

Массивы. Оператор for

НАШИ ПРАВИЛА



Включенная камера



Вопросы по поднятой руке



Не перебиваем друг друга



Все вопросы, не связанные с тематикой курса (орг-вопросы и т. д.), должны быть направлены куратору



Подготовьте свое рабочее окружение для возможной демонстрации экрана (закройте лишние соцсети и прочие приложения)

ЦЕЛЬ

Получить представление о методах работы с массивами, включая цикл `for`

ПЛАН ЗАНЯТИЯ

■
Массивы

■
Явная инициализация массива

■
Устройство массива

■
Оператор for

■
Операторы break и continue

МАССИВЫ

Массив - структура данных, представляющая собой набор элементов одного типа. К каждому элементу можно обратиться по его порядковому номеру (индексу).

Массив не может изменять свой размер, он задается один раз и навсегда.

0	1	2	3	4
7	2	-5	11	1

// объявление и выделение памяти
`int[] a = new int[5];`

// запись значений по индексам

`a[0] = 7;`

`a[1] = 2;`

`a[2] = -5;`

`a[3] = 11;`

`a[4] = 1;`

// получение значения по индексу
`System.out.println(a[2]);` **// вывод -5**

СТАВИМ ПЛЮС, ЕСЛИ ПОНЯТНО

Объявление массива

Неизменность размера

Обращение по индексам (чтение и запись)

ЯВНАЯ ИНИЦИАЛИЗАЦИЯ МАССИВА

Явная инициализация массива сразу указать элементы массива при его объявлении массива вы сразу же указываете его элементы.

```
int[] numbers = {7, 2, -5, 11, 1};
```

```
String words = {"Hello", "Bye", "Java"};
```

Такой подход удобен, когда вы знаете точные значения, которые должны быть в массиве, еще до начала выполнения программы.

СТАВИМ ПЛЮС, ЕСЛИ ПОНЯТНО



Удобство явной инициализации

УСТРОЙСТВО МАССИВА

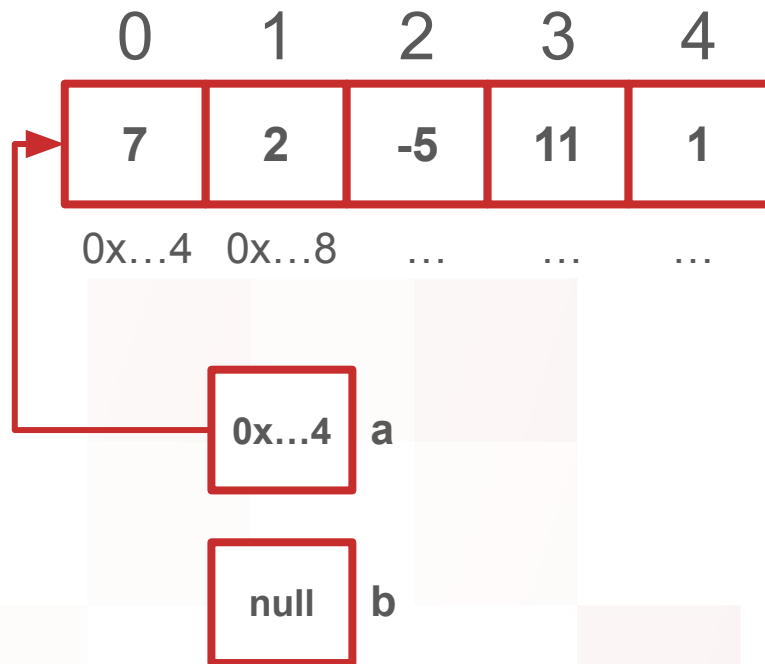
Массив, по аналогии с классами (например, String) является ссылочным типом данных

Переменная массива содержит адрес первого элемента, и, по факту, является указателем на ячейку памяти

Как и переменная любого другого ссылочного типа данных, переменная массива может не содержать никакого адреса и иметь значение **null**

```
int[] a = {7, 2, -5, 11, 1};
```

```
int[] b = null;
```



УСТРОЙСТВО МАССИВА

Следует обратить внимание на то, что происходит при “копировании” переменных массивов

