**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

Выполнил: студент группы БПИ2401

Исламов Эмин Маратович

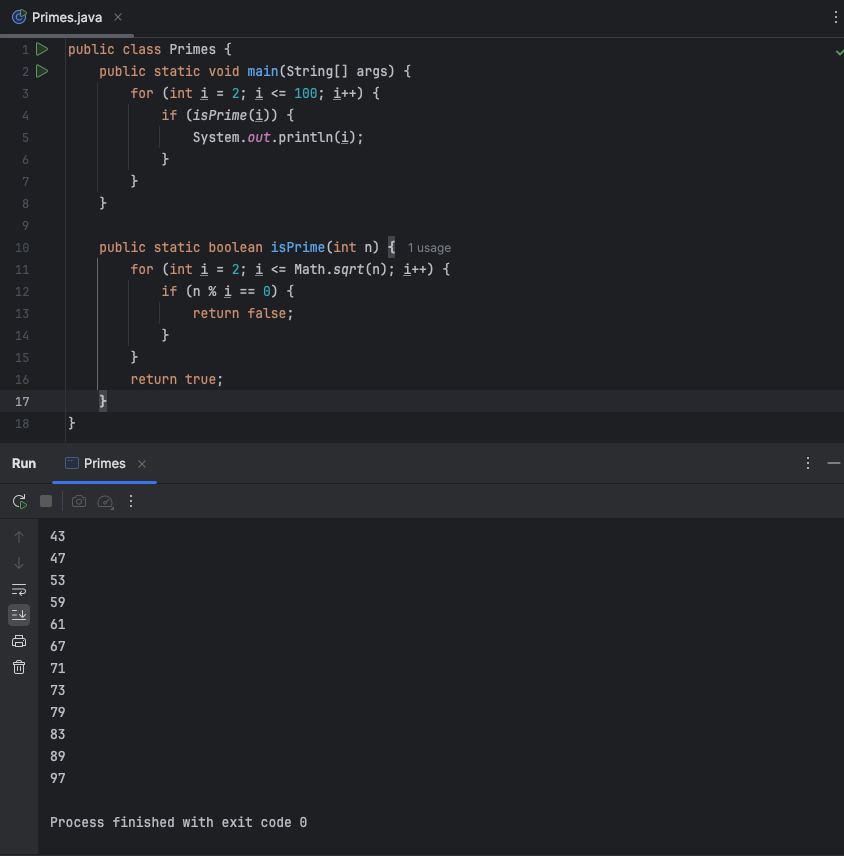
Проверил: Харрасов Камиль Раисович

**Цель работы:** закрепить навыки программирования на Java: использование циклов, методов и строковых операций для решения задач проверки чисел и строк.

**Ход работы:**

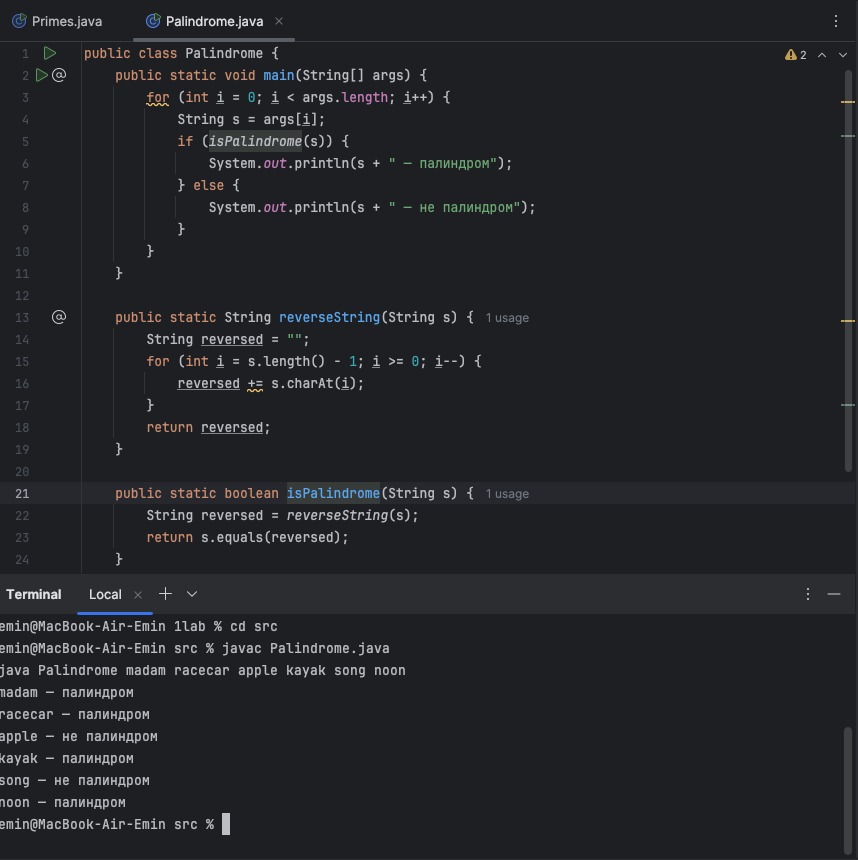
**Задание 1. Программа поиска простых чисел.**

