

Министерство цифрового развития, связи и массовых коммуникаций
Ордена Трудового Красного Знамени федеральное государственное
бюджетное

образовательное учреждение высшего образования
«Московский технический университет связи и информатики»

Кафедра Математической кибернетики и информационных технологий

Отчёт по лабораторной работе № 4

«Реализация стека/дека»

по дисциплине «Структуры и алгоритмы обработки данных»

Выполнил: студент группы

БВТ1902

Мартынов Николай Владимирович

Москва

2021

Оглавление

Введение	3
Листинг программы.....	6
Вывод.....	22

Введение

Цель данной лабораторной – получить знания и навыки реализовав следующие структуры данных:

1. Стек (stack): операции для стека: инициализация, проверка на пустоту, добавление нового элемента в начало, извлечение элемента из начала;
2. 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.

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 А на стержень С, 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 В.

Реализовать алгоритм, используя три стека вместо стержней А, В, С. 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:

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

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

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

Класс Deque

```
package com.company;

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;
    }
}
```

Класс Stacks

```
package com.company;

public class Stacks {
    private int mSize;
    private int[] stackArray;
    private int top;

    public Stacks(int m) {
        this.mSize = m;
        stackArray = new int[mSize];
        top = -1;
    }
}
```

```

    public void addElement(int element) {
        stackArray[++top] = element;
    }

    public int deleteElement() {
        return stackArray[top--];
    }

    public int readTop() {
        return stackArray[top];
    }

    public boolean isEmpty() {
        return (top == -1);
    }

    public boolean isFull() {
        return (top == mSize - 1);
    }
}

```

Класс DoublyLinkedList

```

package com.company;

public class DoublyLinkedList<T>
{
    public class Dek<T>
    {
        private T t;
        private Dek next;
        private Dek prev;

        public Dek()
        {
        }

        public Dek(T t)
        {
            this.t = t;
        }

        public T getT()
        {
            return t;
        }

        public void setT(T t)
        {
            this.t = t;
        }

        public Dek getNext()
        {
            return next;
        }
    }

    public Dek head;
}

```

```

public Dek tail;

//Добавить в начало
public void addHead(T t)
{
    Dek newEl = new Dek(t);

    if (head == null) {
        newEl.next = null;
        newEl.prev = null;
        head = newEl;
        tail = newEl;
    } else {
        head.prev = newEl;
        newEl.next = head;
        newEl.prev = null;
        head = newEl;
    }
}

//Добавить в конец
public void addTail(T t)
{
    Dek newEl = new Dek(t);

    if (head == null) {
        newEl.next = null;
        newEl.prev = null;
        head = newEl;
        tail = newEl;
    } else {
        tail.next = newEl;
        newEl.next = null;
        newEl.prev = tail;
        tail = newEl;
    }
}

//Удалить с начала
public void removeHead()
{
    if (head.next == null) {
        head = null;
        tail = null;
    } else {
        head = head.next;
        head.prev = null;
    }
}

//Удалить с конца
public void removeTail()
{
    if (tail.prev == null) {
        head = null;
        tail = null;
    } else {
        tail = tail.prev;
        tail.next = null;
    }
}

```



```

    }

    /* public int compareTo(Dek other)
    {
        return head.toString().compareTo(other.toString());
    }*/

    @Override
    public String toString()
    {
        String stringBuffer = "Дек {\n";
        Dek dek = head;
        while (dek != null)
        {
            stringBuffer = stringBuffer + dek.getT().toString();
            stringBuffer+="(",\n");
            dek = dek.getNext();
        }
        stringBuffer+=(")");
        return stringBuffer;
    }

    public String toChar()
    {
        String stringBuffer = "";
        Dek dek = head;
        while (dek != null)
        {
            stringBuffer = stringBuffer + dek.getT().toString();
            dek = dek.getNext();
        }
        return stringBuffer;
    }

}

```

Класс Lab4

```

package com.company;
import java.io.*;
import java.util.*;

public class Lab4 {

    public static void write(String fileName, String text) {
        File file = new File(fileName);
        try {
            if(!file.exists()){
                file.createNewFile();
            }
            PrintWriter out = new PrintWriter(file.getAbsolutePath());

            try {
                out.print(text);
            } finally {

