

Основни типови података

У делу који се односи на идентификаторе неколико пута су поменути типови података. Типови податка су уствари особине које описују неки података. У зависности од типа податка мењају се:

1. вредности коју може да има неки податак
2. операције које се могу извршити над тим податком
3. начин представљања податка у меморији.

Типови података могу бити прости и сложени. Сложени типови састоје се од више простих типова.

У програмском језику C++ у основне (просте) типове податка спадају:

1. знаковни подаци (char)
2. цели бројеви (int, long int)
3. реални бројеви (float, double, long double)
4. набројиви тип (enum)
5. празан тип (void).

char – представља слово алфабета или неки специјални знак

int – представља цео број (4, 343, 59823)

long int – представља цео број али велике дужине

float – претстављају реалне бројеве (децимални зарез је обележен тачком 4.15, 65.879)

double – представљају реалне бројеве двоструке прецизности

long double – предстаљају реалне бројеве проширене прецизности

Као посебна група целобројних података јавља је тип **bool** где постоји аутоматска конверзија између типа **bool** и нумеричких типова. Када се користи у изразима логичка вредност **false** се претвара у целобројну вредност нула, а **true** у један. Ако се променљивој типа **bool** додељује нумеричка вредност, вредност нула се претвара у **false**, а било која ненулта вредност у **true**.

Оно што је изузетно битно је да или на почетку сваког програма или кад први пут користи неку променљиву програмер мора да декларише тип те променљиве. За сваку променљиву корисник мора да одреди тип и да је на тај начин обележи. Дакле, програмер мора да води рачуна о томе да уколико врши дељење неког броја или његово кореновање вероватно ће добити реални број. Па према томе ће да одреди који тип променљиве ће дефинисати за ту променљиву.

НАПОМЕНА: програмски језик C++ прави грешку у томе што приликом дељења два цела броја даје као резултат цео број без остатка, па се препоручује да се бројеви који се деле увек декларишу као реални. То није случај са рецимо кореновањем или множењем једног реалног и једног целог броја.

НАПОМЕНА: мора да се дефинише свака променљива у програму. То се односи не само на улазне податке, него и на излазне и на помоћне променљиве у програму.

Декларација променљивих врши се на следећи начин:

tip promenljive imePromenljive;

наравно ако постоје више променљивих истог типа све могу да се наведу у једном реду или за сваку променљиву да се пише тип:

tip promenljiva1, prom_2, prom_n;

tip promenljiva1;

tip prom_2;

tip prom_n;

оно што је важно је да се наведу све променљиве и да се сваки ред заврши са тачка-зарез (;).

Треба нагласити и то да постоји могућност **иницијализације променљиве**. Иницијализација подразумева доделу поченте вредности тој променљивој. То се ради на следећи начин:

tip promenljiva1 = neka vrednost;

рецимо:

int s=0, a, b;

у овом реду декларисали смо три променљиве и све три су истог типа тј. цели бројеви. Само променљива s има почетну вредност која је нула.

int a=0, b=0, c, d;

овог пута декларисане су четири променљиве а две, а и б имају почетну вредност нула.