Introduction to Programming
24.11.14 - 28.11.14

Функции:

Функцията е подпрограма, която може да бъде извикана от главната програма за извършване на определена задача. Например, ако имаме код, който се повтаря на много места в нашата програма, би било добре ако го напишем във функция и я извикваме на всички места, когато ни потрябва вместо да копираме кода отново и отново. Можем да декларираме функции по-същия начин както декларираме и главната функция. Пример за функция, отпечатваща "Hello":

```
1.  void PrintHello()
2.  {
3.      cout << "Hello" << endl;
4.  }
```

<u>1задача</u>: *Напишете функция която отпечатва текста “Hello FMI!” на конзолата.*

От какъв тип ще декларираме функцията, зависи от това какъв тип данни ще връща функцията след изпълнението си. В примера използваме тип void, защото НЕ желаем функцията да връща никаква стойност. Ето как можем да извикаме нашата функция от главната main() функция:

```
1.  int main()
2.  {
3.      PrintHello();
4.      return 0;
5.  }
```

Забележете, че main() функцията е от тип int, защото в края на изпълнението и връшаме цяло число “return 0”. Във лекциите на доц. Павлов главната - main() функция е от тип void, защото в края на изпълнението си не връща стойност, т.е. не се използва “return 0”.

След като можем да правим програми с повече от една функция е хубаво да споменем, че променливите се делят на локални и глобални. Локални са тези променливи, които сме декларирали в някакъв блок “{ }” и са видими само за този блок, глобалните са декларирани извън блока на функциите и са видими за всяка една функция. Не прекалявайте с декларирането на такива променливи!
Пример:

```
1.  int a; // глобална променлива
2.
3.  void MyFunction()
4.  {
5.      int b; // локална променлива
6.      b = 5;
7.      a = 7;
8.  }
9.
```

```
10.  int main()
11.  {
12.     a = 3;
13.     return 0;
14.  }
```

За да прехвърляме стойности между функциите трябва да използваме параметри. Те се дефинират между скобите, които са след името на функцията. Трябва да дефинираме типа данни за всеки параметър поотделно. Това са стойностите с които след това ще викаме функцията, т.е. тя един вид се характеризира с тези променливи. Отново не прекалявайте с променливите, не слагайте променливи които няма да използвате в конкретната функция! Сега ще направим функция, която получава число като параметър и го отпечатва в конзолата:

```
1.   void PrintNumber(int n)
2.   {
3.      cout << n << endl;
4.   }
5.   int main()
6.   {
7.      PrintNumber(6);
8.      return 0;
9.   }
```

Забележете как извикваме функцията от главната main() функция - PrintNumber(6) това означава, че нашата фунция ще бъде извикана с числото 6, това ще бъде и числото, което ще се отпечата в конзолата. Ако искаме да прехвърляме повече от един параметър, трябва да ги разделяме със запетая:

```
1.   void PrintNumber(int n, int m)
2.   {
3.      cout << n << m << endl;
4.   }
5.   int main()
6.   {
7.      PrintNumber(6, 5);
8.      return 0;
9.   }
```

<u>2 Задача:</u> *Напишете функция, която да приема стойностите на две числа от тип double и да връща сбора на двете числа. Например при извикване на вашата функция от main() функцията по следния начин: sum(5.6, 10.1), функцията ви трябва да върне стойност 15.7.*

Досега разглеждахме примери само с void функции, тоест фунции, които не връщат стойност към функцията, която ги е извикала. Нека разгледаме една функция, която връща цели числа (тип int):

```
1.  int GetNumber()
2.  {
3.     return 10; // стойности се връщат с помощта на ключовата дума return
4.  }
5.
6.  int main()
7.  {
8.     int i = GetNumber(); // стойността, която функцията връща се присвоява на
    променливата i
9.     return 0;
10. }
```

След изпълнението на този код стойността на променливата i ще бъде 10. Може да проверите това като я отпечатате.

А сега да приложим наученото в този урок, като направим програма, която пресмята квадрата на число въведено от клавиатурата. Първо пробвайте да я направите сами ☺:

```
1.  #include <iostream>
2.  using namespace std;
3.
4.  double func(double n){
5.      return n*n;
6.  }
7.
8.  int main()
9.  {
10.     double a;
11.     cout<<"a = ";
12.     cin>>a;
13.     cout<<"a^2 = "<<func(a);
14.     return 0;
15. }
```

Задачи:

1. *Напишете функция, която према стойноста на едно цяло n число и пресмята стойноста фактуриел от това число n!*
2. *Да се напише булева функция, която връща стойност true, ако параметърът й е четно число и false в противен случай.*
3. *Да се напише булева функция, която връща стойност true, ако параметърът й е просто число и false в противен случай.*
4. *Напишете функция, която приета като аргумент една буква от латинската азбука и ако е малка връща голямата, ако е голяма връща малката.*
5. *Напишете функция, която получава като аргумент едно четири цифрено число, събира цифрите му и връща буквата намираща се на това място в латинската азбука.*
6. *Да се напише функция за пресмятане на най-голям общ делител на две числа.*
7. *Да се напише функция, която приема като параметри едно число и една цифра и връща броя на срещанията на цифрата в числото.*
8. *Модифицирайте горната задача, така че вместо да връща броя на срещанията на цифрата в числото, връща дали то е просто или не.*
9. *Една вече решаване задача. Имплементирайте я използвайки функции.*
   На стандартния вход въведете число n (2 <= n <= 100) и символ c. На стандартния изход трябва да изведете, равностранен триъгълник (незапълнен), като

страните му са съставени от прочетения символ. В основата на триъгълника символите са разделени с интервали(както е показано на примера).

Вход:
*4 **
Изход:
```
* * * *
 *   *
  * *
   *
```