

## Plan

# 2023-09-22

Повторение:

Циклы:

- while
- do while
- fori

Алгоритмы массивов:

- Bubble Sort
- Binary search

## Theory

# Повторение:

## Циклы:

**Цикл** - это конструкция, которая позволяет повторять выполнение блока кода до тех пор, пока не будет выполнено заданное условие.

- while
- do while
- fori

## while

- Цикл while выполняется, пока условие истинно. Он имеет следующую синтаксис:

```
while (условие) {  
    // ...  
}
```

- Вот пример использования цикла while:

```
int i = 0;
```

```
while (i < 10) {  
    System.out.println(i);  
}
```

```
    i++;  
}
```

## do while

Цикл do-while выполняется, по крайней мере, один раз. Он имеет следующую синтаксис:

```
do {  
    // ...  
} while (условие);
```

Вот пример использования цикла do-while:

```
int i = 0;  
  
do {  
    System.out.println(i);  
    i++;  
} while (i < 10);
```

Этот код также будет выводить на экран числа от 0 до 9.

## fori

Цикл for - это наиболее часто используемый тип цикла в Java. Он имеет следующую синтаксис:

```
for (инициализация; условие; итерация) {  
    // ...  
}
```

1. Инициализация выполняется один раз, когда цикл запускается. Она может использоваться для инициализации переменной цикла.
2. Условие проверяется перед каждым повторением цикла. Если условие истинно, цикл выполняется. Если условие ложно, цикл завершается.
3. Итерация выполняется после каждого повторения цикла. Она может использоваться для изменения переменной цикла или выполнения других действий.

Вот пример использования цикла for:

```
for (int i = 0; i < 10; i++) {  
    System.out.println(i);  
}
```

Этот код будет выводить на экран числа от 0 до 9.

## Итог

### Цикл for

Цикл `for` является наиболее гибким типом цикла в Java. Он может использоваться для повторения блока кода произвольное количество раз.

#### Вот несколько советов по использованию цикла `for`:

- Используйте цикл `for`, когда вы знаете, сколько раз нужно повторить блок кода.
- Используйте переменную цикла для отслеживания количества повторений.
- Используйте итератор для выполнения действий на каждом повторении.

## Цикл `while`

Цикл `while` используется для повторения блока кода, пока условие истинно.

#### Вот несколько советов по использованию цикла `while`:

- Используйте цикл `while`, когда вы не знаете, сколько раз нужно повторить блок кода.
- Используйте условие для проверки, нужно ли повторять блок кода.
- Используйте переменную цикла для отслеживания количества повторений.

## Цикл `do-while`

Цикл `do-while` используется для повторения блока кода, по крайней мере, один раз.

#### Вот несколько советов по использованию цикла `do-while`:

- Используйте цикл `do-while`, когда вы хотите выполнить блок кода хотя бы один раз.
- Используйте условие для проверки, нужно ли повторять блок кода.
- Используйте переменную цикла для отслеживания количества повторений.

## Массив:

**Массив** - это структура данных, которая хранит данные одного типа в виде линейной последовательности.

## Основные характеристики массивов:

**Тип данных:** Массив может содержать только данные одного типа. **Размер:** Массив имеет фиксированный размер, который определяется при его создании. **Индексы:** Каждому элементу массива присваивается уникальный индекс, который используется для доступа к нему.

## Создание массива

Массивы в Java создаются с помощью ключевого слова `new`.

```
int[] array = new int[10];
```

#### Другие способы создания массивов, с заполнением данными при создании

```
int[] marks = {2, 5, 4, 3, 5, 3, 5, 4, 1}; // так тоже можно задавать массив  
int[] array = new int[]{2, 5, 4, 3, 5, 3, 5, 4, 1} // так тоже можно задавать массив
```

Этот код создаст массив из 10 целых чисел.

## Доступ к элементам массива

Элементы массива можно получить по их индексу.

```
int element = array[0];
```

Этот код вернет **первый** элемент массива.

## Изменение элементов массива

```
array[0] = 10;
```

Этот код установит значение 10 для первого элемента массива.

## Значение по умолчанию

**Значение по умолчанию** - это значение, которое присваивается переменной, если оно не было явно задано

В Java значения по умолчанию для переменных определяются их типом данных.

