**Тармоқланиш ( *if* ) буйруғи**

**А**лгоритмнинг тармоқланиш нуқтасидаги навбатдаги қадамни танлаш ***if*** ва ***case*** буйруғларидан бири ёрдамида амалга оширилиши мумкин. ***If*** (тармоқланиш) буйруғи икки имкониятдан бирини, ***case*** эса бир нечта имкониятдан бирини танлайди.

***If* буйруғи** навбатдаги мумкин бўлган икки қадамдан бирини танлашга имкон беради, яъни тармоқланади. Танлаш кўрсатилган шартнинг бажарилишига боғлиқ равишда амалга оширилади. Умумий кўринишда  ***If*** буйруғи қуйидагича ёзилади:

**if** шарт **then**

**begin**

⁄⁄ *бу ерда агар шарт True бўлса бажариладиган буйруқлар ёзилади.*

**end**

**else**

**begin**

⁄⁄ *бу ерда агар шарт False бўлса бажариладиган буйруқлар ёзилади.*

**end;**

**Else** дан олдин (**End** дан кейин) нуқтали вергул қўйилмайди.

***If*** буйруғи қуйидагича бажарилади:

1. Шартнинг қиймати ҳисобланади. (Унинг қиймати True ёки False).

2. Агар шарт рост (шартнинг қиймати - True) бўлса, у ҳолда ***then*** сўзидан кейинги (***begin*** ва ***end*** орасидаги) буйруқлар бажарилади.  ***If*** буйруғини бажариш шу билан тугайди, яъни ***else*** дан кейинги буйруқлар бажарилмайди. Агар шарт ёлғон (шартнинг қиймати - False) бўлса, у ҳолда ***else*** сўзидан кейинги (***begin*** ва ***end*** орасидаги) буйруқлар бажарилади.

3.2-расмда ***if-tnen-else*** га мос алгоритм келтирилган.

буйруқларининг алгоритми

Масалан, агар ***t*** ўзгарувчи электр занжиридаги қаршиликлар уланиши типини билдирса, (***t=1*** кетма-кет уланиш, ***t=2*** - параллел), ***r1*** ва ***r2*** - қаршиликлар миқдори бўлса, қуйидаги ***if*** буйруғи хисоблаш формуласини танлашга имкон беради.

***if t=l then***

 ***begin***

 ***z := r1 + r2;***

 ***end***

 ***else***

 ***begin***

 ***z := (r1 + r2) ⁄(r1\*r2);***

 ***end;***

Агар ***if*** буйруғида ***begin*** ва ***end*** орасида фақат битта буйруқ бўлса, у ҳолда ***begin*** ва ***end*** ларни ёзмаслик ҳам мумкин. Масалан, юқоридаги буйруқларни

***if t = l then z := r1 + r2***

 ***else z := (r1 + r2⁄(r1\*r2);***

кўринишида ёзиш мумкин. Буйруқнинг тугаганлигини билдирувчи нуқтали вергул белгисини қаерга қўйилганлигига эътибор беринг.

Агар бирор амал фақат шарт рост (True) бўлган ҳолдагина бажарилиб, бошқа ҳолларда бажарилиши талаб қилинмаса, бундай буйруғларни умумий кўринишда қуйидагича ёзиш мумкин:

***if*** *шарт* ***then***

***begin***

*{ шарт фақат рост бўлгандагина бажариладиган буйруқлар}*

***end ;***

Масалан

***if n = m***

***then c := c + l;***

буйруғи фақат ***n=m*** бўлгандагина ***c*** нинг қийматини бирга оширади.

***If*** буйруғини амалда қўлланишини квадрат тенгламанинг ҳақиқий ечимларини топиш мисолида келтирамиз.

Маълумки, квадрат тенглама ***a, b, c*** коэффициентлари орқали берилади. Унинг ҳақиқий ечимларини топиш учун дастлаб формула билан дискриминантини топамиз. Агар

**3.3-расм.**  ***if-then*** буйруғига мос алгоритм.

дискриминант нолдан катта ёки тенг бўлса, квадрат тенглама ҳақиқий ечимларга эга бўлади ва бу ечимлар формулалар билан аниқланади. Агар дискриминант нолдан кичик бўлса ечимлар мавжуд эмас.

Илова формасига коэффициентларни киритиш учун 3 та киритиш майдони, изоҳ ва ечимлар учун тўртта чиқариш майдонлари ҳамда битта буйруқли тугма қўйилади. Бу компоненталарнинг қийматлари 3.3-жадвалда берилади.

**3.4-расм.** Квадрат тенглама дастурининг диалог ойнаси

**Эслатма:** Бу ерда ва бундан буён формаларни тавсифлашда компоненталарнинг ўзгарадиган ҳусусият қийматлари келтирилади ҳалос. Компонеталарнинг қолган ҳусусиятлари ўзгармайди.

 **Квадрат тенглама** дастурининг компонентлари 3.3-жадвал

|  |  |
| --- | --- |
| компонента | Мазмуни |
| Edit1 | ***а*** коэффициент учун |
| Edit2 | ***b*** коэффициент учун |
| Edit3 | ***c*** коэффициент учун |
| Label1 | А коэффициентни киритиш учун |
| Label2 | В коэффициентни киритиш учун |
| Label3 | С коэффициентни киритиш учун |
| Label4 | Ечимни экранга чиқариш учун  |
| Button1 | Хисоблаш жараёнини бошлаш учун |

 **Эслатма:** Форма компоненталарининг қийматлари таърифланган жадвалларда компонента номи ва нуқтадан кейин ҳусусияти номи кўрсатилади. Масалан, жадвалдаги **Form1.Caption** сатри илова формасини яратишда форманинг **Caption** ҳусусиятига кўрсатилган матнни киритишни англатади.

