

Федеральное агентство связи

**Государственное бюджетное образовательное учреждение высшего
Образование**

Ордена Трудового Красного Знамени

«Московский технический университет связи и информатики»

Кафедра «МКиИТ»

дисциплина «СиАОД»

Отчет по Лабораторной работе №4

Подготовил студент
группы БВТ1902: Капленко Е. М.
Руководитель: Мкртчян Г. М.

Москва 2020

Задания для данной лабораторной работы:

Лабораторная работа 4. Реализация стека/дека.

Реализовать следующие структуры данных:

- **Стек (stack):**
операции *для стека*: инициализация, проверка на пустоту, добавление нового элемента в начало, извлечение элемента из начала;
- **Дек (двусторонняя очередь, deque):**
операции *для дека*: инициализация, проверка на пустоту, добавление нового элемента в начало, добавление нового элемента в конец, извлечение элемента из начала, извлечение элемента из конца.

Разработать программу обработки данных, содержащихся в заранее подготовленном `txt`-файле, в соответствии с заданиями, применив указанную в задании структуру данных. Результат работы программы вывести на экран и сохранить в отдельном `txt`-файле.

Задания:

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:

$$\langle \text{ЛВ} \rangle ::= \text{T} \mid \text{F} \mid (\text{N}\langle \text{ЛВ} \rangle) \mid (\langle \text{ЛВ} \rangle \text{A} \langle \text{ЛВ} \rangle) \mid (\langle \text{ЛВ} \rangle \text{X} \langle \text{ЛВ} \rangle) \mid (\langle \text{ЛВ} \rangle \text{O} \langle \text{ЛВ} \rangle),$$
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:

$$\langle \text{Формула} \rangle ::= \langle \text{Цифра} \rangle \mid \text{M}(\langle \text{Формула} \rangle, \langle \text{Формула} \rangle) \mid \text{N}(\langle \text{Формула} \rangle, \langle \text{Формула} \rangle)$$

$$\langle \text{Цифра} \rangle ::= 0 \mid 1 \mid 2 \mid 3 \mid 4 \mid 5 \mid 6 \mid 7 \mid 8 \mid 9$$
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:

$$\langle \text{Формула} \rangle ::= \langle \text{Терм} \rangle \mid \langle \text{Терм} \rangle + \langle \text{Формула} \rangle \mid \langle \text{Терм} \rangle - \langle \text{Формула} \rangle$$

$$\langle \text{Терм} \rangle ::= \langle \text{Имя} \rangle \mid (\langle \text{Формула} \rangle)$$

$$\langle \text{Имя} \rangle ::= x \mid y \mid z$$

Реализация стека в виде отдельного класса:

```
public class stackk {

    public String[] Stack;
    public int top;

    public stackk() {
        Stack = new String[550];
        top = -1;
    }

    public void addElement(String element) {
        Stack[++top] = element;
    }

    public boolean isEmpty() {
        if (top== -1) return true;
        else return false;
    }

    public String deleteElement() {
        String last = Stack[top];
        Stack[top--] = null;
    }
}
```

```

        return last;
    }

    public String readLast() {
        String last = Stack[top];
        return last;
    }
}

```

Выполнение заданий на стеке:

```

/*      "Ханойские башни"
    stackk AStack = new stackk();
    stackk BStack = new stackk();
    stackk CStack = new stackk();

    int n = 6;
    int k = 1;
    int m = 3;

    for (int i = n; i > 0; i--) {
        AStack.addElement(String.valueOf(i));
    }

    CStack = piramidki(AStack, BStack, CStack, n, k,m);

    System.out.print("C Стэк - ");
    while (!CStack.isEmpty()) {
        String el = CStack.deleteElement();
        System.out.print(el + " ");
    }

// "Проверка скобок" 4
    stackk Stack = new stackk();
    try(FileReader reader = new FileReader("notes.txt"))
    {
        // читаем посимвольно
        int c;
        while((c=reader.read())!=-1){

            System.out.print((char)c);

            if (c == 40) {
                Stack.addElement(Character.toString(c));
            }
            if (c== 41) {
                if (Stack.isEmpty()) {
                    Stack.addElement("y");
                    break;
                }
                if ("(".equals(Stack.readLast()) ) Stack.deleteElement();
                else break;
            }
        }
    }
    catch(IOException ex){

        System.out.println(ex.getMessage());
    }

    System.out.println("Баланс соблюден? "+ Stack.isEmpty());

