Биз ушбу мавзуни муҳокама қилишдан аввал DELPHI дастурлаш тилида ишлаш мумкин бўлган маълумотларнинг типларини эслайлик.

**Бутун тип.** DELPHI тили еттита типдаги бутун сонли маълумотларни қабул қила олади: Shortint, Smailint, Longint, Int64, Byte, Word ҳамда Longword.

**Бутун типдаги сонлар**  1-жадвал

|  |  |  |
| --- | --- | --- |
| **типи** | **Диапазони** | **ўлчами** |
| Shortin t | -128 ... 127 | 8 бит |
| Smallint | -32 768 ... 32 767 | 16 бит |
| Longint | -2 147 483 648... 2 147 483 647 | 32 бит |
| Int64 | -263 ... 263 - 1 | 64 бит |
| Byte | 0...255 | 8 бит, ишорасиз |
| Word | 0...65 535 | 16 бит, ишорасиз |
| Longword | 0 ... 4 294 967 295 | 32 бит, ишорасиз |

DELPHI тили энг универсал бутун типли маълумот ҳисобланадаиган ва longint типига эквивалент бўлган Integer типидаги маълумотларни ҳам қабул қилади.

**Ҳақиқий тип.** DELPHI тилида олтита ҳақиқий типдаги маълумотлар мавжуд: real48, single, double, extended, comp, currency. Бу типлар бир-биридан қабул қиладиган қийматларининг диапазони, ишончли рақамларининг сони ва компьютер хотирасидан эгаллайдиган хажмлари билан фарқланади. 1-жадвалда бу типлар улар ораидаги фарқ келтирилган.

DELPHI тили Double типига эквивалент бўлган универсал ҳақиқий тип - Real типидаги маълумотларни ҳам қабул қилади.

**Ҳақиқий типдаги маълумотлар** Жадвал-2

|  |  |  |  |
| --- | --- | --- | --- |
|  | Диапазон | Ишончли рақамлари | байт |
| Real48 | 2.9 x 10-39 ... 1.7 x 1038 | 11-12 | 06 |
| Single | 1.5 x 10-45 ... 3.4 х 1038 | 7-8 | 04 |
| Double | 5.0 x 10-324 ... 1.7 x 10308 | 15-16 | 08 |
| Extended | 3.6 x 10-4951 ... 1.1 х 104932 | 19-20 | 10 |
| Comp | 2-63+1 ... 263-1 | 19-20 | 08 |
| Currency | -922 337 203 685 477.5808  ... 922 337 203 685 477.5807 | 19-20 | 08 |

Турли типдаги маълумотлар билан ишлаганда талабалар йўл қўйиши мумкин бўлган ва энг кенг тарқалган ҳатолик Edit ойнасига киритилаётган маълумот билан боғлиқ. Ёш дастурчилар одатда компьютер билан фақат «ақлли» одамлар ишлайди, улар масала шартида кўрсатилган типдаги маълумотни киритишади деб ўйлашади. (1 ва 2-жадвалларга эътибор беринг) Ҳўш, бунинг акси бўлса-чи? деган савол одатда уларнинг ҳаёлларига ҳам келмайди. Шунинг учун, одатда улар ёзган дастлабки дастурларнинг таркибида бундай англашилмовчиликлар ва кутилмаган ҳолатлар умуман ҳисобга олинмайди. Ваҳолангки, ёш дастурчилар бундай вазиятларнинг юзага келиб қолиши мумкинлиги ва унинг олдини олишга илк қадамлариданоқ кўникканлари маъқул.

Талаб қилинган типдаги маълумотни киритиш масаласи ҳам DELPHI тилида ҳимояланган блок механизми try-except ёрдамида ҳал қилиниши мумкин. Шунинг учун, талабаларни, энг содда, чизиқли дастур ёзишни ўрганганларидан кейиноқ, бу механизм билан таништирилганлиги маъқул. Аммо, бу механизм ҳам дастурда вужудга келиши мумкин бўлган барча ҳатоликларни олишда етарли восита бўла олмайди. Демак, дастурчилар киритилаётган маълумотларни назорат қилишни дастурда ҳисобга олганлари мақсадга мувофиқ бўлади. Намуна тариқасида N! ни ҳисоблаш дастурини кўрамиз. Матнда натурал сонларни киритилишга эътибор беринг.

