

Семинар по „Увод в програмирането“

1. Променливи и примитивни типове данни

1) Примитивни типове данни:

- Целичислен тип данни:
 - short, int, long
- Реални числа (числа с плаваща запетая)
 - double, float
- Символен тип:
 - char
- Булев тип:
 - bool

2) Какво е променлива?

3) Декларация на променлива:

<тип данни> <име на променливата> = <стойност>;

4) Дефиниция на променлива

```
int data = 15;
```

5) Литерали: 1, 2, 's', 12.7

6) Размери (в байтове) на основните примитивни типове данни:

Тип данни	Предназначение	Размер(в байтове)
int	Цели числа	4
double	Дробни числа	8
char	Символи	1
bool	Условен тип	1

2. Преобразуване на данни

1) Имплицитно преобразуване (Implicit Type Conversion)

Имплицитно преобразуване на данни е преобразуване на данни, което се извършва от компилатора.

Пример №1:

```
1  #include <iostream>
2
3  int main()
4  {
5      int intNum = 10;
6      double doubleNum;
7
8      doubleNum = intNum;
9
10     std::cout << doubleNum; //10
11
12     return 0;
13 }
```

Пример №2:

```
1  #include <iostream>
2
3  int main()
4  {
5      bool a = true;
6      int justVariable;
7
8      justVariable = a;
9
10     std::cout << justVariable; //1
11
12     return 0;
13 }
```

2) Експлицитно преобразуване (Explicit Type Conversion)

Експлицитното преобразуване е преобразуване на един тип данни към друг, което се случва ръчно от програмиста.

Пример №3:

```
1  #include <iostream>
2
3  int main()
4  {
5      double pi = 3.14;
6      int intValue;
7
8      intValue = (int)pi;
9
10     std::cout << intValue; //3
11
12     return 0;
13 }
```

Преобразуванията могат да бъдат или без загуба на информация, или със загуба на информация. Тези без загуба на информация могат да бъдат като пример **№1** и **№2**, а тази със загуба на информация като пример **№3**.

3. Оператори

Типове оператори:

- 1) Аритметични: +, -, /, *, %, ++, --
- 2) Оператори за присвояване: =, +=, -=, *=, /=, %=
- 3) Оператори за сравнение: >, <, >=, <=, !=
- 4) Логически оператори: &&, ||, !

4. Основни потоци за вход и изход

1) Какво представляват потоците накратко?

2) Входен поток(**Input Stream**)

Процес, при който посоката на байтовете информация е от стандартния вход, чрез клавиатура, към нашата програма.

```
1      #include <iostream>
2
3      int main()
4      {
5          int value;
6
7          std::cin >> value;
8
9          return 0;
10     }
```

3) Изходен поток(**Output Stream**)

Тук посоката на байтовете информация е обратна. От нашата програма към стандартния изход, екран.

```
1      #include <iostream>
2
3      int main()
4      {
5          int value = 5;
6
7          std::cout << value;
8
9          return 0;
10     }
```