**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

Отчет по “Информационные технологии и программирование”

Выполнил: студент группы: БПИ2403

Казанцев Арсений Александрович

Руководитель: Рыбаков Егор Дмитриевич

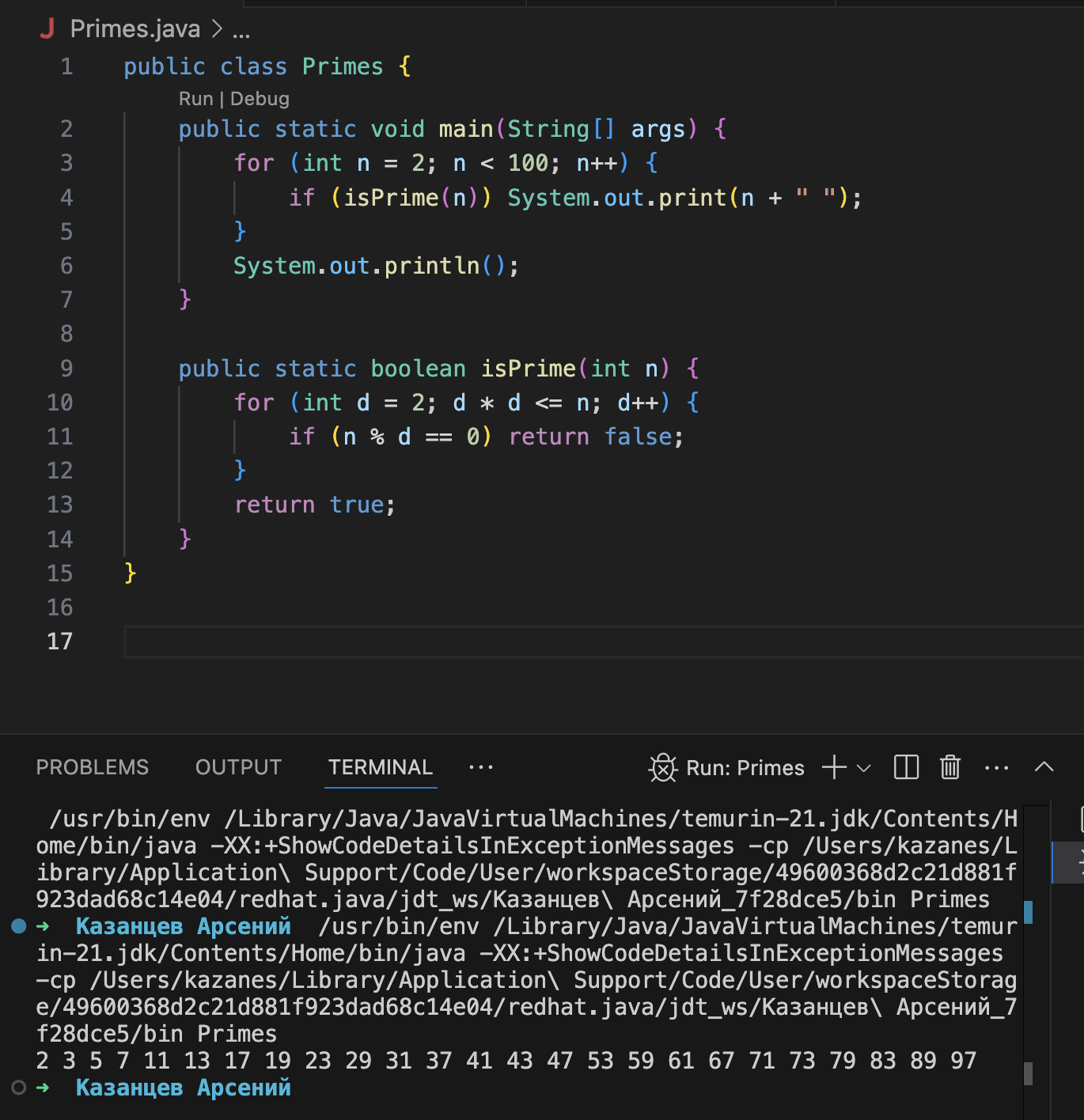
Москва, 2025

Требования к оформлению отчета

Структура отчета:

1. Титульный лист
2. Цель работы
3. Индивидуальное задание
4. Основная часть
5. Заключение

