

Array

Intro

Problem

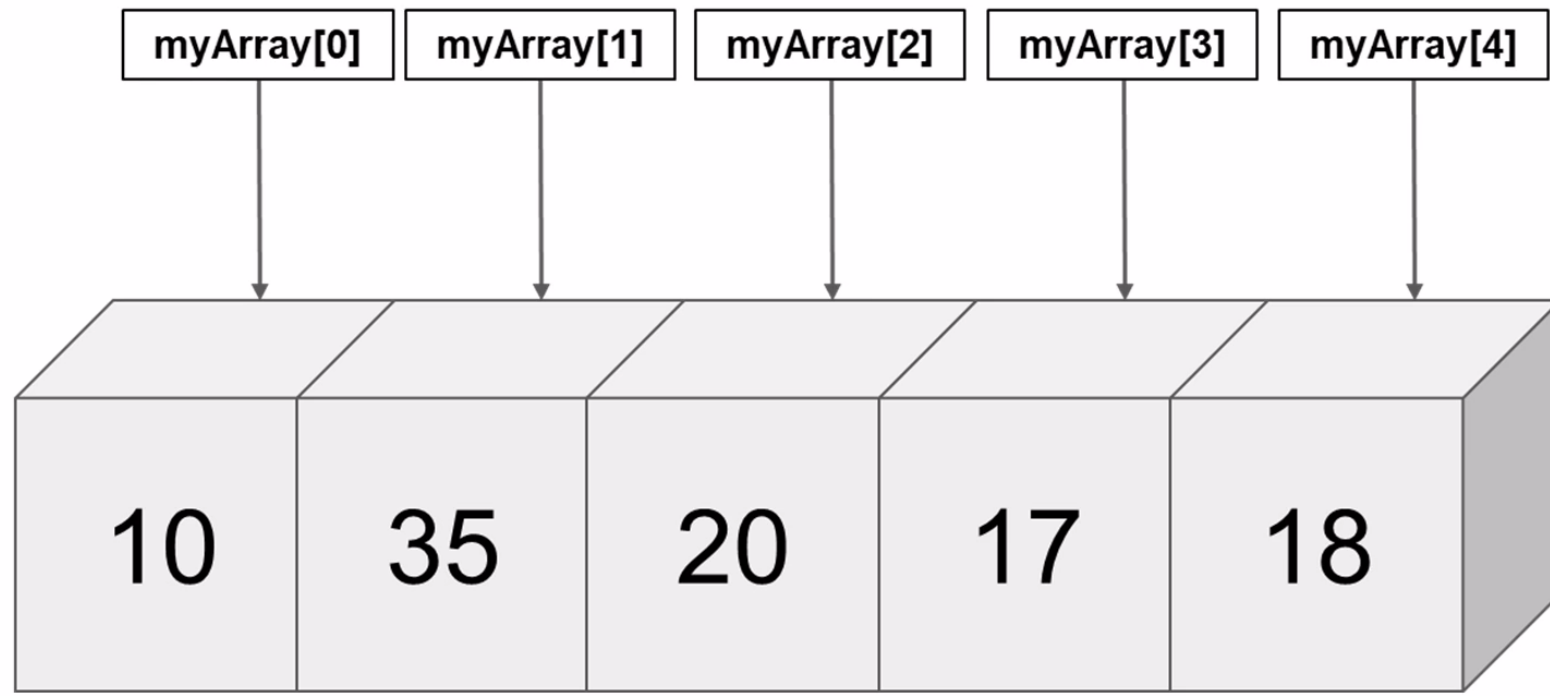
- Что если необходимо хранить в памяти множество однотипных данных, например: фамилии студентов в группе.
- Создадим такое же количество переменных как и студентов!
- А что если появится новый студент? Переписывать программу?!

Solution

Array

Array

Array



Array

- **Массив** — это конечная последовательность упорядоченных по индексу элементов одного типа, доступ к каждому элементу в которой осуществляется по этому индексу.
- Размер или длина массива — это общее количество элементов в массиве. Размер массива задаётся при создании массива и не может быть изменён в дальнейшем.