```

```

        */

        /*6 задание
        stackk Stack = new stackk();
        stackk Numbers = new stackk();
        stackk Letters = new stackk();
        stackk Symbols = new stackk();
        try {
            BufferedReader reader = new BufferedReader(new
FileReader("sort.txt"));
            int c;
            while ((c = reader.read()) != -1) {
                if (c==32) continue;
                if (c == 48 ||c ==49 ||c ==50 ||c ==51 ||c ==52 ||c ==53|| c
==54 ||c ==55 ||c ==56 ||c ==57) {
                    Numbers.addElement(Character.toString(c));
                    continue;
                }
                if (Character.isLetter(c)) {
                    Letters.addElement(Character.toString(c));
                    continue;
                }
                Symbols.addElement(Character.toString(c));
            }
        }
        catch(IOException ex){
            System.out.println(ex.getMessage());
        }

        while (!Symbols.isEmpty()) {
            Stack.addElement(Symbols.deleteElement());
        }
        while (!Letters.isEmpty()) {
            Stack.addElement(Letters.deleteElement());
        }
        while (!Numbers.isEmpty()) {
            Stack.addElement(Numbers.deleteElement());
        }
        try(FileWriter writer = new FileWriter("writesort.txt", false))
        {
            while (!Stack.isEmpty()) {
                String text = Stack.deleteElement();
                writer.write(text);
            }
            writer.flush();
        }
        catch(IOException ex){

            System.out.println(ex.getMessage());

        }
    */

    /*8 задание
    stackk Stack = new stackk();
    try {
        BufferedReader reader = new BufferedReader(new
FileReader("notes.txt"));
        String line = reader.readLine();
        while (line!= null) {
            Stack.addElement(line);
            line = reader.readLine();
        }
    } catch (FileNotFoundException e) {

```

```

        e.printStackTrace();
    } catch (IOException e) {
        e.printStackTrace();
    }

    try(FileWriter writer = new FileWriter("writepere.txt", false))
    {
        // запись всей строки
        while (!Stack.isEmpty()) {
            String text = Stack.deleteElement();
            writer.write(text);
            writer.append('\n');
        }
        writer.flush();
    }
    catch(IOException ex){

        System.out.println(ex.getMessage());
    }
}

public class task9 {
    public static boolean cpv(String str1,String str2){
        boolean y1=false;
        if(str1.charAt(0)==str2.charAt(0))
            y1=true;
        return y1;
    }

    public task9(String Str) throws ScriptException {
        ScriptEngineManager manager = new ScriptEngineManager();
        ScriptEngine engine = manager.getEngineByName("JavaScript");
        StringBuilder str1 = new StringBuilder();
        Stack stk= new Stack();
        Stack stk2 = new Stack();
        for (int i = 0; i < Str.length(); i++) {
            stk.add(Str.charAt(i));
        }
        for (int i = 0; i < Str.length(); i++) {
            stk2.add(stk.pop());
        }
        for (int i = 0; i < Str.length(); i++) {
            if (cpv(stk2.peek().toString(), "T"))
                str1.append("true ");
            if (cpv(stk2.peek().toString(), "F"))
                str1.append("false ");
            if (cpv(stk2.peek().toString(), "N"))
                str1.append("! ");
            if (cpv(stk2.peek().toString(), "A") ||
cpv(stk2.peek().toString(), "*"))
                str1.append("&& ");
            if (cpv(stk2.peek().toString(), "X"))
                str1.append("!= ");
            if (cpv(stk2.peek().toString(), "O") ||
cpv(stk2.peek().toString(), "+"))
                str1.append("|| ");
            if (cpv(stk2.peek().toString(), "("))
                str1.append("(" ");
            if (cpv(stk2.peek().toString(), ")"))
                str1.append(")");
            stk2.pop();
        }

        Object o =engine.eval(str1.toString());
        System.out.println("task9: "+o);
    }
}

