УНИВЕРСИТЕТ СИНЕРГИЯ

РАЗРАБОТКА НА C++

Урок 7.
Массивы и ГПСЧ

Что такое массив в C++?

Массив - это структура данных, которая позволяют хранить множество элементов одного типа под одним именем. Они представляют собой набор элементов, каждый из которых имеет свой **индекс**.

Индекс - номер ячейки конкретного элемента.

Примеры массивов в реальной жизни: Банковские ячейки, школьные шкафчики, списки студентов, дни недели, пиксели на экране монитора.

(в каждом из примеров есть определенный порядок и нумерация, это и есть индексы)

Что такое массив в C++?

Один из ключевых аспектов стандартных массивов - это их **фиксированный размер**, то есть, количество элементов в массиве определено заранее и не может изменяться динамически. Изменение изначального размера требует создания нового массива и копирования в него данных.

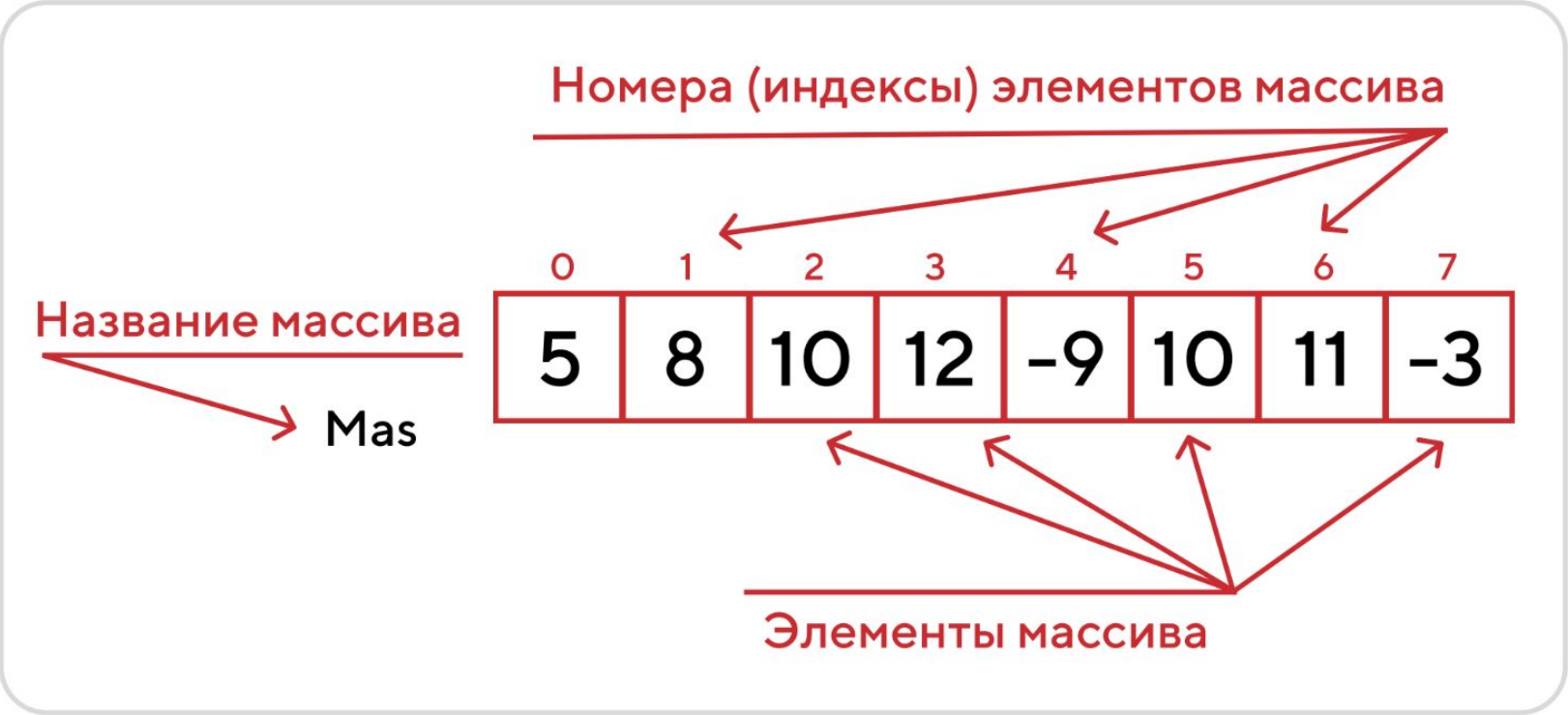
Преимущества:

- Организация данных: Например, вы можете хранить все оценки студентов в одном массиве, что делает их управление более удобным.
- Упорядоченный доступ: Это значит, что вы можете быстро получить доступ к элементу, зная его позицию(индекс).
- Однородность: Все элементы массива имеют один и тот же тип данных, что упрощает работу с ними и обеспечивает структурную целостность данных.

Что такое массив в C++?

Массивы устроены следующим образом:

У всех массивов первый элемент имеет индекс 0!



Как создать массив в C++?

