**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

Шевцов Никита

Вариант №24

Москва 2020

**Содержание**

[1.1 Задание на разработку программы 3](#_Toc53411085)

[1.2 Разработанные функции и классы 4](#_Toc53411086)

[1.2.1 Класс Primes 4](#_Toc53411087)

[1.2.1.1 Метод boolean IsPrime 4](#_Toc53411088)

[1.2.1.2 Метод void main 4](#_Toc53411089)

[1.3 Код программы 5](#_Toc53411090)

[1.4. Результат работы 5](#_Toc53411091)

[2.1. Задание на разработку программы 6](#_Toc53411092)

[2.2. Разработанные функции и классы 8](#_Toc53411093)

[2.2.1. Класс Palindrome 8](#_Toc53411094)

[2.2.1.1. Метод boolean isPalindrome 8](#_Toc53411095)

[2.2.1.2. Метод String reverseString 8](#_Toc53411096)

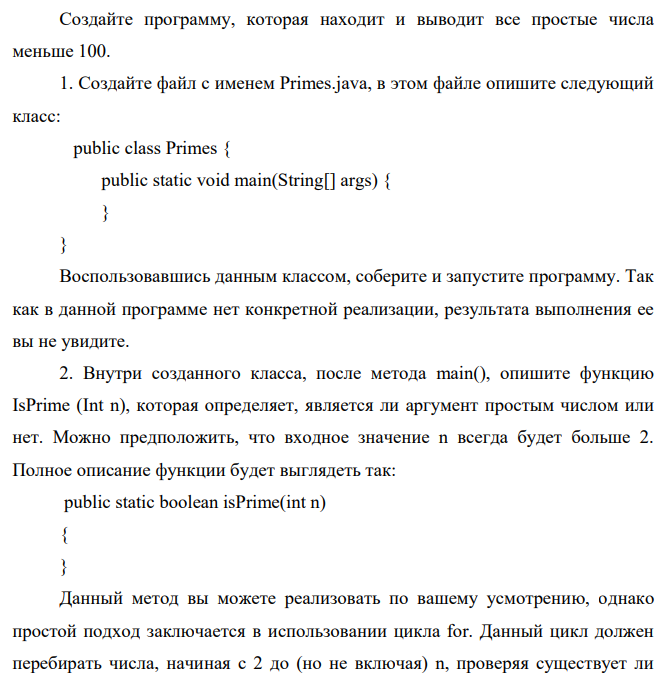
[Метод возвращает перевернутую строку. 8](#_Toc53411097)

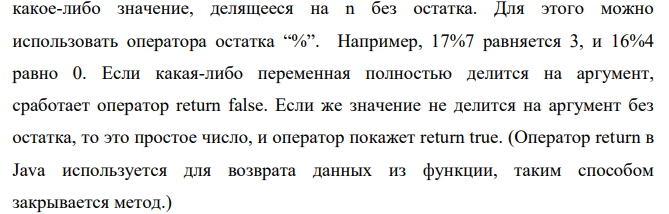
[2.2.1.3. Метод void main 8](#_Toc53411098)

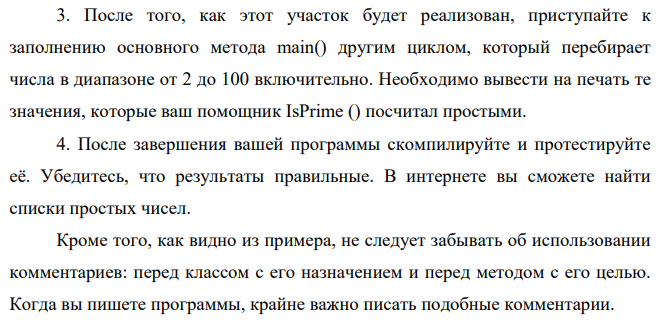
[2.3. Код программы 8](#_Toc53411099)

[2.4. Результат работы 9](#_Toc53411100)

# 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) {

**for** (**int** i = 2; i < 101; i++) {

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

}

}

/\*\*

\* Функция, проверяющая число num на простоту, путем деления его на все числа от 2 до sqrt(num)