```

```

    }
}
public class task10 {
    public static boolean cpv(String str1, String str2) {
        boolean y1 = false;
        if (str1.charAt(0) == str2.charAt(0))
            y1 = true;
        return y1;
    }

    public task10(String Str) throws ScriptException {
        ScriptEngineManager manager = new ScriptEngineManager();
        ScriptEngine engine = manager.getEngineByName("JavaScript");
        //Object o =engine.eval(code);
        StringBuilder str1 = new StringBuilder();
        Stack stk = new Stack();
        Stack stk2 = new Stack();
        for (int i = 0; i < Str.length(); i++) {
            stk.add(Str.charAt(i));
            //System.out.println(stk.peek());
        }
        for (int i = 0; i < Str.length(); i++) {
            stk2.add(stk.pop());
            //System.out.println(stk.peek());
        }
        for (int i = 0; i < Str.length(); i++) {
            if (cpv(stk2.peek().toString(), "0"))
                str1.append("0");
            if (cpv(stk2.peek().toString(), "1"))
                str1.append("1");
            if (cpv(stk2.peek().toString(), "2"))
                str1.append("2");
            if (cpv(stk2.peek().toString(), "3"))
                str1.append("3");
            if (cpv(stk2.peek().toString(), "4"))
                str1.append("4");
            if (cpv(stk2.peek().toString(), "5"))
                str1.append("5");
            if (cpv(stk2.peek().toString(), "6"))
                str1.append("6");
            if (cpv(stk2.peek().toString(), "7"))
                str1.append("7");
            if (cpv(stk2.peek().toString(), "8"))
                str1.append("8");
            if (cpv(stk2.peek().toString(), "9"))
                str1.append("9");
            if (cpv(stk2.peek().toString(), "M"))
                str1.append("Math.max");
            if (cpv(stk2.peek().toString(), "N"))
                str1.append("Math.min");
            if (cpv(stk2.peek().toString(), ","))
                str1.append(",");
            if (cpv(stk2.peek().toString(), "("))
                str1.append("(");
            if (cpv(stk2.peek().toString(), ")"))
                str1.append(")");
            stk2.pop();
        }

        Object o = engine.eval(str1.toString());
        System.out.println("task10: " + o);
    }
}
public class task11 {
    public static boolean cpv(String str1, String str2) {

```

```

        boolean y1 = false;
        if (str1.charAt(0) == str2.charAt(0))
            y1 = true;
        return y1;
    }

    public task11(String Str) throws ScriptException {
        ScriptEngineManager manager = new ScriptEngineManager();
        ScriptEngine engine = manager.getEngineByName("JavaScript");
        //Object o =engine.eval(code);
        StringBuilder str1 = new StringBuilder();
        Stack stk = new Stack();
        Stack stk2 = new Stack();
        for (int i = 0; i < Str.length(); i++) {
            stk.add(Str.charAt(i));
            //System.out.println(stk.peek());
        }
        for (int i = 0; i < Str.length(); i++) {
            stk2.add(stk.pop());
            //System.out.println(stk.peek());
        }
        for (int i = 0; i < Str.length(); i++) {
            if (cpv(stk2.peek().toString(), "x"))
                str1.append("1");
            if (cpv(stk2.peek().toString(), "y"))
                str1.append("2");
            if (cpv(stk2.peek().toString(), "z"))
                str1.append("3");
            if (cpv(stk2.peek().toString(), "("))
                str1.append("(");
            if (cpv(stk2.peek().toString(), ")"))
                str1.append(")");
            if (cpv(stk2.peek().toString(), "+"))
                str1.append("+");
            if (cpv(stk2.peek().toString(), "-"))
                str1.append("-");
            stk2.pop();
        }
        try {
            Object o = engine.eval(str1.toString());
            System.out.println("task11: true");
        } catch (Throwable e) {
            System.out.println("task11: false");
        }
    }
}

