

Министерство цифрового развития, связи и массовых коммуникаций
Российской Федерации

Ордена Трудового Красного Знамени

«Московский технический университет связи и информатики»

Кафедра «Математической кибернетики и информационных технологий»

дисциплина «Структуры и алгоритмы обработки данных»

Отчет по лабораторной работе №4

“ Реализация работы со стеком и деком”

Выполнил: студенты группы БФИ1901

Козырев Сергей Владимирович

Проверил:

Кутейников Иван Алексеевич

Москва 2021

Оглавление

1.Задание на лабораторную работу.	3
2.Листинг программы (Lab4).....	4
2.1 Листинг программы First.....	4
2.2 Листинг программы Second.	5
2.3 Листинг программы Third.	6
2.4 Листинг программы Fourth.....	6
2.5 Листинг программы Fifth.	7
2.6 Листинг программы Sixth.....	8
2.7 Листинг программы Seventh.	9
2.8 Листинг программы Eighth.....	10
2.9 Листинг программы Ninth.	11
2.10 Листинг программы Tenth.....	12
2.11 Листинг программы Eleventh.	14
3.Вывод.....	15
Список использованных источников	16

1.Задание на лабораторную работу.

Задание на лабораторную работу показано на рисунке 1.

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

Рисунок 1 – Задание на работу.

2.Листинг программы (Lab4)

2.1 Листинг программы First.

```
package Lab4;
import java.io.*;
import java.util.*;
//Отсортировать книги по алфавиту
public class First {
    public static void main(String[] args) throws IOException {
        Deque<String> dec = new LinkedList<>();
        Deque<String> answer = new LinkedList<>();
        File file = new File("D:\\VSCode\\Projects\\Lab4\\files\\Books.txt");
        Scanner scanner = new Scanner(file);
        while (scanner.hasNext()) {
            dec.add(scanner.nextLine());
        }
        scanner.close();
        String min = dec.getFirst();
        int m = dec.size();
        for (int i = 0; i < m-1; i++) {
            Iterator iterator = dec.iterator();
            while (iterator.hasNext()) {
                String s = (String) iterator.next();
                if (s.compareTo(min) < 0)
                    min = s;
            }
            answer.addLast(min);
            dec.remove(min);
            min=dec.getFirst();
        }
        answer.addLast(min);
        System.out.println(answer);
        String temp = answer.getFirst();
        try (PrintWriter pw = new PrintWriter("D:\\VSCode\\Projects\\Lab4\\files\\
\\First.txt")) {
            for (int i=0;i<m;i++){
                Iterator iterator = answer.iterator();
                while (iterator.hasNext()) {
                    String s = (String) iterator.next();
                    pw.println(s + "\\n");
                }
                answer.remove(temp);
            }
        } catch (IOException exc) {
            System.out.println(exc);
        }
    }
}
```

2.2 Листинг программы Second.

```
package Lab4;
import java.io.*;
import java.util.Deque;
import java.util.LinkedList;
import java.util.Scanner;
//Дешифратор на 2 элемента назад

public class Second {
    private final static char[] DEK = {'а','б','в','г','д','е','ё','ж','з','и','й',
    ',','к','л','м','н','о','п','р','с','т','у','ф','х','ц','ч','ш','щ','ъ','ы','ь','э',
    ',','ю','я'}; //Дек с алфавитом
    public static void main(String[] args) throws IOException {
        BufferedReader reader = new BufferedReader(new FileReader(new File("D:\\V
SCode\\Projects\\Lab4\\files\\Info.txt")));
        BufferedWriter writer = new BufferedWriter(new FileWriter(new File("D:\\V
SCode\\Projects\\Lab4\\files\\Second.txt")));
        int i = 0;
        while((i = reader.read()) != -1) {
            char ch = (char) i;
            writer.append(switchLetter(ch));
            writer.flush();
        }

        reader.close();
        writer.close();
        File f = new File("D:\\VSCode\\Projects\\Lab4\\files\\Info.txt");
        System.out.println("Зашифрованное сообщение:");
        out(f);
        System.out.println("Расшифрованное сообщение:");
        f=new File("D:\\VSCode\\Projects\\Lab4\\files\\Second.txt");
        out(f);
    }

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

        if(Character.compare(outchar, '0') == 0)
            outchar = ch;
        return outchar;
    }

    public static void out (File f) throws FileNotFoundException {
```

```

        File file = new File(String.valueOf(f));
        LinkedList<String> answer = new LinkedList<>();
        Scanner scanner = new Scanner(file);
        while (scanner.hasNext()) {
            answer.add(scanner.nextLine());
        }
        System.out.println(answer);
    }
}

