Приведение типов

Не путать с привидением!!!

Преобразование типов.

Преобразование (приведение) типов

- Термин приведение – от слова прив**Е**сти, т.е. прив**е**сти один тип данных к другому типу данных.

- Привидение – от слова прив**И**деться, т.е. показаться, не стоит путать два понятия.

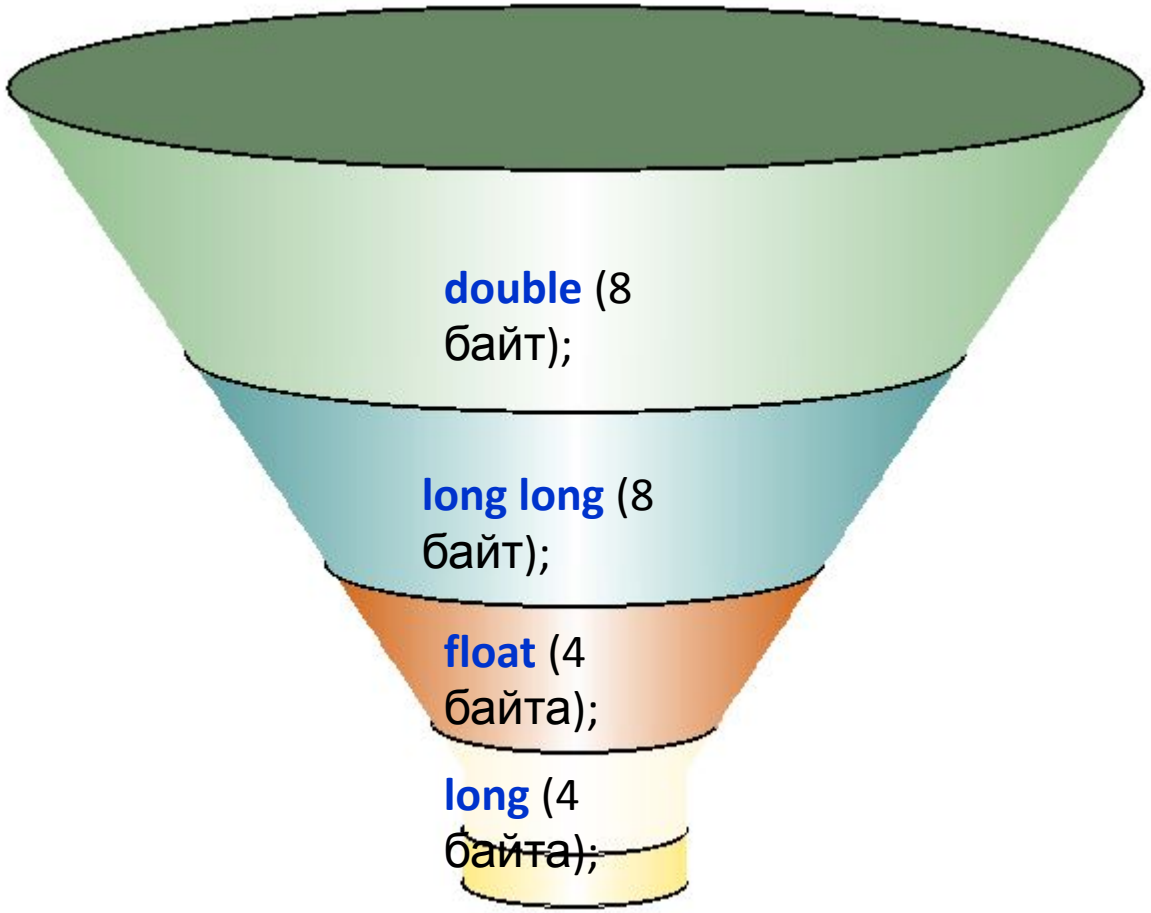
Преобразование (приведение) типов

- Существует так называемая иерархия типов, где все типы размещены по старшинству.
- Для того, что бы разбираться в преобразовании типов, необходимо всегда помнить порядок типов этой иерархии.
- Термин приведение – от слова привЕсти, т.е. привести один тип данных к другому типу.
- Привидение – от слова привИдиться, т.е. показаться, не стоит путать два понятия.

Иерархия типов

- Порядок типов этой иерархии.
- **double** (8 байт);
- **long long** (8 байт);
- **float** (4 байта);
- **long** (4 байта);
- **int** (4 байта);
- **short** (2 байта);
- **char** (1 байт);
- **bool** (1 байт);

Преобразование (приведение) типов



Сужающее преобразование

- **Сужающее преобразование** — при таком преобразовании — больший тип данных в иерархии преобразуется к меньшему типу, безусловно, в этом случае может произойти потеря данных, поэтому с сужающим преобразованием, следует быть осторожными.
- Например:

```
int A=23.5;
cout<<A; // на экране 23
```

Сужающее преобразование

Расширяющее преобразование

- **Расширяющее преобразование** - ведет к так называемому расширению типа данных от меньшего диапазона значений к большему диапазону

Расширяющее преобразование

Виды приведения

- **Неявное преобразование.**
- Все вышеописанные примеры относились к этому типу преобразования.
- Такой вид преобразования также называют автоматическим, т.к. оно происходит автоматически без вмешательства программиста, другими словами, мы ничего не делаем для того, что бы оно произошло.

Явное преобразование

- **Явное преобразование** (приведение типов).
- В данном случае, преобразование производится программистом, тогда, когда это необходимо.
- double z=37.4;
- float **y=z;**
- cout<<z<<"*** "<<y;

Явное преобразование

- **Явное преобразование** (приведение типов).
- В данном случае, преобразование производится программистом, тогда, когда это необходимо.
- double z=37.4;
- float **y=(int) z;**
- cout<<z<<"*** "<<y; *// на экране 37.4 ***37*

Пример

```
int testA = 4, testB = 8;
float testResult;
testResult = testA / testB;
cout<< testResult <<endl;
```

Результат получился не корректный, так как результат выражения всегда приводится к самому старшему типу используемому в выражении, а данном случае, к типу int, в этом случае дробная часть была отброшена и остался только 0.

Пример

Для получения корректного результата, необходимо одну из

переменных в выражении явно привести к типу float:

```
int testA = 4, testB = 8;
float testResult;
testResult = (float)testA / testB;
cout<< testResult <<endl;
```

Преобразование типов в выражении

- Если в каком-либо выражении используются разные типы данных, то результат, приводится к большему из этих типов.
- int I=27;
- short S=2;
- float F=22.3;
- bool B=false;
- float res= I-F+S*B; *// 27-22.3+2*0*
- cout<<res; *// на экране число 4.7*

```
#include <iostream>
using namespace std;
void main(){
 // объявление переменных и запрос на ввод данных
    float digit;
    cout<<"Enter digit:";
    cin>>digit;
 /* Даже, если пользователь ввел число
 с вещественной частью, результат выражения
 запишется в int и вещественная часть будет
 утеряна, разделив число на 100 мы получим
 количество сотен в нем.*/
    int res=digit/100;
    cout<<res<<« сотен в вашем числе!!!\n\n";
}
```