|  |
| --- |
|  |
| МИНОБРНАУКИ РОССИИ |
| *Федеральное государственное бюджетное образовательное учреждение высшего образования* ***«МИРЭА – Российский технологический университет»***  **РТУ МИРЭА** |

**Институт информационных технологий (ИТ)**

**Кафедра инструментального и прикладного программного обеспечения (ИиППО)**

**Дисциплина «Программирование на языке Джава»**

**ОТЧЕТ**

**ПО ПРАКТИЧЕСКОМУ ЗАНЯТИЮ №5**

Выполнил студент группы ИНБО-02-20 Абдусалямова С. К.

Принял Степанов П.В.

Практические работы выполнены «\_\_\_»\_\_\_\_\_\_\_2021г.

«\_\_\_\_\_\_\_\_\_\_\_\_\_\_» «\_\_\_»\_\_\_\_\_\_\_2021г.

Отметка о выполнении

Москва 2021г.

## **Задание**

Упражнения по этой теме выполняются следующим образом: каждый учащийся выполняет от 3 до 5 задач, начиная с номера варианта задания, который соответствует номеру учащегося в журнале группы.

1. Треугольная последовательность

Дана монотонная последовательность, в которой каждое натуральное число k встречается ровно k раз: 1, 2, 2, 3, 3, 3, 4, 4, 4, 4,… По данному натуральному n выведите первые n членов этой последовательности. Попробуйте обойтись только одним циклом for.

1. От 1 до n

Дано натуральное число n. Выведите все числа от 1 до n.

1. От A до B

Даны два целых числа A и В (каждое в отдельной строке). Выведите все числа от A до B включительно, в порядке возрастания, если A < B, или в порядке убывания в противном случае.

1. Заданная сумма цифр

Даны натуральные числа k и s. Определите, сколько существует k-значных натуральных чисел, сумма цифр которых равна d. Запись натурального числа не может начинаться с цифры 0. В этой задаче можно использовать цикл для перебора всех цифр, стоящих на какой-либо позиции.

1. Сумма цифр числа

Дано натуральное число N. Вычислите сумму его цифр. 46 При решении этой задачи нельзя использовать строки, списки, массивы (ну и циклы, разумеется).

1. Проверка числа на простоту

Дано натуральное число n>1. Проверьте, является ли оно простым. Программа должна вывести слово YES, если число простое и NO, если число составное. Алгоритм должен иметь сложность O(logn). Указание. Понятно, что задача сама по себе нерекурсивна, т.к. проверка числа n на простоту никак не сводится к проверке на простоту меньших чисел. Поэтому нужно сделать еще один параметр рекурсии: делитель числа, и именно по этому параметру и делать рекурсию.

## **Ход Работы**

На основе поставленной задачи был написан код программы (варианты 1-3, 5-6):

package com.company;  
  
  
public class Main {  
  
 public static void main(String[] args) {  
// triangularSequence(4);  
// from1ToN(4);  
// fromAToB(2, 4);  
// System.out.println(sumOfNumbers(1234));  
 *checkIsItANaturalNumber*(10, 2);  
 }  
  
 //1  
 static void triangularSequence(int n) {  
 StringBuilder sum = new StringBuilder();  
  
 if (n == 1) System.*out*.print(1 + " ");  
 else {  
 for (int i = 0; i < n; i++) {  
 sum.append(n).append(" ");  
 }  
 *triangularSequence*(n - 1);  
 System.*out*.print(sum);  
 }  
 }  
  
 //2  
 static void from1ToN(int n) {  
 if (n >= 1) {  
 *from1ToN*(n - 1);  
 System.*out*.print(n + " ");  
 }  
 }  
  
 //3  
 static void fromAToB(int a, int b) {  
 if (a > b) {  
 *fromAToB*(a, b + 1);  
 System.*out*.print(b + " ");  
 } else if (a < b) {  
 *fromAToB*(a, b - 1);  
 System.*out*.print(b + " ");  
 } else System.*out*.print(b + " ");  
 }  
  
 //5  
 static int sumOfNumbers(int n) {  
 int sum = n % 10;  
 if (n / 10 > 0) {  
 sum += *sumOfNumbers*(n / 10);  
 return sum;  
 } else return n;  
 }  
  
 //6  
 private static void checkIsItANaturalNumber(int n,int var) {  
 if (n % var != 0) {  
 var++;  
 *checkIsItANaturalNumber*(n, var);  
 } else if (n % var == 0) {  
 if (n==var) System.*out*.print("simple number");  
 else System.*out*.print("not simple number");  
 }  
 }  
  
}

## **ВЫВОД**

В ходе данной работы я ознакомилась с написанием рекурсий на языке java.