```

2.3 Листинг программы Third.

```

package Lab4;
import java.io.*;
import java.util.Deque;
import java.util.LinkedList;
import java.util.Scanner;
import java.util.Stack;

public class Third { //Головоломка с пирадами
    public static void main(String[] args) {
        Stack a = new Stack(); // 3 Башни
        Stack b = new Stack();
        Stack c = new Stack();
        System.out.println("Введите n количество дисков:");
        Scanner scanner = new Scanner(System.in);
        int n = scanner.nextInt();
        for (int i=n;i>0;i--){
            a.add(i);
        }
        swap(a,b,c,n);
        System.out.println("Ответ:");
        System.out.println(c);
    }
    public static void swap (Stack a, Stack b, Stack c,int n){
        //Рекурсивно вызывает сам себя и перезаписывает по разным башням
        if (n == 1) {
            c.add(a.pop());
        } else {
            swap(a,c,b,n-1);
            c.add(a.pop());
            swap(b,a,c,n-1);
        }
    }
}

```

2.4 Листинг программы Fourth.

```

package Lab4;
import java.io.File;
import java.io.FileNotFoundException;
import java.util.*;
//Подсчет количества скобок
public class Fourth {

    public static void main(String[] args) throws FileNotFoundException {
        File file = new File("D:\\VSCode\\Projects\\Lab4\\files\\Fourth.txt");
        Scanner scanner = new Scanner(file);
        Stack temp = new Stack();
        while (scanner.hasNext()) {
            temp.add(scanner.nextLine());
        }
        Iterator iterator = temp.iterator();
        int open=0, close=0;
        while (iterator.hasNext()){
            String s = (String) iterator.next();
            char[] ch= s.toCharArray();
            for (int i=0;i<s.length();i++){ //Проходим по массиву
                char o = '(';
                char c = ')';
                if (Character.compare(ch[i],o)==0) { //Сравниваем элементы со ско
бками
                    open++; //Считаем сколько скобок нашли
                }
                if (Character.compare(ch[i],c)==0) {
                    close++;
                }
            }
        }
        if (open == close) //Если символы равны то совпадает
            System.out.println("Количество символов совпадает");
        else
            System.out.println("Количество символов не совпадает");
    }
}

```

2.5 Листинг программы Fifth.

```

package Lab4;
import java.io.File;
import java.io.FileNotFoundException;
import java.util.*;
//То же самое что и 4, только используем дек вместо стека и квадратные скобки
public class Fifth {

    public static void main(String[] args) throws FileNotFoundException {
        File file = new File("D:\\VSCode\\Projects\\Lab4\\files\\Fifth.txt");
        Scanner scanner = new Scanner(file);

```

```

Deque<String> temp = new LinkedList<String>();
while (scanner.hasNext()) {
    temp.add(scanner.nextLine());
}
Iterator iterator = temp.iterator();
int open=0, close=0;
while (iterator.hasNext()){
    String s = (String) iterator.next();
    char[] ch= s.toCharArray();
    for (int i=0;i<s.length();i++){
        char o = '[';
        char c = ']';
        if (Character.compare(ch[i],o)==0) {
            open++;
        }
        if (Character.compare(ch[i],c)==0) {
            close++;
        }
    }
}
if (open == close)
    System.out.println("Количество символов совпадает");
else
    System.out.println("Количество символов не совпадает");
}
}

```

2.6 Листинг программы Sixth.

```

package Lab4;
import java.io.*;
import java.util.*;
//вывод чисел , букв , символов по отдельности
public class Sixth {

    public static void main(String[] args) throws IOException {
        BufferedReader reader = new BufferedReader(new FileReader(new File("D:\\V
SCode\\Projects\\Lab4\\files\\Sixth.txt")));
        Stack st = new Stack(); //Изначальный стек в нем хранится все
        Stack num = new Stack(); //стек с цифрами
        Stack letter = new Stack(); //стек с буквами
        Stack symbols = new Stack(); //стек с символами
        int i = 0;
        while ((i = reader.read()) != -1) {
            char ch = (char) i;
            st.add(ch);
        }
        Iterator iterator = st.iterator();
        while (iterator.hasNext()) {
            char r=(char) iterator.next();

```



```

        if(Character.isDigit(r)){
            num.push(r);
        }
    }
    Iterator iterator1 = st.iterator();
    while (iterator1.hasNext()) {
        char r=(char) iterator1.next();
        if(Character.isAlphabetic(r)){
            letter.push(r);
        }
    }
    Iterator iterator2 = st.iterator();
    while (iterator2.hasNext()) {
        char r=(char) iterator2.next();
        if(!(Character.isDigit(r)||Character.isAlphabetic(r))){
            symbols.push(r);
        }
    }
    System.out.println("Числа:");
    System.out.println(num);
    System.out.println("Буквы:");
    System.out.println(letter);
    System.out.println("Символы:");
    System.out.println(symbols);
    System.out.println("Изначальный стек:");
    System.out.println(st);
}
}

