



Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет
имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ ИНФОРМАТИКА И СИСТЕМЫ УПРАВЛЕНИЯ

КАФЕДРА КОМПЬЮТЕРНЫЕ СИСТЕМЫ И СЕТИ (ИУ6)

НАПРАВЛЕНИЕ ПОДГОТОВКИ 09.04.01 Информатика и вычислительная техника

МАГИСТЕРСКАЯ ПРОГРАММА 09.04.01/07 Интеллектуальные системы анализа,
обработки и интерпретации больших данных

О Т Ч Е Т

по лабораторной работе № 6

Название: Коллекции. Вариант 9

Дисциплина: Языки программирования для работы с большими
данными

Студент ИУ6-22М
(Группа)

К.Ю. Каташинский
(Подпись, дата) (И.О. Фамилия)

Преподаватель

П.В. Степанов
(Подпись, дата) (И.О. Фамилия)

Москва, 2024

Цель работы

Изучить работу с коллекциями в языке программирования Java.

Ход работы

Задание 1. Задана строка, состоящая из символов '(', ')', '[', ']', '{', '}'. Проверить правильность расстановки скобок. Использовать стек.

Код программы представлен в листинге 1, результат – на рисунке 1.

Листинг 1 – Код программы задания 1

```
package Lab6;

import java.util.Scanner;
import java.util.Stack;

public class Variant18 {

    public static void main(String[] args) {

        System.out.print("Введите строку: ");

        Scanner scanner = new Scanner(System.in);

        String line = scanner.nextLine();

        var stack = new Stack<Character>();

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

            var symbol = line.charAt(i);

            if (symbol == '(' || symbol == '[' || symbol == '{') {

                stack.push(symbol);

                continue;

            }

            var elem = stack.pop();

            if (elem == '(' && symbol == ')') continue;

            if (elem == '[' && symbol == ']') continue;

            if (elem == '{' && symbol == '}') continue;

            stack.push(elem);

        }

        if (stack.isEmpty()) {

            System.out.println("Результат: строка верная");

        }

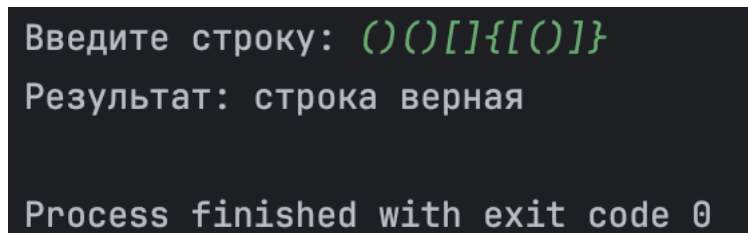
    }

}
```

```

        } else {
            System.out.println("Результат: строка
неверная");
        }
    }
}

```



```

Введите строку: ()()[]{}[(())}
Результат: строка верная

Process finished with exit code 0

```

Рисунок 1 – Результат задания 1

Задание 2. Задан файл с текстом на английском языке. Выделить все различные слова. Слова, отличающиеся только регистром букв, считать одинаковыми. Использовать класс HashSet.

Код программы представлен в листинге 2, результат – на рисунке 2.

Листинг 2 – Код программы задания 2

```

package Lab6;

import java.io.FileReader;
import java.io.BufferedReader;
import java.util.ArrayList;
import java.util.Arrays;
import java.util.HashSet;

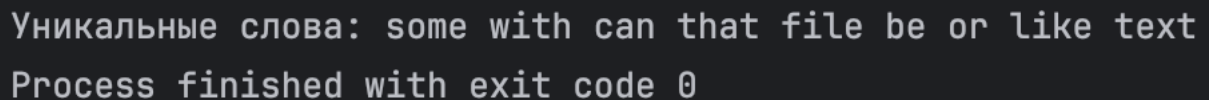
public class Variant19 {
    public static void main(String[] args) {
        var list = new ArrayList<ArrayList<String>>();
        try (BufferedReader br = new BufferedReader(new
FileReader("src/Lab6/input_19.txt"))) {
            for (String line; (line = br.readLine()) !=
null; ) {
                list.add(new
ArrayList<>(Arrays.asList(line.split(" ")))));

```

```

        }
    } catch (Exception e) {
        System.out.println(e.getMessage());
    }
    var hash = new HashSet<String>();
    for (ArrayList<String> line : list) {
        for (String word : line) {
            hash.add(word.toLowerCase());
        }
    }
    System.out.print("Уникальные слова: ");
    for (String word : hash) {
        System.out.print(word + " ");
    }
}
}

```



```

Уникальные слова: some with can that file be or like text
Process finished with exit code 0

```

Рисунок 2 – Результат задания 2

Задание 3. Дана матрица из целых чисел. Найти в ней прямоугольную подматрицу, состоящую из максимального количества одинаковых элементов. Использовать класс Stack.

Код программы представлен в листинге 3, результат – на рисунке 3.