```

Реализация дека:

```

public class Deque {
    int size;
    int head;
    int tail;
    int[] data;

    Deque(int size){
        data = new int [this.size = size];
    }
    void pushBack(int value){
        if (++tail == size)
            tail = 0;
    }
}

```



```

        data[tail] = value;
    }
    int popBack(){
        int ret = data[tail];
        if (--tail < 0)
            tail = size - 1;
        return ret;
    }
    void pushFront(int value){
        data[head] = value;
        if (--head < 0)
            head = size - 1;
    }
    int popFront(){
        if (++head == size)
            head = 0;
        return data[head];
    }
    boolean isEmpty(){
        return head == tail;
    }

    public char peekBack() {
        char c = '\0';
        return c;
    }
}

```

Выполнение заданий, с использованием дека:

```

public class task1 {
    public static boolean cpv(String str1, String str2) {
        boolean y1 = false;
        int len=Math.min(str1.length(),str2.length());
        for (int i = 0; i < len; i++) {
            if(str1.charAt(i)>str2.charAt(i)){
                return true;
            }
        }
        if(len==0)
            return true;
        return y1;
    }
    public static boolean cpv2(String str1, String str2) {
        boolean y1 = true;
        int len=Math.min(str1.length(),str2.length());
        for (int i = 0; i < len; i++) {
            if(str1.charAt(i)!=str2.charAt(i)){
                return false;
            }
        }
        return y1;
    }
}

public task1(String Str) {
    String[] words = Str.split(" ");
    Deque a = new LinkedList();
    Deque b = new LinkedList();
    for (int i = 0; i < words.length; i++) {
        a.push(words[i]);
    }
    String str = " ";
    while (!a.isEmpty()) {

```

```

        for (int i = 0; i < a.size(); i++) {
            if (cpv(str, a.peekFirst().toString())) {
                str = a.peekFirst().toString();
            }
            a.addLast(a.removeFirst());
        }
        for (int i = 0; i < a.size(); i++) {
            if (cpv2(str, a.peekFirst().toString())) {
                a.removeFirst();
                break;
            }
            a.addLast(a.removeFirst());
        }
        b.addLast(str);
        str="";
    }
    System.out.print("task1: ");
    while (!b.isEmpty()){
        System.out.print(b.removeFirst()+" ");
    }
}

public class task2 {
    public static boolean cpv(String str1, String str2) {
        boolean y1 = false;
        if (str1.charAt(0) == str2.charAt(0))
            y1 = true;
        return y1;
    }

    public task2(Deque dec, String str) {
        String str1 = "";
        for (int i = 0; i < str.length(); i++) {
            while (str1.length() < i + 1) {
                if (cpv(Character.toString(str.charAt(i)),
dec.peekLast().toString())) {
                    dec.add(dec.pop());
                    dec.add(dec.pop());

                    str1 += dec.peekLast();
                }
                dec.add(dec.pop());
            }
        }
        System.out.println("task2: " + str1);
    }
}

public class task5 {
    public task5 (String str) {
        Deque stc = new LinkedList();
        for(int i=0;i<str.length();i++){
            if(str.charAt(i)=='[')
                stc.push(str.charAt(i));
            else if(str.charAt(i)==']'&&!stc.isEmpty())
                stc.remove();
        }
        if(stc.size()==0)
            System.out.println("Task5: true");
        else
            System.out.println("Task5: false");
    }
}

public class task7 {
    public task7(int[] arr) {

```

```
Deque deq= new LinkedList();
for (int i=0;i<arr.length;i++){
    if(arr[i]<0)
        deq.addLast(arr[i]);
    else
        deq.addFirst(arr[i]);
}
System.out.print("task7: ");
for (int i=0;i<arr.length;i++){
    System.out.print(" "+deq.removeLast());
}
}
```

Вывод: В ходе выполнения этой лабораторной работы реализованы стек и дек. Ознакомилась с вариациями задач, в которых их уместнее всего использовать.