

**Московский государственный технический  
университет им. Н.Э. Баумана**

**Факультет «Информатика и системы управления»  
Кафедра ИУ5 «Системы обработки информации и управления»**

**Курс «Парадигмы и конструкции языков программирования»**

**Отчет по лабораторной работе № 1**

Выполнил:  
студент группы ИУ5-34Б:  
Суслов Дмитрий Сергеевич  
Подпись и дата:

Проверил:  
преподаватель каф. ИУ5  
Гапанюк Ю. Е.  
Подпись и дата:

Москва, 2023 г.

### Задание:

Написать алгоритм, вычисляющий расстояние Левенштейна

### Код:

Так как Scala является функциональным языком программирования, не будем отходить от функциональной парадигмы и решим задачу с помощью хвостовой рекурсии

```
package Lab2
import scala.annotation.tailrec

object Levenshtein extends App{

  def levenshtein(str1: String, str2: String): Int = {
    val len1 = str1.length
    val len2 = str2.length
    val matrix = Array.ofDim[Int](len1 + 1, len2 + 1)

    for (i <- 0 to len1){
      matrix(i)(0) = i
    }
    for (j <- 0 to len2) {
      matrix(0)(j) = j
    }

    @tailrec
    def distance(i: Int, j: Int): Int = {
      if (i > len1 || j > len2){
        matrix(len1)(len2)
      }
      else if (str1(i - 1) == str2(j - 1)){
        matrix(i)(j) = matrix(i - 1)(j - 1)
        if (j == len2) distance(i + 1, 1) else distance(i, j + 1)
      }
      else{
        matrix(i)(j) = (matrix(i - 1)(j) + 1).min((matrix(i)(j - 1) + 1).min(matrix(i - 1)(j - 1) + 1))
        if (j == len2) distance(i + 1, 1) else distance(i, j + 1)
      }
    }
    distance(1, 1)
  }
  print(levenshtein("mama", "papa"))
}
```

```
"C:\Program Files\Java\jdk-21\bin\
2
Process finished with exit code 0
```