Федеральное агентство связи
Ордена Трудового Красного Знамени федеральное государственное
бюджетное учреждение высшего образования
«Московский технический университет связи и информатики»

Кафедра Математической кибернетики и
информационных технологий

Лабораторная работа №4
по дисциплине: «Структуры и алгоритмы обработки данных»
на тему: «Реализация стека/дека»

Выполнил студент
группы БФИ1902
Гусев Н. С.
Проверил:
Мкртчян Г. М.

Москва, 2021 г.

Оглавление

1. Цель работы

Цель работы: реализовать работу стека и дека, а также рассмотреть их работу.

2. Задание на лабораторную работу

1) Отсортировать строки файла, содержащие названия книг, в алфавитном порядке с использованием двух деков.

2) Дек содержит последовательность символов для шифровки сообщений. Дан текстовый файл, содержащий зашифрованное сообщение. Пользуясь деком, расшифровать текст. Известно, что при шифровке каждый символ сообщения заменялся следующим за ним в деке по часовой стрелке через один.

3) Даны три стержня и n дисков различного размера. Диски можно надевать на стержни, образуя из них башни. Перенести n дисков со стержня A на стержень C, сохранив их первоначальный порядок. При переносе дисков необходимо соблюдать следующие правила: - на каждом шаге со стержня на стержень переносить только один диск; - диск нельзя помещать на диск меньшего размера; - для промежуточного хранения можно использовать стержень B. Реализовать алгоритм, используя три стека вместо стержней A, B, C. Информация о дисках хранится в исходном файле.

4) Дан текстовый файл с программой на алгоритмическом языке. За один просмотр файла проверить баланс круглых скобок в тексте, используя стек.

5) Дан текстовый файл с программой на алгоритмическом языке. За один просмотр файла проверить баланс квадратных скобок в тексте, используя дек.

6) Дан файл из символов. Используя стек, за один просмотр файла напечатать сначала все цифры, затем все буквы, и, наконец, все остальные символы, сохраняя исходный порядок в каждой группе символов.

7) Дан файл из целых чисел. Используя дек, за один просмотр файла напечатать сначала все отрицательные числа, затем все положительные числа, сохраняя исходный порядок в каждой группе.

8) Дан текстовый файл. Используя стек, сформировать новый текстовый файл, содержащий строки исходного файла, записанные в обратном порядке: первая строка становится последней, вторая – предпоследней и т.д.

9) Дан текстовый файл. Используя стек, вычислить значение логического выражения, записанного в текстовом файле в следующей форме:

<ЛВ> ::= T | F | (N) | (A) | (X) | (O), где буквами обозначены логические константы и операции: T – True, F – False, N – Not, A – And, X – Xor, O – Or.

10) Дан текстовый файл. В текстовом файле записана формула следующего вида:

<Формула> ::= <Цифра> | M(<Формула>, <Формула>) | N(<Формула>, <Формула>)

< Цифра > ::= 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 , где буквами обозначены функции:

M – определение максимума, N – определение минимума.

Используя стек, вычислить значение заданного выражения.

11) Дан текстовый файл. Используя стек, проверить, является ли содержимое текстового файла правильной записью формулы вида:

< Формула > ::= < Терм > | < Терм > + < Формула > | < Терм > - < Формула >

< Терм > ::= < Имя > | (< Формула >)