```

```

        out.close();
    }
} catch(IOException e) {
    throw new RuntimeException(e);
}
}

public static void main (String[]args) {
    DoublyLinkedList books = new DoublyLinkedList();
    List<String> list = new ArrayList();
    try {
        File myfile = new File("/Users/k18/Desktop/books.txt");
        BufferedReader reader = new BufferedReader(new
FileReader(myfile));
        String line=null;
        while ((line=reader.readLine())!=null)
        {
            books.addHead(line);
            list.add(line);
        }

    }
    catch (Exception ex)
    {
        System.out.println(ex.getMessage());
    }
    System.out.println(books);
    Collections.sort(list);

    String fileName = "/Users/k18/Desktop/books_out.txt";

    DoublyLinkedList books2 = new DoublyLinkedList();
    for (int i = 0; i < list.size(); i++) {
        books2.addTail(list.get(i));
    }
    System.out.println(books2);
    write(fileName,books2.toString());
}

}

```

Класс Lab4_2

```

package com.company;
import java.io.*;
public class Lab4_2 {
    private static DoublyLinkedList alf = new DoublyLinkedList();
    public static void main(String[] args) throws IOException {
        for(char i = '\u0430'; i <= '\u044f'; i++){
            alf.addTail(i);
        }

        System.out.println(alf);

        BufferedReader reader = new BufferedReader(new FileReader(new
File("/Users/k18/Desktop/message.txt")));
        BufferedWriter writer = new BufferedWriter(new FileWriter(new
File("/Users/k18/Desktop/outMessage.txt")));

```

```

        int i = 0;
        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';
        char[] cc = alf.toChar().toCharArray();
        for(int i = 2; i < cc.length; i++) {
            char c = cc[i] ;
            if(Character.compare(c, ch) == 0) {
                outchar = cc[i-2];
                break;
            }
        }

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

        return outchar;
    }
}

```

Класс Lab4_3

```

package com.company;
import java.util.*;
public class Lab4_3 {

    public static int N;
    public static Stack<Integer>[] tower = new Stack[4];
    public static void main(String[] args)
    {
        Scanner scan = new Scanner(System.in);
        tower[1] = new Stack<Integer>();
        tower[2] = new Stack<Integer>();
        tower[3] = new Stack<Integer>();
        System.out.print("Дисков n=");
        int num = scan.nextInt();
        N = num;
        toh(num);
    }
    public static void toh(int n)
    {
        for (int d = n; d > 0; d--)
            tower[1].push(d);
        display();
        move(n, 1, 2, 3);
    }
    public static void move(int n, int a, int b, int c)
    {
        if (n > 0)
        {
            move(n-1, a, c, b);

```

```

        int d = tower[a].pop();
        tower[c].push(d);
        display();
        move(n-1, b, a, c);
    }
}

public static void display()
{
    for(int i = N - 1; i >= 0; i--)
    {
        String d1 = " ", d2 = " ", d3 = " ";
        try
        {
            d1 = String.valueOf(tower[1].get(i));
        }
        catch (Exception e){
        }
        try
        {
            d2 = String.valueOf(tower[2].get(i));
        }
        catch(Exception e){
        }
        try
        {
            d3 = String.valueOf(tower[3].get(i));
        }
        catch (Exception e){
        }
        System.out.println(" "+d1+"    "+d2+"    "+d3);
    }
    System.out.println(" A    B    C");
    System.out.println("\n");
}

public static void moveRing(char a, char b, char c, int count) {

    if (count==1){
        System.out.println(String.format("Из %s в %s",a,b));
    }else {
        moveRing(a, c, b, count - 1);
        System.out.println(String.format("Из %s в %s",a,b));
        moveRing(c,b,a,count-1);
    }

}

static void towerOfHanoi(int n, char from_rod, char to_rod, char aux_rod)

{
    if (n == 1)
    {
        System.out.println("Сдвинуть диск 1 со стержня " + from_rod + "
на стержень " + to_rod);
        return;
    }
    towerOfHanoi(n-1, from_rod, aux_rod, to_rod);
    System.out.println("Сдвинуть диск " + n + " со стержня " + from_rod
+ " на стержень " + to_rod);
    towerOfHanoi(n-1, aux_rod, to_rod, from_rod);
}

