Занятие 7 — Динамические массивы и указатели. Рекурсивные функции. Строки

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 2009/2010 уч. г.

Содержание занятия

Динамические массивы и указатели — общие понятия

Динамическая память

- Размер массива не известен на момент компиляции
- Он задается на этапе выполнения
- *Динамическое распределение памяти*

Указатели

- Переменная типа *указатель* хранит адрес, по которому расположен массив или какая-то другая переменная
- `имя_типа *идентификатор`
- `int *a;`
- `double *x, *y;`
- Сразу после объявления значение указателя не определено

Выделение памяти

Для создания массива в динамической памяти используется оператор `new` с указанием размера создаваемого массива:

```
double *a;
a = new double[N];
```

Для уничтожения массива (освобождения занимаемой им памяти) служит оператор `delete []`:

```
delete [] a;
```

Динамическая память — пример

```
int main() {
    int N;

    cout << "Enter N: ";
    cin >> N;

    double *a;
    a = new double[N];

    for (int i = 0; i < N; ++i)
        a[i] = rand() / (double)RAND_MAX - 0.5;

    for (int i = 0; i < N; ++i)
        cout << i << "\t" << a[i] << endl;

    delete [] a;

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

Рекурсивные функции

РЕКУРСИЯ (мат.) — см. рекурсия

- Функция внутри себя может обращаться к другой функции
- Но функция также может вызывать и саму себя

Рассмотрим определение факториала:

$$n! = \underbrace{1 \times 2 \times \cdots \times (n-1)}_{(n-1)!} \times n = (n-1)! \times n$$

Таким образом,

$$n! = \begin{cases} 1, & n = 1 \\ (n-1)! \times n, & n > 1 \end{cases}$$

Например,

$$3! = (3-1)! \times 3 = 2! \times 3 = [(2-1)! \times 2] \times 3 = \\ = [1! \times 2] \times 3 = [(1) \times 2] \times 3$$

Рекурсивные функции — пример

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

int Factorial(int N)
{
    if (N == 1)
        return 1;
    else
        return Factorial(N-1) * N;
}

int main() {
    const int N = 5;
    cout << N << "! = " << Factorial(N) << endl;
    system("pause");
    return 0;
}
```

Рекурсивные функции — упражнения

1. Напишите рекурсивную функцию возведения в степень, воспользовавшись свойством: $a^n = a \cdot a^{n-1}$
2. Последовательность Фибоначчи определена следующим образом: $\varphi_0 = 1$, $\varphi_1 = 1$, $\varphi_n = \varphi_{n-1} + \varphi_{n-2}$ при $n > 1$. Начало ряда Фибоначчи выглядит следующим образом: 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, ... Напишите функцию `int Phi(int n)`, которая по данному натуральному n возвращает φ_n.
3. Напишите рекурсивную функцию возведения в степень, которая работала бы и для отрицательных значений n: $a^{-n} = 1/a^n$
4. Напишите функцию быстрого возведения в степень, которая пользовалась бы следующими свойствами: $a^n = \left(a^{n/2}\right)^2$ при четном n, $a^n = a \cdot a^{n-1}$ при нечетном n
5. Для биномиальных коэффициентов (числа сочетаний из n по k) хорошо известна рекуррентная формула: $C_n^k = C_{n-1}^{k-1} + C_{n-1}^k$. Вычислите значение C_n^k, пользуясь этой формулой и учитывая, что $C_n^0 = C_n^n = 1$.

Символьный тип char

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

int main() {
    char c = 'A';
    cout << c << endl;

    while (cin >> c) {
        cout << c;
    }

    cout << endl;

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

Строки в стиле языка C

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

int main() {
    char str[] = "Hello, world";

    cout << str << endl;

    for (int i = 0; str[i] != '\0'; ++i)
        cout << str[i];

    cout << endl;

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

Строки C++ string

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

int main() {
    string s1 = "Hello";
    string s2 = "World";
    string s = s1 + ", " + s2 + "!";

    cout << s << endl;

    for (int i = 0; i < s.length(); ++i)
        cout << s[i];
    cout << endl;

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

Ввод строк C++ string

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

int main() {
    string s1, s2;

    cin >> s1;              // "Lomonosov Moscow State Universi
    cout << s1 << endl; // "Lomonosov"

    getline(cin, s2);
    cout << s2 << endl; // " Moscow State University"

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

Строки — упражнения

1. Даны две строки. Определите, совпадают ли они сравнив их посимвольно. Напишите для этого функцию `bool Compare(string S1, string S2)`
2. Напишите программу, которая по данному числу k от 1 до 120 печатает фразу «Мне k лет», меняя значение k на введенное число, а вместо слова «лет» печатая при необходимости слово «год» в правильном склонении
3. Дана строка, содержащая пробелы. Найдите, сколько в нем слов (слово — это последовательность непробельных символов, слова разделены одним пробелом, первый и последний символ строки — не пробел).
4. Дана строка, содержащая пробелы. Найдите в ней самое длинное слово, выведите на экран это слово и его длину.
5. Даны две строки. Определите, является ли первая строка подстрокой второй строки.

Дополнительные материалы — контейнер vector

```
#include <iostream>
#include <vector>
#include <algorithm>
#include <iterator> // ostream_iterator
using namespace std;

int main() {
    int N = 10;
    vector<double> a(N);

    for (int i = 0; i < N; ++i)
        a.at(i) = rand() / (double)RAND_MAX;

    sort(a.begin(), a.end());

    for (int i = 0; i < N; ++i)
        cout << i << "\t" << a[i] << endl;

    copy(a.begin(), a.end(), ostream_iterator<double>(cout, "\n"));

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

Дополнительные материалы — форматирование вывода

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

int main() {
    const int N = 10;
    const double pi = 4.0 * atan(1.0);

    cout.width(5);
    cout << N << endl;  // "   10"

    cout.width(10);
    cout.precision(3);
    cout << pi << endl; // "      3.14"

    cout.setf(ios::scientific | ios::showpos);
    cout << pi << endl; // "+3.142e+000"

    system("pause");
```

Дополнительные материалы — шаблоны

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

template <class T>
T Abs(T x) {
    if (x >= 0) return x;
    else return -x;
}

int main() {
    cout << "Abs(10):\t" << Abs(10) << endl;
    cout << "Abs(-10):\t" << Abs(-10) << endl;

    cout << "Abs(3.14):\t" << Abs(3.14) << endl;
    cout << "Abs(-3.14):\t" << Abs(-3.14) << endl;

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