< Имя > ::= x | y | z

3. Листинг программы

```
import java.io.BufferedReader;
import java.io.File;
import java.io.FileReader;
import java.io.IOException;
import java.util.ArrayDeque;
import java.util.Arrays;

public class Task1 {
    public static void main(String[] args) {
        ArrayDeque<String> lines = new ArrayDeque<>();
        EnterText(lines);
        String[] text = lines.toArray(new String[0]);
        Arrays.sort(text);
        lines.clear();
        lines.addAll(Arrays.asList(text));
        System.out.println("\nРезультат: " + lines);
    }

    public static void EnterText(ArrayDeque<String> lin1) {
        try {
            File file = new File("D:\\input\\input.txt");
            FileReader fr = new FileReader(file);
            BufferedReader reader = new BufferedReader(fr);
            String line = reader.readLine();
            while (line != null) {
                System.out.println(line);
                lin1.add(line);
                line = reader.readLine();
            }
        } catch (IOException e) {
            e.printStackTrace();
```

```
        }
    }
}

import java.io.*;

public class Task2 {
    private final static char[] DEK = {'a', 'b', 'g', 'u', 'i', 'o', 'e',
't', 'n', 's', 'h', 'v', 'c', 'y'};

    public static void main(String[] args) throws IOException {
        BufferedReader reader = new BufferedReader(new FileReader(new
File("D:\\input\\input2.txt")));
        BufferedWriter writer = new BufferedWriter(new FileWriter(new
File("D:\\input\\output2.txt")));

        int i;
        while ((i = reader.read()) != -1) {
            char ch = (char) i;
            writer.append(switchLetter(ch));
            writer.flush();
        }
        reader.close();
        writer.close();
    }

    private static char switchLetter(char ch) {
        char outchar = '0';
        for (int i = 2; i < DEK.length; i++) {
            char c = DEK[i];
            if (c == ch) {
                outchar = DEK[i - 2];
                break;
            }
        }
        if (outchar == '0')
            outchar = ch;
        return outchar;
    }
}

public class Task3_1 {
    // Структура для представления стека
    static class Stack {
        int capacity;
        int top;
        int[] array;
    }

    // функция для создания стека заданной емкости.
    Stack createStack(int capacity) {
        Stack stack = new Stack();
        stack.capacity = capacity;
        stack.top = -1;
        stack.array = new int[capacity];
        return stack;
    }

    // Стек заполнен, когда вершина равна last index
    boolean isFull(Stack stack) {
        return (stack.top == stack.capacity - 1);
    }

    // Стек пуст, когда вершина равна -1
```

```
boolean isEmpty(Stack stack) {
    return (stack.top == -1);
}

// Функция для добавления элемента в стек. Это увеличивается сверху на 1
void push(Stack stack, int item) {
    if (isFull(stack))
        return;
    stack.array[++stack.top] = item;
}

// Функция для удаления элемента из стека. Это уменьшает вершину на 1
int pop(Stack stack) {
    if (isEmpty(stack))
        return Integer.MIN_VALUE;
    return stack.array[stack.top--];
}

// Функция для реализации легального движения между полюсами
void moveDisksBetweenTwoPoles(Stack src, Stack dest, char s, char d) {
    int pole1TopDisk = pop(src);
    int pole2TopDisk = pop(dest);
    // Когда полюс 1 пуст
    if (pole1TopDisk == Integer.MIN_VALUE) {
        push(src, pole2TopDisk);
        moveDisk(d, s, pole2TopDisk);
    }
    // Когда полюс pole2 пуст
    else if (pole2TopDisk == Integer.MIN_VALUE) {
        push(dest, pole1TopDisk);
        moveDisk(s, d, pole1TopDisk);
    }
    // Когда верхний диск pole1> верхний диск pole2
    else if (pole1TopDisk > pole2TopDisk) {
        push(src, pole1TopDisk);
        push(src, pole2TopDisk);
        moveDisk(d, s, pole2TopDisk);
    }
    // Когда верхний диск pole1 <верхний диск pole2
    else {
        push(dest, pole2TopDisk);
        push(dest, pole1TopDisk);
        moveDisk(s, d, pole1TopDisk);
    }
}

// Функция для отображения движения дисков
void moveDisk(char fromPeg, char toPeg, int disk) {
    System.out.println("Move the disk " + disk +
            " from " + fromPeg + " to " + toPeg);
}

// Функция для реализации загадки ТОН
void tohIterative(int num_of_disks, Stack
        src, Stack aux, Stack dest) {
    int i, total_num_of_moves;
    char s = '1', d = '3', a = '2';
    // Если количество дисков четное, то чередуем
    // полюс назначения и вспомогательный полюс
    if (num_of_disks % 2 == 0) {
        char temp = d;
        d = a;
        a = temp;
    }
```

```
        total_num_of_moves = (int) (Math.pow(2, num_of_disks) - 1);
        // Большие диски будут вставлены первыми
        for (i = num_of_disks; i >= 1; i--)
            push(src, i);
        for (i = 1; i <= total_num_of_moves; i++) {
            if (i % 3 == 1)
                moveDisksBetweenTwoPoles(src, dest, s, d);
            else if (i % 3 == 2)
                moveDisksBetweenTwoPoles(src, aux, s, a);
            else if (i % 3 == 0)
                moveDisksBetweenTwoPoles(aux, dest, a, d);
        }
    }

    // Программа драйвера для проверки вышеуказанных функций
    public static void main(String[] args) {
        // Ввод: количество дисков
        int num_of_disks = 3;
        Task3_1 ob = new Task3_1();
        Stack src, dest, aux;
        // Создаем три стека размером num_of_disks держать диски
        src = ob.createStack(num_of_disks);
        dest = ob.createStack(num_of_disks);
        aux = ob.createStack(num_of_disks);
        ob.tohIterative(num_of_disks, src, aux, dest);
    }
}

import java.io.BufferedReader;
import java.io.File;
import java.io.FileReader;
import java.io.IOException;
import java.util.Stack;

public class Task4 {
    public static void testBrackets(String str) {
        Stack<Character> left_brackets = new Stack<>();
        Stack<Character> right_brackets = new Stack<>();
        for (char c : str.toCharArray()) {
