

# Java Professional module #2

lecture #1. Working with Arrays in Java. Binary search. Sort.

## lecture #1. 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

## Важные моменты

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

- Массивы хранятся как последовательные ячейки памяти.
- Поскольку в Java массивы являются объектами, мы можем найти их длину, используя свойство объекта `length`.
- Переменная массива Java также может быть объявлена, как и другие переменные, с `[]` после типа данных.
- Переменные в массиве упорядочены, и каждая имеет индекс, **начинающийся с 0**.
- Массив Java также можно использовать как локальную переменную или параметр метода.
- Размер массива должен быть указан как целое значение.
- Прямой суперкласс типа массива — `Object`.
- Размер массива не может быть изменен (после инициализации). Однако ссылка на массив может указывать на другой массив.

## Создание, инициализация и доступ к массиву

Объявление массива состоит из двух компонентов: типа и имени.

```
Type[] name = new Type[<size>;
```

```
int[] intArray;  
byte byteArray[];  
short shortsArray[];  
boolean booleanArray[];  
long longArray[];  
float floatArray[];  
double doubleArray[];  
char charArray[];  
Object objectArray[];
```

Элементы в массиве, выделенном new ,  
будут автоматически  
инициализированы:

- нулем (для числовых типов),
- false (для логического значения),
- null (для ссылочных типов).

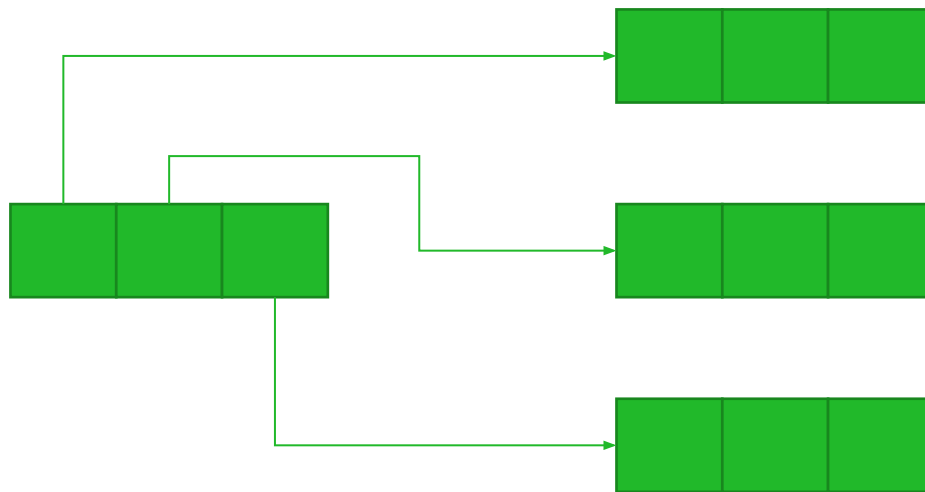
когда размер массива и переменные массива уже известны, можно использовать литералы массива.

```
int[] intArray = {1,2,3,4,5,6,7,8,9,10};
```

## Многомерные массивы:

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

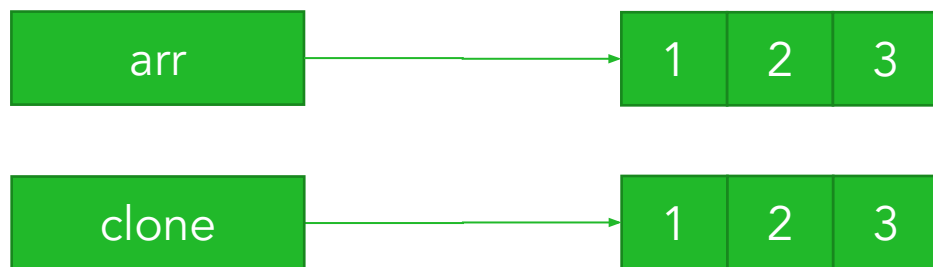
Многомерный массив создается путем добавления одного набора квадратных скобок ([]) для каждого измерения.



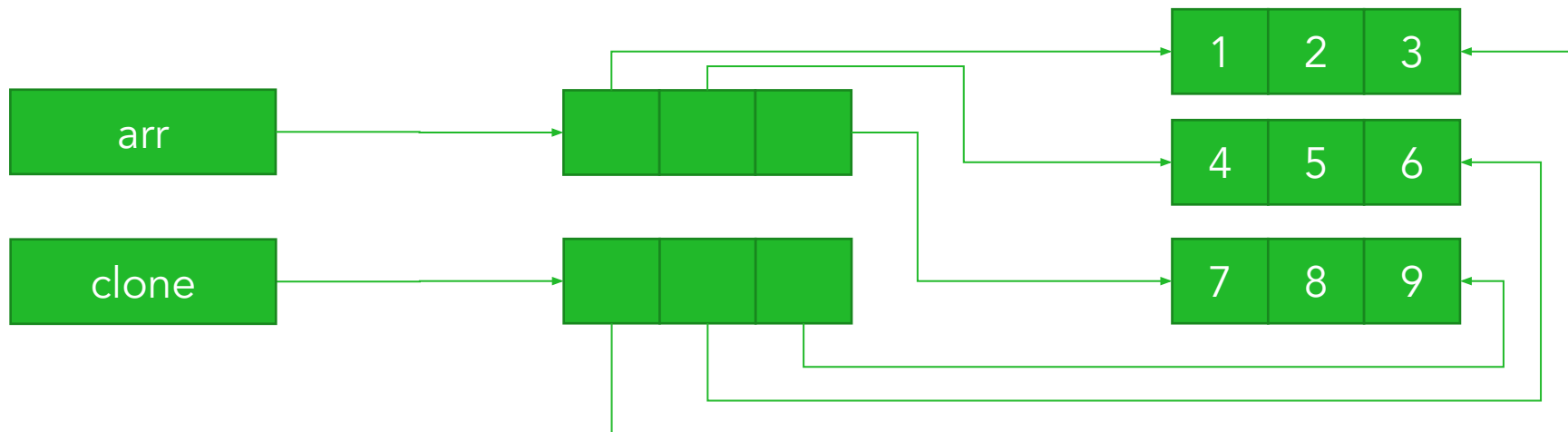
```
int[][] intArray = new int[2][2]; // двумерный массив или матрица  
int[][][] intArray = new int[3][3][3]; // 3D-массив
```

## 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.



## Практика многомерные массивы

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

## Бинарный поиск алгоритм

Алгоритм бинарного поиска:

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.



## Сортировка в Java

В Java есть несколько способов сортировки:

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

## 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.