Оператори за вход и изход. Оператори за присвояване

При изпълнение на програма на С, автоматично се отварят три стандартни потока и те са достъпни за използване. Тези потоци се наричат: стандартен вход - stdin (standard input), стандартен изход - stdout (standard output) и стандартна грешка - stderr (standard error). По подразбиране те са свързани с конзолата – клавиатурата и екрана (файлове с последователен достъп за вход/изход на текстови данни). Поддържат се вътрешно от компилатора и не могат да бъдат променяни. Подобни по предназначение са съответно потоците cin, cout и cerr (от хедърния файл iostream.h), които са валидни и се поддържат само за C++ програмите. Стандартните библиотечни функции за вход/изход на данни, включени в хедърните файлове stdio.h и conio.h, автоматично оперират със stdin и stdout, като осигуряват значителни потребителски удобства и ефективност при съставяне на програмите с конзолен обмен на данни.

1. **Оператор за вход**

   a. Синтаксис

   cin>><променлива>;

   cin – стандартна дума на езика, дефинирана в библиотечния файл iostream.h

   >> - оператор за преместване на данни

   b. Семантика – данните се въвеждат в съответното им място в ОП

   cin>>a;
   cin>>b; ⟺ **cin>>a>>b;**

2. **Оператор за изход**

   1. **Синтаксис**

   cout<< <променлива или израз>;

   cout – стандартна дума на езика, дефинирана в библиотечния файл iostream.h

   << - оператор за преместване на данни

2. Семантика – данните се извеждат на изходно устройство. По подразбиране това устройство е екрана

cout<<a;
cout<<b; ⟺ **cout<<a<<b;**

3.Форматиране на входа и изхода

a. Форматиране на входа – прегледност на въвежданите данни – оператор за изход на подходящо съобщение

b. Форматиране на изхода – манипулатори за форматиране на изхода

- **setw**
  - **синтаксис**

setw (<цял израз>);

  - **семантика**

задава брой позиции на следващото извеждане, подравнено отляво

Пример:

primer.cpp

```
#include <iostream.h>
#include <conio.h>
#include <iomanip.h>
int main()
{
    cout<<215<<'\n';
    cout<<setw(5)<<215<<'\n';
    cout<<setw(8)<<215<<'\n';
    getch();
    return 0;
}
```

C:\Users\Банкова\Desktop\primer.exe

```
215
  215
     215
```

- **setprecision**
  - **синтаксис**

setprecision (<цял израз>)<< setiosflags(ios::fixed);

  - **семантика**

задава броя цифри след десетична точка, с които ще бъде изведено следващото реално число, като прави закръгление.

Би могло да се форматира и самото число

hex //извежда числото в шестнадесетична бройна система

dec // извежда числото в десетична бройна система

oct // извежда числото в осмична бройна система

primer.cpp

```
#include <iostream.h>
#include <conio.h>
#include <iomanip.h>
int main()
{
    double b=215.2345;
    int a=10;
    cout<<215<<'\n';
    cout<<setprecision(2)<<setiosflags(ios::fixed)<<b<<'\n';
    cout<<dec<<215<<'\n';
    cout<<hex<<32<<'\n';
    cout<<oct<<8<<'\n';

    getch();
    return 0;
}
```

C:\Users\Банкова\Desktop\primer.exe

```
215
215.23
215
20
10
```

4. Оператор за присвояване

a. Синтаксис

<променлива> = <израз>;

където

- <променлива> е идентификатор, дефиниран вече като променлива,
- <израз> е израз от тип, съвместим с типа на <променлива>.

b. Семантика - Намира се стойността на <израз>. Ако тя е от тип, различен от типа на <променлива>, конвертира се, ако е възможно, до него и се записва в именуваната с <променлива> памет. Този оператор е дясноасоциативен.

Допустим е операторът:

X = y = 5;

Отначало променливата y се свързва със 5, което е стойността на израза y = 5. След това x се свързва с 5, което е стойността на целия израз.

Примери:

```
-x = y;

m + n = p;

4 = sin(p + 5)
```

това не са оператори за присвояване

c. Забележки: величините участващи в израза могат да са от различни типове. Понякога това води до загуба на точност (компилатора предупреждава за това). Това е начинът да се осъществи неявно преобразуване на типове

Пример:

int b=3.52; //получава се загуба на точност

Ще отбележим, че в рамките на дефиницията на една функция не са възможни две дефиниции на една и съща променлива, но на една и съща променлива може да й бъдат присвоявани многократно различни стойности.