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

*var n, i, fak : integer;*

*begin*

*try*

*n := strtoint(edit1.text);*

*except*

*on econverterror do*

*showmessage(' Butun son kirirting'');*

*end;*

*if n<1 then begin*

*showmessage('Natural son kirirting'');*

*exit end*

*else*

*begin*

*fak := 1;*

*for i := 1 to n do fak := fak\*i;*

*label1.caption := inttostr(fak);*

*end;*

*end;*

Ушбу матнда  *econverterror* методи *edit1.text* ойнасига киритилган сатрни бутун сонга айлантириш қай даражада амалга оширилганлигини аниқлайди.

Сонли маълумотларни қайта ишлашда энг кўп учрайдиган навбатдаги муаммо - бу олинаётган натижаларнинг шу натижа мансуб бўлган тип диапазонидан четга чиқишидир (1 ва 2-жадвалларга қаранг). Юқоридаги дастур Edit1 ойнасига киритилган 12 сони учун 479 001 600 натижа бергани ҳолда, 13 сони учун 5447020800 ўрнига 1 932 053 504 натижани экранга чиқаради. Бундай катта сонлар билан ишлаш, айниқса олинган натижаларнинг тўғри ёки нотўғрилигини аниқлаш мушкул вазифа ҳисобланади. Бошловчи дастурчилар кўпинча бу кўринишдаги натижаларни тўғри деб қабул қилишади. Чунки, агар улар ёзган дастур нотўғри бўлганда эди, компьютер ҳатолик ҳақида ахборот берган бўлур эди. Компьютер дастурни бажариб, қандайдир натижа бердими, демак «дастур тўғри ёзилган» қабилидаги қарашлар ёш дастурчиларни кўпинча ҳушёрликдан бир оз чалғитади.

Бутун сонлар билан ишлашда бу каби ҳатоликнинг юзага келишининг асосий сабаби – бу маълумотларнинг белгиланган диапазондан четга чиқишидир. Бундай ҳатоликнинг олдини *Erangeerror* класси ҳам ола олмайди. Олинган маълумот диапазондан четга чиқса, у ҳолда компьютер ҳатолик ҳақида ахборот бериш ўрнига, олинган натижани тўғридан тўғри [***юқори диапазон+1***] га бўлиб, қолган қолдиқни натижа сифатида қабул қилади.

Дастурда *fak* ўзгарувчисини *longword* типида деб белгилаш ҳам вазиятни қутқара олмайди. Чунки, *n* сони етарлича катта бўлиши мумкин. Яна юқоридаги фикримизга қайтиб, дастурчиларга нафа=ат киритилаётган маълумотларни, балки олинаётган натижаларни ҳам дастурда назорат қилишни таъкидлаб ўтамиз. Факториалнинг ҳисоблаш дастурини қуйидагича ёзишга тавсия этамиз.

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

*var n,i, fak: integer;*

*begin*

*try*

*n := strtoint(edit1.text);*

*except*

*on econverterror do*

*showmessage('Butun son kirirting');*

*end;*

*if n<1 then begin*

*showmessage('Natural son kirirting');*

*exit*

*end*

*else*

*begin*

*fak := 1;*

*for i := 1 to n do begin*

*if fak>(2147483647 div i) then*

*begin*

*showmessage ('Bunday katta sonni yoza olmayman');*

*exit*

*end*

*else begin*

*fak := fak\*i;*

*label1.caption := inttostr(fak);*

*end;*

*end;*

*end;*

*end;*

Бу матнда *2147483647* сони *longword* типидаги энг катта сон.

Қиймати бутун сонлар диапазонидан четга чиқадиган маълумотлар билан ишлашга эҳтиёж пайдо бўлган, берилган масалани ҳал учун (масалан катта соннинг факториалини ҳисоблаш учун) сатрли маълумотлар билан ишлаш усулларидан фойдаланишни тавсия этиш мумкин.

Ҳақиқий типдаги маълумотлар билан ишлаш бутун сонларга қараганда анча содда. Бунинг сабаби шуки, биринчидан, киритилган ихтиёрий сонни ҳақиқий сон сифатида қараш мумкин; иккинчидан, 2-жадвалда кўрсатилган ҳақиқий типдаги ихтиёрий маълумот ўзи мансуб бўлган типнинг юқори диапазонидан четга чиқса, бу ҳақда

*Floating point overflow*

кўринишидаги ахборот чиқарилади. Қуйи диапазондан четга чиқадиган маълумотлар ўрнига эса нол қўйилади.

