**История и особенности Java**

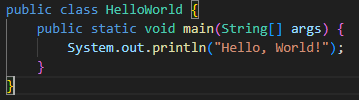
Java — это универсальный объектно-ориентированный язык программирования, разработанный компанией Sun Microsystems (ныне принадлежащей Oracle) в 1995 году. Его ключевая концепция — «Write Once, Run Anywhere» (WORA), что означает способность одного и того же кода исполняться на любых устройствах, поддерживающих виртуальную машину Java (JVM).

Язык был создан с упором на простоту, безопасность и кроссплатформенность. Это сделало Java выбором номер один для корпоративного программирования, мобильных приложений (например, Android), разработки серверных решений и веб-приложений.

**Основные преимущества Java:**

* Платформонезависимость за счет JVM.
* Поддержка объектно-ориентированного программирования.
* Богатая стандартная библиотека (Java Standard Edition, SE).
* Автоматическое управление памятью (Garbage Collection).

Каждое Java-приложение начинается с класса и метода main, который является точкой входа. Пример простой программы:



**Объяснение кода:**

1. public class HelloWorld — объявление класса с именем HelloWorld.
2. public static void main(String[] args) — метод main запускает приложение.
3. System.out.println("Hello, World!"); — вывод строки в консоль.

**Основы объектно-ориентированного программирования (ООП)**

Java следует принципам ООП, которые помогают создавать гибкие и масштабируемые программы. Четыре основополагающих принципа ООП:

**Инкапсуляция**  
Позволяет скрывать детали реализации и предоставлять доступ к данным через методы. Это достигается с помощью модификаторов доступа:

* + public — доступ открыт для всех.
  + private — доступ только внутри класса.
  + protected — доступ в пределах пакета или наследуемых классов.

**Наследование**  
Позволяет создавать новый класс на основе существующего. Новый класс (наследник) получает доступ к полям и методам родительского класса. Используется ключевое слово extends.

public class Animal {

    public void makeSound() {

        System.out.println("Animal makes a sound");

    }

}

public class Dog extends Animal {

    @Override

    public void makeSound() {

        System.out.println("Dog barks");

    }

}

**Полиморфизм**  
Способность объектов одного типа принимать различные формы. Реализуется через переопределение методов и интерфейсы. Пример

Animal animal = new Dog();

animal.makeSound(); // Вызывает метод Dog

**Структура программы в Java**

Программы состоят из следующих элементов:

1. **Классы и объекты.**  
   Класс — это шаблон, объект — экземпляр класса.
2. **Поля и методы.**  
   Поля — переменные внутри класса, методы — функции, выполняющие действия.
3. **Пакеты.**  
   Пакеты группируют классы для структурирования программы.

**Абстракция**  
Сокрытие деталей реализации с помощью абстрактных классов или интерфейсов.

public abstract class Shape {

    abstract void draw();

}

public class Circle extends Shape {

    @Override

    void draw() {

        System.out.println("Drawing a circle");

    }

}