**Федеральное агентство связи**

Ордена Трудового Красного Знамени

федеральное государственное бюджетное образовательное учреждение высшего образования   
«Московский технический университет связи и информатики»

Кафедра «Информатика»

Лабораторная работа №1: Основы синтаксиса Java

по дисциплине «Кроссплатформенные технологии программирования»

В данной лабораторной работе вы изучите основы синтаксиса Java с помощью нескольких простых задач программирования. Далее вы узнаете, как использовать компилятор Java и виртуальную машину Java для запуска программы.

Выполнил: студент

группы БСТ1803

Корнеев Александр

Вариант 12

Москва 2020

**Содержание**

[1.1 Задание на разработку программы 3](#_Toc41571144)

[1.2 Разработанные функции и классы 4](#_Toc41571145)

[1.2.1 Класс Primes 4](#_Toc41571146)

[1.2.1.1 Метод boolean IsPrime 4](#_Toc41571147)

[1.2.1.2 Метод void main 4](#_Toc41571148)

[1.3 Код программы 5](#_Toc41571149)

[1.4. Результат работы 5](#_Toc41571150)

[2.1. Задание на разработку программы 6](#_Toc41571151)

[2.2. Разработанные функции и классы 8](#_Toc41571152)

[2.2.1. Класс Palindrome 8](#_Toc41571153)

[2.2.1.1. Метод boolean isPalindrome 8](#_Toc41571154)

[2.2.1.2. Метод String reverseString 8](#_Toc41571155)

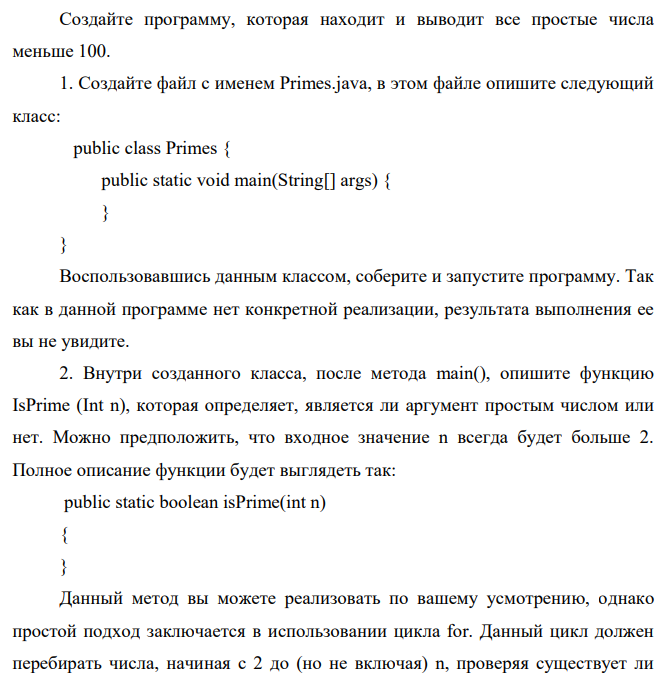
[Метод возвращает перевернутую строку. 8](#_Toc41571156)

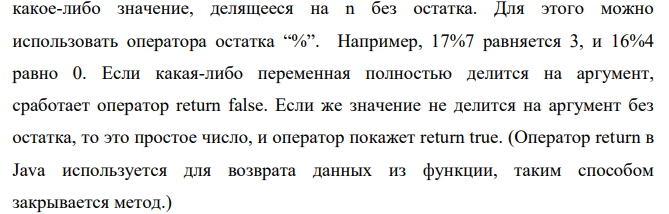
[2.2.1.3. Метод void main 8](#_Toc41571157)

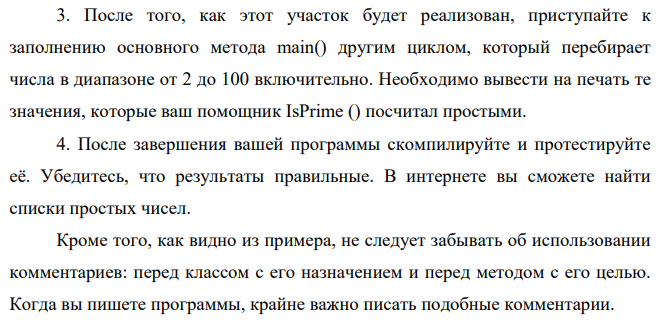
[2.3. Код программы 8](#_Toc41571158)

[2.4. Результат работы 10](#_Toc41571159)

# Задание на разработку программы







# Разработанные функции и классы

## Класс Primes

Класс, выводящий все простые числа меньше 100, содержит следующие методы:

* + - 1. Метод boolean IsPrime

Имеет один целочисленный аргумент, осуществляет проверку данного числа на “простоту”, возвращает **true** если число простое. Иначе **false**.

* + - 1. Метод void main

Осуществляет проверку чисел от 2 до 100 на простоту с помощью метода **IsPrime** и выводит простые числа.