**Цель работы:** закрепить основы программирования на Java через работу с числами и строками.

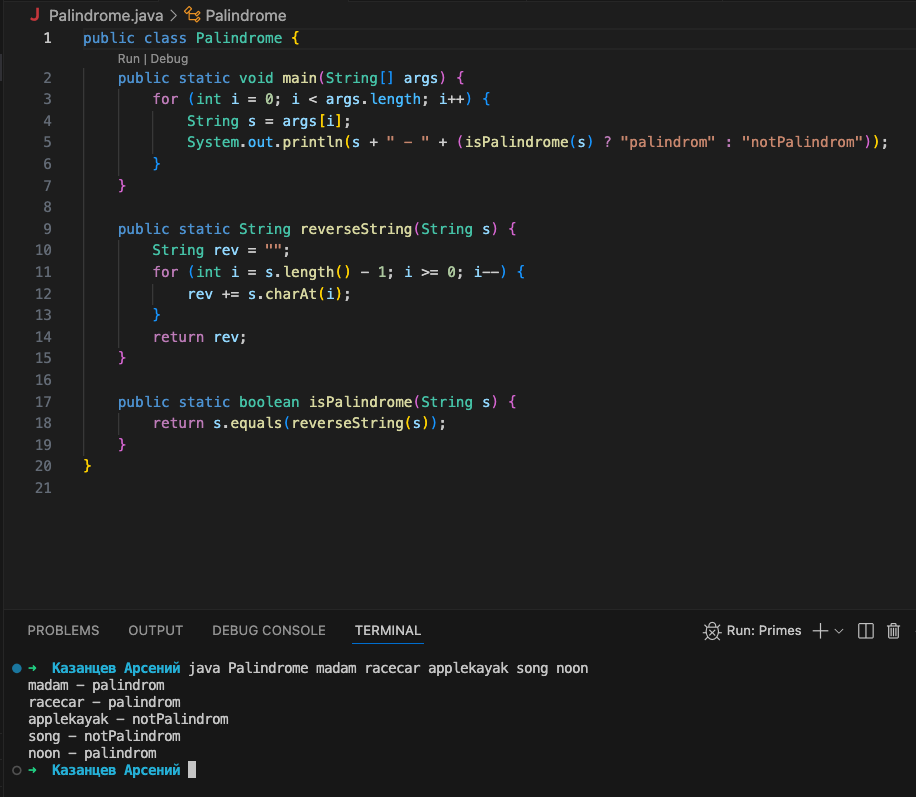
1) 

**Алгоритм**

1. Перебрать числа от 2 до 99.
2. Для каждого числа вызвать метод isPrime(int n).
3. В isPrime проверить делители от 2 до ⌊√n⌋: если нашли делитель без остатка - составное; иначе - простое.
4. Вывести простые числа в одну строку.

## Пояснение методов

* public static void main(String[] args) - точка входа. Запускает цикл по n = 2..99, печатает только простые числа.
* public static boolean isPrime(int n) - возвращает true, если n простое: перебирает делители d от 2 до √n (условие d \* d <= n). Если n % d == 0 - возвращает false, иначе в конце - true.

2) 

## Алгоритм

1. Считать аргументы командной строки как слова для проверки.
2. Для каждого слова:
   * Получить перевёрнутую строку (reverseString).
   * Сравнить исходную и перевёрнутую через equals.
3. Вывести результат для каждого слова.

## Пояснение методов

* public static void main(String[] args) - если аргументов нет, печатает подсказку по использованию и завершает работу; иначе перебирает слова из args, печатает результат проверки.
* public static String reverseString(String s) - формирует и возвращает перевёрнутую копию строки s, проходя символы от конца к началу (метод charAt, длина length()).
* public static boolean isPalindrome(String s) - возвращает true, если s равна своей перевёрнутой версии. Для сравнения используется s.equals(...), а не ==.

java Palindrome madam racecar applekayak song noon

**1. Java является компилируемым или интерпретируемым языком?**  
Java сочетает оба подхода: код компилируется в **байт-код** (.class), а затем интерпретируется и частично компилируется в машинный код виртуальной машиной JVM .

**2. Что такое JVM и для чего предназначается?**  
**JVM (Java Virtual Machine)** - виртуальная машина, которая выполняет байт-код Java. Она обеспечивает платформенную независимость: один и тот же код может запускаться на разных ОС .

**3. Жизненный цикл программы на языке Java**

1. Написание кода (.java).
2. Компиляция в байт-код (.class).
3. Запуск на JVM - интерпретация и JIT-компиляция .

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

* **Примитивные:** byte, short, int, long, float, double, char, boolean.
* **Ссылочные:** классы, строки, интерфейсы, массивы .

**5. Чем примитивные типы данных отличаются от ссылочных?**  
Примитивы хранят конкретное значение, а ссылочные типы содержат адрес объекта в памяти .

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

* **Неявное (widening):** byte → short → int → long → float → double.
* **Явное (narrowing):** long → int, double → float (может теряться точность) .

**7. Что такое байт-код в Java, и почему он важен?**  
**Байт-код** - промежуточное представление программы после компиляции. Он платформенно независим, выполняется на любой JVM .

**8. Какой тип данных используется для хранения символов в Java?**  
Тип **char** (16-бит Unicode) - хранит один символ .

**9. Что такое литералы в Java? Примеры.**  
Литералы - фиксированные значения в коде:

* Целые: 42 (int), 42L (long).
* Вещественные: 3.14, 2.7f.
* Символьные: 'a', '1'.
* Логические: true, false.
* Строковые: "hello" .

**10. Почему Java считается строго типизированным языком?**  
Все переменные и выражения имеют фиксированный тип, который проверяется на этапе компиляции, что уменьшает количество ошибок .

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

* Потеря точности (например, при преобразовании double → int).
* Переполнение диапазона.
* Ошибки при работе с разными типами данных (например, float k = 123456789f; int l = (int)k; даёт неточное значение) .