тип_данных имя_массива[размер];

- тип_данных - например, int, double, char, string, или пользовательский тип данных.
- имя_массива - это имя, которое вы присваиваете вашему массиву, чтобы обращаться к нему в коде.
- размер - это количество элементов, которое массив будет хранить. Размер массива должен быть целым положительным числом.

int numbers[5]; // Массив целых чисел с 5 элементами

double prices[10]; // Массив чисел с плавающей точкой с 10 элементами

char letters[26]; // Массив символов с 26 элементами

string names[3]; // Массив строк с 3 элементами

Примеры массивов

```
int array[5] = {1, 2, 3, 4, 5}; // Массив целых чисел с 5 элементами (заданными по умолчанию)

int numbers[] = {1, 2, 3, 4, 5}; // Правильный вариант, так как числа в {} позволяют понять размер массива

char symbols[] = {‘k’}; // Массив символов с 1 элементом


string str_arr[3] = { "Hello" }; // Массив строк с 3 элементами (но известен по умолчанию только один)
str_arr[1] = "World";
str_arr[2] = "!";
cout << str_arr[0] + " " + str_arr[1] + str_arr[2] << endl; // вывод: Hello World!
cin >> str_arr[1]; // пользователь вводит слово Synergy
cout << str_arr[0] + " " + str_arr[1] + str_arr[2] << endl; // вывод: Hello Synergy!


int array1['A']; //Лучше так не делать :) но это правильно, ведь ‘A’ это число 65
```

Какие бывают ошибки?

```
int numbers[-1]; // Ошибка! Размер не может быть отрицательным
int numbers[“hello”]; // Ошибка! Размер не может быть строкой
int numbers(10); // Это не ошибка, но и не массив (одно и то же int numbers = 10;)
int numbers[]; // Ошибка! Не указан размер (и его невозможно определить)
int numbers{10}; // Ошибка! Компилятор не видит []

int numbers[1]; // Всё хорошо
numbers[1] = 1512; // Ошибка! (в массиве из одного элемента есть только [0] индекс)
numbers[0] = 1512; // Правильно
cout << numbers[1]; // -858993460 (в памяти за пределами массива хранится “мусор”)
cout << numbers[0]; // 1512
```

Быстрое заполнение массива

В данном примере мы заполняем массив с помощью счетчика цикла.
При этом значение элемента равно его индексу.

```
arr[0] = 0;
arr[1] = 1;
// …и так далее
arr[19] = 19; //последний
// так как 20 элементов
// мы заполнили
// не от 1 до 20 а от 0 до 19 :)
```

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

int main()
{
    int arr[20];
    for (int i = 0; i < 20; i++) {
        arr[i] = i;
        cout << arr[i] << " ";
    }
}
```

Консоль отладки Microsoft Visual Studio

```
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19
```

Быстрое заполнение массива

Те же 20 элементов можно заполнить любыми числами, изменив выражение после знака “=”

можно даже поставить

arr[i] = 1;

тогда все 20 элементов будут равны единице

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

int main()
{
    int arr[20];
    for (int i = 0; i < 20; i++) {
        arr[i] = (i + 2) * (i - 2);
        cout << arr[i] << " ";
    }
}
```

Консоль отладки Microsoft Visual Studio

```
-4 -3 0 5 12 21 32 45 60 77 96 117 140 165 192 221 252 285 320 357
```

Быстрое заполнение массива

Массив char хранит символы, а по таблицам ASCII:

числу 65 соответствует буква ‘A’
66 - ‘B’
67 - ‘C’
…и так далее

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

int main()
{
    char arr[20];
    for (int i = 0; i < 20; i++) {
        arr[i] = i+65;
        cout << arr[i] << " ";
    }
}
```

Консоль отладки Microsoft Visual Studio

A B C D E F G H I J K L M N O P Q R S T

Случайность в программировании

В программировании часто приходится моделировать случайные события (генерация пароля, бросание игральных костей и т.п.)

Так как чисто случайным мы считаем событие, которое нельзя спрогнозировать, а компьютер полностью предсказуем, то в программировании чистой случайности не существует

Единственным выходом остаётся написание сложного алгоритма, результат выполнения которого трудно предсказать

Это и называется ГПСЧ - Генератор **Псевдо**Случайных Чисел

Генератор псевдослучайных чисел

ГПСЧ – программа, которая принимает стартовое значение, выполняет с ним определенные математические операции, а затем повторяет эти операции для каждого полученного результата

Пример простейшего ГПСЧ, выводящего 100 случайных чисел от 0 до 100

Заметьте что при перезапуске программы мы получим точно такие же результаты нашего псевдорандома

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

int main()
{
	unsigned int r = 4541;
	for(int i = 0; i < 100; i++){
		r = (r * 8253729 + 2396403);
		cout << r % 100 << " ";
	}
}
```

Функции srand() и rand()

В библиотеке **<cstdlib>** есть встроенный ГПСЧ

srand() – устанавливает передаваемое пользователем значение в качестве стартового, вызывается один раз.
Это тот самый “сид” в майнкрафте :)

rand() – генерирует следующее случайное число в последовательности

Для каждого стартового числа всегда будет одинаковая последовательность