Бу ҳолда ҳам, зарур бўлса, дастурни ҳақиқий ўрнига ҳақиқий бўлмаган маълумотларни киритишдан ҳимоялаш учун econverterror классидан фойдаланиш мумкин.

Экспоненциал кўринишдаги сонларни киритишда эса соннинг мантиссаси учун битта майдон, тартиби учун иккинчи майдон ташкил қилингани маъқул. Шундан кейин бу майдонлардаги маълумотларни

*c := edit1.text+'e-'+edit2.Text; a := strtofloat(c);*

тарзида экспоненциал кўринишдаги сонга айлантириш мумкин.

Белгили ва сатрли маълумотлар билан ишлаганда одатда, муаммолар кўп бўлмайди. Паскал тилида сатрли катталикларнинг узунлиги 256 тагача бўлиши мумкин эди. DELPHI тилида эса сатрли катталикларнинг узунлиги етарлича катта бўлиши мумкин. Чунки, DELPHI даги *shortstring, longstring* ва *widestring* каби сатрли типлар учун хотирадан статик ва динамик тарзда жой ажратилиши мумкин. Бу типлар бир-биридан ўзгарувчилар учун хотирадан жой ажратиш, белгиларнинг кодлаштириш усуллари щамда сатрларнинг узунлиги билан фарқланади. Сатрли ўзгарувчиларга статик жой ажратиш фақат дастур ишга туширилмасдан аввал содир бўлади. Дастур ишга туширилганидан ўзгарувчиларга хотирадан жой динамик тарзда ажратилади. Демак, старли маълумотлар учун ажратиладиган жой амалий жиҳатдан компьютернинг бўш хотираси билан чегараланади. Сатрли маълумотлар билан ишлаганда универсал *String* сатрли типи билан ишлашни тавсия этамиз.

Жуда ҳам узун бўлган сатрли катталикларни дастур ёрдамида қайта ишлашга фақат назарий жиҳатдан тўғри келиши мумкин, аммо амалиётда бундай масалалар жуда кам учрайди. Шунинг учун сатрли катталикларнинг узунлиги билан боғлиқ муаммони четга суриб қўйиш мумкин. DELPHI тилида консолли иловалар яратгиш имконияти мавжуд. Аммо, шуни ёдда тутиш керакки, консолли иловаларни Windows муҳитида яратилади ва бу иловалар одатда DOS дастури каби ишлатилади. Биламизки, DOS системасида ASCII кодлаш усули, Windows да эса ANSI усули қўлланади. Бу кодлаш усулларида кирилл алифбесининг харфлари турли кодларга эга. Шунинг сабабли, консолли иловалар дастури матнидаги кирилл алифбесида ёзилган изоҳлар ўрнига бошқа матн экран чиқарилади. Демак, консолли иловаларда турли матнларни чиқаришга зарурат бўлса, бу матнларни лотин алифбесида чиқариш тавсия қилинади.

Агар консолли иловаларда рус алибесидаги ахборотларни экранга чиқариш зарур бўлса, дастур матнида ANSI-сатрни ASCII-сатрига ўтказувчи қайта кодлаш амалини қўшимча ташкил қилишга тўғри келади. ANSI кодлаш усулида кириллча ҳарфлар 192 дан 255 гача кодланиши ҳамда ASCII да - 128 дан 175 гача (А..Яа..п) ва 224 дан 239 гача (р..я) кодланишини ҳисобга олсак, дастур матнига қуйидаги буйруқларни қўшиб қўйиш тавсия қилинади.

*for i := 1 to length(matn) do*

*case matn[i] of*

*'А'..'п' : matn[i] := Chr(Ord(matn[i]) - 64);*

*'р'..'я' : matn[i] := Chr (Ord(matn[i] ) -16);*

*end;*

Биз юқорида фақат энг кўп қўлланадиган бутун, ҳақиқий ва сатрли типларга оид типик маълумотларни келтирдик ҳалос. Қолган типлар билан DELPHI тилида назарий ёки амалий машғулотлар вақтида эҳтиёжга қараб талабаларни таништириш мумкин.