Компоненталарнинг ҳусусиятлари жадвали 3.4-жадвал

|  |  |
| --- | --- |
| **Компонента** | **Мазмуни** |
| Edit1.text |  |
| Edit2.text |  |
| Edit3.text |  |
| Label1 | А ни киритинг |
| Label2 | В ни киритинг |
| Label3 | С ни киритинг |
| Label4 | Ечимлар |
| Button1 | Ечимларни хисоблаш  |

Дастур **Ечимларни хисоблаш** тугмаси босилиши билан хисоблашни бошлайди. Бунда ***onclick*** ходисаси рўй беради. Уни ***TForm1.Button1Click*** процедураси ёрдамида қайта ишланади.

**3.1-листинг. Квадрат тенглама** ечимларини хисоблаш

**unit** kw\_teng;

**interface**

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

**type**

TForm1 = class(TForm)

 Edit1: TEdit;

 Edit2: TEdit;

 Edit3: TEdit;

 Label1: TLabel;

 Label2: TLabel;

 Label3: TLabel;

 Label4: TLabel;

 Button1: TButton;

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

 **private { Private declarations }**

 **public { Public declarations }**

 **end;**

**var**

 **Form1: TForm1;**

**implementation**

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

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

var a1,b1,c1,d,x1,x2:real;

**begin**

 a1 := strtofloat(edit1.Text);

 b1 := strtofloat(edit2.Text);

 c1 := strtofloat(edit3.Text);

 d := b1\*b1-4\*a1\*c1;

 if d >= 0 then

 begin

 x1 := (-b1 + sqrt(d)) ⁄ (2\*a1);

 x2 := (-b1 - sqrt(d)) ⁄ (2\*a1);

 label4.caption :='x1='+floattostr(x1)+#13+'x2='+floattostr(x2);

 end

 else

 label4.caption := 'Тенгламанинг ҳақиқий ечимлари йўқ';

end;

**end.**

Кўпинча дастурда иккита эмас, балки ундан кўп вариантлардан бирини танлашга тўғри келади. Масалан, спортчи мусобақада қатнашиб, А ўринни эгаллаган бўлса, у қандай медал олган? Бу масала учун мумкин бўлган 4 та вариантдан бири ўринли бўлишини ҳисобга олиб, ҳулоса чиқаришга тўғри келади. Ҳулоса эса А нинг қийматига боғлиқ.

**3.5-расм. Мусобақа** дастурининг диалог ойнаси

**Мусобақа** дастурининг алгоритми қуйидагича:

1. Бошлансин
2. Киритилсин А
3. Агар А=1 бўлса В := "олтин медал олган" ; 8 га ўтилсин
4. Агар А=2 бўлса В := "кумуш медал олган" ; 8 га ўтилсин
5. Агар А=3 бўлса В := "бронза медал олган" ; 8 га ўтилсин
6. В := "медал олмаган"
7. Чиқарилсин В
8. Ишни тугатилсин

**3.2-листинг.** Мусобақа дастурида спортчи олган медални аниқлаш.

unit Unit1;

interface

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

type

 TForm1 = class(TForm)

 Edit1: TEdit;

 Label1: TLabel;

 Label2: TLabel;

 Button1: TButton;

 procedure Button1Click(Sender: TObject);

 private

 { Private declarations }

 public

 { Public declarations }

 end;

var

 Form1: TForm1;

implementation

{$R \*.dfm}

procedure TForm1.Button1Click(Sender: TObject);

 var a:integer;

 b:string;

begin

 a := strtoint(edit1.Text);

 if a = 1

 then b := 'олтин медал олган'

 else if a = 2

 then b := 'кумуш медал олган'

 else if a = 3

 then b := 'бронза медал олган'

 else b := 'медал олмаган';

 label2.Caption := 'Спортчи '+b;

end;

end.

Бу мисолда бир нечта ***if*** лар ичма-ич жойлашган. Бундай дастурларни ўқиш ва тушуниш ҳамда компьютер хотирасига киритиш дастурчи учун бир оз қийинроқ.

**3.3.** ***Case* буйруғи**

Аввалги мисолда спортчининг олган медалини аниқлаш учун мумкин бўлган тўртта вариантдан бирини танлаш ичма-ич жойлашган ***if*** буйруқлари орқали амалга оширилган эди. Аммо, кўп вариантлардан бирини танлашда масалага ***if*** ёрдамида ёндашиш ярамайди. Буни айниқса танлаш вариантларининг сони катта бўлганда янада яққол кўриш мумкин.

Delphi тилида вариантлардан бирини осонгина танлашга имкон берадиган ***Case*** буйруғи мавжуд. У умумий кўринишда қуйидагича ёзилади:

***case Селектор of***

 **рўйхат1** :  **begin**  {1-буйруқлар кетма-кетлиги} end;

 **рўйхат2** : **begin** {2-буйруқлар кетма-кетлиги} end;

 . . . . . . . . . . .

 **рўйхатN** : **begin** {N-буйруқлар кетма-кетлиги} **end;**

 ***else***

 **begin** { N+1 -буйруқлар кетма-кетлиги) **end;**

***end;***

Бу ерда ***Селектор*** — қиймати дастурнинг навбатдаги қадамини (яъни, бажарилиши керак бўлган навбатдаги буйруқлар кетма-кетлигини) аниқлайдиган ифода; **рўйхатN** — константалар рўйхати. Агар константалар бирор диапазондаги сонлардан иборат бўлса, улардан биринчиси ва охиргисини икки нуқта билан ажратиб кўрсатиш мумкин. Масалан, 1, 2, 3, 4, 5, 6 константалри ўрнига 1..6 деб ёзса бўлади.

***Case*** буйруғи қуйидагича бажарилади:

1. Дастлаб селектор-ифоданинг қиймати хисобланади.

2. Селектор-ифоданинг қиймати константалар рўйхатидаги константалар билан кетма-кет солиштирилади.

3. Агар селектор-ифоданинг қиймати рўйхатдаги бирор константа билан устма-уст тушса, шу константага мос буйруқлар кетма-кетлиги бажарилади. Шу билан ***саsе*** буйруғининг бажарилиши тугайди.

4. Агар селектор-ифоданинг қиймати константалар рўйхатидаги бирор константага тенг бўлмаса, у ҳолда ***else*** дан кейин турган буйруқлар кетма-кетлиги бажарилади.

