МИНИСТЕРСТВО ЦИФРОВОГО РАЗВИТИЯ СВЯЗИ И МАССОВЫХ КОММУНИКАЦИЙ

Ордена Трудового Красного Знамени

Федеральное государственное бюджетное образовательное учреждение высшего образования

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

Кафедра «Информационные технологии»

Информационные технологии и программирование

Лабораторная работа №1

Выполнил: Студент группы

БВТ2207

Кривенок Вадим

Москва

2023

**Выполнение.**

**Задание 1.**

Создайте программу, которая находит и выводит все простые числа меньше 100.

Для начала пишем public (модификатор доступа) class Main.

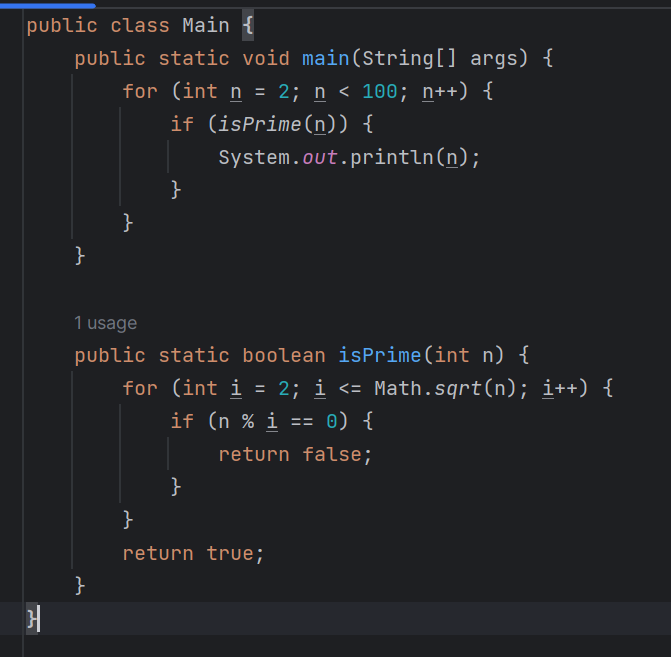
****Имя исходного файла java должно совпадать с именем public класса.

Рис.1 – public class Main.

Описываю условие задания.

for (int n=2; n < 100; n++) – задаем, что изначально n=2, и что она меньше 100 (не включительно). Каждый раз проходя «круг» n увеличивается на 1.

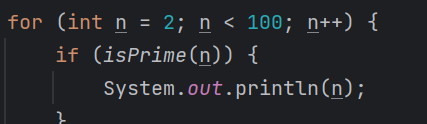
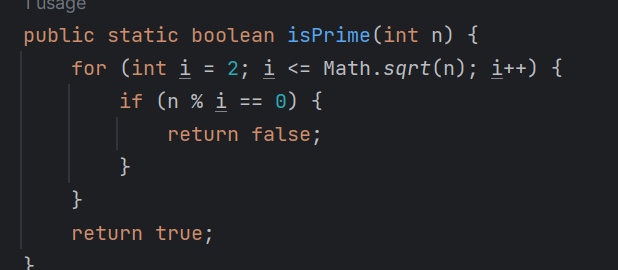
if (isPrime(n)) – если проходится проверка на простое число, то выводится.

Рис.2 – описание условия задачи.

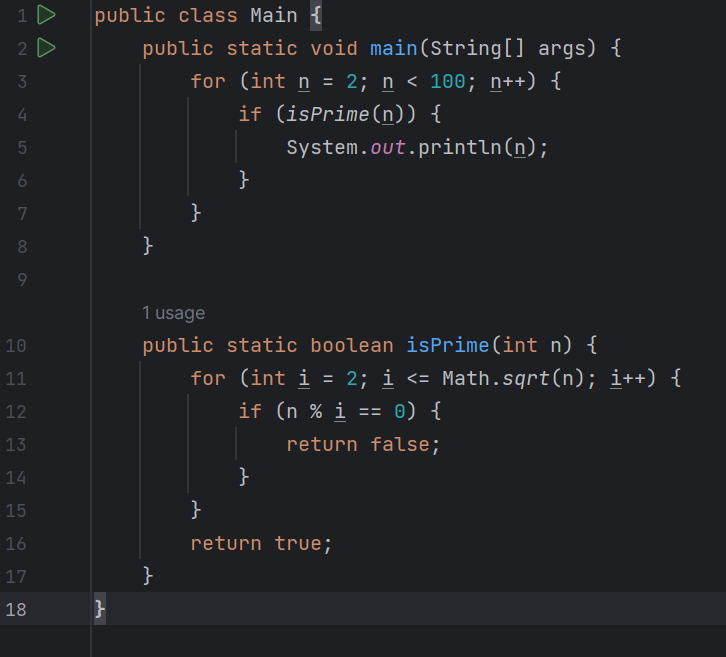
Внутри созданного класса, после метода main(), описываю метод IsPrime (int n), который определяет, является ли аргумент простым числом или нет. Можно предположить, что входное значение n всегда будет больше 2.

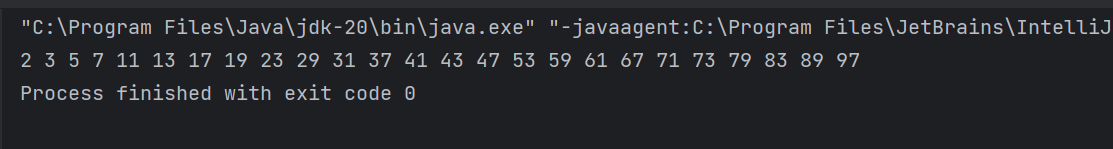
Рис.3 – проверка простых чисел.

Boolean — это тип данных, переменные которого принимают одно из значений: истина и ложь. Значение типа boolean возвращают операции сравнения, логические операции и их сочетания. Выражения с ними — это по сути условные высказывания, которые могут быть правдивы или нет.

Условие i <= Math.sqrt(n) используется в функции isPrime() для оптимизации проверки на простоту числа.

Когда мы проверяем, является ли число n простым, нам не нужно проверять делители больше, чем его квадратный корень. Это связано с математической особенностью простых чисел: если у числа есть делитель, который больше его квадратного корня, то у него также будет делитель, который меньше квадратного корня, и наоборот. Таким образом, если мы проверяем только числа до Math.sqrt(n), то мы уменьшаем количество проверок и повышаем производительность программы.

 Рис.4 – готовый код 1 задание.

Рис.5 – вывод 1 задание.

**Задание 2.**

Создайте программу, которая определяет, является ли введенная строка палиндромом.

****Рис.6 – готовый код задания 2.

В следующих строках выводятся результаты вызовов функции isPalindrome() с разными словами на консоль.

public static String reverseString(String word) – объявление метода reverseString(), который используется для разворота строки.

String p = "" – Создание пустой строки p, которая будет содержать развернутую версию слова.

В цикле for происходит разворот строки word и добавление символов к строке p в обратном порядке.

public static boolean isPalindrome(String p) – Объявление метода isPalindrome(), который определяет, является ли строка палиндромом.

return (p.equals(reverseString(p))) – Сравнивание исходной строки p с ее развернутой версией с помощью метода equals(), который возвращает true, если строка является палиндромом, и false в противном случае.

<https://github.com/Azrael3MU/ITIP_labs.git>