### Значения по умолчанию для примитивных типов данных:

- Целочисленные типы: '0'
- Действительные типы: '0.0'
- Логический тип: 'false'
- Тип char: '\u0000'
- ссылочных типы, объекты: 'null'

Например, следующий код создаст массив из 10 целых чисел, все элементы которого будут иметь значение 0:

```
int[] array = new int[10];
```

В этом примере элементы массива будут иметь значение по умолчанию для целочисленного типа данных, то есть 0.

## Массив целых чисел:

```
int[] array = new int[10];
```

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

## Массив вещественных чисел:

```
double[] array = new double[10];

for (int i = 0; i < array.length; i++) {
    System.out.println(array[i]); // Выведет 0.0 0.0 0.0 0.0 ... 0.0
}
```

## Массив логических значений:

```
boolean[] array = new boolean[10];

for (int i = 0; i < array.length; i++) {
    System.out.println(array[i]); // Выведет false false false false ... false
}
```

## Массив символов:

```
char[] array = new char[10];

for (int i = 0; i < array.length; i++) {
    System.out.println(array[i]); // Выведет \u0000 \u0000 \u0000 \u0000 ... \u0000
}
```

## Удаление элементов массива

Элементы массива можно удалить с помощью цикла for.

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

Этот код удалит все элементы массива, равные 10.

## Вот несколько советов по использованию массивов:

- Используйте массивы, когда вам нужно хранить большое количество данных одного типа.
- Используйте цикл for, чтобы получить доступ к каждому элементу массива.

## Алгоритмы массивов:

- Bubble Sort
- Binary search

## Bubble Sort

Метод сортировки пузырьком - это простой алгоритм сортировки, который работает путем сравнения соседних элементов массива и обмена их местами, если они находятся в неправильном порядке. Этот процесс повторяется до тех пор, пока все элементы массива не будут отсортированы.

# Binary search

Бинарный поиск - это алгоритм сортировки, который работает путем последовательного уменьшения области поиска. Алгоритм начинается с середины массива и сравнивает целевое значение с элементом в середине. Если целевое значение меньше элемента в середине, область поиска сужается до левой половины массива. Если целевое значение больше элемента в середине, область поиска сужается до правой половины массива. Этот процесс повторяется до тех пор, пока целевое значение не будет найдено или пока область поиска не станет пустой.

## Homework

### ► English

### ▼ На русском

## Задача 1

Определить закономерность, согласно которой формируется та или иная числовая последовательность. Написать методы, которые формируют первые N элементов данной последовательности в виде целочисленного массива и выводит элементы массива на экран.

### Дано:

- последовательность 1: 2, 4, 6, 8, 10, 12, 14, 16, 18, 20
- последовательность 2: 0, 0, 2, 0, 4, 0, 6, 0, 8, 0
- последовательность 3: 0, 2, 0, 6, 0, 10, 0, 14, 0, 18

## Задача 2

Найти элемент **N** в массиве из сгенерированных чисел и заменить его на минимальный элемент в массиве. Если **N** не найден в массиве вывести сообщение, что элемент не найден!

пример: дано: {10,4,**2**,3, 1}, **N** = 2;. Результат: {10,4,**1**,3, 1}

## Задача 3

Является ли массив строго возрастающей последовательностью

пример: 30, **31**, **31**, 39 - не является строго возрастающей последовательностью  
пример: 30, 31, 32, 39 - является строго возрастающей последовательностью

## Задача 4

Дан массив чисел, найдите все четные числа в массиве и запишите эти числа в другой массив. Отсортируйте только \*\*второй\*\* массив и распечатайте оба массива.

## Задача 5

Дано два массива. Нужно в первом массиве найти индексы всех чисел из второго массива. Найти индексы всех чисел из второго массива в первом массиве

пример: arr1 = {1, -10, 70, 101} arr2 = {1, 99}

Вывод: Число 1, найдено, индекс 0 Число 99, индекс не найдено

[//: # \(\)](#) [//: # \(\)](#) [//: # \(\)](#) [//: # \(\)](#) [//: # \(\)](#) [//: # \(\)](#)