```

```

    }
}

```

Класс Lab4_4

```

package com.company;
import java.io.BufferedReader;
import java.io.File;
import java.io.FileReader;
import java.util.*;
public class Lab4_4 {
    public static void main(String args[]) {
        try {
            File myfile = new File("/Users/kl8/Desktop/testProg.txt");
            BufferedReader reader = new BufferedReader(new
FileReader(myfile));
            String line=null;
            String progr = "";
            while ((line=reader.readLine())!=null)
            {
                progr = progr+line;
            }
            System.out.println(progr);
            System.out.println(balancedParenthensies(progr));
        }
        catch (Exception ex)
        {
            System.out.println(ex.getMessage());
        }

        /*
        System.out.println(balancedParenthensies("{ (a,b) }"));
        System.out.println(balancedParenthensies("{ (a},b) }"));
        System.out.println(balancedParenthensies("{} (a,b) }"));*/
    }

    public static boolean balancedParenthensies(String s) {
        Stack<Character> stack = new Stack<Character>();
        for(int i = 0; i < s.length(); i++) {
            char c = s.charAt(i);
            if(c == '[' || c == '(' || c == '{' ) {
                stack.push(c);
            } else if(c == ']') {
                if(stack.isEmpty() || stack.pop() != '[') {
                    return false;
                }
            } else if(c == ')') {
                if(stack.isEmpty() || stack.pop() != '(') {
                    return false;
                }
            } else if(c == '}') {
                if(stack.isEmpty() || stack.pop() != '{') {
                    return false;
                }
            }
        }
        return stack.isEmpty();
    }
}

```

Класс Lab4_5

```
package com.company;
import java.io.BufferedReader;
import java.io.File;
import java.io.FileReader;
import java.util.Deque;
import java.util.LinkedList;
public class Lab4_5 {
    public static void main(String args[]) {
        /* try {
            File myfile = new File("/Users/kl8/Desktop/testProg.txt");
            BufferedReader reader = new BufferedReader(new
FileReader(myfile));
            String line=null;
            String progr = "";
            while ((line=reader.readLine())!=null)
            {
                progr = progr+line;
            }
            System.out.println(progr);
            System.out.println(balancedParenthensies(progr));
        }
        catch (Exception ex)
        {
            System.out.println(ex.getMessage());
        }*/

        System.out.println(isBalanced("[[][]]"));
        System.out.println(isBalanced("[[]]"));
        System.out.println(isBalanced("[[] [")");

    }
    public static boolean isBalanced(String str) {
        if (null == str || ((str.length() % 2) != 0)) {
            return false;
        } else {
            char[] ch = str.toCharArray();
            for (char c : ch) {
                if (!(c == '{' || c == '[' || c == '(' || c == '}' || c ==
']' || c == ')')) {
                    return false;
                }
            }
        }
    }

    Deque<Character> deque = new LinkedList<>();
    for (char ch : str.toCharArray()) {
        if (ch == '{' || ch == '[' || ch == '(') {
            deque.addFirst(ch);
        } else {
            if (!deque.isEmpty() && ((deque.peekFirst() == '{' && ch ==
'}') || (deque.peekFirst() == '[' && ch == ']') || (deque.peekFirst() == '('
&& ch == ')')))) {
                deque.removeFirst();
            } else {
                return false;
            }
        }
    }
}
```

```

    }

    return true;
}
}

```

Класс Lab4_6

```

package com.company;
import java.io.*;
import java.util.Stack;

public class Lab4_6 {

    public static void main(String args[]) {
        Stack<String> strOut = new Stack<String>();
        try {
            File myfile = new File("/Users/kl8/Desktop/6.txt");
            BufferedReader reader = new BufferedReader(new
FileReader(myfile));
            String line=null;
            String progr = "";
            while ((line=reader.readLine())!=null)
            {
                progr = progr+line;
            }
            System.out.println(progr);

            String[] parts = progr.split("[^0-9]");
            String[] parts2 = progr.split("[^a-zA-z]");
            String[] parts3 = progr.split("[^&\\/\\\\\\#,#,+()$~%. ' \":*?<>{}])");

            for (int i = 0; i < parts.length; i++) {
                if (!parts[i].equals("")) {
                    strOut.addElement(parts[i]);
                }
            }
            for (int i = 0; i < parts2.length; i++) {
                if (!parts2[i].equals("")) {
                    strOut.addElement(parts2[i]);
                }
            }
            for (int i = 0; i < parts3.length; i++) {
                if (!parts3[i].equals("")) {
                    strOut.addElement(parts3[i]);
                }
            }

            System.out.println(strOut);

        }
        catch (Exception ex)
        {
            System.out.println(ex.getMessage());
        }

    }

}