Зарурат бўлмаса, ***else*** ни ёзмаслик ҳам мумкин. Бу ҳолда агар селектор-ифоданинг қиймати рўйхатда кўрсатилган контанталарнинг бирортасига тенг бўлмаса, у ҳолда ***case*** буйруғидан кейинги навбатда турган буйруқ бажарилади.

**3.7-расм. *Case*** буйруғининг алгоритми

***Case*** буйруғига мисолар келтирамиз.

**case** n\_day **of**

1,2,3,4,5: day:='Иш куни ' ;

6: day := 'Шанба';

7: day := 'Якшанба';

**end;**

**case** n\_day **of**

1..5: day := 'Иш кун';

6: day := 'Шанба';

7: day := 'Якшанба';

**end;**

**case** n\_day **of**

6: day := 'Шанба';

7: day := 'Якшанба';

**else** day := 'Иш куни';

**end;**

Мисол тариқасида оғирлик ўлчов бирликларидан фунтни килограммга ўтказувчи дастурни кўрамиз. Дастур турли давлатларда фунтни турли белгилашларни ҳам хисобга олади. Масалан, Россияда бир фунт - 409,5 грамм, Англияда — 453,592 грамм, Германия, Дания ва Исландияда эса - 500 граммни ташкил қилади.

3.8-расмдаги диалог ойнасига давлатларни танлаш учун **Давлатлар** рўйхатикиритилган.

**3.8-расм. *Case*** данфойдаланувчи дастурнинг диалог ойнаси

Давлатни танлаш учун рўйхат - ***ListBox*** компонентасидан фойдаланилмоқда. ***ListBox*** компонентасининг нишони **Standart** қуроллар панелида (3.9-расм) жойлашган.

**3.9-расм.**  **ListBox** компонентаси

Рўйхат формага бошқа объектлар каби жойланади. 3.5-жадвалда **ListBox** компонентасининг ҳусусиятлари келтирилган.

**ListBox** компонентасининг ҳусусиятлари 3.5-жадвал

|  |  |
| --- | --- |
| Ҳусусияти | Мазмуни |
| Name | Компонентанинг номи. Дастурда компонента ҳусусиятларига мурожаат қилиш учун фойдаланилади.  |
| Items | Рўйхат элементлари |
| Itemindex | Рўйхатдан танланган элемент номери. Рўйхатдаги биринчи элементнинг номери нолга тенг.  |
| Left | Рўйхатнинг чап чегарасидан форманинг чап чегарасигача бўлган масофа. |
| Top | Рўйхатнинг юқори чегарасидан форманинг юқори чегарасигача бўлган масофа. |
| Height | Рўйхат майдонининг баландлиги |
| Width | Рўйхат майдонининг кенглиги |
| Font | Рўйхат элементларини кўрсатиш учун шрифт  |
| Parent-Font | Она формадан шрифт ҳусусиятларини мерос олиш қилиб белгиси  |

**Items** ва **Itemindex** ҳусусиятлари алоҳида эътиборга сазовор. **Items** ҳусусияти рўйхат элементларини сақлайди.

**Itemindex** рўйхатдан танланган элемент номерини сақлайди. Агар бирорта ҳам элемент танланмаган бўлса, унинг номери минус бирга тенг.

Рўйхат форма яратилаётганда ҳам, дастур ишлаётган вақтда ҳам ҳосил қлиниши мумкин. Форма яратилаётганда рўйхат ҳосил қилиш учун **Object Inspector** ойнасида **Items** ҳусусиятини танлаб, сатрлар рўйхатининг муҳааррири устида сичқонча тугмасини чертиш лозим. (3.10-расм).

Очилган **String List Editor** диалог ойнасида (3.11-расм) ҳар бирини алоҳида сатрга ёзиб, рўйхат элементлари киритилади. Ҳар бир элемент киритилганидан сўнг, янги сатрга ўтиш учун <Enter> тугмаси босилади. Рўйхат тугаганидан сўнг, **ОК** тугмаси чертилади.

**3.10-расм.** Рўйхат муҳарририни ишга тушириш

3.6-жадвалда формадаги компоненталар рўйхати, 3.7-жадвалда эса компоненталар ҳусусиятларининг қийматлари берилган.

**3.11-расм.** Рўйхат муҳаррири

Форманинг компоненталари. 3.6-жадвал

|  |  |
| --- | --- |
| **Компонента** | **Мазмуни** |
| ListBoxl | Давлатлардан бирини танлаш учун  |
| Editl | Оғирликни фунтларда киритиш учун |
| Label1, Label2, Label3 | Киритиш майдонларини изоҳлаш учун  |
| Label4 | Хисоблаш натижасини чиқариш учун  |
| Button1 | Хисоблаш процедурасини активлаштириш учун  |

**Компоненталар ҳусусиятларининг қийматлари 3.7-жадвал**

|  |  |
| --- | --- |
| **ҳусусият** | **Қиймати** |
| Form1 .Caption | ***Case***  дан фойдаланишга мисол дастури |
| Editl. Text |  |
| Label1 . Caption | Давлатни танланг  |
| Label2 .Caption | Давлатлар |
| Label3 . Caption | Фунтлар |
| Button1 . Caption | Хисоблаш |

Қайта хисоблаш процедураси **Хисоблаш** тугмаси босилганда ишга тушади ва фунтлардаги оғирликни 1 килограммдаги фунтлар қоэффициентига кўпайтиради. Бу коэффициентни рўйхатдан танланган элементга қараб аниқланади.

3.3.-листингда фунтни килограммга ўтказиш дастурининг матни келтирилган.

