

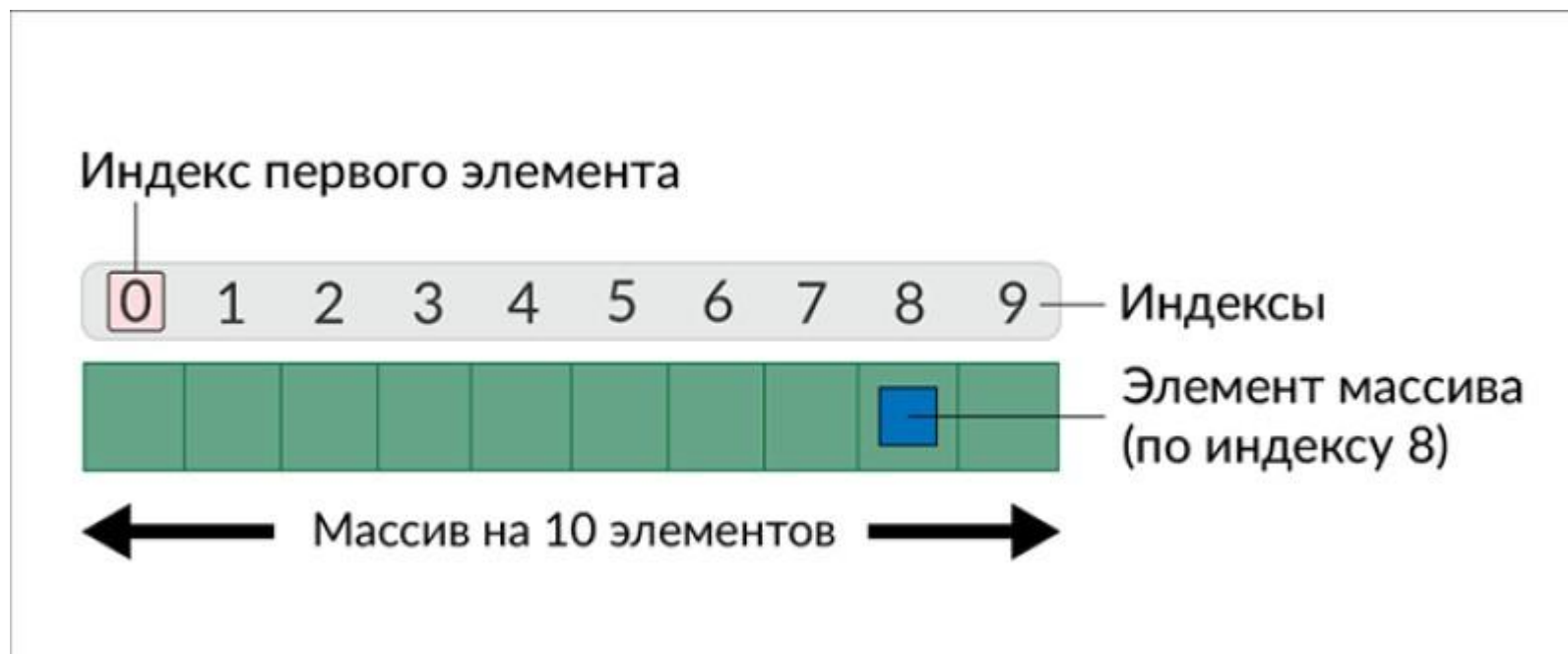
# Java Professional module #2

Working with Arrays in Java. Binary search. Sort.  
Mentor: Rustam Khakov

## Working with Arrays in Java. Binary search. Sort.

- Arrays in Java
  - Important points
  - Creating, initializing and accessing
  - Multidimensional Arrays
  - Cloning of arrays
- Binary search
  - Binary Search Algorithm
- Sort
  - Ways of sorting in Java

# Массивы



# Массивы

№	Объявление массива, Java-синтаксис	Примеры	Комментарий
1.	<code>dataType[] arrayName;</code>	<code>int[] myArray;</code>  <code>Object[] arrayOfObjects;</code>	Желательно объявлять массив именно таким способом, это Java-стиль
2.	<code>dataType arrayName[];</code>	<code>int myArray[];</code>  <code>Object arrayOfObjects[];</code>	Унаследованный от C/C++ способ объявления массивов, который работает и в Java

