



МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
"МИРЭА - Российский технологический университет"

**РТУ МИРЭА**

---

Институт информационных технологий (ИТ)  
Кафедра инструментального и прикладного программного обеспечения

**ОТЧЁТ  
ПО ПРАКТИЧЕСКОЙ РАБОТЕ №5  
по дисциплине  
«Программирование на языке Java»**

**Тема: Программирование рекурсии в Java**

Выполнил студент группы ИКБО-16-20

Пак С.А.

Принял ассистент кафедры ИиППО

Русляков А.А.

Практические работы выполнены «\_\_\_\_\_» 2021г.

«Зачтено» «\_\_\_\_\_» 2021г

Москва 2021

## СОДЕРЖАНИЕ

ТЕОРЕТИЧЕСКОЕ ВВЕДЕНИЕ.....	3
РЕШЕНИЕ ЗАДАЧИ.....	4
1. Постановка задачи.....	4
2. Программный код.....	4
3. Вывод программы.....	6
ВЫВОД.....	8

## ТЕОРЕТИЧЕСКОЕ ВВЕДЕНИЕ

В контексте языка программирования рекурсия — это некий активный метод (или подпрограмма) вызываемый сам по себе непосредственно, или вызываемой другим методом (или подпрограммой) косвенно. В первую очередь надо понимать, что рекурсия — это своего рода перебор. Вообще говоря, всё то, что решается итеративно можно решить рекурсивно, то есть с использованием рекурсивной функции.

Так же, как и у перебора (цикла) у рекурсии должно быть условие остановки — базовый случай (иначе также, как и цикл, рекурсия будет работать вечно — infinite). Это условие и является тем случаем, к которому рекурсия идет (шаг рекурсии). При каждом шаге вызывается рекурсивная функция до тех пор, пока при следующем вызове не сработает базовое условие и не произойдет остановка рекурсии (а точнее возврат к последнему вызову функции). Всё решение сводится к поиску решения для базового случая. В случае, когда рекурсивная функция вызывается для решения сложной задачи (не базового случая) выполняется некоторое количество рекурсивных вызовов или шагов, с целью сведения задачи к более простой. И так до тех пор, пока не получим базовое решение.

Итак, рекурсивная функция состоит из:

- условие остановки или же базового случая или условия;
- условие продолжения или шага рекурсии — способ сведения сложной задачи к более простым подзадачам.

# РЕШЕНИЕ ЗАДАЧИ

## 1. Постановка задачи

### Задания:

4. 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.

5. Дано натуральное число  $N$ . Вычислите сумму его цифр.

При решении этой задачи нельзя использовать строки, списки, массивы (ну и циклы, разумеется).

6. 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  $O(\log n)$ .

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.

## 2. Программный код

Файл Task4.java:

```
package ru.mirea;

import java.util.*;

public class Task4 {
    private static final Scanner IN = new Scanner(System.in);

    /**
     * Вычисляет сумму цифр числа
     * @param number    число, сумму цифр которого нужно найти
     * @return          сумма цифр целого числа
     */
    private static int sumOfDigits(int number) {
        if (number < 10) {
            return number;
        }

        return number % 10 + sumOfDigits(number / 10);
    }
}
```

```

public static void main(String[] args) {
    System.out.print("Введите k: ");
    int k = IN.nextInt();

    System.out.print("Введите s: ");
    int s = IN.nextInt();

    int start = (int) Math.pow(10, k - 1);
    int end = (int) Math.pow(10, k);

    int count = 0;

    for (int num = start; num < end; ++num) {
        if (sumOfDigits(num) == s) {
            ++count;
        }
    }

    System.out.println();
    System.out.println("Количество " + k + "-значных чисел, сумма
цифр которых равна " + s + ": " + count);
}
}

```

Файл Task5.java:

```

package ru.mirea;

import java.util.*;

public class Task5 {
    private static final Scanner IN = new Scanner(System.in);

    private static int sumOfDigits(int number) {
        if (number < 10) {
            return number;
        }

        return number % 10 + sumOfDigits(number / 10);
    }

    public static void main(String[] args) {
        System.out.print("Введите число: ");
        int n = IN.nextInt();

        System.out.println("Сумма цифр числа " + n + ": " +
sumOfDigits(n));
    }
}