**3.3-листинг.** Оғирликни **фунтдан килограммга ўтказиш.**

unit funt;

interface

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

type

 TForm1 = class(TForm)

 Label1: TLabel;

 Label2: TLabel;

 Label3: TLabel;

 Edit1: TEdit;

 Label4: TLabel;

 Button1: TButton;

 ListBox1: TListBox;

 procedure FormCreate(Sender: TObject);

 procedure Button1Click(Sender: TObject);

 private

 { Private declarations }

 public

 { Public declarations }

 end;

 var Form1: TForm1;

implementation

{$R \*.dfm}

procedure TForm1.FormCreate(Sender: TObject);

begin

{

ListBox1.items.add('Россия');

ListBox1.items.add('Австрия');

ListBox1.iterns.add('Англия');

ListBox1.items.add('Германия');

ListBox1.iterns.add ('Дания');

ListBoxl.iterns.add('Исландия');

ListBox1.iterns.add ('Италия');

ListBox1.items.add ('Нидерландия'); }

ListBox1.itemindex := 0;

end;

procedure TForm1.Button1Click(Sender: TObject);

var

funt:real; ⁄⁄ фунтдаги оғирлик

kg:real; ⁄⁄ килограммдаги оғирлик

k:real; ⁄⁄ қайта хисоблаш коэффициенти

begin

case ListBox1.Itemindex of

0: k := 0.4095; ⁄⁄ Россия

1: k := 0.453592; ⁄⁄ Англия

2: k := 0.56001; ⁄⁄ Австрия

3..5,7: k := 0.5; ⁄⁄ Германия, Дания, Исландия, Нидерландия

6: k := 0.31762; ⁄⁄ Италия

end;

funt := StrToFloat(Edit1.Text);

kg := k\*funt;

label4.caption := Edit1.Text + ' ф. - бу '

+ FloatToStrF(kg,ffFixed, 6,3) + 'кг.';

end;

end.

 **FormCreate** ходисаларни қайта ишлаш процедурасига эътибор беринг. Бу ходиса форма очилган заҳоти автоматик тарзда содир бўлади. Бу процедурани ўзгарувчиларга бошланғич қиймат беришда, шу жумладан рўйхатга элементларни қўшиш учун фойдаланиш ҳам мумкин. Келтирилган мисолда рўйхат муҳаррири томонидан дастурнинг иши давомида ташкил қилинган.

**3.4.** **Цикллар**

Кўплаб масалаларнинг ечиш алгоритмлари циклик жараёнларни ўз ичига олади, яъни мақсадга эришиш учун маълум бир буйруқлар кетма-кетлиги бир неча марта такроран бажаришни назарда тутади.

Масалан, билимларни назорат қилиш дастури савол беради, жавобни қабул қилади, жавобнинг баҳосини баллар йиғиндисига қўшади, сўнгра бу амалларни то назорат қилинувчи токи ҳамма саволларга жавоб бериб бўлмагунча яна бир неча марта такроран бажаради. Бошқа мисол. Рўйхатдан бирор фамилияни қидириш талаб қилинган бўлсин. Дастлаб, рўйхатдан биринчи фамилия текширилади, сўнгра иккинчиси, учинчиси ва х.к. Бу жараён токи қидирилаётган фамилия топилгунча ёки рўйхат тугагунча давом этади.

Айрим буйруқлари бир неча марта такроран бажарилиши талаб қилинган алгоритмларни циклли алгоритм, шу буйруқларнинг ўзи эса цикл деб аталади.

Delphi да циклли жараёнларни дастурлаш учун уч ҳил кўринишдаги буйруқлардан фойдаланиш мумкин: ***for, while*** ва ***repeat*** буйруқлари.

**3.5.** ***For***  **цикли**

Қуйидаги масалани кўрайлик. *у = 3х2 - 2х + 7* функциянинг қийматларини в точках —1, 0, 1, 2 ва 3 нуқталардаги қийматларини хисоблаш талаб қилинган бўлсин. (хисобланган қийматлар жадвал кўринишида, форманинг Label майдонига чиқариш керак). Бу масаланинг ечиш дастури қуйидагича бўлиши мумкин:

**3.4-листинг.**

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

var **у: real; ⁄⁄** функциянинг қиймати

**x: real; ⁄⁄** функциянинг аргументи

**dx: real; ⁄⁄** аргументнинг орттирмаси

st: **string; ⁄⁄** жадвални ифодалаш

**begin**

st := ''; x := -1; dx := 1;

у := 3\*х\*х – 2\*х + 7;

st := st + FloatToStr(x) + ' ' + FloatToStr(y) + chr(13);

x := x + dx;

у := 3\*х\*х – 2\*х + 7;

st := st + FloatToStr(x) + ' ' + FloatToStr(y) + chr(13);

x := x + dx;

у := 3\*х\*х – 2\*х + 7;

st := st + FloatToStr(x) + ' ' + FloatToStr(y) + chr(13);

x := x + dx;

у := 3\*х\*х – 2\*х + 7;

st := st + FloatToStr(x) + ' ' + FloatToStr(y) + chr(13);

x := x + dx;

у := 3\*х\*х – 2\*х + 7;

st := st + FloatToStr(x) + ' ' + FloatToStr(y) + chr(13);

x := x + dx;

Label1.Caption := st;

**end;**

Процедура матнидан кўриниб турибдики,

у := 3\*х\*х – 2\*х + 7;

st := st + FloatToStr(x) + ' ' + FloatToStr(y) + chr(13);

x := x + dx;

буйруқлари кетма-кетлиги беш марта кўрсатилган ва беш марта бажарилади.

Ечилаётган масала етарлича содда бўлишига қарамай, унинг дастурини ёзиш ва компьютер хотирасига киритиш дастурчилар учун бир оз ноқулайликлар пайдо қилади. Биз кўрган масалада х нинг 5 та қийматини хисобга олиш талаб қилинган эди. Агар х нинг 100 та ёки 1000 та нуқтадаги қийматларини топиш талаб қилинса, бу ноқулайлик янада кучайган бўлар эди.

