*Delphi* тилида дастурлашда дастурчининг асосий вазифаси ходисаларни қайта ишлаш процедураларини (қисм дастурларни) ишлаб чиқишдан иборат бўлади.

Ходиса рўй берганда, дастурчининг шу ходиса учун мўлжаллаган процедураси автоматик тарзда ишга тушиши керак бўлади. Ходисалар рўй берганда, уларга мос келган ходисаларнинг қайта ишлаш процедураларини ишга туширишни *Delphi* ўз зиммасига олади.

*Object Pascal* тилида дастурнинг асосий бирлиги қисм дастур ҳисобланади. Қисм дастурлар икки турга бўлинади: процедура ва функция. Процедура ҳам, функция ҳам бирор вазифани бажариш учун кўрсатма - буйруқлар кетма-кетлигидан иборат бўлади. Қисм дастурдаги буйруқларни бажариш учун бу қисм дастурга мурожаат қилиш (чақириш) лозим. Функциянинг процедурадан фарқи шундаки, функция номи билан қиймат боғланган бўлади ва бу функциянинг номидан ифодаларда алоҳида операнда сифатида фойдаланиш мумкин.

**Процедура структураси.** Процедура сарлавҳадан бошланади. Сўнгра константаларни эълон қилиш бўлими, типларни эълон қилиш бўлими, ўзгарувчиларни эълон қилиш бўлими, кўрсатма-буйруқлар бўлими келади. Процедуралар умумий кўринишда қуйидагича ёзилади:

***procedure ном*** (Параметрлар рўйхати);

***const***

⁄⁄ бу ерда номланган константалар рўйхати берилади;

***type***

⁄⁄ бу ерда типлар ***var*** ёрдамида эълон қилинади;

⁄⁄ бу ерда ўзгарувчилар ва уларнинг типлари эълон қилинади;

***begin***

⁄⁄ бу ерда дастурнинг буйруқлари ёзилади;

***end;***

Процедура сарлавҳаси ***procedure*** дан бошланади. Ундан кейин процедуранинг номи кўрсатилади. Бу ном шу процедурага мурожаат қилиш (ишга тушириш ёки чақириш) учун хизмат қилади. Агар процедурада параметрлар қатнашса, улар процедура номидан кейин қавслар ичида кўрсатилади. Сарлавҳа "нуқтали вергул" белгиси билан тугайди.

Агар дастурда ностандарт, яъни янги типларни яратишга эҳтиёж пайдо бўлса, бу типлар ***type*** сўзидан кейин эълон қилинади.

Ўзгарувчиларни эълон қилиш бўлимида шу процедура учун хос бўлган барча ўзгарувчилар ва уларнинг типлари рўйхати келтирилади. Бу рўйхат ***var***  сўзидан кейин бошланади.

Буйруқлар бўлими ***begin*** сўзи билан бошланади ва ***end*** сўзи билан тугайди. Бу ерда процедуранинг буйруқлари кетма-кетлиги жойлашади. ***End*** сўзидан кейин "нуқтали вергул" белгиси қўйилади.

Қуйидаги процедурада умумий харид суммасини топиш масаласи ҳал қилинган. Агар 5000 сўмдан ортиқ суммага харид қилинса, умумий суммадан 10% чегириб ташланади.

***procedure*** Summa;

***var***

baho: real; ⁄⁄ нархи

miqdori: integer; ⁄⁄ ҳарид қилинган бир ҳил буюмлар сони

s: real; ⁄⁄ сумма

mes: string[255]; ⁄⁄ хабарнома

***begin***

baho := StrToFloat(Form1.Edit1.Text);

miqdori := StrToInt(Form1.Edit2.Text);

s := baho \* miqdori;

**if s >** 500 **then begin**

s := s \* 0.9;

mes := '10% ли чегирма айриб ташланди.'+#13;

**end;**

mes := mes+ 'Харид нархи : '+ FloatToStrF(s,ffFixed,4,2) +' сум';

Forml.Label3.Caption := mes;

***end;***

**Функция структураси** функциянинг сарлавҳаси, константалар, типлар, ўзгарувчиларни эълон қилиш бўлимлари ҳамда буйруқлар бўлимидан иборат бўлади. Функция умумий ҳолда қуйидагича кўринишда ташкил қилинади:

**function ном** (Параметрлар рўйхати) : Тип;

**const ⁄⁄** константалар рўйхати бўлими

**type ⁄⁄** типларни эълон қилиш бўлими

var ⁄⁄ ўзгарувчиларни эълон қилиш бўлими

**begin ⁄⁄** буйруқлар бўлими

Result := қиймат; ⁄⁄ функция номини қиймат билан боғлаш

**end;**

Функциянинг сардавҳаси ***function*** сўзи билан бошланади, ундан кейин функциянинг номи келади. Сўнгра қавслар ичида функциянинг параметр-аргументлари ва уларнинг типларининг рўйхати ёзилади. Қавсдан кейин икки нуқта (:) қўйиб, функциянинг қабул қиладиган қийматининг типи кўрсатилади. Сарлавҳа "нуқтали вергул" билан тугайди.

