

Функции в C++.

Предаване на параметри. Глобални и локални променливи

Базово програмиране

Лекция 5

Милена Карова

кат. КНТ ТУ-Варна

Програмиране „отгоре-надолу“

- Проектиране на програми „отгоре-надолу“
- Работа с функции като изискване за работа в екип

Всяка програма може да се раздели на части (subparts): модул за въвеждане на данните, модул за пресмятане на данни. Ако такава част може да се програмира отделно, може да се компилира самостоятелно, то тя се нарича функция.

- Съществуват 2 вида функции:
 - Предефинирани функции (намират се в различни библиотеки на езика)
 - Потребителски функции (програмистът самостоятелно ги разработва)

Предефинирани функции

- С++ се инсталира с библиотеки, съдържащи предефинирани функции, които могат да се ползват в потребителските програми
- Аргумент
- Стойност, която функцията връща

```
the_root = sqrt(9.0);
```

```
bonus = sqrt(sales)/10;
```

```
bonus = (sqrt(sales))/10;
```

- Извикване на функцията Аргументът може да бъде константа, променлива или по-сложен израз
- Извикване на функция в оператор за извеждане

```
cout << "The side of a square with area " << area  
      << " is " << sqrt(area);
```

Пример за използване на предефинирани функции хедърен файл `math.h` и `cmath.h`

Name	Description	Type of Arguments	Type of Value Returned	Example	Value	Library Header
<code>sqrt</code>	square root	<i>double</i>	<i>double</i>	<code>sqrt(4.0)</code>	2.0	<code>cmath</code>
<code>pow</code>	powers	<i>double</i>	<i>double</i>	<code>pow(2.0,3.0)</code>	8.0	<code>cmath</code>
<code>abs</code>	absolute value for <i>int</i>	<i>int</i>	<i>int</i>	<code>abs(-7)</code> <code>abs(7)</code>	7 7	<code>cstdlib</code>
<code>labs</code>	absolute value for <i>long</i>	<i>long</i>	<i>long</i>	<code>labs(-70000)</code> <code>labs(70000)</code>	70000 70000	<code>cstdlib</code>
<code>fabs</code>	absolute value for <i>double</i>	<i>double</i>	<i>double</i>	<code>fabs(-7.5)</code> <code>fabs(7.5)</code>	7.5 7.5	<code>cmath</code>
<code>ceil</code>	ceiling (round up)	<i>double</i>	<i>double</i>	<code>ceil(3.2)</code> <code>ceil(3.9)</code>	4.0 4.0	<code>cmath</code>
<code>floor</code>	floor (round down)	<i>double</i>	<i>double</i>	<code>floor(3.2)</code> <code>floor(3.9)</code>	3.0 3.0	<code>cmath</code>

• `abs` и `labs` са функции на `cstdlib.h`

Функция `pow`. Typecasting

- Функцията `pow()` работи с аргументи от тип `float` или `double`.
- Ако се налага преобразуване от един тип в друг използвайте typecasting

```
double result, x = 3.0, y = 2.0;  
result = pow(x, y);  
cout << result;
```

```
double answer;  
answer = static_cast<double>(9)/2;
```

```
candy_per_person =  
    static_cast<double>(total_candy)/number_of_people;
```

```
candy_per_person =  
    double(total_candy)/number_of_people;
```

- Могат да се използват стари форми на typecasting.

Програмно дефинирани функции (потребителски функции)

- Дефиниции на функции
- Прототипи на функции

```
double total_cost(int number_par, double price_par);
```

- Видове аргументи: формални и фактически
- Тип на връщаната стойност
- Възможно е в прототипа, тъй като не се знаят всички аргументи да се поставят само типовете на параметрите без техните имена

Анатомия на функцията

```
#include <iostream>
using namespace std;

double total_cost(int number_par, double price_par);
//Computes the total cost, including 5% sales tax,
//on number_par items at a cost of price_par each.

int main()
{
    double price, bill;
    int number;

    cout << "Enter the number of items purchased: ";
    cin >> number;
    cout << "Enter the price per item $";
    cin >> price;

    bill = total_cost(number, price);

    cout.setf(ios::fixed);
    cout.setf(ios::showpoint);
    cout.precision(2);
    cout << number << " items at "
         << "$" << price << " each.\n"
         << "Final bill, including tax, is $" << bill
         << endl;

    return 0;
}

double total_cost(int number_par, double price_par)
{
    const double TAX_RATE = 0.05; //5% sales tax
    double subtotal;

    subtotal = price_par * number_par;
    return (subtotal + subtotal*TAX_RATE);
}
```

function declaration

function call

function heading

function body

function definition

Дефиниция на функция
Заглавен ред на
функцията
Тяло на функцията
Оператор return
Извикване на функцията

```
Enter the number of items purchased: 2
Enter the price per item: $10.10
2 items at $10.10 each.
Final bill, including tax, is $21.21
```

```
return (subtotal + subtotal*TAX_RATE);
```

```
return subtotal + subtotal*TAX_RATE;
```

```
bill = total_cost(number, price);
```

