**МИНИСТЕРСТВО ЦИФРОВОГО РАЗВИТИЯ, СВЯЗИ И МАССОВЫХ КОММУНИКАЦИЙ РОССИЙСКОЙ ФЕДЕРАЦИИ**

**Ордена трудового Красного Знамени федеральное государственное бюджетное**

**образовательное учреждение высшего образования**

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

Кафедра Математическая кибернетика и информационные технологии

Лабораторная работа №1

**История языка программирования Java. Типы данных**

**по дисциплине  
«Информационные технологии и программирование»**

Выполнил: студент гр. БВТ2403

Куликов А.А.

Руководитель:

Москва, 2025 г.

**Цель работы:**изучить историю языка программирования Java, его особенности и типы данных. Реализовать программы для нахождения простых чисел и проверки строк на палиндром.

**Теория**:

Язык Java был разработан в начале 1990-х годов компанией Sun Microsystems. Основным разработчиком языка является Джеймс Гослинг. Изначально язык создавался для управления устройствами и интерактивного телевидения. Принцип WORA (Write Once, Run Anywhere) — это ключевая концепция, связанная с языком программирования Java и его экосистемой. Этот принцип подразумевает, что программа, написанная на Java, может быть скомпилирована в байт-код и затем выполнена на любой платформе, где установлена виртуальная машина Java (Java Virtual Machine, JVM), без необходимости в изменениях или повторной компиляции. Первая публичная версия языка была выпущена в 1995 году (Java 1.0). Язык Java быстро завоевал популярность благодаря своей платформенной независимости, обеспечиваемой виртуальной машиной Java. В последующие годы язык претерпел множество изменений и обновлений, включая добавление новых библиотек и улучшение производительности. В 2006 году Sun Microsystems открыла исходный код Java, что способствовало её дальнейшему развитию и распространению. В 2010 году Oracle Corporation приобрела Sun Microsystems и продолжила развивать Java.

Особенности Java:

* Простой
* Объектно-ориентированный
* Надежный
* Безопасный
* Не зависящий от архитектуры компьютера
* Переносимый
* Многопоточный

Стоит запомнить, что:

• исходный файл Java может содержать несколько классов, но допускается только один public (это модификатор доступа) класс;

• имя исходного файла Java должно совпадать с именем public класса. Поэтому имя файла программы — JavaHelloWorld Program.java;

• когда мы компилируем код, он генерирует байт-код и сохраняет его с расширением Class Name.class. Если посмотреть на каталог, в котором мы скомпилировали java-файл, то можно заметить новый созданный файл JavaHelloWorldProgram.class;

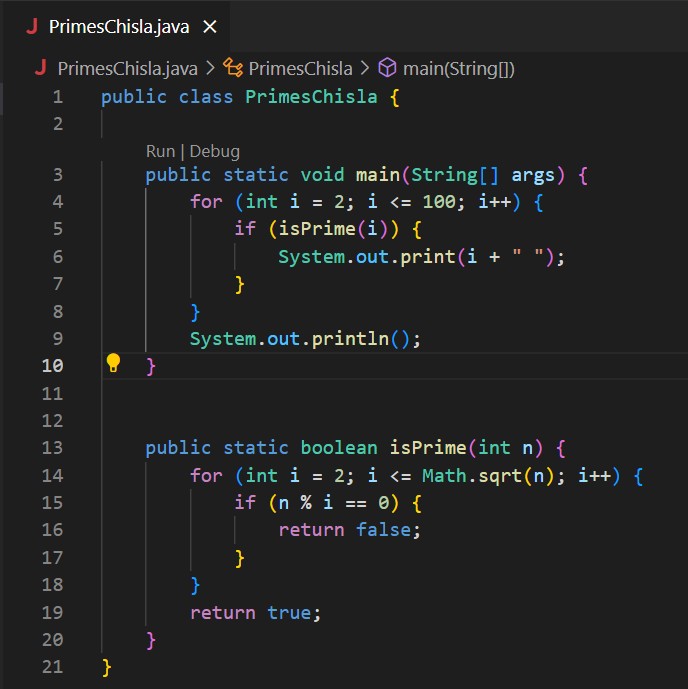
• при выполнении файла класса не нужно указывать полное имя файла. Нужно использовать только публичное имя класса;

• при запуске программы с помощью команды java она загружает класс в JVM, ищет в классе метод main и запускает его. Синтаксис метода main должен быть таким же, как указан в примере, иначе программа не будет запущена и выбросит исключение Exception в потоке "main".

**Индивидуальное задание:**

**Задание 1**

Условие: написать программу Primes.java, которая выводит все простые числа от 2 до 100.



