtonlClick(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;**

**Button1click** функцияси тезликни ҳисоблаб, натижани **Label4** майдонига чиқаради. Бошланғич маълумотлар эса **Editl** ва **Edit2** киритиш-таҳрирлаш майдонларининг **Text** ҳусусиятига мурожаат қилиб олинади. **Text** ҳусусияти дастурнинг ишлаши жараёнида фойдаланувчи киритган белгилар кетма-кетлигидан иборат бўлади. Дастурнинг иши тўғри бўлиши учун бу кетма-кетлик фақат рақамлар ва вергулдан иборат бўлиши лозим. Сатрни сонга айлантириш учун **StrToInt** ва **StrToFloat** функцияларидан фойдаланилган. ***StrToInt*** функцияси берилган сатрдаги (***Edit1.Text*** - майдонидаги) белгиларни текширади. Агар ҳамма белгилар рақамлардан иборат бўлса, улар ифодалайдиган бутун сонни қиймат қилиб олади ва бу қийматни **masofa** ўзгарувчисига беради. ***StrToFioat*** функцияси ҳам худди шу каби ишлайди. У ***Edit2.Text*** майдонидаги рақамлар кетма-кетлигига мос келадиган ҳақиқий сонни ***t*** ўзгарувчига қиймат қилиб беради.

Бошланғич маълумотлар **masofa** ва **v** ўзгарувчиларга берилганидан сўнг, хисоблаш жараёни бошланади. Дастлаб ***Trunc*** функцияси ёрдамида соннинг каср қисми ташлаб юборилади ва *t* — нинг бутун қисми (минутлар) аниқланади. ***Trunc(t\*100) mod 100*** ифодасининг қиймати секундлар миқдорини топади. Бу қуйидагича ҳисобланади: дастлаб ***t*** сони 100 га кўпайтирилади, ***Trunc*** функцияси ёрдамида кўпайтманинг каср қисми ташлаб юборилади, ҳосил бўлган сонни 100 га бўлиб, ***mod*** ёрдамида қолдиғи топилади.

Ҳамма маълумотлар тайёр бўлганидан сўнг, хисоблаш бошланади. Тезлик км⁄соат ларда ифодалангани учун, метрда берилган масофа ҳамда минут ва секундларда берилган вақт километр ва соатларга айлантирилади.

Тезликнинг хисобланган қиймати **Label4** майдонига, унинг **Caption** ҳусусиятини ўзгартириб чиқарилади. Сонларни сатрга айлантириш учун ***IntToStr*** ва ***FloatToStr*** функцияларидан фойдаланиш мумкин.

**Якунлаш** тугмаси босилганида дастур ўз ишини тугатиши лозим. Бунинг учун экрандан дастурнинг бош ойнасини ёпиб, олиб ташлаш керак. Бу вазифани ***close*** методи ёрдамида бажариш мумкин. Якунлаш тугмаси учун ***Onclick*** ходисаларни қайта ишлаш процедурасининг матни 2-листингда келтирилган.

**2-листинг. Якунлаш** тугмаси учун ***Onclick*** процедурасининг матни.

⁄⁄ *Якунлаш тугмаси босилганда*

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

**begin**

Form1.Close; ⁄⁄ *дастурнинг бош ойнасини ёпиш*

**end;**

менюга

**1.4. Кодлар муҳаррири**

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

менюга

**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.**

**Дастурни ишга тушириш**

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.**

**Дастурни ишга тушириш**

Ўзгаришлар киритилганидан сўнг, лойихани сақлаш лозим. Бунинг учун **File** менюсидан **Save all** буйруғи танланади.

менюга

**1.8. Иловани якуний созлаш**

Дастур ҳамма талабларга жавоб берадиган бўлганидан сўнг, уни якуний созлаш, яъни дастурга ном ва нишон тайинлаш лозим. Бу ном ва нишон папкадаги файллар рўйхати орасида, ишчи столда, дастур ишлаётган бўлса масалалар панелида кўриниб туради.

Иловани созлаш **Project** менюсидан **Options** буйруғи танланганда очиладиган **Project Options** диалог ойнасининг **Application** пункти ёрдамида бажарилади. (40-расм),

**Title** майдонига илова номи ёзилади. Бу майдонга киритилган матн Windows масалалр панелида, ишлаётган дастур нишони билан ёнма-ён чиқарилади.

Иловага стандарт бўлмаган нишон танлаш учун **Load Icon** тугмаси босилади. Сўнгра папкалар кўришнинг стандарт ойнаси ёрдамида дастурга мос келадиган нишон қидирилади. (Нишонлар .**ico** кенгайтмали файлларда сақланади).

**Илова учун нишон яратиш.** Delphi таркибига **Image Editor** (тасвир муҳаррири) кирган. У дастурчига иловалар учун ўзининг нишонини яратишга имкон беради. **Image Editor** дастури **Tools** менюсидан ёки Windows муҳитида — **Пуск ⁄ Программы Borland Delphi 7 ⁄ Image Editor** буйруқлари билан ишга туширилади.

Янги нишон яратиш учун **File** менюсидан **New** буйруғи очадиган рўйхатдан **Icon File** тугмаси танланади. Яратилаётган файлнинг типи кўрсатилганидан сўнг, **Icon Properties** ойнаси очилади. Унда янги нишоннинг аломатлари белгиланади: size (ўлчами) — 32x32 (Windows нишонларининг стандарт ўлчами) ва Colors (ранглар) — 16 ҳил ранг. **ОК** тугмаси босилгандан кейин **Icon1.ico** ойнаси очилади. Унда стандарт қуроллар ва ранглар ёрдамида керакли нишонни чизиш мумкин.

Image Editor да расм чизиш Microsoft Paint дан фарқ қилмайди. Аммо бу ерда бир "нозик жой" бор. Дастлаб тасвир майдони шаффоф (transparent) ранг билан бўялган бўлади. Агар нишонни шу фонда чизилса, уни кейинчалик экранга чиқарилганда, шаффоф рангга бўялган қисми нишон жойлашадиган фон рангини қабул қилади.

**40-расм. Application** пункти ёрдамида дастур учун ном ва нишон танланади

Расмни чизиш жараёнида ҳато чизилган элементларни шаффоф рангга бўяш орқали "ўчириш" мумкин. Унга ранглар палитрасининг қуйи қаторидаги чап квадрат тўғри келади. Шаффоф рангдан ташқари, ранглар палитрасида "инверсли" (қарама-қарши) ранг ҳам мавжуд. Бу ранг билан чизилган расмлар экранга чиқарилганда фон рангига нисбатан инверсион рангга бўялади.

Яратилган нишон **File** менюсидаги **Save** буйруғи билан сақлаб қўйилади.

менюга

**1.9. Иловаларни бошқа компьютерга ўтказиш**

Битта ЕХЕ-файлдан иборат бўлиб, фақат стандарт компоненталардан фойдаланадиган унчалик катта бўлмаган иловаларни дискеталар ёрдамида бошқа компьютерларга ўтказиш мумкин. Одатда, бошқа компьютерларда бу иловани муаммоларсиз ишга тушириш мумкин.

 Модуллар кутубхоналари, драйверлар ва бошқа дастурий компоненталарни ўз ичига олган иловаларни бошқа компьютерларга ўтказиш мураккаброқ. Бундай иловалар учун ўрнатувчи диск (CD-ROM) яратиш мақсадга мувофиқ бўлади. Бу ишни Delphi таркибига кирган **InstallShield Express** пакети ёрдамида ҳал қилиш мумкин. Ўрнатувчи дискларни яратиш жараёни ҳақида кейинги бобларда тўхталамиз.

менюга