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»

(МТУСИ)

Кафедра «Математическая кибернетика и информационные технологии»

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №1

по дисциплине

«Информационные технологии и программирование»

на тему

«Начало работы с языком программирования Java»

Выполнил: студент группы БВТ2302

Никитин Андрей Александрович

Проверил:

Харрасов Камиль Раисович

Москва, 2024

**Цель работы**

Изучить базовые принципы работы и синтаксиса языка программирования Java.

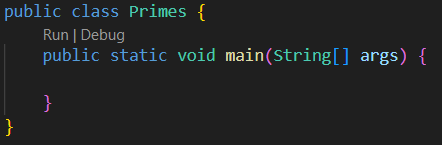
**Задачи**

1. Создать программу, которая находит и выводит все простые числа меньше 100;
2. Создать программу, которая определяет, является ли введенная строка палиндромом.

**Ход работы**

**Задание №1**

Создадим файл *Primes.java,* так как имя исходного файла java должно совпадать с именем public класса. В этом файле создадим класс Primes:



Разберем этот участок:

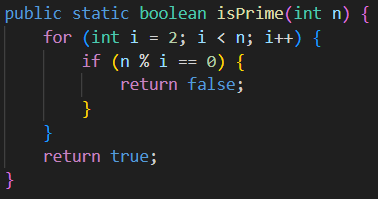
Public в Java — это модификатор доступа, который определяет видимость классов, полей и методов для других классов и пакетов. Когда элемент объявлен с модификатором *public*, он доступен из любого места в программе, даже за пределами текущего пакета.

Модификатор *static* в Java используется для объявления методов и переменных, которые не нуждаются в объекте класса. Таким образом, методы и переменные вызываются от класса, а не от объекта.

Ключевое слово *void* в Java позволяет создавать методы, не производящие возврат значения.

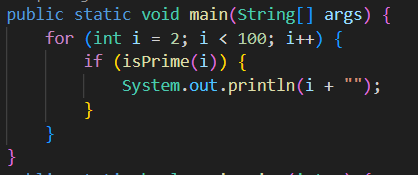
Параметр *String [] args* в методе *main* представляет собой массив строк, которые передаются в приложение при его запуске.

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

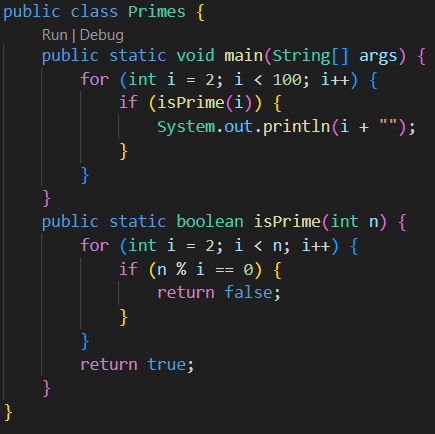


Данный метод мы можем реализовать с использованием цикла *for*. Данный цикл должен перебирать числа, начиная с 2 до (но не включая) n, проверяя существует ли какое-либо значение, делящееся на n без остатка. Для этого можно использовать оператора остатка «%». Если какая-либо переменная полностью делится на аргумент, сработает оператор *return false*. Если же значение не делится на аргумент без остатка, то это простое число, и сработает оператор *return true*.

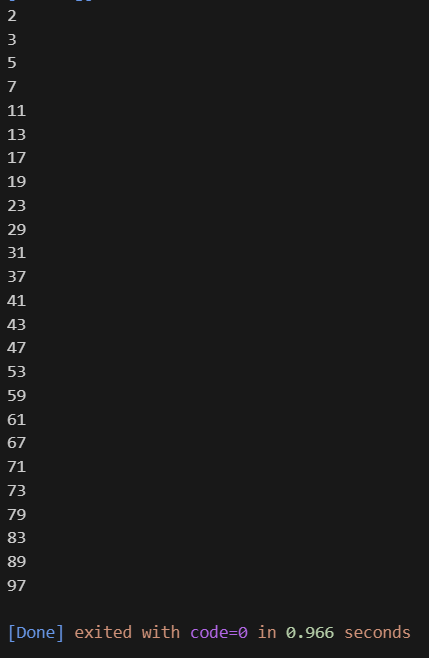
Перейдем к заполнению основного метода *main()* другим циклом, который перебирает числа в диапазоне от 2 до 100 включительно. Необходимо вывести на экран те значения, которые метод *IsPrime()* посчитал простыми.



У нас получается следующая программа:



Запустим и проверим её работу:

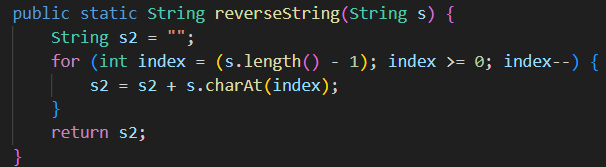


Программа работает исправно.

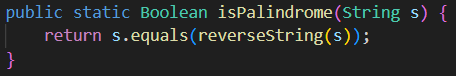
**Задание №2**

Для этой программы создадим класс с именем *Palindrome* в файле под названием *Palindrome.java*.

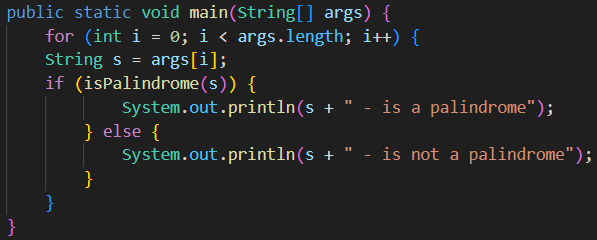
Наша первая задача состоит в том, чтобы создать метод, позволяющий полностью изменить символы в строке. Назовем его *reverseString*. Суть его работы заключается в прибавлении к пустой строке символов изначальной строки в обратном порядке.



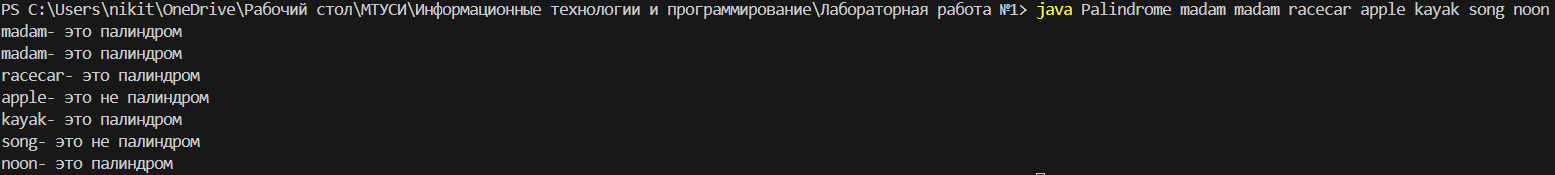
Создадим еще один метод *public static boolean isPalindrome(String s),* который перевернёт слово *s*, а затем сравнит его с первоначальными данными. Используем метод *equals (Object)* для проверки значения равенства. Если значения будут равны, то слово является палиндромом, иначе - нет.



Допишем *main:*



Скомпилируем и протестируем программу:



Программа работает исправно.

**ВЫВОД**

Мы изучили основные принципы синтаксиса и работы строго типизированного объектно-ориентированного языка программирования java.