

Занятие 6 — Ссылки. Функции

Программирование, численные методы и информатика

А. В. Позднеев

Кафедра автоматизации научных исследований
Факультет вычислительной математики и кибернетики
Московский государственный университет имени М. В. Ломоносова
<http://ani.cmc.msu.ru/geol>

Осенний семестр 2010/2011



Содержание занятия

1 Ссылки

- Понятие ссылки
- Модификатор `const`

2 Функции

- Определение функции в C++
- Передача параметров по ссылке
- Указатели на функции

Понятие ссылки

- ▶ Ссылка — синоним имени объекта (другое имя уже существующего объекта)
- ▶ тип `&имя_ссылки` = инициализатор;
- ▶ Инициализатором должно быть имя некоторого объекта, имеющего место в памяти
- ▶ Функционально ссылка ведет себя подобно обычной переменной

Ссылки — пример

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

int main()
{
    int i = 42;
    int &ref_i = i;

    ++i;
    cout << "i: " << i << endl;
    cout << "ref_i: " << ref_i << endl;

    ++ref_i;
    cout << "i: " << i << endl;
    cout << "ref_i: " << ref_i << endl;

    system("pause");
    return 0;
}
```

Модификатор const

Переменные-константы

- ▶ Модификатор `const` перед именем типа сообщает компилятору, что объект нельзя модифицировать
- ▶ `const` тип имя_переменной = инициализатор;
- ▶ `const double pi = 4.0 * atan(1.0);`

Ссылки-константы

- ▶ По константной ссылке нельзя модифицировать объект, на который она указывает
- ▶ `const` тип `&имя_ссылки` = инициализатор;
- ▶ `int num_students = 6;`
`const int &ref = num_students;`
`++ref;` // компилятор выдаст ошибку!

Функции — определение функции в C++

```
тип_функции имя_функции(спецификация_формальных_параметров)
{ тело_функции }
```

- ▶ тип_функции — тип возвращаемого значения (void — если функция ничего не возвращает)
- ▶ имя_функции — идентификатор
- ▶ спецификация_формальных_параметров
 - ▶ пустой, либо void
 - ▶ список спецификаций отдельных параметров, отделенных запятыми
тип_1 имя_параметра_1, тип_2 имя_параметра_2
- ▶ возврат в точку вызова
 - ▶ return; // если функция имеет тип void
 - ▶ return выражение;

Примеры функций

```
int Min(int a, int b)
{
    if (a < b)
        return a;
    else
        return b;
}

double Cube(double a)
{
    return a * a * a;
}

int main()
{
    cout << Min(1, 2) << endl;
    cout << Cube(3.0) << endl;
    system("pause");
    return 0;
}
```

Упражнение

- ▶ Напишите функцию `double Square(double x)`, вычисляющую квадрат числа.
- ▶ С ее помощью напишите функцию `double Dist(double x1, double y1, double x2, double y2)`, вычисляющую расстояние между двумя точками (x_1, y_1) и (x_2, y_2) на координатной плоскости. (Воспользуйтесь стандартной функцией `sqrt()` из модуля `cmath` для вычисления квадратного корня.)
- ▶ Выведите на экране значение расстояние между точками
 - ▶ $(0, 0)$ и $(0, 1)$
 - ▶ $(0, 0)$ и $(1, 0)$
 - ▶ $(1, 2)$ и $(4, 6)$
- ▶ Определите квадрат расстояния между точками $(0, 0)$ и $(1, 1)$.

Передача параметров по ссылке

```
bool Root2(
    double a, double b, double c,
    double &x1, double &x2)
{
    double D = b * b - 4 * a * c;
    if (D < 0.0) return false;

    x1 = (-b - sqrt(D)) / (2*a);
    x2 = (-b + sqrt(D)) / (2*a);
    return true;
}

int main()
{
    double x1, x2;
    if (Root2(1.0, 5.0, 6.0, x1, x2))
        cout << "x1: " << x1 << endl << "x2: " << x2 << endl;
    else
        cout << "No roots\n";
    system("pause");
}
```

Упражнения

Напишите функцию `Swap()`, аргументами которой являются два целых числа. Функция меняет значения своих аргументов и не возвращает никакого значения. В функции `main()` проверьте, правильность работы функция `Swap()`.

Указатели на функции

Функции можно передавать в качестве аргументов функций

```
typedef int Fun_t(int x);

void Print(Fun_t f)
{
    for (int i = 0; i <= 4; ++i)
        cout << "i: " << i << " f: " << f(i) << endl;
}

int Square(int i) { return i*i; }
int Cube(int i) { return i*i*i; }

int main()
{
    Print(Square);
    Print(Cube);
    system("pause");
    return 0;
}
```