

Программирование на языке C++

Лекция 2

Многомерные массивы

Александр Смаль

Многомерные встроенные массивы

- C++ позволяет определять многомерные массивы:

```
int m2d[2][3] = { {1, 2, 3}, {4, 5, 6} };  
for( size_t i = 0; i != 2; ++i ) {  
    for( size_t j = 0; j != 3; ++j ) {  
        cout << m2d[i][j] << ' ' ;  
    }  
    cout << endl;  
}
```

1 2 3
4 5 6

- Элементы m2d располагаются в памяти “по строчкам”.
- Размерность массивов может быть любой, но на практике редко используют массивы размерности > 4 .

```
int m4d[2][3][4][5] = {};
```

Динамические массивы

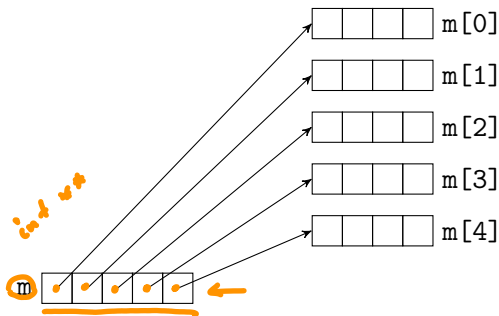
- Для выделения одномерных динамических массивов обычно используется оператор `new []`.

```
int * m1d = new int[100];
```

- Какой тип должен быть у указателя на двумерный динамический массив? $m[i] = *(m+i)$
 - Пусть m — указатель на двумерный массив типа `int`.
 - Значит m[i][j] имеет тип `int` (точнее `int &`).
 - m[i][j] \Leftrightarrow *(m[i] + j), т.е. тип m[i] — `int *`.
 - аналогично, m[i] \Leftrightarrow *(m + i), т.е. тип m — `int **`.
- Чему соответствует значение m[i]?
Это адрес строки с номером i.
- Чему соответствует значение m?
Это адрес массива с указателями на строки.

Двумерные массивы

Давайте рассмотрим создание массива 5×4 .



```
int ** m = new int * [5];  
for (size_t i = 0; i != 5; ++i)  
    m[i] = new int[4];
```

Двумерные массивы

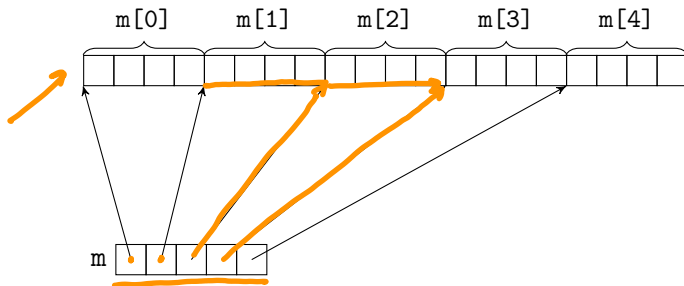
Выделение и освобождение двумерного массива размера $a \times b$.

```
→ int ** create_array2d(size_t a, size_t b) {  
    → int ** m = new int *[a];  
    |   for (size_t i = 0; i != a; ++i)  
    |       → m[i] = new int[b]; ←  
    → return m;  
}  
  
void free_array2d(int ** m, size_t a, size_t b) {  
    |   for (size_t i = 0; i != a; ++i)  
    |       delete [] m[i]; ←  
    |       delete [] m;  
}
```

При создании массива оператор new вызывается (a + 1) раз.

Двумерные массивы: эффективная схема

Рассмотрим эффективное создание массива 5×4 .



```
int ** m = new int * [5];  
m[0] = new int[5 * 4];  
for (size_t i = 1; i != 5; ++i)  
    m[i] = m[i - 1] + 4;
```

Двумерные массивы: эффективная схема

Эффективное выделение и освобождение двумерного массива размера $a \times b$.

```
int ** create_array2d(size_t a, size_t b) {  
    → int ** m = new int *[a];  
    → m[0] = new int[a * b];  
    for (size_t i = 1; i != a; ++i)  
        → m[i] = m[i - 1] + b;  
    return m;  
}  
  
void free_array2d(int ** m, size_t a, size_t b) {  
    → delete [] m[0];  
    → delete [] m;  
}
```

При создании массива оператор new вызывается 2 раза.