Шунга ўхшаш ноқулайликларнинг олдини олиш учун Delphi дастурлаш тилига ***For*** буйруғини киритилган. У умумий кўринишда қуйидагича ёзилади:

***For x0 := x1 to x2 do***

***begin*** цикл буйруқлари кетма-кетлиги ***end;***

Бу ерда ***х0*** - цикл жараёнини бошқарадиган ўзгарувчи бўлиб, циклнинг бажарилиш ёки бажарилмаслиги унинг жорий қийматига боғлиқ; ***х1 - х0*** нинг бошланғич қиймати; ***х2 - х0*** нинг охирги қиймати. ***х0, х1, х2*** – параметрлар бутун типдаги бўлиши шарт.

***For*** буйруғи қуйидагича бажарилади: Дастлаб ***х0*** – ўзгарувчига ***х1*** - бошланғич қиймат берилади. ***х0*** нинг қиймати учун ***x0≥x2*** шарти текширилади. Агар бу шарт ўринли бўлмаса, цикл буйруқлари кетма-кетлиги бир марта тўла бажарилади. Шундан сўнг, ***х0*** - нинг қиймати бирга орттирилади ва яна ***x0≥x2*** шарти текширилади. Бу шарт ўринли бўлмаса, цикл буйруқлари кетма-кетлиги яна бир марта тўла бажарилади. Бу жараён токи ***x0≥x2*** шарти ўринли бўлиб қолгунча давом этаверади. ***x0≥x2*** шарти рост, яъни True бўлганидан кейингина ***for*** дан кейинги навбатда турган буйруқнинг бажаришга ўтилади. Бу фикрларни блок-схемалар (3.12-расм) орқали ифодаланса, ***for*** буйруғининг бажарилиши тартиби янада тушунарлироқ бўлади. Бу блок-схемадан ***x1*** параметри ***x2*** дан катта бўлса, цикл бир марта ҳам бажарилмаслиги ҳамда цикл буйруқлари кетма-кетлиги бир марта бажарилганда, ***х0*** – нинг қиймати автоматик тарзда бирга ортиши кўриниб турибди. Демак, ***For*** буйруғидан такрорланишлар сони олдиндан маълум бўлган ҳолатдагина фойдаланиш мумкин.

Келтирилган фикрларни хисобга олиб, юқоридаги масалани ***for*** буйруғи ёрдамида қуйидагича ёзиш мумкин:

**3.12-расм**. ***For*** буйруғининг бажарилиш алгоритми

**3.5-листинг.**

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

var **у: real; ⁄⁄** функциянинг қиймати

**x: real; ⁄⁄** функциянинг аргументи

**dx: real; ⁄⁄** аргументнинг орттирмаси

st: **string; ⁄⁄** жадвални ифодалаш

**begin**

st :=''; x := -1; dx := 0.5;

**for** I := l **to 5 do begin**

у := 3\*х\*х – 2\*х + 7;

st := st + FloatToStr(x) + ' ' + FloatToStr(y) + chr(13);

x := x + dx; **end;**

Label1.Caption := st;

End.

Ҳар икки дастур вариантини солиштириб кўрилса, иккинчи вариантнинг ҳар томонлама қулайлиги кўриниб турибди. Биринчидан, бу дастур матнини компьютер хотирасига киритиш оз вақт талаб қилади. Иккинчидан, бу дастур матнини ўқиш ва тушуниш аввалгисига қараганда енгил. Учинчидан, зарурат бўлса, ***х*** нинг параметрларини осонгина ўзгартириш мумкин.

***For*** циклининг такрорланишлари сони *х2-х1+1* га тенг бўлади.

 Агар циклда такроран бажарилиши керак бўлган буйруқлар кетма-кетлиги фақат битта буйруқдан иборат бўлса, у ҳолда ***do*** хизматчи сўзидан кейинги ***begin*** ва ***end*** ларни ёзмаслик мумкин. Бу ҳолда буйруқнинг умумий кўриниши

 ***For x0 := x1 to x2 do***  буйруқ ***;***

тарзида бўлади. Қуйидаги мисолда иккита берилган ***a*** ва ***b*** бутун сонлари орасидаги йиғинди, яъни ***a + (a+1) + ... + b*** ифоданинг қиймати хисобланмоқда. Бу масала учун яратилган формада иккита **Edit**, биттадан **Label** ва **Button** компоненталари қатнашади.

**3.6-листинг.**

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

 **var i,s,a,b:integer;**

**begin**

 **s := 0;**

 **a := strtoint(edit1.Text);**

 **b := strtoint(edit2.Text);**

 **for i := a to b do s := s+i;**

 **label1.Caption := inttostr(s);**

**end;**

Цикл жараёнини бошқарувчи ***х0*** параметрдан циклнинг ичида фойдаланмаслик ҳам мумкин. Бу ҳолда унинг вазифаси талаб қилинган сондаги такрорланишларни таъминлашдан иборат бўлади. Масалан,

ифоданинг қийматини хисоблаш талаб қилинган бўлсин. Бу ерда ***х*** – ўзгарувчидан ташқари яна иккита ўзгарувчи керак бўлади. Биринчиси янги қўшилувчини, иккинчиси эса умумий йиғиндини хисоблаш учун керак. Дастур формасида иккита ***Edit*** (***x*** ва ***n*** учун), биттадан ***Label*** ва ***Button*** компоненталари (зарурат бўлса, ***x*** ва ***n*** ўзгарувчиларни изоҳлаш учун иккита Label ни формага киритиш мумкин) қатнашади.

**3.7-листинг.**

procedure TForm1.Button1Click(Sender: TObject);

 var n,i:integer;

 x,s1,s2:real;

begin

 x := strtofloat(edit1.Text);

 n := strtoint(edit2.Text);

 s1 := x;

 for i := 1 to n do begin

 s1 := sin(s1);

 s2 := s2 + s1; end;

 label1.caption := floattostr(s2);

end;

 Юқорида айрилганидек, бу дастурдаги ***i*** ўзгарувчининг вазифаси цикл жараёнини ***n*** марта такрорланишини таъминлашдан иборат.