Syntax

```
тип_данных[] название_массива;  
// либо  
тип_данных [ ]название_массива;  
// либо  
тип_данных название_массива[];
```


Example

```
int[] array1; // best practice  
int []array2;  
int array3[];
```

Example

```
int nums[]; // обявили  
nums = new int[4]; // инициализирали (масив из 4 чисел)
```

Example

```
// объявим и инициализируем массив из 4 чисел  
int nums[] = new int[4];  
// объявим и инициализируем массив из 15 чисел  
int[] nums2 = new int[15];
```

Example

```
// объявим и инициализируем массив из 4 чисел
int[] nums = new int[4];
// инициализируем 0-ой элемент массива
nums[0] = 1;
// инициализируем 1-ый элемент массива
nums[1] = 2;
// инициализируем 2-ой элемент массива
nums[2] = 4;
// инициализируем 3-ий элемент массива
nums[3] = 100;
System.out.println(nums[2]); // 4
```

Example

```
// эти два способа равноценны  
int[] nums = new int[] {1, 2, 3, 5};  
int[] nums2 = {1, 2, 3, 5}; // anonymous array  
System.out.println(nums.length); // длина массива 4
```

Example

```
int[] array = new int[10];
Random rand = new Random();
for (int i = 0; i < array.length; i++) {
    array[i] = rand.nextInt(10);
}

System.out.println("initial array:");
for (int i = 0; i < array.length; i++) {
    System.out.print(array[i] + " ");
}

for (int i = 0; i < array.length; i++) {
    array[i] += 10;
}

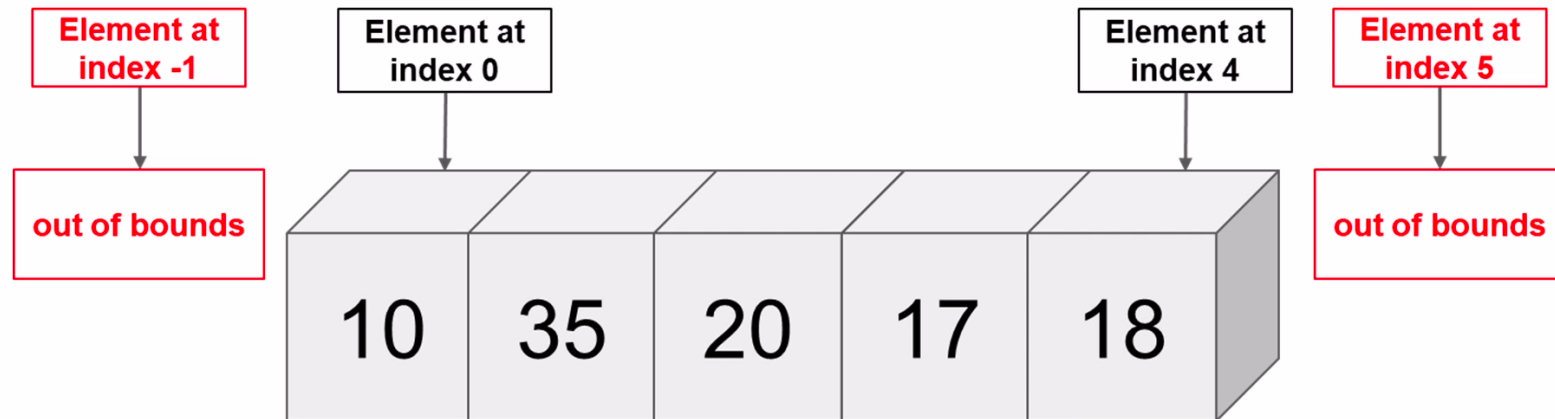
System.out.println("new array:");
for (int i = 0; i < array.length; i++) {
    System.out.print(array[i] + " ");
}
```

What can go wrong?