### Code

code/HWSolution08/src/CarRentalService.java

```
public class CarRentalService {
    public static void main(String[] args) {
        char carType = 'E';
        int days = 5;
        int age = 25;
        int experience = 3;
        char insurance = '0';
        boolean isVip = true;
        boolean hasCoupon = false;
        boolean wantsGps = true;
        boolean wantsChildSeat = false;

        if (!canRent(carType, age, experience, isVip)) {
            System.out.println("Клиент не может арендовать этот тип автомоб

        } else {
            double baseCost = calculateBaseCost(carType, days);
            int discount = calculateDiscount(days, isVip, hasCoupon);
            double insuranceCost = calculateInsurance(insurance);
            double extras = calculateExtras(wantsGps, wantsChildSeat);

            double totalCost = baseCost * (1 - discount / 100.0) + insuranceCost;

            if (discount > 25) {
                System.out.println("Вы получили купон на следующий прокат!"
                totalCost = baseCost * 0.75 + insuranceCost + extras;
```

```
    }

    System.out.println("Общая стоимость: " + totalCost);
}

}

public static double calculateBaseCost(char carType, int days) {
    double dailyRate = 0;
    switch (carType) {
        case 'E':
            dailyRate = 100;
            break;
        case 'B':
            dailyRate = 200;
            break;
        case 'L':
            dailyRate = 300;
            break;
    }
    return dailyRate * days;
}

public static boolean canRent(char carType, int age, int experience, boolean isVip) {
    if (carType == 'L') {
        if (isVip && age >= 20) {
            return true;
        }
        if (age >= 23 && experience >= 5) {
            return true;
        }
        return false;
    }
    return true;
}

// return carType == 'L' ? (isVip && age >= 20) || (age >= 23 && experience >= 5) : true;

public static int calculateDiscount(int days, boolean isVip, boolean hasLicense) {
    int discount = 0;
    if (days > 30) {
        discount += 10;
    } else if (days > 7) {
        discount += 6;
    }
}
```



```
    } else if (days > 3) {
        discount += 4;
    }

    if (isVip) {
        discount += 12;
    }

    if (hasCoupon) {
        discount += 5;
    }

    return discount;
}

public static double calculateInsurance(char insurance) {
    switch (insurance) {
        case 'O':
            return 20;
        case 'K':
            return 50;
        default:
            return 0;
    }
}

public static double calculateExtras(boolean wantsGps, boolean wantsChildSeat) {
    double extras = 0;
    if (wantsGps) {
        extras += 10;
    }
    if (wantsChildSeat) {
        extras += 5;
    }
    return extras;
}
}
```

code/HWSolution08/src/CinemaTicket.java

```
/**
 * @author Andrej Reutow
 * created on 21.09.2023
 */
```

```

* <p>
* В кинотеатре есть разные типы билетов: стандартный, студенческий и для п
* стоимость. Клиент может также выбрать место в VIP-зоне, что увеличит сто
* Требования:
* Создать метод calculateTicketPrice, который принимает возраст клиента, с
* Использовать if-else и & для определения стоимости билета.
* </p>
*/

```

```

public class CinemaTicket {
    public static void main(String[] args) {
        System.out.println("Цена билета: " + calculateTicketPrice(25, false
        System.out.println("Цена билета: " + calculateTicketPrice(17, false
        System.out.println("Цена билета: " + calculateTicketPrice(65, false
        System.out.println("Цена билета: " + calculateTicketPrice(20, true,

    }

    public static int calculateTicketPrice(int age, boolean isStudent, bool
        int price = 0;

        if (age < 18) {
            price = 100;
        } else if (age >= 18 && age <= 64) {
            if (isStudent) {
                price = 150;
            } else {
                price = 200;
            }
        } else {
            price = 150;
        }

        if (isVip) {
            price += 50;
        }

        return price;
    }
}

```

code/HWSolution08/src/PhoneShop.java

```

/**
 * @author Andrej Reutow
 * created on 21.09.2023