```

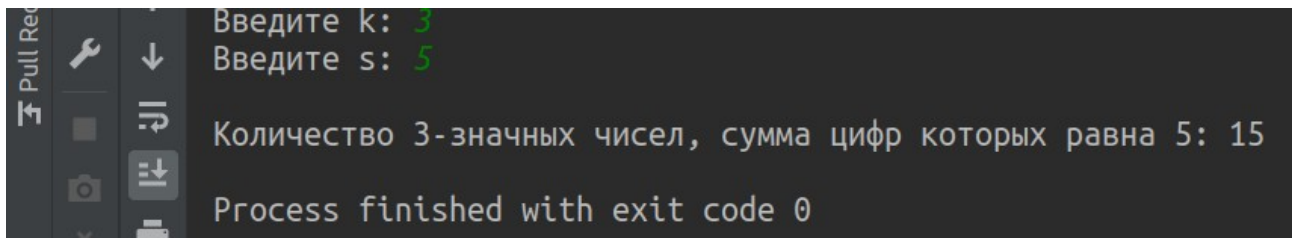
```
}  
}
```

Файл Task6.java:

```
package ru.mirea;  
  
import java.util.*;  
  
public class Task6 {  
    private static final Scanner IN = new Scanner(System.in);  
  
    /**  
     * Проверяет, является ли число простым  
     * @param number      проверяемое число  
     * @param divisor      делитель (начинается с числа 2)  
     */  
    private static boolean isSimple(int number, int divisor) {  
        if (number == 1 || number == 2) {  
            return true;  
        }  
  
        if (divisor == number - 1) {  
            return (number % divisor != 0);  
        }  
  
        return (number % divisor != 0) && isSimple(number, divisor +  
1);  
    }  
  
    public static void main(String[] args) {  
        System.out.print("Введите число: ");  
        int n = IN.nextInt();  
  
        if (isSimple(n, 2)) {  
            System.out.println("Число " + n + " является простым!");  
            return;  
        }  
  
        System.out.println("Число " + n + " не является простым!");  
    }  
}
```

### 3. Вывод программы

Результаты работы программы для задания 4 (рис.1).



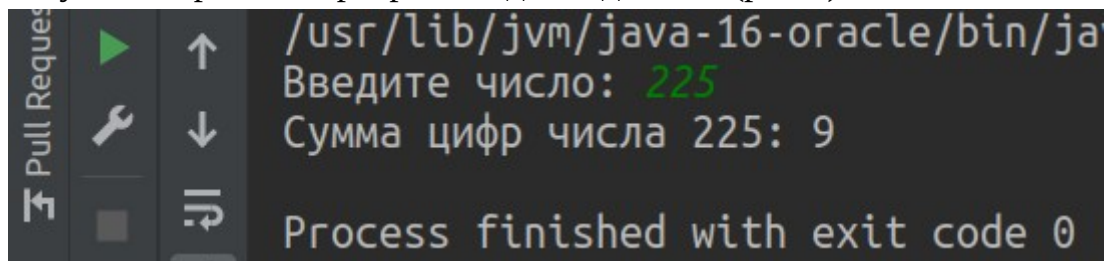
```
Введите k: 3
Введите s: 5

Количество 3-значных чисел, сумма цифр которых равна 5: 15

Process finished with exit code 0
```

Рис.1 Задание 4

Результаты работы программы для задания 5 (рис.2).

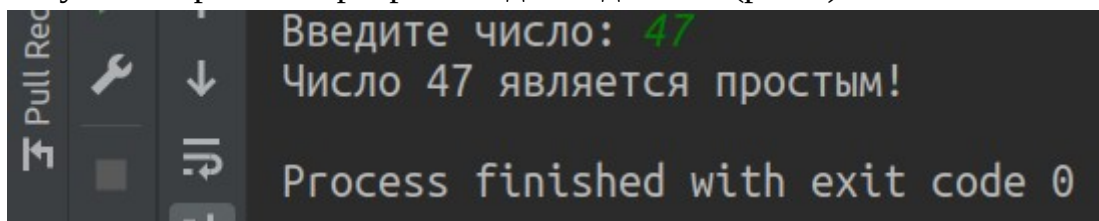


```
/usr/lib/jvm/java-16-oracle/bin/java
Введите число: 225
Сумма цифр числа 225: 9

Process finished with exit code 0
```

Рис.2 Задание 5

Результаты работы программы для задания 6 (рис.3).



```
Введите число: 47
Число 47 является простым!

Process finished with exit code 0
```

Рис.3 Задание 6

## **ВЫВОД**

В ходе выполнения работы разработал и запрограммировал рекурсивные алгоритмы на языке Java.