Юқорида айтиб ўтилдики, ***for*** фойдаланганда ***х0*** –параметрнинг ўзгариш қадами бирга тенг. Бу бууйруқдан ***х0*** – нинг ўзгариш қадами -1 га тенг бўлганда ҳам фойдаланиш мумкин. Бу ҳолда ***for*** буйруғининг умумий кўриниши

***For x0 := x1 Downto x2 do***

***begin*** цикл буйруқлари кетма-кетлиги ***end;***

тарзида бўлади.

 Қуйидаги ифоданинг қийматини ҳисоблаш талаб қилинган бўлсин. Кўриниб турибдики, бу ифоданинг қийматини хисоблаш жараёнини энг ички илдиздан бошланиши керак. Агар дастурда йиғиндини хисоблаш учун ***S*** ўзгарувчи киритилган бўлса, унинг бошланғич қиймати 0 га тенг. Навбатдаги йиғинди формула бўйича топилади. ***i*** нинг қийматлари эса ***n*** дан бошлаб ҳар бир қадамда 1 га қараб камайиб боради. Қўйилган масаланинг дастур матни қуйидагича ёзилади:

**3.8-листинг.**

procedure TForm1.Button1Click(Sender: TObject);

 var n,i:integer;

 s:real;

begin

 n := strtoint(edit1.text);

 s := 0;

 for i := n downto 1 do s := sqrt(s + 3\*i);

 label1.Caption := floattostr(s);

end;

менюга

**3.6.** ***While* цикли**

***While*** (токи) буйруғи такрорланишлар сони олдиндан маълум бўлмаган ҳолларда циклларни ташкил қилиш учун мўлжалланган.

Одатда ***while*** буйруғи билан берилган аниқликдаги хисоблаш, массив ва файллардан берилган элементни қидириш каби масалаларни дастурлашда фойдаланиш мумкин.

***While*** буйруғи умумий кўринишда қуйидагича ёзилади:

***while шарт******do begin***

 ⁄ ⁄ бу ерда циклнинг буйруқлари кетма-кетлиги ёзилади.

***end* ;**

Бу ерда ***шарт*** – такрорланиш ёки такрорланмасликни аниқлайдиган мантиқий ифода.

***While*** буйруғи қуйидаги алгоритм асосида бажарилади:

1. Дастлаб ***шарт*** қиймати хисобланади.

2. Агар ***шарт*** нинг қиймати ***False*** (яъни ***шарт*** ўринли бўлмаса), у ҳолда ***while*** буйруғини бажариш шу ерда тугатилади ва ундан кейинги навбатда турган буйруқни бажаришга ўтилади.

3. Агар ***шарт*** нинг қиймати True (яъни, ***шарт*** ўринли) бўлса, у ҳолда ***begin*** ва ***end*** лар ўртасида кўрсатилган циклнинг буйруқлари кетма-кетлиги бир марта тўла бажарилади. Шундан кейин ***шарт*** яна бир марта текширилади. Агар ***шарт*** ўринли бўлса, циклнинг буйруқлари кетма-кетлиги яна бир марта тўла бажарилади. Бу жараён токи ***шарт*** ўринли бўлмай (False) қолгунча давом этаверади.

***While*** буйруғига мос алгоритм қуйидагича ёзилади:

**3.13-расм. While** буйруғининг ишлаш алгоритми

Алгоритмдан қуйидаги ҳолатлар кўриниб турибди:

1. ***While*** цикли бир марта ҳам бажарилмаслиги мумкин. Бунинг учун унда кўрсатилган шарт биринчи марта текширилгандаёқ ўринли бўлмаслиги етарли.
2. ***While*** циклидаги буйруқлар кетма-кетлиги ҳеч бўлмаганда бир марта бажарилиши учун, ***while*** буйруғидаги шартда қатнашадиган параметрларга аввалдан ***шарт*** ўринли бўладиган қийматларни бериб қўйиш лозим.
3. Циклнинг қачондир тугашини таъминлаш учун, цикл буйруқлари кетма-кетлиги цикл шартида қатнашадиган параметрларнинг қийматларига таъсир этиб бориши (шартда қатнашадиган ўзгарувчиларнинг қийматларини цикл ичида ўзгартириб бориш) керак.

Берилган N натурал сонининг туб ёки туб эмаслигини аниқлаш масаласини кўрайлик. Биз иш бошламасдан аввал бу сонни туб деб фараз қиламиз. Энди қилган фаразимизнинг тўғри ёки нотўғрилигини аниқлаймиз. Бунинг учун ҳар қандай натурал соннинг 1 дан бошқа бўлувчилари оралиқда ётишини билган ҳолда, N сонини шу интервалдаги барча k бутун сонларга сонларга бўлиб кўрамиз. Агар N сони шу сонлардан бирортасига бўлиниб қолса, у туб эмас, яъни бизнинг фаразимиз нотўғри. Такрорлаш жараёни то k бўлувчининг қиймати сонидан катта бўлиб қолгунча ёки фаразимиз бузилгунча этади. Бу масаланинг дастури қуйидагича ёзилади:

**3.9-листинг. N натурал сонининг туб ёки туб эмаслигини аниқлаш.**

procedure TForm1.Button1Click(Sender: TObject);

 var n,m,k:integer;

 y:string;

begin

 n := strtoint(edit1.Text);

 y := 'Туб';

 m := trunc(sqrt(n));

 k := 1;

 while (k <= m) and (y = 'Туб') do

 begin

 k := k +1;

 if n mod k = 0 then y := 'Туб эмас';

 end;

 label2.caption := 'Берилган сон' + edit1.text + #13 + 'У ' + y;

end;

Энди π сонини фойдаланувчи танлаган аниқликда ҳисоблаш дастурини кўрайлик. Бу масаланинг алгоритми асосида

йиғиндининг етарлича катта номерли хадлар учун π⁄4 га яқинлашиши ётади.