```

2.7 Листинг программы Seventh.

```

package Lab4;
import java.io.*;
import java.util.*;
//вывод сначала отрицательных , потом положительных
public class Seventh {

    public static void main(String[] args) throws IOException {
        Deque temp = new ArrayDeque();
        Stack s = new Stack();
        File file = new File("D:\\VSCode\\Projects\\Lab4\\files\\Seventh.txt");
        Scanner scanner = new Scanner(file);
        String str = scanner.nextLine();
        String [] str1 = str.split(" ");
        for (int i = 0; i < str1.length; i++) {
            s.push(Integer.valueOf(str1[i]));
        }
        Iterator iterator = s.iterator();
        Iterator iterator1 = s.iterator();
    }
}

```

```

while (iterator1.hasNext()) { //Проходимся по стеку
    int r = (int) iterator1.next(); //Записываем 1 элемент в переменную
    if (r<0){ //Проверяем меньше нуля и записываем в дек
        temp.add(r);
    }
}
while (iterator.hasNext()) {
    int r = (int) iterator.next();
    if (r>0){ //Проверяем меньше нуля и записываем в дек
        temp.add(r);
    }
}
System.out.println(temp); //Выводим дек
}
}

```

2.8 Листинг программы Eighth.

```

package Lab4;
import java.io.*;
import java.util.Scanner;
import java.util.Stack;
//Запись строки в обратном порядке в файл
public class Eighth {
    public static void main(String[] args) throws FileNotFoundException {
        File file = new File("D:\\VSCode\\Projects\\Lab4\\files\\Eighth.txt");
        Scanner scanner = new Scanner(file);
        Stack temp = new Stack(); //Создаем стек
        while (scanner.hasNext()) {
            temp.add(scanner.nextLine()); //Записывается через LIFO (порядок меня
ется)
        }
        try (PrintWriter pw = new PrintWriter("D:\\VSCode\\Projects\\Lab4\\files\\
\\Eighth_answer.txt")) {
            {
                for (int i=0;i<=temp.size();++i) { //Запись стека в файл
                    String s ;
                    s = (String) temp.pop();
                    pw.println(s);
                }
                pw.println(temp.pop());
            }
            temp.remove(temp);
        } catch (IOException exc) {
            System.out.println(exc);
        }
    }
}

```

2.9 Листинг программы Ninth.

```
package Lab4;
import java.io.*;
import java.util.Iterator;
import java.util.Stack;
// Выполнение логических операций над числами с использованием , лог.опер
public class Ninth {
    public static void main(String[] args) throws IOException {
        BufferedReader reader = new BufferedReader(new FileReader(new File("D:\\V
SCode\\Projects\\Lab4\\files\\Ninth.txt")));
        Stack st = new Stack(); //Основной стек в который все записываем
        Stack num = new Stack(); //Стек для чисел
        Stack letter = new Stack(); //Стек для букв
        Stack preanswer = new Stack(); //Стек для пред ответа (значения скобок, с
читаем за скобки)
        int i = 0;
        while ((i = reader.read()) != -1) {
            char ch = (char) i;
            st.add(ch);
        }
        Iterator iterator = st.iterator();
        while (iterator.hasNext()) { //записывает в виде t f
            char r=(char) iterator.next();
            if(Character.isDigit(r)){
                if (r=='0')
                    num.push(false);
                else
                    num.push(true);
            }
        }
        Iterator iterator1 = st.iterator();
        while (iterator1.hasNext()) { //записывает буквы в стек с буквами
            char r=(char) iterator1.next();
            if(Character.isAlphabetic(r)){
                letter.push(r);
            }
        }
        while (preanswer.size()!=6) {
            char s =(char) letter.pop();
            switch (s) { // на букву срабатывает кейс с такой же буквой
                case ('O'): //Производится логическая операция над числами из мас
сива чисел
                    boolean num1 = (boolean) num.pop(); //Числа вынимаются по оче
реди с конца в обратном порядке
                    boolean num2 = (boolean) num.pop();
                    preanswer.add(num1 || num2); //и записывается true или в fals
e в preanswer
                    break;
                case ('X'):
```

```

        boolean num3 = (boolean) num.pop();
        boolean num4 = (boolean) num.pop();
        if (num3 != num4) {
            preanswer.add(true);
        } else {
            preanswer.add(false);
        }
        break;
    case ('A'):
        boolean num5 = (boolean) num.pop();
        boolean num6 = (boolean) num.pop();
        preanswer.add(num5 && num6);
        break;
    case ('N'):
        boolean num7 = (boolean) num.pop();
        if (num7 == true) {
            preanswer.add(false);
        } else {
            preanswer.add(true);
        }
        break;
    case ('F'):
        preanswer.add(false);
        break;
    case ('T'):
        preanswer.add(true);
        break;
    }

}

Iterator iterator2 = preanswer.iterator();
boolean answer=false;
while (iterator2.hasNext()) {
    if ((boolean)iterator2.next() == true){
        //Если находит хоть 1 true то все равно true
        answer = true;
    }
}

System.out.println("Лв = " + answer); //Вывод функции
}
}