# Массивы

`int [ ] array1 = { 5, 17, 350 }`

↑      ↑      ↑      ↑

тип данных    квадратные скобки    имя массива    в фигурных скобках  
массива      указывают, что    массива      указаны числа, которые  
                 это массив                                    мы присвоили в array1

0    1    2  
↓   ↓   ↓

`int [ ] array1 = { 5, 17, 350 }`

↑      ↑      ↑      ↑

тип данных    квадратные скобки    имя массива    в фигурных скобках  
массива      указывают, что    массива      указаны числа, к  
                 это массив                                    мы присвоили в ;

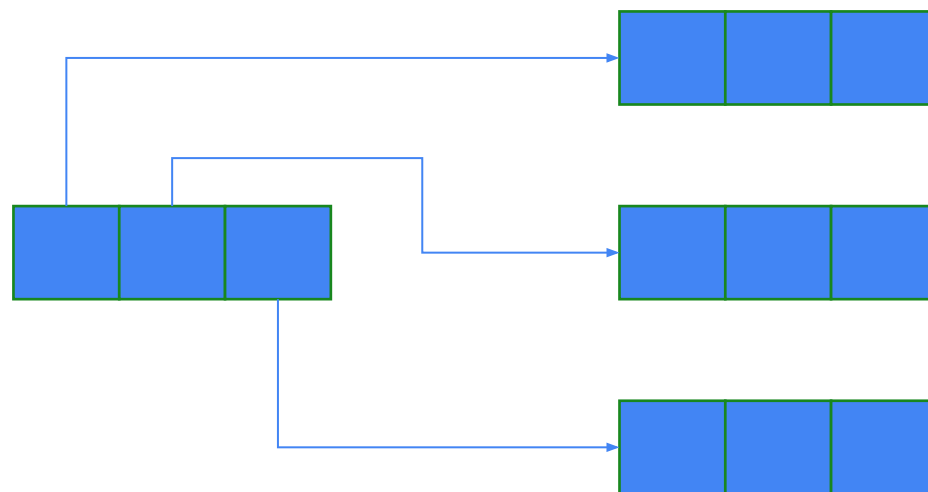
## *Важные моменты*

Массив в Java — это группа переменных одинакового типа, на которые указывает ссылка (общее имя).

- Последовательные ячейки памяти.
- Можем найти длину, используя свойство объекта `length`.
- Переменные в массиве упорядочены
- Размер массива должен быть указан как целое значение.
- Размер массива не может быть изменен (после инициализации)

# Многомерные массивы

Многомерные массивы — это массивы массивов, в которых каждый элемент массива содержит ссылку на другие массивы.



```
int[][] intArray = new int[2][2]; // двумерный массив или матрица  
int[][][] intArray = new int[3][3][3]; // 3D-массив
```

# Двумерные массивы

	столбец 0	столбец 1
строка 0	1	2
строка 1	3	4

```
1 class Test {  
2  
3     public static void main(String[] args) {  
4         int[][] array = new int[2][2];  
5     }  
6 }
```



## ДВУМЕРНЫЙ МАССИВ

Индексы строк ----  
↓  
0  
1  
2

0	1	2	3	←---- Индексы столбцов
5	7	3	17	
7	0	1	12	twoDimArray[1][2] //1
8	1	2	3	twoDimArray[2][3] //3

Всего в массиве **3 строки** и **4 столбца**, то есть  $3 \times 4 = 12$  элементов