Берилган қаторнинг ҳар бир n-чи ҳадини топиш учун ни ҳисоблаймиз ва йиғиндига қўшамиз. Хисоблаш навбатдаги ҳаднинг қиймати берилган аниқликдан кичик бўлгунча давом эттирилади. -1 нинг даражасини ҳисоблаш учун қуйидагича йўл тутамиз. Маълумки, -1 нинг даражалари ишорасини навбати билан ўзгартириб, бир марта +1, бир марта -1 га тенг бўлади. Шунинг учун, дастурда қиймати -1 га тенг бўлган битта ўзгарувчи оламиз. Ҳар гал цикл бир марта бажарилганда, биз шу ўзгарувчининг қийматини тескарисига ўзгартириб борамиз. Бу бизга -1 нинг даражаларини беради. Олинган ҳадларнинг йиғиндиларини S га йиғиб борамиз. Кўрсатилган аниқликка эришганимиздан сўнг, S нинг қийматини 4 га кўпайтириб қўямиз. Чунки қаторнинг йиғиндиси π⁄4 га тенг эди.

Бу дастурнинг диалог ойнаси қуйидагича: Формага **Edit1** (аниқликни кўрсатиш учун), **Label1** (аниқликни изоҳлаш учун), **Label2** (натижани чиқариш учун) ҳамда **Button1** (хисоблашни бошлаш учун) компоненталарни жойлаймиз.

**3.14-расм. π сонини хисоблаш** дастурининг диалог ойнаси

**3.10-листинг. π сонини хисоблаш**

procedure TForm1.Button1Click(Sender: TObject);

 var eps,m,s,t:real ;

 n : longint;

**begin**

 eps := strtofloat(edit1.text);

 m := -1;

 s := 0;

 n := 0;

 t := 1; ⁄ ⁄ цикли камида бир марта бажарилиши учун

 **while** abs(t) >= eps **do begin**

 n := n +1; ⁄ ⁄ янги ҳаднинг номери

 m := -m; ⁄ ⁄ (-1) нинг навбатдаги даражаси

 t := m ⁄ (2\*n-1);

 S := S + t; **end;**

 S := S\*4;

 label2.caption := floattostr(s) + #13 +

 'Ҳисоблаш ' + inttostr(n) + '-чи ҳадда ' + #13 + 'тугатилди';

**end;**

 **Эслатма:** Агар ***while*** буйруғи билан ташкил қилинган циклда такроран бажариладиган буйруқлар кетма-кетлиги битта буйруқдан иборат бўлса, ***do*** хизматчи сўзидан кейин, шу буйруқнинг ўзини ***begin*** ва ***end*** ларсиз ёзиш мумкин. У ҳолда буйруқнинг умумий кўриниши

***While шарт do буйруқ;***

тарзида ёзилади.

**3.7.** **Repeat цикли**

***Repeat*** буйруғи худди ***while*** каби такрорлашлар сони олдиндан маълум бўлмаган вақтда, дастурнинг айрим буйруқлари кетма-кетлигини такроран бажарилишини таъминлаш мақсадида фойдаланилади. Такрорланиш сони бу усулдаги циклда ҳам дастурнинг бажарилиши давомида аниқланади.

 ***Repeat*** буйруғи умумий кўринишда қуйидагича ёзилади:

***repeat***

⁄ ⁄ буйруқлар кетма-кетлиги

***until шарт***

Бу ерда ***шарт*** — мантиқий типдаги ифода бўлиб, циклнинг бажариш ёки бажармасликни ҳал қилади.

***Repeat*** буйруғи қуйидагича бажарилади:

1. ***Repeat*** ва ***until*** орасидаги буйруқлар кетма-кетлиги бажарилади.

2. Сўнгра мантиқий ифоданинг қиймати хисобланади. Агар мантиқий ифоданинг қиймати ёлғон (False) бўлса, 1-қадамга ўтилади, акс ҳолда 3-қадамга ўтилади.

3. Циклни бажариш тўхтатилади ва навбатда турган буйруқни бажаришга ўтилади.

Шундай қилиб, **repeat** ва ***until***  орасида турган буйруқлар ктема-кетлиги мантиқий ифоданинг қиймати ёлғон (False) бўлса, такрор ва такрор бажарилаверади. Унинг ишлаш алгоритми 4.4-расмда келтирилган.

**3.14.-расм. Repeat буйруғининг ишлаш а**лгоритми

**Масала:** Фибоначчи сонлари формулалар билан хисобланади. Олдиндан берилган К сонидан катта бўлган биринчи фибоначчи сонини топинг.

Бу масалани ечиш учун қуйидагича фикр юритамиз. К сони шундай катта бўлиши мумкинки, шунча фибоначчи сонларини сақлаш учун компьютер хотираси кичиклик қилиб қолиши мумкин. Демак, оддий йўл билан бу масалани ҳал қилиш қийин. Бошқача мулоҳаза юритамиз. Фомулалардан кўриниб турибдики, янги ҳадни топишда ундан аввалги икки ҳад ишлайди ҳалос. Қолганларини керак бўлмагани учун ташлаб юбориш ҳам мумкин. Шунинг учун белгилаш киритамиз. Топилаётган янги ҳадни ***f2*** ўзгарувчи билан, ундан битта олдинги ҳадни ***f1*** билан, иккита олдинги ҳадни эса ***f0*** билан белгилаймиз. Нечанчи ҳад топилаётганлигидан қатъий назар, биз шу белгилашга содиқ қоламиз. Ишни бошлаш учун ***f0=f1=1*** қийматдан фойдаланамиз. Янги ***f2*** ҳад топилганидан кейин, у ўзидан кейинги ҳад учун ***f1*** вазифасини ўташи лозим. Уни топишда қатнашган ***f1*** эса ***f0*** вазифасини бажаради.

**3.11-листинг. К дан катта бўлган биринчи Фибоначчи сони.**

procedure TForm1.Button1Click(Sender: TObject);

 var k,f0,f1,f2:longint;

begin

 k := strtoint(edit1.Text);

 f0 := 1; f1 := 1;

 repeat

 f2 := f1 + f0; f0 := f1; f1 := f2;

 until f2 >= k;

 label1.caption := inttostr(f2);

end;

 Бу масалани ***while*** цикли билан ҳам ҳал қилиш мумкин эди. Аммо, бу ҳолда ***repeat*** ва ***until***  лар орасидаги буйруқларни циклга кирмай туриб ҳам ёзишга тўғри келган бўлар эди, яъни бир ҳил буйруқларни икки марта ёзилади. Биринчи мартасида циклга киришга тайёргарлик учун, иккинчи мартасида эса навбатдаги ҳадларни ҳисоблаш учун. ***repeat*** ва ***until***  циклида эса бундай ноқулайлик йўқ, ҳамма буйруқлар бир мартадан ёзилган.

