# **Лабораторная работа №2**

**Разработка алгоритмов**

Гусев В.Е.

ИС-З-21

|  |
| --- |
|  |

**Задание №11**

Разработать программу, осуществляющую:

- Генерацию текста длиной N символов;

- Поиск в данном тексте подстроки длиной m символов, при этом любой один символ в подстроке может быть искаженным.

|  |
| --- |
|  |

Код программы:

import java.util.\*;

public class lab2 {

    private static final String strAZ = "ABCDEFGHIJKLMNOPQRSTUVWXYZ ";

    public static void main(String[] args) {

        Scanner scanner = new Scanner(System.in);

        System.out.print("Введите длину текста (N): ");

        int N = scanner.nextInt();

        String text = generateRandomText(N);

        System.out.println("Сгенерированный текст: " + text);

        System.out.print("Введите подстроку для поиска (длина m): ");

        String pattern = scanner.next();

        int m = pattern.length();

        List<Integer> matches = findString(text, pattern);

        System.out.println("\nРезультаты поиска:");

        for (int pos : matches) {

            String foundSubstring = text.substring(pos, pos + m);

            System.out.printf("Найдено: '%s' (позиция %d)\n", foundSubstring, pos);

        }

        System.out.println("\nВсего найдено: " + matches.size());

    }

    private static String generateRandomText(int length) {

        Random random = new Random();

        StringBuilder sb = new StringBuilder();

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

            sb.append(strAZ.charAt(random.nextInt(strAZ.length())));

        }

        return sb.toString();

    }

    private static List<Integer> findString(String text, String pattern) {

        List<Integer> matches = new ArrayList<>();

        int m = pattern.length();

        for (int i = 0; i <= text.length() - m; i++) {

            String substring = text.substring(i, i + m);

            if (checkString(substring, pattern)) {

                matches.add(i);

            }

        }

        return matches;

    }

    private static boolean checkString(String str, String pattern) {

        int errors = 0;

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

            if (str.charAt(i) != pattern.charAt(i)) {

                errors++;

                if (errors > 1) return false;

            }

        }

        return true;

    }

}

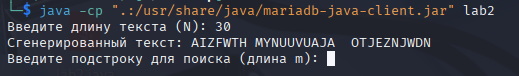
Компилируем программу



Запускаем программу и вводим длину текста (30)



Программа случайным образом сгенерировала строку из набора символов "ABCDEFGHIJKLMNOPQRSTUVWXYZ ". Вводим подстроку, которую мы хотим найти («AI»)



Программа отображает результаты поиска (полное совпадение и неполное совпадение (с одним искаженным символом)): найденные значения, их позиция(смещение от начала сгенерированной строки) и общее количество найденных совпадений.