            if (c == ')') {
                right_brackets.push(c);
            } else if (c == '(') {
                left_brackets.push(c);
            }
        }
        while (!left_brackets.empty() && !right_brackets.empty()) {
            char left = left_brackets.peek();
            char right = right_brackets.peek();
            if (left == '(' && right == ')') {
                left_brackets.pop();
                right_brackets.pop();
            } else
                break;
        }
        if (left_brackets.empty() && right_brackets.empty())
            System.out.println("OK");
        else
            System.out.println("FAIL");
    }

    public static void main(String[] args) {
        try {
            File file = new File("D:\\input\\input4.txt");
            FileReader fr = new FileReader(file);
```

```
            BufferedReader reader = new BufferedReader(fr);
            StringBuilder line = new StringBuilder(reader.readLine());
            String tempLine = "";
            boolean bool = true;
            while (bool) {
                line.append(tempLine);
                tempLine = reader.readLine();
                if (tempLine == null)
                    bool = false;
            }
            System.out.println("Получившаяся строка: " + line);
            System.out.print("Проверка: ");
            testBrackets(line.toString());
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
}

import java.io.BufferedReader;
import java.io.File;
import java.io.FileReader;
import java.io.IOException;
import java.util.ArrayDeque;
import java.util.Deque;

public class Task5 {
    public static boolean testBrackets(String str) {
        Deque<Character> brackets = new ArrayDeque<>();
        for (char c : str.toCharArray()) {
            switch (c) {
                case '(':
                    brackets.addFirst(c);
                    break;
                case ')':
                    if (brackets.isEmpty() ||
!brackets.removeFirst().equals('('))
                        return false;
                    break;
                default:
                    break;
            }
        }
        return brackets.isEmpty();
    }

    public static void main(String[] args) {
        try {
            File file = new File("D:\\input\\input4.txt");
            FileReader fr = new FileReader(file);
            BufferedReader reader = new BufferedReader(fr);
            StringBuilder line = new StringBuilder(reader.readLine());
            String tempLine = "";
            boolean bool = true;
            while (bool) {
                line.append(tempLine);
                tempLine = reader.readLine();
                if (tempLine == null)
                    bool = false;
            }
            System.out.println("Получившаяся строка: " + line);
            System.out.println("Проверка: " + (testBrackets(line.toString())
? "OK" : "FAIL"));
        } catch (IOException e) {
```

```
            e.printStackTrace();
        }
    }
}

import java.io.BufferedReader;
import java.io.File;
import java.io.FileReader;
import java.io.IOException;
import java.util.Stack;

public class Task6 {
    public static void main(String[] args) {
        try {
            File file = new File("D:\\input\\input5.txt");
            FileReader fr = new FileReader(file);
            BufferedReader reader = new BufferedReader(fr);
            String line = String.valueOf(reader.readLine());
            Chain(line);
        }
        catch(IOException e) {
            e.printStackTrace();
        }
    }

    public static void Chain(String line) {
        Stack<Character> chain = new Stack<>();
        for(int i = 0; i < line.length(); i++) {
            if(Character.isDigit(line.charAt(i))) {
                chain.push(line.charAt(i));
            }
        }
        for(int i = 0; i < line.length(); i++) {
            if(Character.isLetter(line.charAt(i))) {
                chain.push(line.charAt(i));
            }
        }
        for(int i = 0; i < line.length(); i++) {
            if(!Character.isDigit(line.charAt(i)) &&
!Character.isLetter(line.charAt(i)) ) {
                chain.push(line.charAt(i));
            }
        }
        System.out.println(chain);
    }
}

import java.io.BufferedReader;
import java.io.File;
import java.io.FileReader;
import java.io.IOException;
import java.util.ArrayDeque;
import java.util.Deque;
import java.util.Stack;

public class Task7 {
    public static void main(String[] args) {
        try {
            File file = new File("D:\\input\\input7.txt");
            FileReader fr = new FileReader(file);
            BufferedReader reader = new BufferedReader(fr);
            String line = String.valueOf(reader.readLine());
            Chain1(line);
        } catch (IOException e) {
```

```
            e.printStackTrace();
        }
    }

    public static void Chain1(String line) {
        Deque<Integer> chain = new ArrayDeque<>();
        String[] strArr = line.split(" ");
        int[] numArr = new int[strArr.length];
        for (int i = 0; i < strArr.length; i++) {
            numArr[i] = Integer.parseInt(strArr[i]);
        }
        for (int j : numArr) {
            if (j < 0) {
                chain.addLast(j);
            }
        }
        for (int j : numArr) {
            if (j > 0) {
                chain.addLast(j);
            }
        }
        System.out.println(chain);
    }
}

import java.io.*;
import java.util.ArrayList;
import java.util.Arrays;
import java.util.Scanner;
import java.util.Stack;

public class Task8 {
    public static void main(String[] args) throws FileNotFoundException,
UnsupportedEncodingException {
        PrintWriter writer = new PrintWriter("D:\\input\\output8.txt", "UTF-
8");
        Stack<String> list = new Stack<>();
        try (Scanner scan = new Scanner(new File("D:\\input\\input8.txt"))) {
            while (scan.hasNextLine()) {
                list.push(scan.nextLine());
            }
            while(!list.empty()) {