Листинг 3 – Код программы задания 3

```

package Lab6;

import java.util.Stack;

public class Variant29 {
    public static void main(String[] args) {
        int[][] matrix = new int[][] {
            {1, 0, 1, 0, 1},

```

```

        {3, 2, 2, 2, 3},
        {4, 2, 2, 2, 4},
        {5, 2, 2, 2, 5},
        {1, 0, 1, 0, 1},
    };

    Stack<Integer> stack = new Stack<Integer>();
    int n = 5, currentXCount = 0, maxCount = 0;
    var firstLine = true;
    for (int oy = 0; oy < n; ++oy) {
        for (int ox = 0; ox < n; ++ox) {
            for (int i = oy; i < n; ++i) {
                if (stack.empty()) {
                    stack.push(matrix[i][ox]);
                } else {
                    var elem = stack.pop();
                    stack.push(elem);
                    if (matrix[i][ox] == elem)
stack.push(matrix[i][ox]);
                }
                var XCount = 0;
                var validLine = true;
                for (int j = ox + 1; j < n; ++j) {
                    var elem = stack.pop();
                    stack.push(elem);
                    if (XCount == currentXCount &&
!firstLine) break;

                    if (elem != matrix[i][j]) {
                        validLine = firstLine;
                        break;
                    }
                }
                if (firstLine) {

```

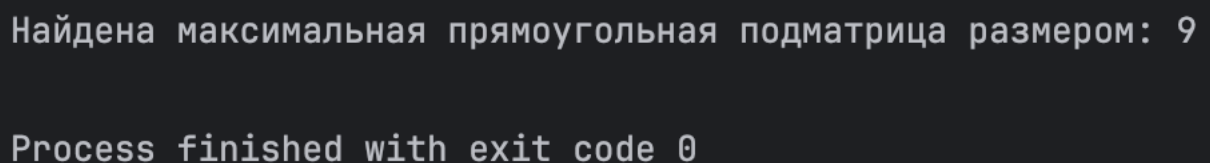
```

        ++currentXCount;
    } else {
        ++XCount;
    }
    stack.push(elem);
}
if (!validLine) break;
if (stack.size() >= maxCount) maxCount
= stack.size();

    firstLine = false;
}
firstLine = true;
currentXCount = 0;
stack.clear();
}
}

System.out.println("Найдена максимальная
прямоугольная подматрица размером: " + maxCount);
}
}

```



```

Найдена максимальная прямоугольная подматрица размером: 9
Process finished with exit code 0

```

Рисунок 3 – Результат задания 3

Задание 4. На прямой гоночной трассе стоит N автомобилей, для каждого из которых известны начальное положение и скорость. Определить, сколько произойдет обгонов.

Код программы представлен в листинге 4, результат – на рисунке 4.

Листинг 4 – Код программы задания 4

```
package Lab6;
```

```

import java.util.ArrayList;
import java.util.Comparator;
import java.util.Random;
class Auto {
    private String name;
    private int currentPosition, speed, beMoreFirst;
    public Auto(String name) {
        this.name = name;
        this.currentPosition = new Random().nextInt(11);
        this.speed = new Random().nextInt(7) + 1;
    }
    public int getCurrentPosition() {
        return this.currentPosition;
    }
    public int getSpeed() {
        return this.speed;
    }
    public int getBeMoreFirst() {
        return this.beMoreFirst;
    }
    public void setBeMoreFirst(int beMoreFirst) {
        this.beMoreFirst = beMoreFirst;
    }
    public void setCurrentPosition(int currentPosition) {
        this.currentPosition = currentPosition;
    }
}
class AutoComparator implements Comparator<Auto> {
    @Override
    public int compare(Auto o1, Auto o2) {

```

```

        return Integer.compare(o1.getCurrentPosition(),
o2.getCurrentPosition());
    }
}

public class Variant210 {
    public static void main(String[] args) {
        ArrayList<Auto> autos = new ArrayList<Auto>();
        for (int i = 0; i < 4; i++) {
            autos.add(new Auto("Автомобиль: " + i));
        }
        autos.sort(new AutoComparator());
        int i = 0;
        for (Auto auto : autos) {
            auto.setBeMoreFirst(i++);
        }
        boolean isEndOfRace = false;
        while (!isEndOfRace) {
            isEndOfRace = true;
            for (Auto auto : autos) {
                if (auto.getCurrentPosition() +
auto.getSpeed() >= 100) {
                    auto.setCurrentPosition(100);
                    continue;
                }
            }

            auto.setCurrentPosition(auto.getCurrentPosition() +
auto.getSpeed());
            isEndOfRace = false;
        }
        autos.sort(new AutoComparator());
    }
}

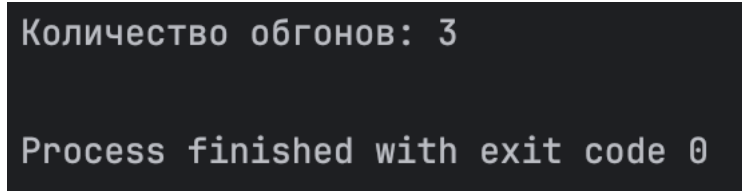
```



```

        i = 0;
        int count = 0;
        for (Auto auto : autos) {
            if (i > auto.getBeMoreFirst()) {
                count = count + (i -
auto.getBeMoreFirst());
            }
            i++;
        }
        System.out.println("Количество обгонов: " + count);
    }
}

```



```

Количество обгонов: 3

Process finished with exit code 0

```

Рисунок 4 – Результат задания 4

Вывод

В ходе выполнения лабораторной работы была изучена работа с коллекциями в языке программирования Java.