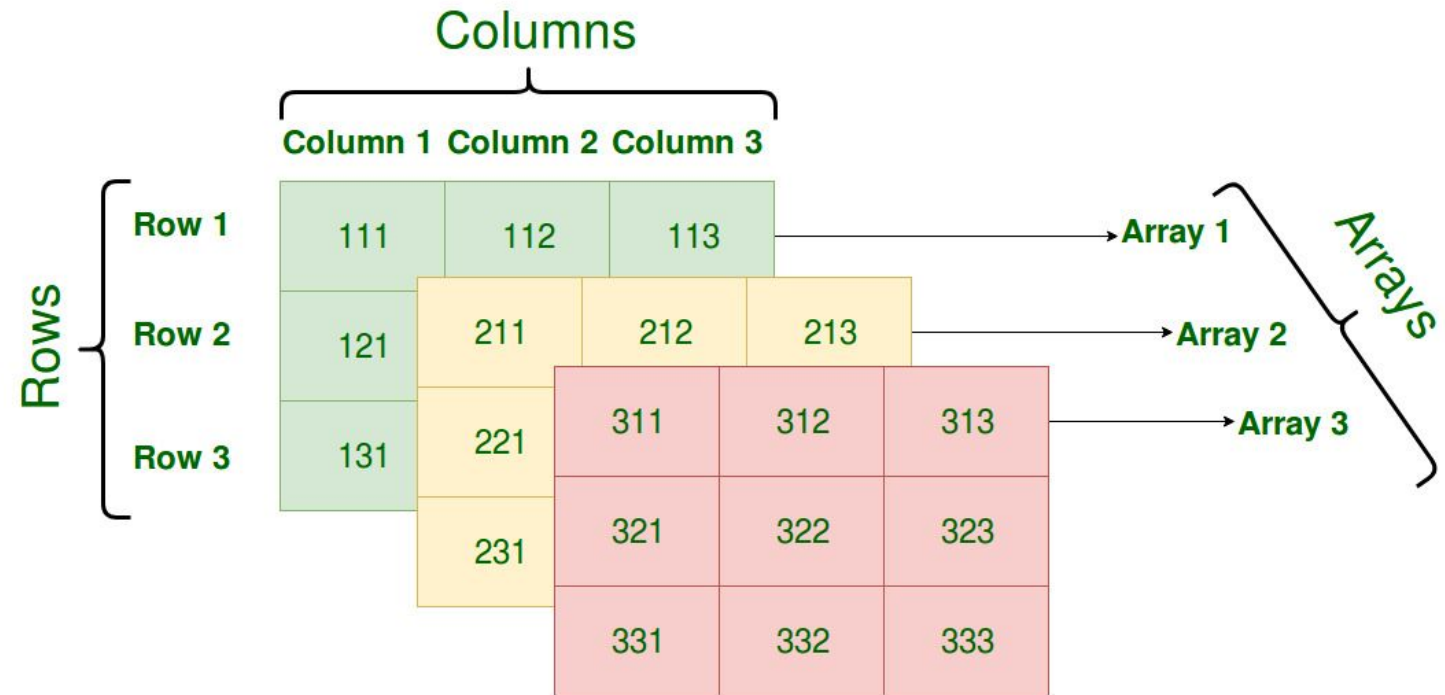
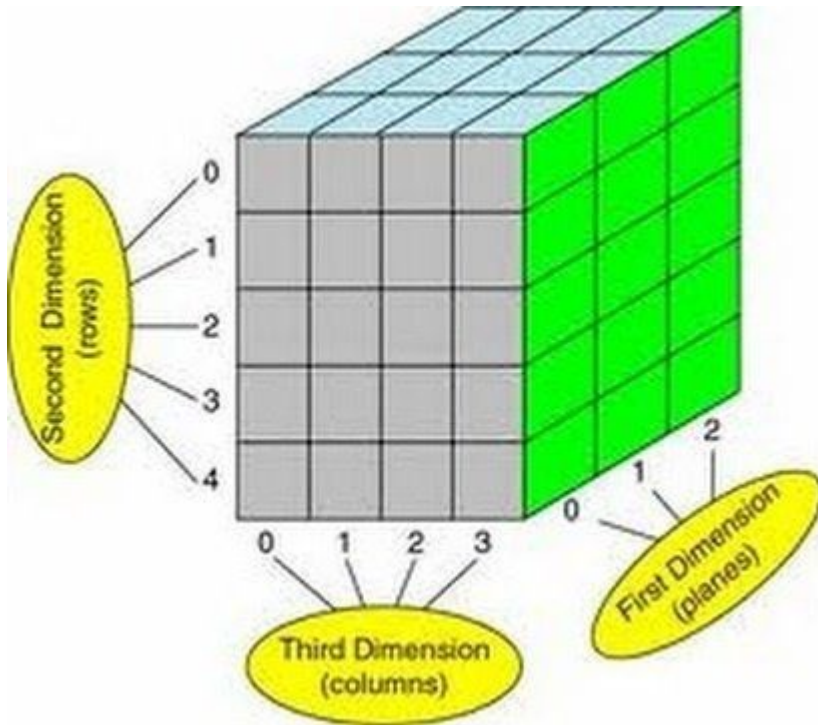
$$A = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1j} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2j} & \dots & a_{2n} \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ a_{i1} & a_{i2} & \dots & a_{ij} & \dots & a_{in} \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ a_{m1} & a_{m2} & \dots & a_{mj} & \dots & a_{mn} \end{pmatrix}$$