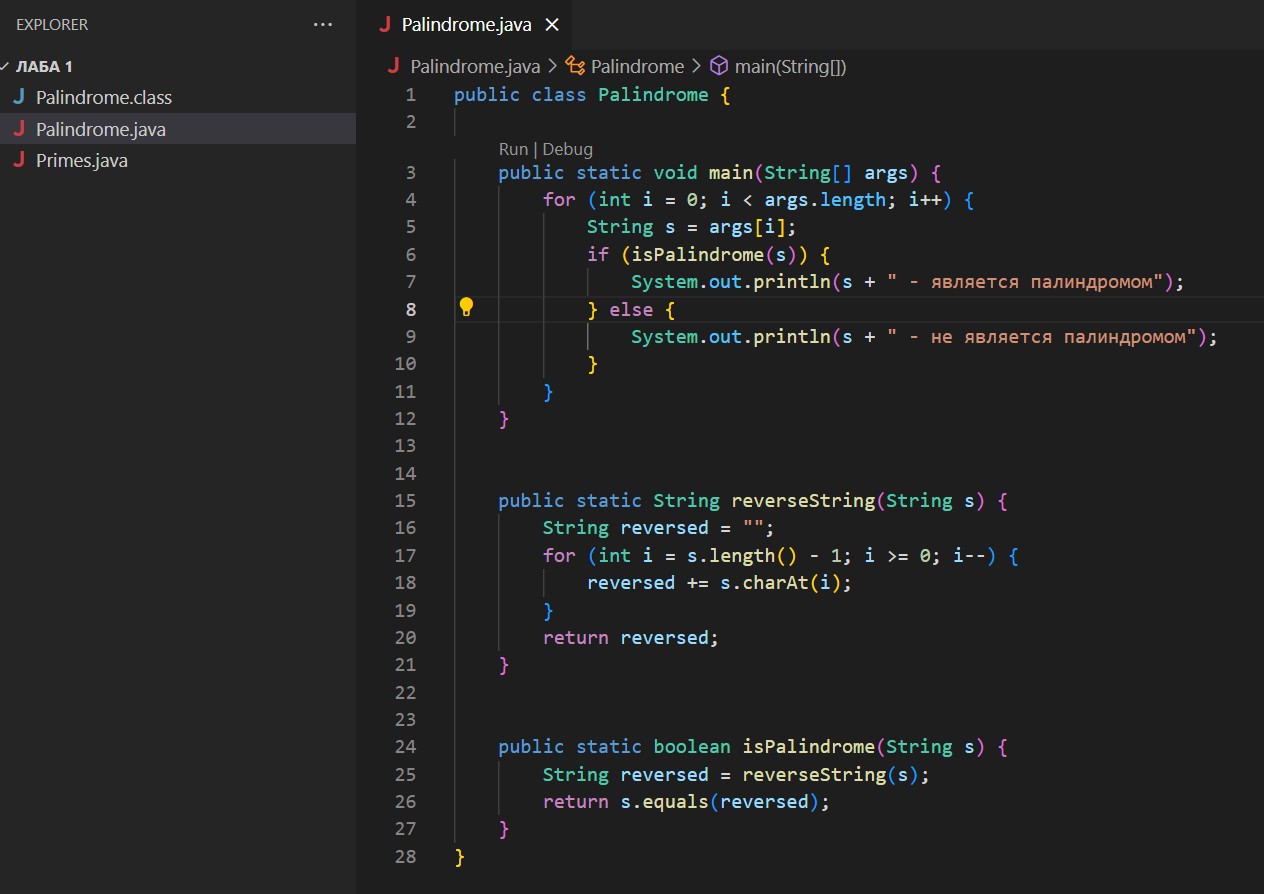
В этом задании я разработал программу для поиска простых чисел в диапазоне до 100. В основном методе организовал перебор чисел от 2 до 99, а для проверки каждого числа на простоту создал отдельный метод isPrime(). Этот метод проверяет, делится ли число нацело на какие-либо значения от 2 до корня из проверяемого числа.



Во втором задании мне нужно было создать программу, которая проверяет, является ли введённая строка палиндромом. Для этого я написал класс Palindrome, где в методе main() перебираются все аргументы, переданные программе через командную строку. Каждая строка проверяется методом isPalindrome(String s).

**Задание 2**

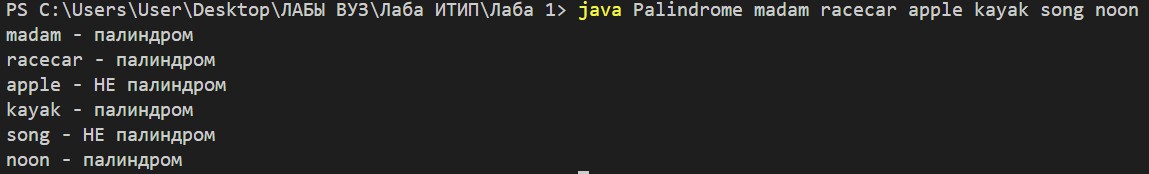
Условие: написать программу Palindrome.java, которая определяет, является ли строка палиндромом.



Во втором задании мне предстояло написать программу, которая проверяет, являются ли введенные строки палиндромами. Для этого я создал класс Palindrome с методом main(), который принимает аргументы командной строки. В программе я реализовал два вспомогательных метода:

* Метод reverseString(String s) создает перевернутую версию исходной строки, последовательно добавляя символы от конца к началу в новую строку.
* Метод isPalindrome(String s) сравнивает исходную строку с перевернутой версией с помощью метода equals()

Основной метод перебирает все аргументы командной строки и для каждого выводит сообщение о том, является ли строка палиндромом или нет.



После компиляции и запуска программы с тестовыми данными я получил этот результат.

**Вывод:**

В ходе работы я научился использовать методы и циклы, работать со строками и числами, а также компилировать и запускать код через JVM.

GitHub

https://github.com/LoopLip/ITIP-/tree/main/First%20Lab