                String out = list.pop();
                writer.println(out);
            }
            System.out.println("Выполнено!");
            writer.close();
        } catch (FileNotFoundException e) {
            e.printStackTrace();
        }
    }
}

import java.io.*;
import java.util.Iterator;
import java.util.Stack;

public class Task9 {
    public static void main(String[] args) throws IOException {
        BufferedReader reader = new BufferedReader(new FileReader(new
File("D:\\input\\input9.txt")));
        Stack<Character> st = new Stack<Character>();
        Stack<Boolean> num = new Stack<Boolean>();
        Stack<Character> letter = new Stack<Character>();
```

```
Stack<Boolean> preanswer = new Stack<Boolean>();
int i = 0;

while ((i = reader.read()) != -1) {
    char ch = (char) i;
    st.add(ch);
}

for (char r : st) {
    if (Character.isDigit(r)) {
        if (r == '0')
            num.push(false);
        else
            num.push(true);
    }
}

for (char r : st) {
    if (Character.isAlphabetic(r)) {
        letter.push(r);
    }
}

while (preanswer.size() != 6) {
    char s = letter.pop();
    switch (s) {
        case ('O'):
            boolean num1 = num.pop();
            boolean num2 = num.pop();
            preanswer.add(num1 || num2);
            break;
        case ('X'):
            boolean num3 = num.pop();
            boolean num4 = num.pop();
            if (num3 != num4) {
                preanswer.add(true);
            } else {
                preanswer.add(false);
            }
            break;
        case ('A'):
            boolean num5 = num.pop();
            boolean num6 = num.pop();
            preanswer.add(num5 && num6);
            break;
        case ('N'):
            boolean num7 = num.pop();
            if (num7) {
                preanswer.add(false);
            } else {
                preanswer.add(true);
            }
            break;
        case ('F'):
            preanswer.add(false);
            break;
        case ('T'):
            preanswer.add(true);
            break;
    }
}

Iterator <Boolean> iterator2 = preanswer.iterator();
boolean answer = false;
```

```
        while (iterator2.hasNext()) {
            if (iterator2.next()){
                answer = true;
            }
        }
        System.out.println(answer);
    }
}

import java.util.Stack;

public class Task10 {

    public static Stack<String> slov = new Stack<>();
    public static int first=-1;
    public static int second=-1;
    public static int top=-1;

    public static void zapoln(String [] s) {
        for (String value : s) {
            slov.push(value);
        }
    }

    public static void start() {
        String d;
        int l=slov .size();
        for (int i = 0; i <l; i++) {
            d=slov .pop();

            if(first==-
1&&(d.equals("0")||d.equals("1")||d.equals("2")||d.equals("3")||d.equals("4")
||d.equals("5")||d.equals("6")||d.equals("7")||d.equals("8")||d.equals("9")))
            {
                first=Integer.parseInt(d);
                continue;
            }

            if(second==-
1&&(d.equals("0")||d.equals("1")||d.equals("2")||d.equals("3")||d.equals("4")
||d.equals("5")||d.equals("6")||d.equals("7")||d.equals("8")||d.equals("9")))
            {
                second=Integer.parseInt(d);
                continue;
            }

            if(d.equals("m")&&first!=-1&&second!=-1){
                top=Math.min(first,second);
                first=-1;
                second=-1;
                continue;
            }

            if(d.equals("m")&&first!=-1){
                top=Math.min(first,top);
                first=-1;
                continue;
            }

            if(d.equals("M")&&first!=-1&&second!=-1){
                top=Math.max(first,second);
                first=-1;
                second=-1;
                continue;
```

```
            }

            if(d.equals("M")&&first!=-1){
                top=Math.max(first,top);
                first=-1;
            }
        }
    }

    public static void main(String[] args) {
        zapoln(new String[]{"M", "(", "5", ",", "m", "(", "6", ",", "8", ")",
")"});
        start();
        System.out.println(top);
    }
}

import java.io.*;
import java.util.Iterator;
import java.util.Stack;
public class Task11 {
    public static void main(String[] args) throws IOException {
        BufferedReader reader = new BufferedReader(new FileReader(new
File("D:\\input\\input11.txt")));
        Stack<Character> st = new Stack<>();
        Stack<Character> letter = new Stack<Character>();
        Stack<Character> symbols = new Stack<Character>();
        int open = 0,close = 0;
        int i = 0;
        while ((i = reader.read()) != -1) {
            char ch = (char) i;
            st.add(ch);
        }
        for (char r : st) {
            if (Character.isAlphabetic(r)) {
                letter.push(r);
            }
        }
        for (char r : st) {
            if (!(Character.isDigit(r) || Character.isAlphabetic(r))) {
                symbols.push(r);
            }
        }
        int kol = 0;
        while (symbols.size() != 0) {
            char s = symbols.pop();
            switch (s) {
                case ('+'):
                case ('-'):
                    kol++;
                    break;
                case ('('):
                    open++;
                    break;
                case (')'):
                    close++;
                    break;
            }
        }

        int w = 0;
        while (letter.size() != 0) {
            char s = letter.pop();
            switch (s) {
```

```
                case ('x'):
                case ('y'):
                case ('z'):
                    w++;
                    break;
            }
        }

        if ((w-1 == kol) && (close==open))
            System.out.println("Формула имеет правильный вид");
        else
            System.out.println("Формула имеет не правильный вид");
    }
}
```