 Цикллар билан ишлаганда, уларнинг қачондир тугаши лозимлигини назарда тутиш керак. Айрим ҳолларда, бошланғич маълумотларнинг нотўғри киритилганлиги, циклни тугашини ҳал қиладиган мантиқий ифодаларда қатнашадиган ўзгарувчиларнинг қийматларининг ўзгармаслиги ёки бошқа қандайдир сабаблар "чексиз цикл" нинг юзага келтириши мумкин. Бу ҳолда компьютер хеч қачон бу циклни тугата олмайди, демак, масалани охиригача еча олмайди. Бундай ҳатолик синтактик бўлмагани учун, компьютер уни сезмай қолади. Қуйидаги дастур парчаларига эътибор беринг:

А) x := 3; y := x mod 2;

 While y <> 0 do begin x := x + 2; y := x mod 2; end;

Б) x := 3;

 repeat y := x mod 2; x := x + 2; until y = 0;

Ҳар икки мисолда "чексиз цикл" юзага келганлигини кўриш мумкин.

**Диққат! 1. Repeat** ва ***until***  орасидаги буйруқлар кетма-кетлиги ҳеч бўлмаганда бир марта ишлайди. 2. Цикл қачондир тугаши учун **Repeat** ва ***until***  орасидаги буйруқлар кетма-кетлиги мантиқий ифода таркибига кирган ўзгарувчиларнинг қийматларини ўзгартирсин. 3. ***Repeat*** цикли камида бир марта ишлайди, ***For*** ва ***While*** цикллари эса бир марта ҳам ишламаслиги мумкин.

**3.8.** ***Goto*** **буйруғи**

Биз юқорида ***if*** ва ***case*** буйруқлари навбатдаги буйруқлар кетма-кетлигини бажариш ёки бажармаслиги бирор шартнинг ўринли бўлишига боғлиқлигини кўрган эдик. Шунинг учун у буйруқларни шарти ўтиш буйруқлари деб ҳам юритилади. Амалларнинг бажарилиш тартибини буза оладиган яна бир буйруқ - ***goto*** шартсиз ўтиш буйруғи ҳам мавжуд. У умумий кўринишда қуйидагича ёзилади:

***goto тамға***;

Бу ерда ***тамға*** – ***goto*** буйруғидан кейин бажарилиши керак бўлган оператордан олдин жойлашган идентификатор.

***goto*** буйруғида фойдаланиладиган тамға тамғаларни эълон қилиш бўлимида эълон қилинган бўлиши лозим. Бу бўлим ***label*** сўзи билан бошланади ва дастур матнида ўзгарувчиларни эълон қилиш бўлимидан олдин жойлаштирилади.

Дастурда тамға ***goto*** буйруғидан кейин бажарилиши керак бўлган буйруқдан олдин қўйилади ва ундан икки нуқта билан ажратиб қўйилади.

3.14-листингда натурал соннинг тублигини текширадиган дастур матни келтирилмоқда. Биз илгари юқорида келтирган натурал соннинг туб ёки туб эмаслигини текшириш дастури айрим камчиликлардан ҳоли эмас эди. Унда киритилган маълумотларни назорат қилиш ҳамда 2 сони учун дастур нотўғри натижа берар эди. ***Goto*** буйруғи ёрдамида ана шу камчиликларни бартараф қилишга уриниб кўрамиз. Бунинг учун 1) агар фойдаланувчи манфий сон киритса, бу ҳақида махсус ахборотни экранга чиқариб, дастур ишини тугатишини ташкил қиламиз; 2) 1 сони киритилган бўлса, уни текширмаймиз; 3) фойдаланувчи 2 сонини киритса, уни тўғридан-тўғри туб деб, дастур ишини тўхтатамиз. 4) Қолган сонларни эса эски алгоритмимиз ёрдамида текширамиз.

**3.12-листинг. Туб сон дастури.**

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

**label Bye, javob; ⁄⁄** тамғаларни эълон қилиш бўлими

var n: integer; ⁄⁄ текшириладиган сон

k: integer; ⁄⁄ бўлувчи

y: string; ⁄⁄ натижа

m : integer; ⁄⁄ мумкин бўлган энг катта бўлувчи

**begin**

n := StrToInt(Edit1.text);

**if** n <= 0 **then begin**

MessageDlg('Сон нолдан катта бўлиши керак ', mtError, [mbOk], 0);

Edit1.text := **";**

**goto** bye;

**end;**

⁄⁄ мусбат сон киритилган бўлса

If n = 1 then begin

y := '1 сони туб ҳам, мураккаб ҳам эмас');

goto javob;

end

else

if n = 2 then begin

y :='2 cони туб. ';

goto javob;

end

else

⁄⁄ 2 дан катта бутун сони киритилган бўлса

y := 'Туб';

 m := trunc(sqrt(n));

 k := 1;

 while (k <= m) and (y = 'Туб') do

 begin

 k := k +1;

 if n mod k = 0 then y := 'Туб эмас';

 end;

 javob : label2.caption := 'Берилган сон ' + edit1.text + #13

 + 'Ó ' + y;

Bye :

end;

 Дастурлаш бўйича адабиётларда ***goto*** буйруғидан умуман фойдаланмасликка ёки камроқ фойдаланишга чақирилади. Чунки, ундан, айниқса ноўрин фойдаланилганда дастур матни чалкашиб кетади. Аммо, бундай ҳулоса қатъий эмас. ***Goto*** буйруғини ўринли қўллаш имкониятлари мавжуд. Юқоридаги масала бунга мисол бўлиши мумкин.