What can go wrong?

```
int[] nums = {1, 2, 3, 5};  
System.out.println(nums[-1]);  
System.out.println(nums[4]);
```


ArrayIndexOutOfBoundsException



Multidimensional Array

Multidimensional Array

Массивы бывают:

- одномерные
- двумерные
- n-мерные

Multidimensional Array

	Column 1	Column 2	Column 3	Column 4
Row 1	<code>a[0][0]</code>	<code>a[0][1]</code>	<code>a[0][2]</code>	<code>a[0][3]</code>
Row 2	<code>a[1][0]</code>	<code>a[1][1]</code>	<code>a[1][2]</code>	<code>a[1][3]</code>
Row 3	<code>a[2][0]</code>	<code>a[2][1]</code>	<code>a[2][2]</code>	<code>a[2][3]</code>

Syntax

```
int[][] array1;  
int [][]array2;  
int array3[][];
```

Example

```
int[] nums1 = new int[] {0, 1, 2, 3, 4, 5};  
int[][] nums2 = {  
    {0, 1, 2},  
    {3, 4, 5}  
};
```

Example

```
int[][] nums2 = {{0, 1, 2}, {3, 4, 5}};  
  
// установим элемент первого столбца второй строки  
nums2[1][0] = 44;  
System.out.println(nums2[1][0]);
```

Multidimensional Arrays

	Column 1	Column 2	Column 3	Column 4
Row 1	<div>1</div> <div>a[0][0]</div>	<div>2</div> <div>a[0][1]</div>	<div>3</div> <div>a[0][2]</div>	
Row 2	<div>4</div> <div>a[1][0]</div>	<div>5</div> <div>a[1][1]</div>	<div>6</div> <div>a[1][2]</div>	<div>9</div> <div>a[1][3]</div>
Row 3	<div>7</div> <div>a[2][0]</div>			

Example

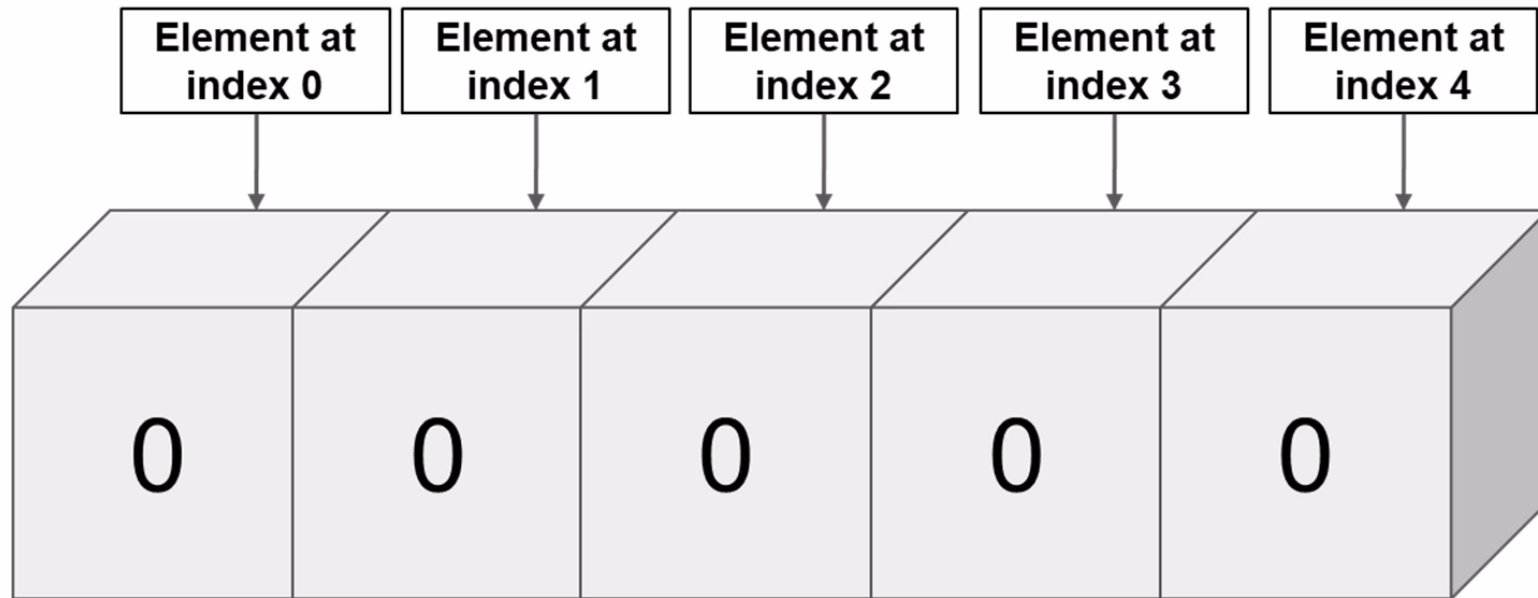
```
int[][] nums = new int[3][];  
nums[0] = new int[2];  
nums[1] = new int[3];  
nums[2] = new int[5];  
...
```

