**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

Лашков Ю.Е.

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.

**Индивидуальное задание:**1. Написать программу Primes.java, которая выводит все простые числа от 2 до 100.

2. Написать программу Palindrome.java, которая определяет, является ли строка палиндромом.

**Основная часть:**  
Язык программирования Java появился в начале 1990-х годов благодаря компании Sun Microsystems. Его главным разработчиком считается Джеймс Гослинг. Первоначально язык создавался для управления бытовыми устройствами и систем интерактивного телевидения. Однако ключевая идея Java — принцип WORA (Write Once, Run Anywhere) — сделала его универсальным. Это значит, что однажды написанная программа может выполняться на любой платформе, где установлена виртуальная машина Java (JVM), без необходимости изменения кода.

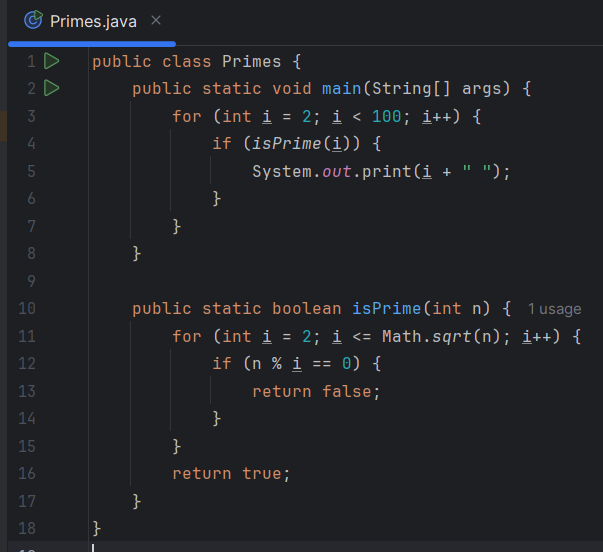
Первая версия Java была представлена в 1995 году и быстро завоевала популярность благодаря своей независимости от платформы. Позже язык получил развитие: в 2006 году исходный код был открыт, а в 2010 году Java перешла под руководство компании Oracle. Сегодня Java продолжает активно развиваться и остаётся одним из самых востребованных языков программирования.

Жизненный цикл программы на Java выглядит следующим образом: разработчик пишет исходный код в файле с расширением .java, затем с помощью компилятора javac этот код преобразуется в байт-код (.class). Дальше программу запускает JVM, которая интерпретирует или компилирует байт-код «на лету» в машинные инструкции (механизм JIT-компиляции). Такой подход обеспечивает переносимость и высокую производительность.

Java сочетает в себе несколько важных особенностей. Во-первых, это простота синтаксиса, унаследованного от C и C++. Во-вторых, объектно-ориентированная модель, позволяющая использовать наследование, полиморфизм и инкапсуляцию. В-третьих, надёжность и безопасность: строгая типизация, обработка исключений и автоматическая сборка мусора защищают программы от множества ошибок. Важной чертой является кроссплатформенность и возможность многопоточной работы.

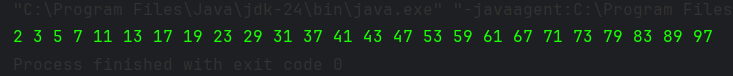
Типы данных в Java делятся на две группы: примитивные и ссылочные. Примитивные включают восемь базовых типов (byte, short, int, long, float, double, char, boolean) и хранят значения непосредственно в памяти. Ссылочные типы — это классы, массивы, строки и интерфейсы, которые содержат ссылки на объекты, расположенные в динамической памяти (куче). Такой подход позволяет объединять эффективность примитивов и гибкость работы с объектами.