Создаётся класс Primes с методом isPrime(int n), который с помощью цикла проверяет делимость числа от 2 до корня из n. В методе main() организован цикл перебора чисел от 2 до 100 и вывод только простых чисел.



**Задание 2. Программа проверки палиндрома.**

Создаётся класс Palindrome с методами reverseString(String s) для переворота строки и isPalindrome(String s) для сравнения исходной и перевёрнутой строки. В методе main() программа принимает слова из аргументов командной строки и выводит, является ли каждое из них палиндромом.



**Вывод:** в ходе работы были реализованы две программы: для поиска простых чисел и для проверки строк на палиндром. В процессе выполнения закреплены навыки создания классов, написания методов, работы с циклами и условиями, а также обработки строк и аргументов командной строки. Работа помогла лучше понять базовые конструкции языка Java и особенности его компиляции и запуска.

**Контрольные вопросы и ответы:**

**1. Java является компилируемым или интерпретируемым языком?**

Java сочетает оба подхода: исходный код сначала компилируется в байт-код (.class файлы), а затем этот байт-код интерпретируется (и частично компилируется JIT-компилятором) внутри JVM.

**2. Что такое JVM и для чего предназначается?**

JVM (Java Virtual Machine) — виртуальная машина Java, которая исполняет байт-код. Она обеспечивает платформенную независимость, управление памятью и безопасность выполнения программ.

**3. Каков жизненный цикл программы на языке Java?**

1. Написание исходного кода (.java).
2. Компиляция в байт-код (.class).
3. Загрузка байт-кода в JVM.
4. Исполнение программы JVM (интерпретация и JIT-компиляция).

**4. Какие виды типов данных есть в языке Java?**

* Примитивные типы (8 штук: byte, short, int, long, float, double, char, boolean).
* Ссылочные типы (классы, интерфейсы, массивы, объекты).

**5. Чем примитивные типы данных отличаются от ссылочных?**

Примитивные типы хранят конкретные значения в памяти. Ссылочные типы хранят только ссылку на объект, а сами данные размещаются в куче (heap).

**6. Как происходит преобразование примитивных типов в Java?**

* Неявное (widening conversion): при расширении диапазона (int → long).
* Явное (casting): при сужении диапазона (double → int). Для явного преобразования используется оператор (тип).

**7. Что такое байт-код в Java, и почему он важен для платформенной независимости?**

Байт-код — промежуточное представление программы, генерируемое компилятором Java. Он не зависит от операционной системы и процессора. Благодаря этому один и тот же .class файл может выполняться на любой платформе, где есть JVM.

**8. Какой тип данных используется для хранения символов в Java? Как представляются символы в памяти?**

Тип данных char. Символы в Java хранятся в формате Unicode (16 бит), что позволяет поддерживать символы разных языков.

**9. Что такое литералы в Java? Приведите примеры литералов для разных типов данных.**

Литералы — это фиксированные значения в коде.

Примеры:

* Целые: 42, 0x2A, 010.
* Вещественные: 3.14, 2.5e3.
* Символьные: 'a', '\n'.
* Строковые: "Hello".
* Логические: true, false.

**10. Почему Java считается строго типизированным языком?**

Потому что каждый объект и переменная имеют определённый тип, и компилятор проверяет совместимость типов на этапе компиляции. Это снижает риск ошибок во время выполнения.

**11. Какие проблемы могут возникнуть при использовании неявного преобразования типов?**

* Потеря точности (например, при преобразовании double в int).
* Переполнение при переходе в меньший тип (long в int).
* Неожиданные результаты при смешанных операциях (например, при работе с char и int).