4. Результат работы программы

```
"C:\Program Files\Java\jdk1.8.0_261\bin\java.exe" ...
Властелин колец
Гордость и предубеждение
Тёмные начала
Автостопом по галактике
Гарри Поттер и Кубок огня
Убить пересмешника
Винни Пух
Лев, колдунья и платяной шкаф
Джейн Эйр
Уловка-22
Грозовой перевал
Пение птиц
Ребекка
Над пропастью во ржи

Результат: [Автостопом по галактике, Винни Пух, Властелин колец, Гарри Поттер и Кубок огня, Гордость и предубеждение, Грозовой перевал, Джейн Эйр, Лев, колдунья и платяной шкаф, Над пропастью во ржи, Пение птиц, Ребекка,
 Тёмные начала, Убить пересмешника, Уловка-22]
```

Рисунок 1 – Сортировка книг

```
"C:\Program Files\Java\jdk1.8.0_261\bin\java.exe" ...
Получившаяся строка: алг Сумма квадратов (арг цел n, рез цел S)  дано | n > 0  надо | S = 1*1 + 2*2 + 3*3 + … + n*nнач цел i|  ввод n; S:=0|  нц для i от 1 до n|  |  S := S + i * i|  кц|  вывод "S = ", Skon
Проверка: ОК
```

Рисунок 2 – Проверка на скобки

```
"C:\Program Files\Java\jdk1.8.0_261\bin\java.exe" ...
[3, 2, 4, 7, 3, 2, 9, 4, 8, 4, 8, 9, 3, 2, 8, 7, 4, 0, 7, 8, o, r, g, h, r, b, o, u, h, r, g, o, r, g, h, r, i, g, g, d, r, j, h, v, d, r, v, j, r, d, o, v, r, g, i, ;, [, -, -, -, -, -, -, -, =, -, =, -, =, -, -, =, -, =]
```

Рисунок 3 – Сортировка символов

5. Вывод

Я рассмотрел реализацию стека и дека и научился работать с ними.