Предаване на параметри – по стойност и по адрес

- Предаване на параметрите по стойност

```
#include<iostream>
using namespace std;

int suma(int x, int y);
void main()
{   int a(5),b(6);
    int rez;
    rez=suma(a,b);
    cout<<a;
    cout<<endl<<rez;
}
int suma(int x, int y)
{ int s=x+y;
  x++;
return s;
}
```

Предаване на параметрите по адрес

```
#include<iostream>
using namespace std;

int suma(int &x, int y);
void main()
{   int a(5),b(6);
    int rez;
    rez=suma(a,b);
    cout<<a;
    cout<<endl<<rez;
}
int suma(int &x, int y)
{ int s=x+y;
  x++;
return s;
}
```


Предефиниране на функции

```
#include <iostream>
```

```
double ave(double n1, double n2);  
//Returns the average of the two numbers n1 and n2.
```

```
double ave(double n1, double n2, double n3);  
//Returns the average of the three numbers n1, n2, and n3.
```

```
int main()  
{  
    using namespace std;  
    cout << "The average of 2.0, 2.5, and 3.0 is "  
          << ave(2.0, 2.5, 3.0) << endl;  
  
    cout << "The average of 4.5 and 5.5 is "  
          << ave(4.5, 5.5) << endl;  
  
    return 0;  
}
```

```
double ave(double n1, double n2)  
{  
    return ((n1 + n2)/2.0);  
}
```

```
double ave(double n1, double n2, double n3)  
{  
    return ((n1 + n2 + n3)/3.0);  
}
```

```
The average of 2.0, 2.5, and 3.0 is 2.50000  
The average of 4.5 and 5.5 is 5.00000
```

Ако програмата съдържа функции с едно и също име, тогава говорим за предефиниране на функции (overloading).

Това означава функциите с едни и същи имена да имат различен брой формални параметри или формални параметри от различен тип.

Задача

- В една печатница съществува следната система за заплащане. За черно-бяло отпечатване всеки лист хартия струва 0,01 лв. За всяка страница се прави плака, струваща 7 лв. При цветен печат – всяка страница струва 0,04 лв. и плаката струва 28 лева. Подвързването струва 2 лева на книга. Към тази цена трябва да се добави 20% ДДС. Отпечатването е двустранно.
- Пример за черно-бели 400 книги, всяка от по 100 страници:
- $100 \cdot 7 \text{ лв} = 700 \text{ лв}$ за плаки
- $(100/2) \cdot 400 \cdot 0,01 = 200 \text{ лв}$ за хартия от по 2 страници на лист
- $400 \cdot 2 \text{ лв.} = 800 \text{ лв}$ за подвързване
- Общо: 1700 без ДДС 2040 лв. с ДДС

Задача

- Намерете решение на задачата без ползването на глобални променливи, а само предаване на параметри. Напишете 3 функции:
- Функция, която намира цената на книга при дадени брой страници, цена на печатен лист, цена на плака и брой книги.
- Функция, която изчислява цената на поръчка при подадени брой книги и брой страници за черно-бял печат. Тази функция трябва да вика първата функция.
- Функция, която изчислява цената на поръчка при подадени брой книги и брой страници за цветен печат. Тази функция трябва да вика първата функция.
- Функция `main()`, която изчислява крайната цена за отпечатването на 1000 цветни книги с по 32 страници всяка, 2000 черно бели книги с 40 страници всяка и 400 черно-бели книги – всяка с по 160 страници. Функция `main()` трябва да вика функции 2 и 3.

Решение на задачата

```
#include<iostream>
using namespace std;
double pricing (int num,double sh, int pl,int cop); //намиране цената на поръчката
double prblack_white(int pages, int copies); //намиране цената за черно-бял печат
double prcolor(int pages,int copies);//намиране цената за цветен печат
int main()
{
    double total_price=0;
    total_price=prcolor(32,1000)+prblack_white(40,2000)+prblack_white(160,400);
    cout<<endl<<"obshto:"<<total_price<<endl;

    system("pause");
    return 0;
}

double prcolor(int pages,int copies)
{ double pricecolor=pricing(pages,0.04,28,copies);
  return pricecolor;
}

double prblack_white(int pages,int copies)
{ double pricebw=pricing(pages,0.01,7,copies);
  return pricebw;
}

double pricing (int num,double sh, int pl,int cop)
{
    int bin=2; //цена за подвързване
    double vat=20.0/1000.0; // ДДС
    int sheets; //листи хартия
    sheets=(num+1)/2; //ако броят на страниците е нечетен
    double q=((sheets*cop*sh)+(pl*num)+(cop*bin));
    return (q*(1+vat));
}
```