Example

```
int[][] array = {
    {1, 1, 1, 1},
    {1, 1, 1, 1},
    {1, 1, 1, 1}
};
int sum = 0;
for (int i = 0; i < array.length; i++) {
    for (int j = 0; j < array[0].length; j++) {
        sum += array[i][j];
    }
}
System.out.println(sum);
```

Default initialization for Array

Default initialization for Array



Default initialization for Array

Initialization	Content
<code>new boolean[3]</code>	<code>{false, false, false}</code>
<code>new int[3]</code>	<code>{0, 0, 0}</code>
<code>new double[3]</code>	<code>{0.0, 0.0, 0.0}</code>
<code>new String[3]</code>	<code>{null, null, null}</code>
<code>new int[3][]</code>	<code>{null, null, null}</code>

Сортировка массивов

Сортировка массивов

- Сортировка выбором.
- Сортировки с помощью обменов:
 - Пузырьковая сортировка.
 - Шейкерная сортировка.
- Сортировка с помощью включения.
- Сортировка слиянием.
- Сортировка с помощью разделения.

Сортировка выбором

```
int[] array = {8, 5, 2, 6, 9, 3, 1, 4, 0, 7};

for (int i = 0; i < array.length; i++) {
    int minIndex = i;
    for (int j = i + 1; j < array.length; j++) {
        if (array[j] < array[minIndex]) {
            minIndex = j;
        }
    }
    int temp = array[i];
    array[i] = array[minIndex];
    array[minIndex] = temp;
}

for (int i = 0; i < array.length; i++) {
    System.out.print(array[i] + " ");
}
```


Пузырьковая сортировка

```
int[] array = {5, 4, 3, 1, 2};
for (int i = array.length - 1; i > 0; i--) {
    for (int j = 0; j < i; j++) {
        if (array[j] > array[j + 1]) {
            int temp = array[j];
            array[j] = array[j + 1];
            array[j + 1] = temp;
        }
    }
}
for (int i = 0; i < array.length; i++) {
    System.out.print(array[i] + " ");
}
```

Class **Arrays**

Methods for class **Arrays**

- `String toString(T[])`
- `String toString(T[])`
- `T[] copyOf(T[], int)`
- `T[] copyOfRange(T[], int, int)`
- `void sort(T[])`
- `void sort(T[], int, int)`
- `int binarySearch(T[], T)`
- `int binarySearch(T[], int, int, T)`

Methods for class **Arrays**

- `void fill(T[], T)`
- `void fill(T[], int, int, T)`
- `boolean equals(T[], T[])`
- `boolean equals(T[], int, int, T[], int, int)`
- `int compare(T[], T[])`
- `int compare(T[], int, int, T[], int, int)`
- `boolean deepEquals(Object[], Object[])`
- `List<T> asList(T ...)`