**Реализация заданий лабораторной работы:**

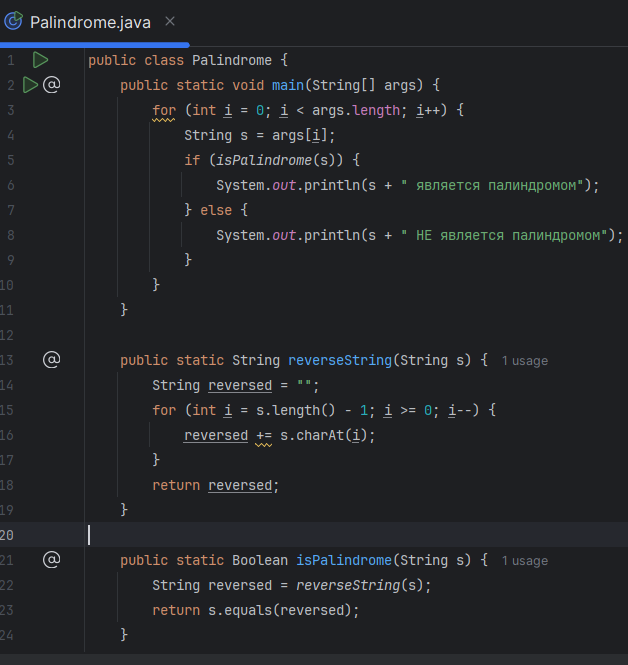
В первом задании мне предстояло написать программу, которая находит и выводит все простые числа меньше 100. Для этого я создал класс Primes с методом main(), в котором реализовал цикл for для перебора чисел от 2 до 99. Каждое число проверяется с помощью отдельного метода isPrime(int n).****

Метод isPrime выполняет проверку на простоту числа. Он перебирает все возможные делители числа от 2 до квадратного корня из n. Если находится хотя бы один делитель без остатка, метод возвращает false, что указывает на то, что число не является простым. Если же делителей не найдено, возвращается true, и число считается простым.

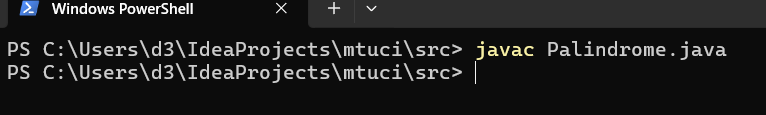
После компиляции и запуска программы я получил следующий результат:

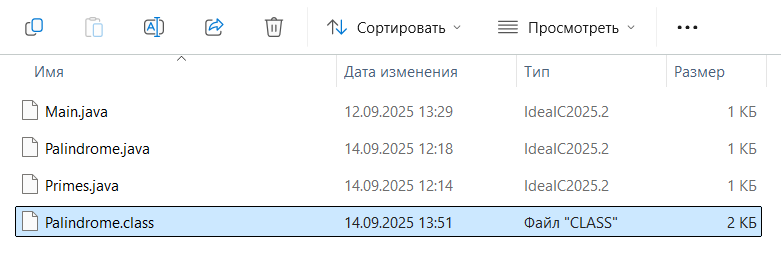
****

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).

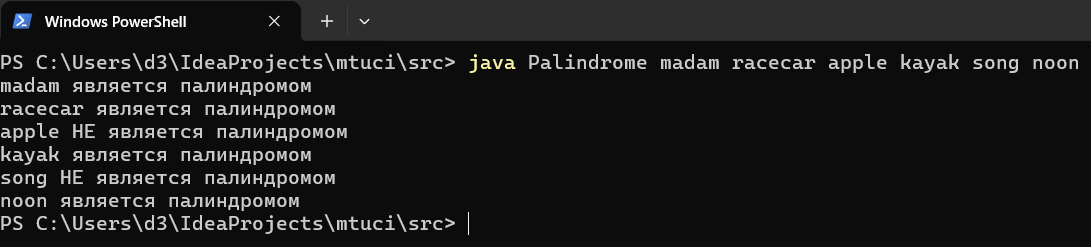
****

Метод isPalindrome использует вспомогательный метод reverseString(String s), который переворачивает строку посимвольно. Затем исходная строка сравнивается с перевёрнутой с помощью метода equals(). Если строки совпадают, возвращается true, и программа выводит, что строка является палиндромом; иначе — false и строка не является палиндромом.





После компиляции программы (javac Palindrome.java) и проверки содержимого директории, я убедился, что появился файл Palindrome.class — байт-код, готовый к выполнению на JVM. Далее я запустил программу в PowerShell с передачей нескольких строк:



Вывод программы соответствует ожидаемым результатам: все палиндромы определяются корректно, а строки, не являющиеся палиндромами, отмечаются как таковые.

**Вывод:**

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

<https://github.com/deylon999/ITAP/tree/master/1lab>