```

Класс Lab4_7

```
package com.company;

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

public class Lab4_7 {

    static void RearrangePosNeg(int arr[], int n)
    {
        int key, j;
        for (int i = 1; i < n; i++) {
            key = arr[i];
            if (key > 0)
                continue;
            j = i - 1;
            while (j >= 0 && arr[j] > 0) {
                arr[j + 1] = arr[j];
                j = j - 1;
            }
            arr[j + 1] = key;
        }
    }

    public static void main(String args[]) {
        DoublyLinkedList strOut = new DoublyLinkedList();
        try {
            File myfile = new File("/Users/k18/Desktop/7.txt");
            BufferedReader reader = new BufferedReader(new
FileReader(myfile));
            String line=null;
            String progr = "";
            while ((line=reader.readLine())!=null)
            {
                progr = progr+line;
            }
            System.out.println(progr);

            String[] parts = progr.split("[^0-9-]+");

            int arr[] = new int[parts.length];
            for (int i = 0; i < parts.length; i++) {
                if (!parts[i].equals("")) {
                    arr[i] = Integer.parseInt(parts[i]);
                }
            }
            RearrangePosNeg(arr,arr.length);
            for (int i = 1; i < arr.length; i++) {
                strOut.addTail(arr[i]);
            }
            System.out.println(strOut);

        }
        catch (Exception ex)
        {
            System.out.println(ex.getMessage());
        }
    }
}
```



```

    }
}

```

Класс Lab4_8

```

package com.company;
import java.io.BufferedReader;
import java.io.File;
import java.io.FileReader;
import java.util.Stack;
public class Lab4_8 {
    public static String reverse(String str)
    {
        if (str == null || str.equals("")) {
            return str;
        }
        Stack<Character> stack = new Stack<Character>();
        char[] ch = str.toCharArray();
        for (int i = 0; i < str.length(); i++) {
            stack.push(ch[i]);
        }
        int k = 0;
        while (!stack.isEmpty())
        {
            ch[k++] = stack.pop();
        }
        return String.copyValueOf(ch);
    }

    public static void main(String[] args)
    {
        try {
            File myfile = new File("/Users/k18/Desktop/textReverse.txt");
            BufferedReader reader = new BufferedReader(new
FileReader(myfile));
            String line=null;
            String progr = "";
            while ((line=reader.readLine())!=null)
            {
                progr = progr+line;
            }
            System.out.println(progr);
            System.out.println(reverse(progr));
        }
        catch (Exception ex)
        {
            System.out.println(ex.getMessage());
        }
    }
}

```

Класс Lab4_9

```

package com.company;
import java.io.BufferedReader;

```

```

import java.io.File;
import java.io.FileReader;
import java.util.*;
public class Lab4_9 {
    public static boolean evaluateBool(String s)
    {
        Stack<Object> stack = new Stack<>();
        StringBuilder expression =new StringBuilder(s);
        expression.chars().forEach(ch->
        {
            if(ch=='F') stack.push(false);
            else if(ch=='T') stack.push(true);
            else if(ch=='A' || ch=='R' || ch=='X')
            {
                boolean op1 = (boolean) stack.pop();
                boolean op2 = (boolean) stack.pop();
                switch(ch)
                {
                    case 'A' : stack.push(op2&&op1); break;
                    case 'O' : stack.push(op2 || op1); break;
                    case 'X' : stack.push(op2^op1); break;
                }//endSwitch
            }else
            if(ch=='N')
            {
                boolean op1 = (boolean) stack.pop();
                stack.push(!op1);
            }//endIF
        });
        return (boolean) stack.pop();
    }

    public static void main(String[] args)
    {
        try {
            File myfile = new File("/Users/k18/Desktop/logicFunc.txt");
            BufferedReader reader = new BufferedReader(new
FileReader(myfile));
            String line=null;
            String progr = "";
            while ((line=reader.readLine())!=null)
            {
                progr += line;
            }
            System.out.println(progr);
            System.out.println(evaluateBool(progr));

            //System.out.println(logicalExpressionEvaluation(progr));

        }
        catch (Exception ex)
        {
            System.out.println(ex.getMessage());
        }
    }

    // [N, [[0,A, [N,1]], O, [N, [[N,0], A,1]]]]
    // (F+(T*F+(F+T)))X(NT)