\* **@param** num - целое число для проверки

\*/

**public** **static** **boolean** isPrime(**int** num) {

**int** lim = (**int**) Math.*ceil*(Math.*sqrt*(num+1));

**for** (**int** i = 2; i < lim; i++) {

**if** (num%i == 0) **return** **false**;

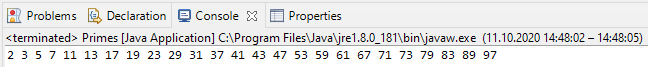
}

**return** **true**;

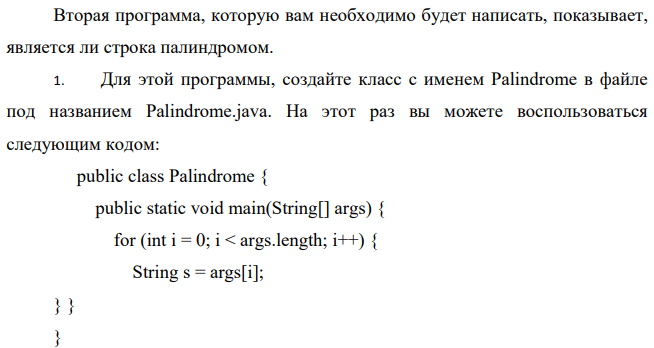
}

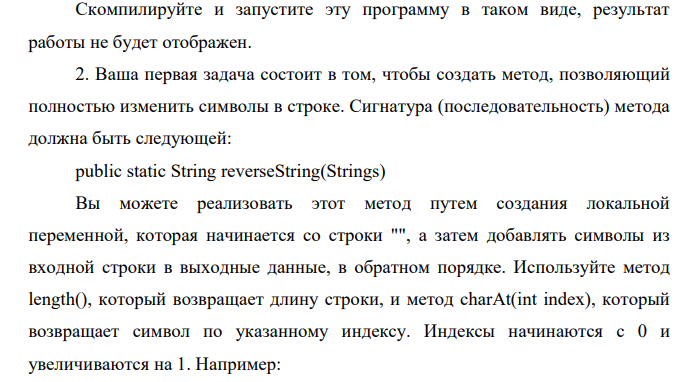
}

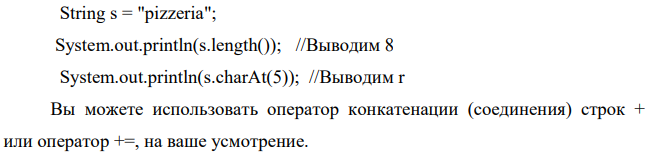
# 1.4. Результат работы

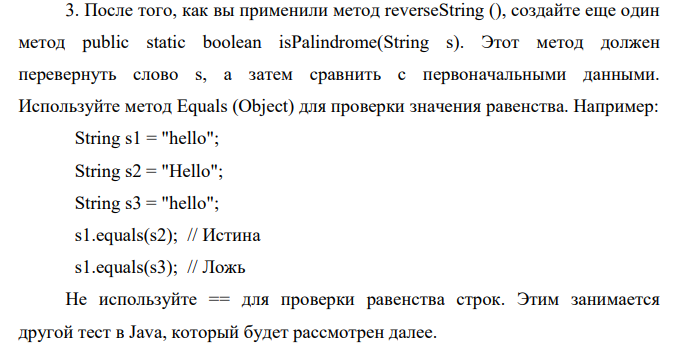


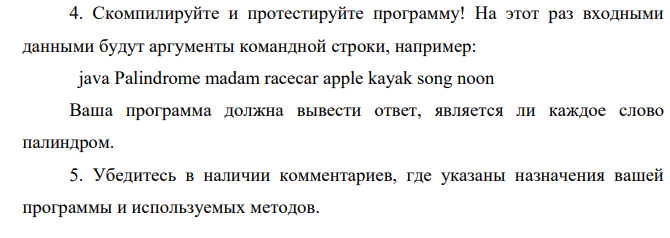
# Задание на разработку программы











# 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

**public** **class** Palindrome {

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

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

System.***out***.print(args[i] + " ");

**if** (*isPalindrome*(args[i])) System.***out***.print(**true** + "\n");

**else** System.***out***.print(**false** + "\n");

}

}

/\*\*

\* Функция, возвращающая реверсированную строку

\* **@param** orig - строка для обработки

\* **@return** Строка, обратная orig

\*/

**public** **static** String reverseString(String orig) {

String rev = "";

**for** (**int** i = orig.length()-1; i > -1; i--) {

rev += orig.charAt(i);

}

**return** rev;

}

/\*\*

\* Функция, проверяющая является ли строка палиндромом,

\* сравнивая ее с реверсированной версеей себя

\* **@see** Palindrome#reverseString(String)

\* **@param** str - строка для проверки

\* **@return** <code>true</code>, если является, <code>false</code>, если не является

\*/

**public** **static** **boolean** isPalindrome(String str) {

**if** (str.equals(*reverseString*(str))) **return** **true**;

**return** **false**;

}

}

# Результат работы

