

**МИНИСТЕРСТВО ЦИФРОВОГО РАЗВИТИЯ СВЯЗИ И МАССОВЫХ  
КОММУНИКАЦИЙ**

**Ордена Трудового Красного Знамени**

**Федеральное государственное бюджетное образовательное учреждение  
высшего образования**

**«Московский технический университет связи и информатики»**

**Кафедра «Математическая Кибернетика и Информационные технологии»**

**Дисциплина «Информационные технологии и программирование»**

**Лабораторная работа №1**

**Знакомство с Java**

**Выполнила: Студентка группы**

**БВТ2303**

**Васюкова Наталья**

**Москва**

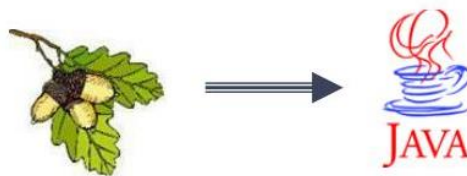
**2024**

## Цель работы:

Знакомство с синтаксисом и основными концепциями языка Java – классами, методами, объектами. Выполнение тренировочных заданий.

## Основная теория:

Язык Java был разработан в 1991 году компанией Sun Microsystems, впоследствии приобретенной компанией Oracle. Ведущий разработчик - Джеймс Гослинг. Изначально язык назывался Oak (Дуб), но позднее был переименован в Java. Java получил своё название от марки кофе Java, поэтому на эмблеме языка изображена чашка.



Java – объектно-ориентированный язык программирования (компилируемый и интерпретируемый). Синтаксис языка унаследован от C++.

В настоящий момент язык используется в:

- 1) Enterprise: тяжелые серверные приложения для банков, корпораций, инвестфондов и т.д.
- 2) Web backend
- 3) Big Data: распределенные вычисления в кластерах из тысяч серверов.
- 4) Embedded Systems: микропроцессоры
- 5) Smart Devices: программы для умного дома, электроники, холодильников с выходом в интернет

Важные определения:

### 1. Класс

Класс в Java — это шаблон или чертеж для создания объектов. Можно представить класс как план здания: он описывает, как должно выглядеть здание, но само здание еще не построено.

Пример класса:

```
public class Car {  
    // Поля класса  
    String color;  
}
```

```

String model;
int year;

// Конструктор
public Car(String color, String model, int year) {
    this.color = color;
    this.model = model;
    this.year = year;
}

// Метод
public void displayInfo() {
    System.out.println("Модель: " + model + ", Цвет: " + color + ",
Год: " + year);
}

```

## 2. Объект класса

Объект — это конкретный экземпляр класса. Если класс — это чертеж, то объект — это фактическое здание, построенное по этому чертежу.

Создание объекта:

```

public class Main {
    public static void main(String[] args) {
        // Создание объекта класса Car
        Car myCar = new Car("Красный", "Toyota", 2020);

        // Использование метода
        myCar.displayInfo();
    }
}

```

## 3. Метод

Метод — функция, описанная внутри класса. Он определяет поведение объекта. Метод может что-то делать (в данном случае выводить информацию об автомобиле) и, при необходимости, принимать параметры и возвращать значения.

Как работает метод в примере:

Мы объявили метод `displayInfo()` внутри класса `Car`, который выводит информацию о конкретном автомобиле.

Когда мы вызываем `myCar.displayInfo();`, программа выполняет код внутри метода и выводит информацию об объекте `myCar`.

Метод main или точка входа в программу

```
public class JavaHelloWorldProgram {  
    public static void main(String args[]) {  
        System.out.println("Hello World");  
    }  
}
```

public static void main – комбинация "psvm" должна работать в IDE как горячие клавиши.

1) public (+-) - модификатор доступа: доступ ничем не ограничен. Метод доступен как внутри самого класса, так и из других классов.

2) static (+-) - метод статический: он принадлежит только классу, а не объекту (а не экземплярам). Инициализируется первым. Статический метод можем вызывать без создания объекта. Обычно он используется для выполнения операций, которые не зависят от состояния конкретного объекта. К ним можно обращаться через имя класса, а не через объект. Класс описывает структуру, а объект - конкретный элемент класса, имеющий какие-то свойства.

Не статический (экземплярный) метод принадлежит конкретному объекту класса. Чтобы его вызвать, сначала нужно создать объект.

3) void (+) - "пустота", тип возвращаемого значения, он должен быть описан всегда. Void в случае если ничего не возвращаем. Вместо него может быть int например. Если мы что-то возвращаем, то нужен return, а void не обязателен.

4) main - точка входа в программу. При запуске JVM ищет именно этот метод, чтобы начать выполнение программы. Внутри метода main код, который будет выполняться при запуске приложения.

### Ход работы:

**Задание 1:** Создайте программу, которая находит и выводит все простые числа меньше 100.

```
public class Primes { // объявляем класс Primes  
    public static void main(String[] args) { // void - метод не  
        // возвращает никакого значения.  
        // String[] - метод  
        // принимает массив строк.
```

```

        for (int i = 2; i < 100 + 1; i++) { // цикл создает переменную i=2
и будет выполняться, пока i<100. i++ – после каждой итерации значение i
будет увеличиваться на 1.
            if (isPrime(i)) { // вызываем метод isPrime(i), чтобы
проверить, является ли это число i простым
                System.out.println(i); // если i простое, то оно
выводится на экран
            }
        }
    }

    public static boolean isPrime(int n) { // Определяем метод isPrime,
он принимает целое n и возвращает boolean (true/false) и говорит является
ли n простым
        if (n < 2) // синтаксис "меньше либо равно" - (n <= 2)
            return false; // простые числа начинаются с 2
        for (int i = 2; i < n; i++) { // (ЛИБО i <= Math.sqrt(n)); цикл
перебирает числа от 2 до квадратного корня из n
            if (n % i == 0) {
                return false; // если n делится хотя бы на одно из этих
i, то оно не простое (составное)
            }
        }
        return true; // если не нашлся ни один делитель, то число
простое
    }
}

```

**Задание 2:** Создайте программу, которая определяет, является ли введенная строка палиндромом.

```

// Программа может работать как с заглавными, так и со строчными буквами!
public class Palindrome {
    public static void main(String[] args) { // args - параметр который
можно указать при запуске через cmd. Например java Palindrome.java
Amogusugoma 848 rar
        for (int i = 0; i < args.length; i++) {
            String s = args[i];
            if (isPalindrome(s)) {
                System.out.println(s + " - yes");
            } else {
                System.out.println(s + " - no");
            }
        }
    }

    public static String reverseString(String s) {

```

```
        String reversed = "";
        for (int i = s.length() - 1; i >= 0; i--) { // с индекса
последнего символа (s.length() - 1) до индекса первого (0)
            reversed += s.charAt(i);
        }
        return reversed;
    }

    public static boolean isPalindrome(String s) {
        String reversed = reverseString(s);
        s = s.toLowerCase(); // .toLowerCase() - перевести всю строку в
нижний регистр
        reversed = reversed.toLowerCase();
        return s.equals(reversed); // .equals() сравнивает строки
    }
}
```

### **Вывод:**

Мы познакомились с языком программирования Java, узнали о его особенностях, изучили понятия класс, объект, метод, а также решили две задачи с использованием языка.