# Двумерный массив

	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>

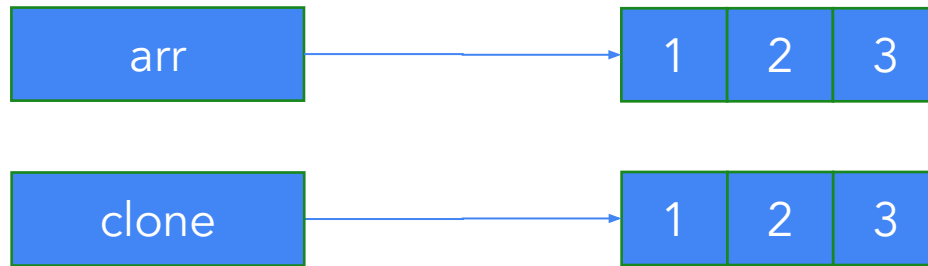
	Column 1	Column 2	Column 3	Column 4
Row 1	<div>1 <code>a[0][0]</code></div>	<div>2 <code>a[0][1]</code></div>	<div>3 <code>a[0][2]</code></div>	
Row 2	<div>4 <code>a[1][0]</code></div>	<div>5 <code>a[1][1]</code></div>	<div>6 <code>a[1][2]</code></div>	<div>9 <code>a[1][3]</code></div>
Row 3	<div>7 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