```

```

//TRUE

//XOR !=
//AND *
//OR +

/*public static void main(String[] args)
{
    String str = "[[0,&,1],|,[!,1]]";
    System.out.println(logicalExpressionEvaluation(str));

}*/
}

```

Класс Lab4_10

```

package com.company;

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

public class Lab4_10 {
    public static int i=0;

    public static int M(int a, int b){
        return Integer.max(a,b);
    }

    public static int N(int a, int b){
        return Integer.min(a,b);
    }

    public static int Numb(String str){
        int x = 0;
        while (true)
        {
            if (str.charAt(i)>='0'&& str.charAt(i)<='9')
            {
                x=(x*10)+str.charAt(i)-'0';
                ++i;
            }
            else
                return x;
        }
    }

    public static Stack<Integer> out = new Stack<>();
    public static int Cacl(String str)
    {
        int x,y;
        if(str.charAt(i)=='N'&&str.charAt(i+1)=='(')
        {
            i+=2;
            x=Cacl(str);
            ++i;
            y=Cacl(str);
            ++i;

```

```

        return out.push(N(x,y));
        // return N(x,y);
    }
    else if(str.charAt(i)=='M'&&str.charAt(i+1)=='(')
    {
        i+=2;
        x=Cacl(str);
        ++i;
        y=Cacl(str);
        ++i;
        return out.push(M(x,y));
        //return M(x,y);
    }
    else return out.push(Numb(str));
    //else return Numb(str);
}
//M(N(M(3,N(9,N(3,4))),6),1),0)
//M(N(7,4),N(3,6))
public static void main(String[] args)
{
    try {
        File myfile = new File("/Users/k18/Desktop/10.txt");
        BufferedReader reader = new BufferedReader(new
FileReader(myfile));
        String line=null;
        String progr = "";
        while ((line=reader.readLine())!=null)
        {
            progr += line;
        }
        System.out.println(progr);
        Cacl(progr);
        System.out.println(out.peek());
        // System.out.println(Cacl(progr));
    }
    catch (Exception ex)
    {
        System.out.println(ex.getMessage());
    }
}
}

```

Класс Lab4_11

```

package com.company;

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

public class Lab4_11 {

    public static void main(String[] args){
        try {
            File myfile = new File("/Users/k18/Desktop/11.txt");
            BufferedReader reader = new BufferedReader(new
FileReader(myfile));
            String line = null;
            String form = "";
            while ((line = reader.readLine()) != null) {
                form += line;
            }
        }
    }
}

```

```

    }
    form = form.replaceAll("\\s+", "");
    System.out.println(form);
    Boolean str = Lab4_4.balancedParenthensies(form);
    if (str == true) {
        Stack<Character> out = new Stack<>();
        for (int i = 0; i < form.length(); i++) {
            // if ((form.charAt(i) == 'x') || (form.charAt(i) ==
'(') || (form.charAt(i) == 'y') || (form.charAt(i) == '{') || (form.charAt(i)
== 'z') || (form.charAt(i) == '[') || (form.charAt(i) == '+') ||
(form.charAt(i) == '-')) {

                if(((form.charAt(i) == 'x') || (form.charAt(i) == 'y')
|| (form.charAt(i) == 'z'))&&((form.charAt(i+1) == '(') || (form.charAt(i+1)
== '[') || (form.charAt(i+1) == '{'))
                {
                    out.push(form.charAt(i));
                }
                if ((form.charAt(i) == ')') || (form.charAt(i) == ']')
|| (form.charAt(i) == '}')) {
                    out.push(form.charAt(i));
                }
                if (form.charAt(i)==form.charAt(i+1))
                {
                    break;
                }
                else {break;}
            // if(((form.charAt(i) == 'x') || (form.charAt(i) ==
'y') || (form.charAt(i) == 'z'))&&((form.charAt(i+1) == '(') ||
(form.charAt(i+1) == '[') || (form.charAt(i+1) == '{'))
            /* if (form.charAt(i)
            {

            }*/
        }
        if (out.size() == form.length()) System.out.println("Запись
правильная!");
        else System.out.println("Запись неверна!");

        /* System.out.println(out.size());
        System.out.println(form.length());
        System.out.println(out);*/

    } else {System.out.println("Запись неверна!");}
}
catch(Exception ex)
{
    System.out.println(ex.getMessage());
}

}

}

```

Вывод

В результате выполненной лабораторной работы я реализовал две структуры данных Стек и Дек и использовал эти структуры для решения одиннадцати задач на java.