



Динамические переменные и массивы

Динамические переменные

- Это переменные, которые созданы в процессе выполнения программы. Для них существует функция удаления, тоже в процессе выполнения программы.
- Чтобы мы могли полноценно создавать динамические переменные, нам понадобится использовать оператор — `new`, после его выполнения в оперативной памяти компьютера выделяются ячейки на тот тип данных, который мы указали.
-

Создание динамической переменной

- Для создания динамических переменных нам понадобится применять конструкцию ниже:

```
<тип данных указателя> *<имя указателя> = new <тип данных>(<первоначальное значение>);
```

Удаление динамических переменных

- `delete <имя переменной>;`

- **<тип данных указателя>** — указанный тип данных почти ни на что не повлияет. Читайте ниже.
- **new** — это оператор, который и будет заключительным звеном для создания нашей переменной.
- **<тип данных>** — здесь нам понадобится указать тип, какой будет храниться в переменной. Он обязательно должен совпадать с типом указателя.
- **<первоначальное значение>** — с помощью круглых скобок можно указать значение переменной еще при ее инициализации. Использование круглых скобок в этой конструкции обязательно.

Пример использования динамических переменных

```
1  #include <iostream>
2
3  using namespace std;
4
5  int main() {
6      setlocale(0, "");
7      int *a = new int;
8
9      int b = 10;
10
11     *a = b;
12
13     cout << "Теперь переменная a равна " << *a << endl;
14
15     cout << "Пришло время удалить эту переменную!";
16     delete a;
17
18     system("pause");
19     return 0;
20 }
```

- **строке 7:** мы объявили переменную, оперируя конструктором `new`.
- **Дальше в строке 11:** значение нашей переменной становится равно 10.
- **И в самом конце, в строке 15:** выводим значение нашей переменной на экран.

Важно помнить! Динамические переменные — это указатели, и поэтому перед ними обязательно должен стоять оператор `*`.

Динамический массив

Это массив, у которого размер можно задавать и переменной, и числовой константой а процессе выполнения программы

<тип данных> *<имя массива> = new <тип переменных> [<количество ячеек>];

- <тип данных> — без разницы какой тип данных тут будет находиться, но лучше тот, который будет совпадать с типом переменных.
- <тип переменных> — указанный сюда тип и будут иметь ячейки массива.
- <количество ячеек> — здесь мы задаем размер массива

Динамический массив полностью идентичен обычному массиву, кроме:

- Своей инициализации
- Возможностью своевременно освободить память.
- Для удаления динамического массива нам понадобится уже знакомый оператор — `delete`.

```
delete [] <имя массива>;
```

Пример с использованием динамического массива

```
1 int main() {
2     setlocale(0, "");
3
4     int n;
5
6     cout << "Введите количество чисел, которое вы хотите ввести: ";
7     cin >> n;
8
9     cout << "Введите " << n << " чисел: ";
10
11     int *dinamich_array = new int [n]; // создаем
12                                     // динамический массив
13     for (int i = 0; i < n; i++) {
14         cin >> dinamich_array[i]; // считываем числа в ячейки массива
15     }
16
17     cout << "Теперь давайте выведем элементы массива в обратном
18 порядке: ";
19
20     for (int i = n - 1; i >= 0; i--) {
21         cout << dinamich_array[i] << " "; // выводим значение всех ячеек
22     }
23
24     cout << endl << "Удаляем массив!";
25
26     delete [] dinamich_array; // удаляем динамический массив
27
28     return 0;
}
```

dinamic_array.cpp

Задайте количество чисел, которое вы хотите ввести: 5

Введите 5 чисел: 2 4 6 8 16

Теперь давайте выведем элементы массива в обратном порядке: 16
8 6 4 2

Удаляем массив!

Process returned 0 (0x0) execution time : 0.010 s

Press any key to continue.

Как создать двумерный динамический массив

- Для создания двумерного динамического массива мы будем использовать похожую конструкцию (как и в одномерном динамическом массиве):
- `<тип данных> **<имя массива> = new <тип данных массива>* [<количество ячеек>];`

Нужно обратить внимание на:

- Дополнительный оператор `*` перед `<имя массива>` и после `<тип данных массива>`.
- Дальше для каждой ячейки мы должны создать одномерный массив. Чтобы это сделать, нам понадобится цикл `for` и конструктор `new`.

```
1  for (int i = 0; i < n; i++) {  
2  
3      <имя массива>[i] = new <тип ячеек> [<количество  
4  ячеек>];  
5  
}
```

Пример двумерного динамического массива

```
#include <iostream>

using namespace std;

int main() {
    setlocale(0, "");

    int **dynamic_array2 = new int* [5]; // создаем
    for (int i = 0; i < 5; i++) {        // двумерный
        dynamic_array2[i] = new int [i + 1]; // массив
    }                                    // !

    for (int i = 0; i < 5; i++) {
        cout << "Введите числа" << "(" << i + 1 << ")" << ":\n";
        for (int j = 0; j < i + 1; j++) {
            cin >> dynamic_array2[i][j];
        }
    }

    for (int i = 0; i < 5; i++) {
        int sum = 0;
        for (int j = 0; j < i + 1; j++) {
            sum += dynamic_array2[i][j];
        }
        cout << "Сумма " << i + 1 << " массива равна " << sum << endl;
    }

    for (int i = 0; i < 5; i++) {
        delete [] dynamic_array2[i]; // удаляем массив
    }

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