# Трёхмерный массив

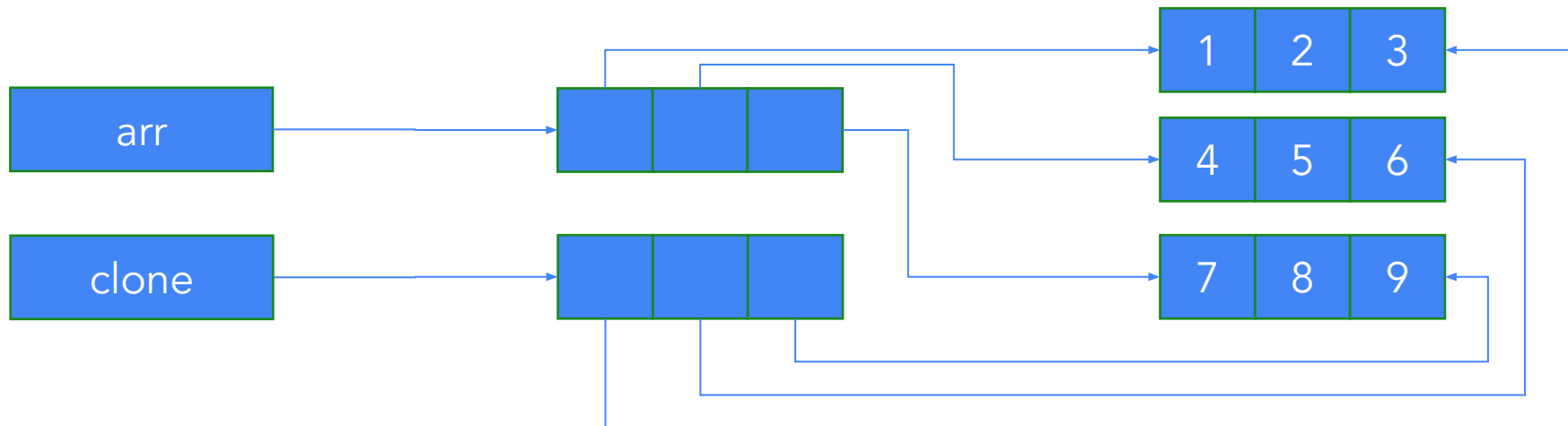


## Клонирование массивов

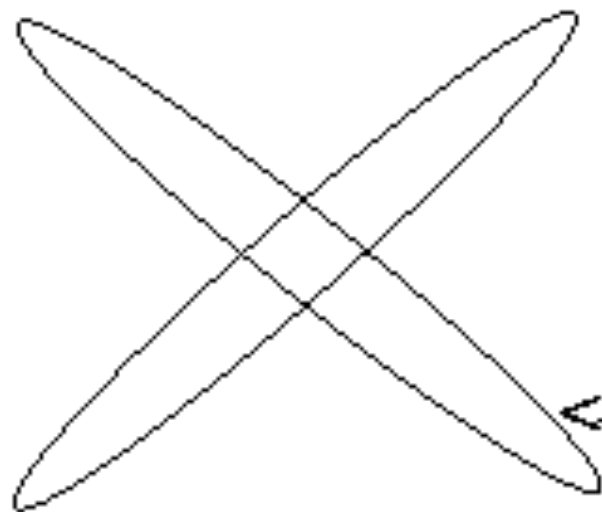
При клонировании одномерного массива выполняется «глубокая копия», содержащим копии элементов исходного массива!



При клонировании многомерного массива, клон является «поверхностной копией», то есть создается только один новый массив, в котором каждый массив элементов является ссылкой на исходный массив элементов, но подмассивы являются общими.



# Двумерные массивы



← **Элементы побочной диагонали**

← **Элементы главной диагонали**

Главная диагональ  $i == j$  матрица  $n$  на  $n$

Выше диагонали  $i < j$ , ниже  $i > j$

Побочная диагональ  $i == n-1-j$

# *Практика*

1. Доступ к элементам двумерных массивов
2. Доступ к элементам трехмерного массива
3. Морской бой

# Морской бой

1. Консольное приложение.
2. Одновременно в игре могут участвовать только два человека.
3. Игроки вводят имена.
4. У каждого игрока есть своё поле - квадрат 10x10 клеток
5. По очереди расставляют свои корабли.
  1. 4 однопалубных корабля,
  2. 3 двухпалубных,
  3. 2 трехпалубных,
  4. 1 четырёхпалубный.
6. Корабли можно располагать только по горизонтали или по вертикали.
7. Между кораблями должна быть минимум одна клетка
8. Игроки не видят расположение кораблей друг друга.
9. Начинается игра. Первый игрок делает выстрел, сообщая нашему приложению координаты предполагаемой цели - номер клетки по горизонтали и номер клетки по вертикали.
10. Если выстрел первого игрока оказался удачным, и он поразил цель, то возможно два варианта развития событий.
  1. Первый вариант: в указанной игроком клетки находится корабль, то, если корабль однопалубный, игрок "убил" корабль, если не однопалубный, то ранил. Следующий ход за первым игроком.
  2. Второй вариант: игрок не попал ни в какой корабль, ход переходит второму игроку.
11. Таким образом, возвращаемся в пункт 8, передавая ход друг другу, игроки пытаются как можно раньше уничтожить корабли друг друга. Тот, кто первым, уничтожит все корабли - победитель. Печатаем поздравление и выход из игры.



# Сортировка

- Использование циклов
- Использование метода `sort()` класса `Arrays`
- Использование метода `sort` класса `Collections`

# *Бинарный поиск*

Алгоритм:

1. Начните со среднего элемента в массиве, в качестве ключа поиска.
2. Если значение ключа поиска равно элементу, вернуть индекс ключа поиска.
3. Если значение ключа поиска меньше элемента, продолжаем поиск в правой половине.
4. В противном случае продолжаем поиск в левой половине.
5. Если не найден возвращаемся в пункт 2, пока значение не будет найдено или интервал не станет пустым.