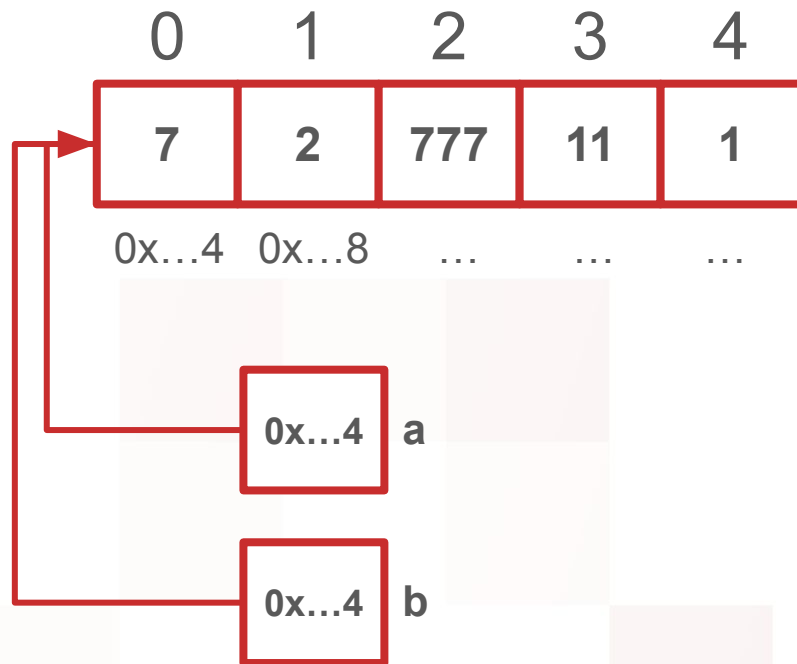
```
int[] a = {7, 2, -5, 11, 1};
```

```
int[] b = null;
```

```
b = a; // копирование ссылки
```

```
b[2] = 777;
```

```
System.out.println(a[2]); // 777
```



СТАВИМ ПЛЮС, ЕСЛИ ПОНЯТНО

■
Массив как ссылочный тип

■
Особенности работы с переменными
массива

Оператор for

Синтаксический сахар над циклом while, который предоставляет простой и читаемый способ работы с массивами

```
for (инициализация; условие окончания;  
инкремент/декремент) {  
    // блок кода  
}
```

```
int i = 0;  
while (i < 5) {  
    System.out.println(i);  
    i++;  
}
```

```
for (int i = 0; i < 5; i++) {  
    System.out.println(i);  
}
```

Итеративный оператор for

Если есть какой-то массив, то можно итерироваться сразу по его значениям в for

```
for (тип переменная:  
массив_указанного_типа) {  
    // блок кода  
}
```

```
int[] a = {7, 2, -5, 11, 1};
```

```
for (int i = 0; i < 5; i++) {  
    System.out.println(a[i]);  
}
```

```
for (int n: a) {  
    System.out.println(n);  
}
```

СТАВИМ ПЛЮС, ЕСЛИ ПОНЯТНО

Структура цикла for

Использование for при работе с массивами


Итеративный for

Оператор break

break прерывает выполнение цикла, даже если условие окончания его работы не было нарушено. Позволяет избежать лишних итераций.

Найти первое отрицательное число

```
for (int i = 0; i < a.length; i++) {  
    if (a[i] < 0) {  
        System.out.println(a[i]);  
        break; // прерывание цикла  
    }  
}  
// код после цикла  
// выполнение программы будет продолжено здесь  
...
```




Оператор continue

continue используется для перехода к следующей итерации цикла, пропуская оставшуюся часть кода в текущей итерации. В отличие от break, который полностью прерывает цикл, continue только пропускает одну итерацию или часть её.

