

INOPOLIS
UNIVERSITY



Массивы. Циклы

Цели: работать с типом данных массив

План урока:

- массивы
- свойства массивов
- работа с индексами
- циклы
- итерация по массиву

Массивы

Массив — структура данных, хранящая набор значений (элементов массива), идентифицируемых по индексу или набору индексов, принимающих целые значения из некоторого заданного непрерывного диапазона.

Массив – это совокупность данных, которая обладает следующими свойствами:

- все элементы массива имеют один и тот же тип;
- массив имеет одно имя для всех элементов;
- доступ к конкретному элементу массива осуществляется по индексу (индексам).



Массивы

Объявление массива похоже на объявление переменной за тем исключением, что после указания типа ставятся квадратные скобки:

```
тип_переменной[] название_массива;  
int[] numbers;
```

После определения переменной массива мы можем присвоить ей определенное значение:

```
numbers = new int[4];
```

Заполнение массива данными:

```
int[] nums2 = new int[] { 1, 2, 3, 5 };  
int[] nums3 = { 1, 2, 3, 5 };
```

Для обращения к элементам массива используются **индексы**.

```
nums[0] = 6;  
int a = nums[1];
```

Массивы

Мы рассматривали одномерные массивы - такие массивы можно представлять в виде горизонтального ряда элемента.

```
int[] nums1 = new int[] { 0, 1, 2, 3, 4, 5 };
```

0	1	2	3	4	5
---	---	---	---	---	---

Но массивы также бывают многомерными.

Массивы которые имеют два измерения называют **двумерными**. Двумерный массив можно представить в виде таблицы

```
int[][] nums2 = new int[][] { { 0, 1, 2 }, { 3, 4, 5 } };  
int[][] nums3 = { { 0, 1, 2 }, { 3, 4, 5 } };
```

0	1	2
3	4	5



Массивы

Задача:

Создать массив books. Тип данных в массиве – строка;

Поместить 3 книги (строки) в массив ;



Массивы

Задача:

Создать класс Car;

Создать массив garage;

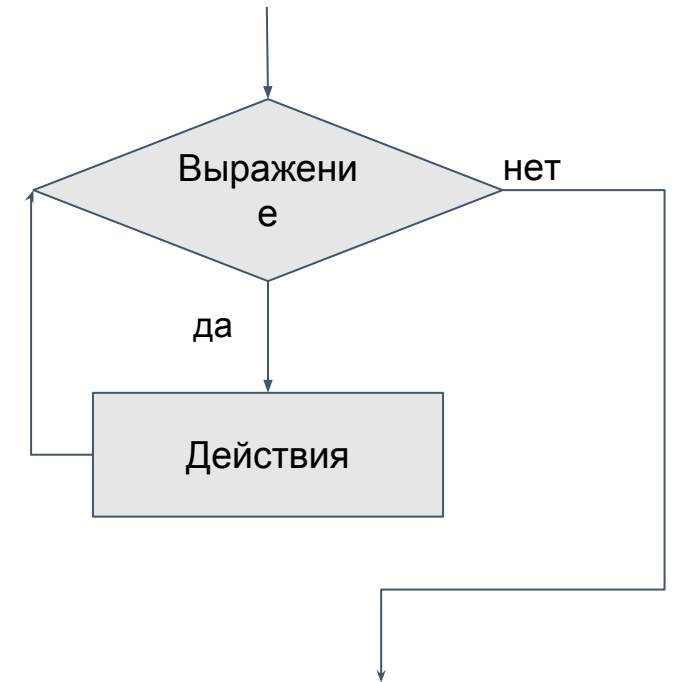
Поместить 5 автомобилей в гараж;



Циклы

Цикл — разновидность управляющей конструкции в высокоуровневых языках программирования, предназначенная для организации многократного исполнения набора инструкций

- **Циклы со счетчиком:** количество итераций определяется смоделированным счетчиком. `for`
- **Безусловные циклы:** циклы, которые выполняются бесконечно. `for each..`
- **Циклы с предусловием:** условие выполнения определяется перед первой итерацией. `while`
- **Циклы с постусловием:** условие выполнения определяется после первой итерации (поэтому они всегда выполняются минимум один раз). `do..while`



Циклы

```
for (ТипСсылки имя_ссылки_на_элемент : ссылка_на_массив) {  
    //тело цикла (последовательность действий)  
}
```

* Для работы с массивами (и другими структурами данных).



Циклы

```
for (ТипСсылки имя_ссылки_на_элемент : ссылка_на_массив) {  
    //тело цикла (последовательность действий)  
}
```

* Для работы с массивами (и другими структурами данных).

```
for (int i = начальное_значение_счетчика; условие; алгоритм_изменения_счетчика) {  
    //тело цикла (последовательность действий)  
}
```

* можно прервать раньше времени словом break или "проскочить" итерацию словом continue.



INNOVATION
UNIVERSITY



Вопросы?