```

```
* <p>
* Вася хочет купить себе новый телефон. В магазине он видит телефоны разные
* Задание:
* Напишите программу, которая поможет Васе выбрать телефон, подходящий по цене
*/

public class PhoneShop {
    public static void main(String[] args) {

        // Вводим цену, которую хочет заплатить Вася
        int budget = 40000;

        boolean found = false;
        if (budget >= 10000) {
            System.out.println("Вася может купить телефон за 10000 евро.");
            found = true;
        }
        if (budget >= 20000) {
            System.out.println("Вася может купить телефон за 20000 евро.");
            found = true;
        }
        if (budget >= 30000) {
            System.out.println("Вася может купить телефон за 30000 евро.");
            found = true;
        }
        if (budget >= 40000) {
            System.out.println("Вася может купить телефон за 40000 евро.");
            found = true;
        }

        if (!found) {
            System.out.println("К сожалению, Вася не может позволить себе ни один из этих телефонов.");
        }

        sellPhone();
    }

    public static void sellPhone() {
        int[] price = {10000, 20000, 30000, 40000};
        int budget = 40000;
        for (int i = 0; i < price.length; i++) {
            if (budget >= price[i]) {
                System.out.println("Вася может купить телефон за " + price[i]);
            }
        }
    }
}
```

```
    }  
  
    }  
}
```

code/HWSolution08/src/PizzaDelivery.java

```
public class PizzaDelivery {  
  
    public static boolean canDeliver(int distance, boolean isRegularCustomer,  
        int deliveryDistance = isRegularCustomer ? 10 : 5;  
    if (distance <= deliveryDistance) {  
        return true;  
    } else {  
        return false;  
    }  
}  
  
    public static void main(String[] args) {  
        System.out.println("Можем доставить? " + canDeliver(4, false)); //  
        System.out.println("Можем доставить? " + canDeliver(7, true)); //  
        System.out.println("Можем доставить? " + canDeliver(7, false)); //  
        System.out.println("Можем доставить? " + canDeliver(11, true)); //  
    }  
}
```

code/HWSolution08/src/Seller.java

```
/**  
 * @author Andrej Reutow  
 * created on 21.09.2023  
 * <p>  
 * У продавца есть товар, который он хочет продать. Он может продать товар :  
 * эту цену. Если покупатель не согласен на эту цену, продавец может предло  
 * Если покупатель не  
 * согласен и на эту цену, продавец может предложить скидку в 20%. Если пок  
 * отказывается продавать товар.  
 */  
public class Seller {  
  
    public static void main(String[] args) {
```

```
        sell(100); // Товар продан
        sell(90); // Скидка 10%
        sell(70); // Товар не продан
    }

    public static void sell(int offer) {
        int price = 100;

        if (offer >= price) {
            System.out.println("Товар продан");
        } else if (offer > price * 0.9) {
            System.out.println("Скидка 10%");
            System.out.println("Новая цена: " + price * 0.9);
        } else if (offer > price * 0.8) {
            System.out.println("Скидка 20%");
            System.out.println("Новая цена: " + price * 0.8);
        } else {
            System.out.println("Товар не продан");
        }
    }
}
```

code/Lesson\_14/src/WhileLoop.java

```
import java.util.Scanner;

/**
 * @author Andrej Reutow
 * created on 22.09.2023_000_000
 */
public class WhileLoop {

    public static void main(String[] args) {
        // 0 ... 3_000_000

        //      int number = 10;
        //      boolean condition = number <= 100;
        //      while (condition) {
        //          System.out.println(number); // 10 ... 100
        //          number++;
        //      }
    }
}
```

```
//      }

//      do {
//          System.out.println(number);
//          number++;
//      } while (false); // 10 <= 5 - false

Scanner scanner = new Scanner(System.in);
int selectedMenu = 3;

int arrayL = scanner.nextInt(); // 5
int[] array = new int[arrayL]; // 1 = 5

System.out.println("Введите желаемый индекс:");
int setIndex = scanner.nextInt(); // 2
int value = scanner.nextInt(); // 777
array[setIndex] = value; // {0, 0, 777, 0, ... }

//      do {
//          System.out.println("1 - спать");
//          System.out.println("2 - вставать");
//          System.out.println("3 - выход");
//          selectedMenu = scanner.nextInt();
//      } while (selectedMenu != 3); // false, 3 -> 1 1 != 3 = true

//      while (selectedMenu != 3) { // false
//          System.out.println("1 - спать");
//          System.out.println("2 - вставать");
//          System.out.println("3 - выход");
//          selectedMenu = scanner.nextInt();
//      }

menu();
System.out.println("END");

int a = 1;
int b = 1;
add(a, b);
}

public static void menu() {
```

```

Scanner scanner = new Scanner(System.in);
int selectedMenu = 0;
while (true) { // false
    System.out.println("1 - спать");
    System.out.println("2 - вставать");
    System.out.println("3 - выход");
    selectedMenu = scanner.nextInt();
    if (selectedMenu == 3) {
        break;
    }
}

public static void add(int a, int b) {
    int result = a + b;
    System.out.println(result);
}
}