Найти сумму всех четных положительных

```
for (int i = 0; i < a.length; i++) {  
    // для нечетного - переходим к следующему  
    if (a[i] % 2 == 1) continue;  
  
    // для отрицательного - переходим к следующему  
    if (a[i] < 0) continue;  
  
    sum = sum + a[i];  
}  
  
// выполнение программы будет продолжено здесь  
// после прохода всех итераций  
...
```

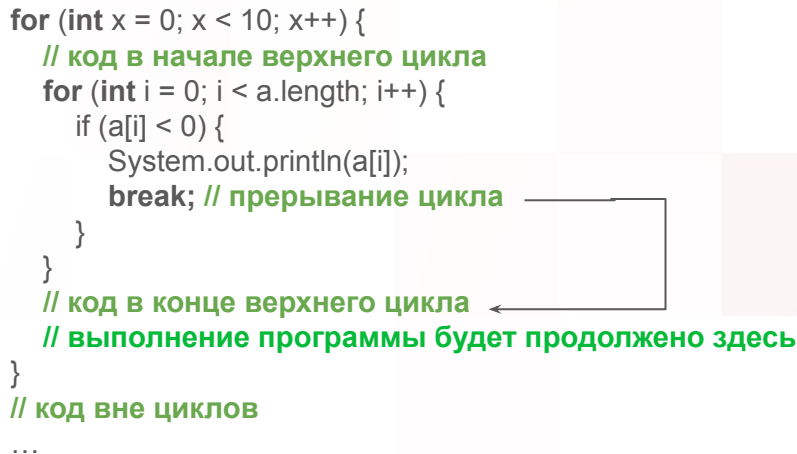


Оператор break во вложенном цикле

break завершает цикл, в котором был вызван:

Найти первое отрицательное число в цикле
который итерируется от 0 до 10

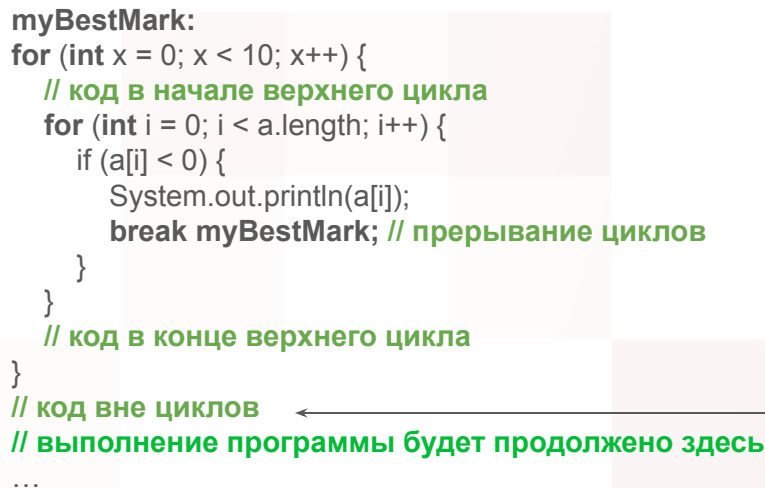
```
for (int x = 0; x < 10; x++) {  
    // код в начале верхнего цикла  
    for (int i = 0; i < a.length; i++) {  
        if (a[i] < 0) {  
            System.out.println(a[i]);  
            break; // прерывание цикла  
        }  
    }  
    // код в конце верхнего цикла  
    // выполнение программы будет продолжено здесь  
}  
// код вне циклов  
...
```



Если надо завершить текущий цикл и все
вышестоящие, то можно воспользоваться **break с
меткой**:

Найти первое отрицательное число в цикле
который итерируется от 0 до 10

```
myBestMark:  
for (int x = 0; x < 10; x++) {  
    // код в начале верхнего цикла  
    for (int i = 0; i < a.length; i++) {  
        if (a[i] < 0) {  
            System.out.println(a[i]);  
            break myBestMark; // прерывание циклов  
        }  
    }  
    // код в конце верхнего цикла  
}  
// код вне циклов  
// выполнение программы будет продолжено здесь  
...
```



СТАВИМ ПЛЮС, ЕСЛИ ПОНЯТНО

■
Случаи использования break

■
Случаи использования continue

Break с меткой

ПОИГРАЕМ ;)

■
Массив

■
Явная инициализация

■
Структура массива

■
for

■
break

ДОМАШНЕЕ ЗАДАНИЕ

1. Доделайте прошлые домашние задания
2. В CurrencyConverter храните курсы валют в массивах, для вывода списка доступных валют используйте for в одном случае и итеративный for - в другом.
3. Порешайте задачи на странице <https://w3schools.com/java/exercise.asp> раздел Java Loops (все задания) и Java Arrays (1-5 задания)



Ваша новая IT-профессия – Ваш новый уровень жизни

Программирование с нуля в
немецкой школе AIT TR GmbH