Буйруқлар бўлимида ўзгарувчиларни эълон қилиш бўлимда кўрсатилган ўзгарувчилардан ташқари, r***esult*** ўзгарувчисидан ҳам фойдаланиш мумкин. Функциядаги буйруқлар бажариб бўлинганидан сўнг, бу ўзгарувчининг қиймати функциянинг қийматига айланади. Шунинг учун, функциянинг буйруқлари орасида албатта ***result*** ўзгарувчисига қиймат берувчи буйруқнинг бўлиши шарт. Одатда, бу буйруқ функциянинг энг охирги бажариладиган буйруғи бўлади.

Қуйидаги мисолда градусларни радианга айлантириш функцияси келтирилган:

***function GradToRad****(grad:integer):real;*

***begin***

*result:= (grad\*pi)⁄180;*

***end;***

**2.12****. Дастурда буйруқларни ёзиш.**

Ҳар бир буйруқ бошқасидан нуқтали вергул билан ажратилади. Бошқача айтганда, ҳар бир буйруқдан кейин нуқтали вергул белгиси қўйилади.

Дастурнинг ҳар бир сатрида бир ёки бир нечта буйруқларни кўрсатиш мумкин.

Айрим буйруқларни (***if, case, repeat, while*** ва х.к.) бир нечта сатрга ёзиш қабул қилинган. Уларнинг структурасини бошқаларидан ажратиш мақсадида сатрнинг чап чегарасидан буйруқларни бир оз чекинтириб ёзиш тавсия этилади. Бу дастур матнини ўқиш ва тушунишни осонлаштиради. Масалан:

**if** d >= 0 **then**

**begin**

x1:=(-b+Sqrt(d))⁄(2\*a);

x2:=(-b-Sqrt(d))⁄(2\*a);

ShowMessage('x1=' + FloatToStr(xl) + 'x2=' + FloatToStr(x2)) ;

**end**

**else**

ShowMessage('Тенглама ҳақиқий ечимларга эга эмас.');

***Then*** ва ***else*** бир-бирларининг остига ҳамда ***if*** га нисбатан бир ҳил масофада чекинтириб ёзилганига эътибор беринг. ***End*** сўзи ***begin*** остига ёзилган. ***begin*** ва ***end*** лар орасида буйруқлар ***begin*** га нисбатан бир-бирининг остида, бир масофада чекинтириб жойлаштирилган. Юқоридаги буйруқларни қуйидагича ҳам ёзиш мумкин:

**if** d >= 0 **then begin** x1:=(-b+Sqrt(d))⁄(2\*a); x2:=(-b-Sqrt(d))⁄(2\*a);

ShowMessage('x1=' + FloatToStr(xl) + 'x2=' + FloatToStr(x2)) ; **end**

**else** ShowMessage('Тенглама ҳақиқий ечимларга эга эмас.')

Аммо, биринчи вариант қулайроқ, чунки, унда алгоритм структураси яхшироқ кўринади.

Айрим узун ифодаларни бир нечта сатрга бўлиб ёзиш мумкин. Бундай ифодаларни ихтиёрий белгисидан бошлаб узиш ва қолган қисмини кейинги сатрга ўтказиш мумкин. Ўзгарувчиларнинг номларини, сонли ва матнли константаларни, шунингдек таркибий операторларни узиш, масалан, қиймат бериш операторини мумкин эмас. Қуйида бир нечта сатрга ёзилган буйруққа мисол келтирамиз:

st:= 'Тенгламанинг илдизлари'+ #13

+'x1=' + FloatToStr(x1)+ #13 +'х2=' + FloatToStr(x2);

Шуни ҳисобга олиш керакки, компилятор ортиқча "бўш жой" белгиси ҳамда бўш сатрларга эътибор бермайди. Шунинг учун у сатр бошидаги барча "бўш жой" белгиларини "кўрмайди". Бу эса сатрлардаги буйруқларини чекинтириб ёзишга имкон беради. Арифметик, мантиқий ифодаларни (шартларни), параметрлар рўхатини ёзишда "бўш жой" белгиларини қўйиш талаб қилинмайди, аммо улардан фойдаланиш дастур матнини дастурчи учун ўқишни осонлаштиради. Қуйидаги икки ифодани солиштиринг:

x1:=(-b+Sqrt(d))⁄(2\*a); ҳамда x1 := (-b + Sqrt(d)) ⁄ (2 \* а);

Иккинчи вариантнинг осон ўқилиши кўриниб турибди.

