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 лабораторной работе №1

По дисциплине “Информационные технологии и программирование”

Выполнил: студент Петунин Алексей

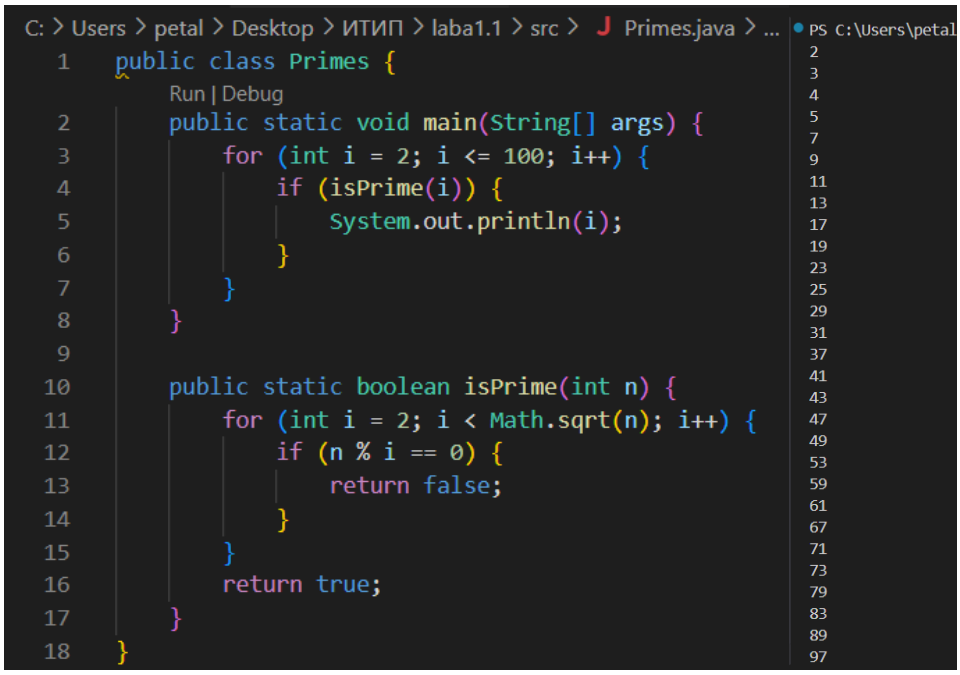
группы БПИ2401

Проверил: Харрасов Камиль Раисович

Москва

2025

Задание 1: Программа для поиска простых чисел



Объяснение кода:

1. Класс Primes:

Объявление класса

```
public class Primes {
```

2. Метод main():

Точка входа в программу

```
public static void main(String[] args) {
```

Цикл перебора чисел от 2 до 100

```
for (int i = 2; i <= 100; i++)
```

Вызов метода проверки на “простоту”

```
if (isPrime(i))
```

Вывод простого числа

```
System.out.println(i);
```

3. Метод isPrime():

Метод возвращает true/false

```
public static boolean isPrime(int n)
```

Цикл до корня из n (пока i меньше корня из n)

```
for (int i = 2; i < Math.sqrt(n); i++)
```

Проверка делимости

```
if (n % i == 0)
```

- return false - если найден делитель

- return true - если число простое

```
            return false;
        }
    }
    return true;
```

Задание 2: Программа для проверки палиндромов

```
C: > Users > petal > Desktop > ИТИП > laba1.2 > Palindrome.java > Palindrome > main(String[])
1   public class Palindrome {
        Run | Debug
2       public static void main(String[] args) {
3           String[] words = {"madam", "racecar", "apple", "kayak", "song", "noon"};
4
5           for (int i = 0; i < words.length; i++) {
6               String s = words[i];
7               if (isPalindrome(s)) {
8                   System.out.println(s);
9               }
10          }
11      }
12
13      public static String reverseString(String s) {
14          String result = "";
15          for (int i = s.length() - 1; i >= 0; i--) {
16              result += s.charAt(i);
17          }
18          return result;
19      }
20
21      public static boolean isPalindrome(String s) {
22          String reversed = reverseString(s);
23          return s.equals(reversed);
24      }
25  }
```

```
PS C:\Users\petal
madam
racecar
kayak
noon
```

Объяснение кода:

1. Метод reverseString():

Создание пустой строки

```
String result = "";
```

Обратный цикл

```
for (int i = s.length() - 1; i >= 0; i--)
```

Добавление символов в обратном порядке

```
result += s.charAt(i);
```

Возврат перевернутой строки

```
return result;
```

2. Метод isPalindrome():

Переворачиваем строку

```
String reversed = reverseString(s)
```

Сравнение с оригиналом

```
return s.equals(reversed);
```

3. Метод main():

- String[] words = {...} - массив тестовых слов
- for-цикл для перебора всех слов
- if (isPalindrome(s)) - проверка каждого слова

```
for (int i = 0; i < words.length; i++)
    String s = words[i];
    if (isPalindrome(s)) {
```

Ответы на вопросы:

1. Java компилируемый или интерпретируемый?

Оба. Сначала компилируется в байт-код, затем байт-код интерпретируется JVM.

2. Что такое JVM?

Виртуальная машина Java - выполняет байт-код на любой платформе. Как "переводчик" для разных компьютеров.

3. Жизненный цикл программы?

Пишем код (.java)

Компилируем в байт-код (.class)

JVM запускает байт-код

4. Виды типов данных?

Примитивные - простые значения (числа, символы, true/false)

Ссылочные - сложные объекты (строки, массивы, классы)

5. Отличие примитивных от ссылочных?

Примитивные хранят значение прямо в переменной.

Ссылочные хранят ссылку на объект в памяти.

6. Преобразование типов?

Неявное - автоматически (маленький тип в большой)

Явное - вручную, с риском потери точности (большой в маленький)

7. Что такое байт-код?

Промежуточный код между Java и машинным кодом. Позволяет запускать одну программу на разных компьютерах.

8. Тип для символов?

char - хранит один символ в кодировке Unicode.

9. Что такое литералы?

Конкретные значения в коде: 10, "hello", 3.14, true

10. Почему Java строго типизирована?

Потому что проверяет типы при компиляции - не даст смешать числа со строками без явного указания.

11. Проблемы неявного преобразования?

Потеря точности - например, при преобразовании большого числа в маленький тип.

ВЫВОД

В ходе лабораторной работы были изучены:

- Базовые конструкции языка Java

- Работа с методами и циклами

- Обработка строк и математических операций

Обе программы работают корректно и демонстрируют практическое применение изученных концепций.

Github: https://github.com/AlexleGGend/ITIP-labi