```

code/Lesson\_14/src/Task1.java

```

import java.util.Arrays;

/**
 * @author Andrej Reutow
 * created on 22.09.2023
 * <p>
 * Определить закономерность, согласно которой формируется та или иная числ
 * Написать методы, которые формируют первые N элементов данной последовате
 * <p>
 * Дано:
 * <p>
 * последовательность 1: 2, 4, 6, 8, 10, 12, 14, 16, 18, 20
 * последовательность 2: 0, 0, 2, 0, 4, 0, 6, 0, 8, 0
 * последовательность 3: 0, 2, 0, 6, 0, 10, 0, 14, 0, 18
 * </p>
 */
public class Task1 {

    public static void main(String[] args) {
        int arrayLength = 10; // число N из условия задачи

        int[] array = new int[arrayLength]; // длина массива 10
        // 2, 4, 6, 8, 10, 12, 14, 16, 18, 20
    }
}

```

```

        sequence1(array);

        System.out.println(Arrays.toString(array));
    }

    // 2, 4, 6, 8, 10, 12, 14, 16, 18, 20
    public static void sequence1(int[] array) {
        array[0] = 2;
        for (int i = 0; i < array.length - 1; i++) {
            array[i + 1] = array[i] + 2;
        }

        //      for (int i = 1; i < array.length; i++) {
        //          array[i] = array[i - 1] + 2;
        //      }
    }
}

```

code/Lesson\_14/src/Task2.java

```

import java.util.Arrays;

/**
 * @author Andrej Reutow
 * created on 22.09.2023
 *
 * <p>
 * <p>
 * Найти элемент N в массиве из сгенерированных чисел и заменить его на мин
 * <p>
 * пример: дано: {10,4,2,3, 1}, N = 2;. Результат: {10,4,1,3, 1}
 * </p>
 */
public class Task2 {
    //1. создать массив и заполнить его элементами (с случайными числами)

    //2. указать значение числа N, n - это искомое число

    //3. найти индекс минимально числа в массиве и сохранить найденный индекс

    //4. найти в массиве число n и если оно найдено, сохранить индекс найденного

```



//4.1 если число n не найдено, вывести в консоль сообщение, о том что ,

//5. заменить число n в массиве на минимальное число из массива.

//6. распечатать массив используя (System.out.println(Arrays.toString(a

```
public static void main(String[] args) {
    int[] array = {10, 4, 2, 3, 1};
    int n = 10;

    int minIndex = findMin(array);

    int searchNumberIndex = findIndex(array, n); // метод findIndex(...)

    if (searchNumberIndex >= 0) { // заменяем элемент, только если индекс найден
        // 5. заменить число n в массиве на минимальное число из массива
        //todo заменить элементы
    }

    System.out.println(Arrays.toString(array));
}
```

//3. найти индекс минимально числа в массиве и сохранить найденный индекс

```
private static int findMin(int[] array) {
    // int[] copy = Arrays.copyOf(array, array.length); // копирует массив
    /*
    Arrays.copyOf(array, array.length)
```

по капотом происходит копирования данных входящего массива в новый.  
примерно вот так:

```
    public static int[] copyOf(int[] array, int newLength) {
        int[] copy = new int[newLength];
        for (int i = 0; i < newLength; i++) {
            copy[i] = array[i];
        }
        return copy;
    }
    */
```

```
//todo найти минимальный элемент в массиве array, и вернуть его индекс
return 0;
```

```
/*
```

Как найти минимальный элемент в массиве:

Шаг 1: Установка начального значения минимума

Инициализируйте переменную `minElement`, которая будет хранить

Шаг 2: Обход массива

Пройдитесь по всем элементам массива, начиная со второго (и

Шаг 3: Вывод результата

После завершения цикла `minElement` будет содержать минимальный

\*/

}

//4. найти в массиве число `n` и если оно найдено, сохранить индекс найденного

//4.1 если число `n` не найдено, вывести в консоль сообщение, о том что не

`private static int findIndex(int[] array, int value) {`

`// todo найти элемент по его значению и вернуть его индексу.`

`return -1;`

`}`

`}`