Дастур ишини тушунишни енгиллаштириш учун дастур матнига изоҳларни киритиш мумкин. Умумий ҳолда изоҳларни фигурали қавслар орасида кўрсатилади. Очилаётган қавс изоҳнинг бошланганлигини, ёпилаётгани эса тугаганлигини билдиради. Агар изоҳ бир сатрли ёки бирор буйруқдан кейин келса, ундан аввал ⁄⁄ (қўш чизиқча) белгилари қўйилади. Масалан:

**var**

{ квадрат тенгламанинг коэффициентлари }

a:real; ⁄⁄ биринчи даражали номаълум олдидаги коэффициент

b:real; ⁄⁄ иккинчи даражали номаълум олдидаги коэффициент

с:real; ⁄⁄ нолинчи даражали номаълум олдидаги коэффициент

{ тенгламанинг илдизлари } x1,x2:real;

**2.13.** **Дастурлаш усули**

Дастур устида ишлар экан, дастурчи ўзи ёзаётган дастур аввало дастурчи учун, қолаверса бошқалар учун ҳам мўлжалланганлигини яхши тасаввур қилиш лозим. Дастур матни биринчи ўринда, дастурчи учун, сўнгра у билан бирга лойиха устида ишлаётган ҳамкасблари учун керак бўлади. Шунинг учун дастурчилар ишининг самарасини кучайтириш мақсадида дастур матнини осон ўқилиши, тушунарли бўлиши ҳамда дастурнинг структураси ечилаётган масала структураси ва алгоритмига мос бўлиши зарур деган критериялар қабул қилинган. Бунга қандай эришиш мумкин? Бунинг учун яхши дастурлаш усулига риоя қилиш лозим. дастурлаш усули – бу дастурчи ўз иши жараёнида (онгли равишда ёки "бошқалар шундай қилгани учун" ) риоя қиладиган қонун-қоидалар мажмуасидир. Табиийки, яхши дастурчи яхши дастурлаш усулларига риоя қилиши керак. Дастурлашнинг яхши усули қуйидагиларни назарда тутади:

* изоҳлардан фойдаланиш;
* ўзгарувчи, процедура, функцияларни маъносига қараб номлаш ;
* чекинишлардан фойдаланиш;
* бўш сатрлар ва "бўш жой" белгиларидан фойдаланиш .

Дастурлашнинг яхши усули дастур матнини компьютер хотирасига киритишда ҳатоликларни камайтиришга, дастурни ўқиш ҳатоликларни аниақлаш ва тузатмалар киритишни енгиллаштиришда ёрдам беради.

Дастурлаш усулининг яхшилигини баҳолаш учун аниқ бир критерия ўйлаб топилмаган. Аммо, дастур матнига бир марта қарашда дастурнинг яхши ёки ёмон ёзилганлигини кўриш мумкин. +олаверса, яхши дастур фақат матнининг ёзилиши усули билан белгиланмайди. Яхши дастур ишончли, фойдаланувчига нисбатан дўстона муносабат, бажарилиш тезлиги билан фарқланади.

Дастурнинг ишончлилиги деганда дастур фойдаланувчининг онгли эканлигига қарамасдан, бошланғич маълумотларни назорат қилиши, бажарилган амалларнинг натижаларини текшириши (мисол учун, бирор бир сабаб билан бажарилмаслиги мумкин бўлган амаллар, маслан, файллар устидаги амаллар) назарда тутилади.

Дўстона муносабат деганда эса яхши режалаштирилган диалог ойнаси, ёрдамчи маълумотномалар системасининг мавжудлиги, дастурнинг фойдаланувчи нуқтаи назаридан онгли ва олдиндан айтиш мумкин бўлган ҳулқи тушунилади.

**Эслатма:** Ушбу электрон қўлланмадабиз келтирган дастурларни яхши дастурлаш усулига мисол қилиб олиш мумкин.