```

2.10 Листинг программы Tenth.

```

package Lab4;
import java.io.*;
import java.util.Iterator;
import java.util.Stack;

```

```

//То же самое что и в 9том только используем другие буквы в case и действия други
e
public class Tenth {
    public static void main(String[] args) throws IOException {
        BufferedReader reader = new BufferedReader(new FileReader(new File("D:\\V
SCode\\Projects\\Lab4\\files\\Tenth.txt")));
        Stack st = new Stack(); //Основной стек, для записи из файла
        Stack num = new Stack(); // Стек чисел
        Stack letter = new Stack(); //Стек букв
        Stack answer = new Stack(); // Стек ответов
        int i = 0;
        while ((i = reader.read()) != -1) {
            char ch = (char) i;
            st.add(ch);
        }
        Iterator iterator = st.iterator();
        while (iterator.hasNext()) {
            char r=(char) iterator.next();
            if(Character.isDigit(r)){
                num.add(r);
            }
        }
        Iterator iterator2 = st.iterator();
        while (iterator2.hasNext()) {
            char r=(char) iterator2.next();
            if(Character.isAlphabetic(r)){
                letter.push(r);
            }
        }
        while (answer.size() != 1){
            char s =(char) letter.pop();
            switch (s){ // на букву срабатывает кейс с такой же буквой
                case ('M'): //Производятся логические операции над числами
                    char num1 =(char) num.pop(); //Производится логическая опера
ция над числами из массива чисел
                    char num2 =(char) num.pop();
                    if (num1 > num2)
                        answer.add(num1); //записывается в стек answer
                    else
                        answer.add(num2);
                    break;
                case ('N'):
                    char num3 = (char) num.pop();
                    char num4 = (char) num.pop();
                    if (num4 > num3)
                        answer.add(num3);
                    else
                        answer.add(num4);
                    break;
                case ('F'):
                    char num6 = (char) num.pop();

```

```

        answer.add(num6);
    }
}
System.out.println(answer);
}
}

```

2.11 Листинг программы Eleventh.

```

package Lab4;
import java.io.*;
import java.util.Iterator;
import java.util.Stack;
//Проверка формулы на правильность
public class Eleventh {
    public static void main(String[] args) throws IOException {
        BufferedReader reader = new BufferedReader(new FileReader(new File("D:\\V
SCode\\Projects\\Lab4\\files\\Eleventh.txt")));
        Stack st = new Stack();
        Stack letter = new Stack();
        Stack symbols = new Stack();
        int i = 0;
        while ((i = reader.read()) != -1) {
            char ch = (char) i;
            st.add(ch);
        }
        Iterator iterator1 = st.iterator();
        while (iterator1.hasNext()) {
            char r = (char) iterator1.next();
            if (Character.isAlphabetic(r)) {
                letter.push(r);
            }
        }
        Iterator iterator2 = st.iterator();
        while (iterator2.hasNext()) {
            char r = (char) iterator2.next();
            if (!(Character.isDigit(r) || Character.isAlphabetic(r))) {
                symbols.push(r);
            }
        }
        int kol = 0; //Записываем количество + или -
        int open = 0, close = 0; //Скобки
        while (symbols.size() != 0) {
            char s = (char) symbols.pop();
            switch (s) { //сравниваем
                case '+':
                case '-':
                    kol++;
                    break;
                case '(':

```

```

        open++;
        break;
    case (')'):
        close++;
        break;
    }
}
int w = 0; //записываем кол во букв
while (letter.size() != 0) {
    char s = (char) letter.pop();
    switch (s) { //сравниваем
        case ('x'):
        case ('y'):
        case ('z'):
            w++;

            break;
    }
}

if ((w-1 == kol) && (close==open))
    //Логика в том , что букв на 1 больше чем знаков ,
    // по этому вычитаем 1 их букв и сравниваем , потом сравниваем со скопбк
ми
    System.out.println("Формула имеет правильный вид");
else
    System.out.println("Формула имеет не правильный вид");
}
}

```

3.Вывод

Выполнив данную лабораторную работу, мы научились работать с такой структурой данной, как стеки деки и очереди, применять их в поставленных задачах, а именно, реализовывать определенные логические , которые можно реализовывать записывать данные в вышеупомянутые структуры данных.

Список использованных источников

1 ГОСТ 7.32-2017 Система стандартов по информации, библиотечному и издательскому делу. Отчет о научно-исследовательской работе. Структура и правила оформления.

2 ГОСТ 7.1-2003 Библиографическая запись. Библиографическое описание. Общие требования и правила составления.