# Код программы

public class Primes

{

public static void main(String[] args)

{

// Перебор чисел от 2 до 100 и вывод простых

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

{

if (isPrime(i)) System.out.print(i + " ");

}

}

public static boolean isPrime(int n)

// Функция проверки простоты числа

{

for(int i = 2;i<n;i++)

{

if (n%i==0) return false;

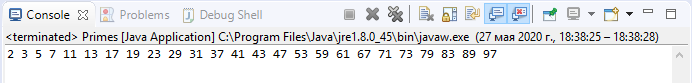
}

return true;

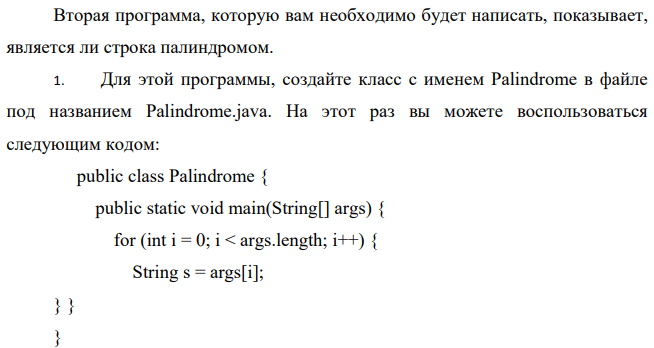
}

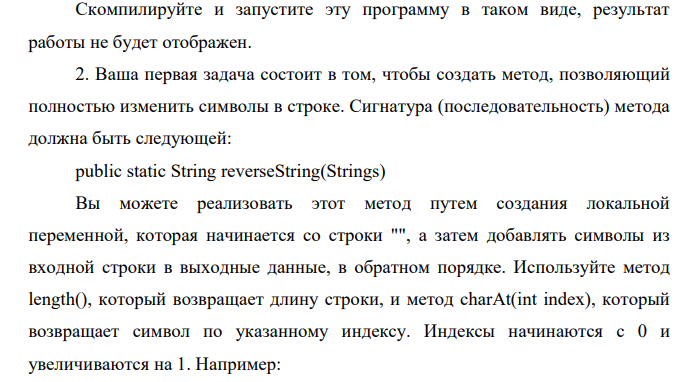
}

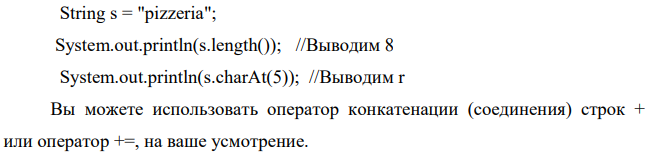
# 1.4. Результат работы

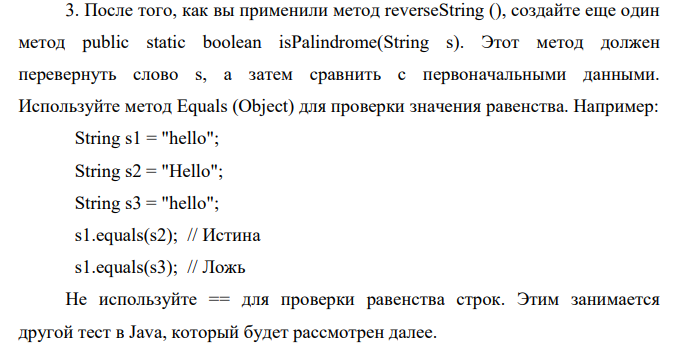


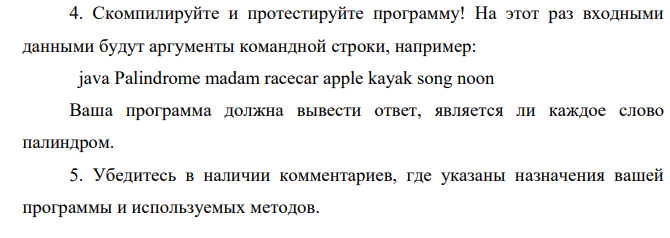
# Задание на разработку программы











# Разработанные функции и классы

* + 1. Класс Palindrome

Данный класс осуществляет проверку являются ли слова, содержащиеся во входных данных палиндромами

* + - 1. Метод boolean isPalindrome

Метод осуществляет проверку является ли входящая строка палиндромом. Возвращает **true**, если это так, иначе **false.**

* + - 1. Метод String reverseString

Метод возвращает перевернутую строку.

* + - 1. Метод void main

Проверяет входные строки на то, являются ли они палиндромами и выводит отчет по каждому из слов

# Код программы

/\*

\* Данный класс осуществляет проверку являются ли слова

\* содержащиеся во входных данных палиндромами

\*/

public class Palindrome

{

public static void main(String[] args)

{

// Проверка являются ли слова из входных аргуемнтов палиндромами

// И вывод отчета по каждому слову

for (int i = 0; i < args.length; i++)

{

String s = args[i];

System.out.println(s + " " + isPalindrome(s));

}

}

public static String reverseString(String text)

// Метод возвращает перевернутую строку

{

String reverse = "";

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

{

reverse+=text.charAt(i);

}

return reverse;

}

public static boolean isPalindrome(String s)

{

// Метод осуществляет проверку является ли строка палиндромом

String rs = reverseString(s);

return s.equals(rs);

}

}

# Результат работы

