

Министерство цифрового развития, связи и массовых коммуникаций
Российской Федерации

Бюджетное образовательное учреждение высшего образования
Московский технический университет связи и информатики

Кафедра «Математической кибернетики и информационных технологий»

Дисциплина «Структуры и алгоритмы обработки данных»

Отчёт по лабораторной работе №4

«Реализация стека/дека»

Выполнил студент
Группы БФИ1901
Курбатов А.О.

Проверил
Кутейников И. А.

Москва 2021

Задание

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:

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

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

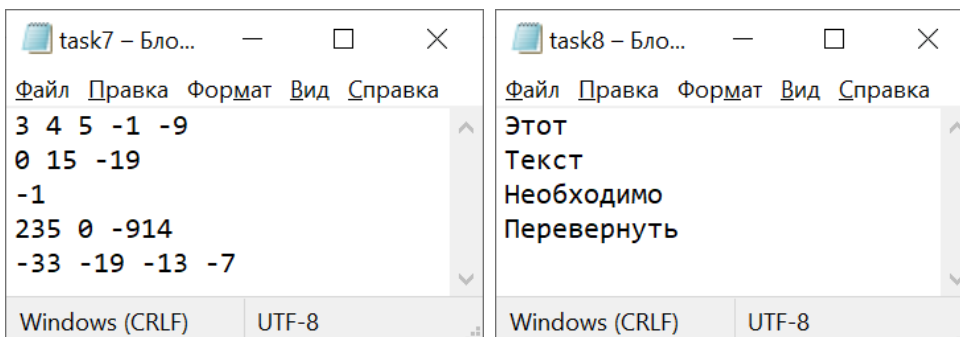
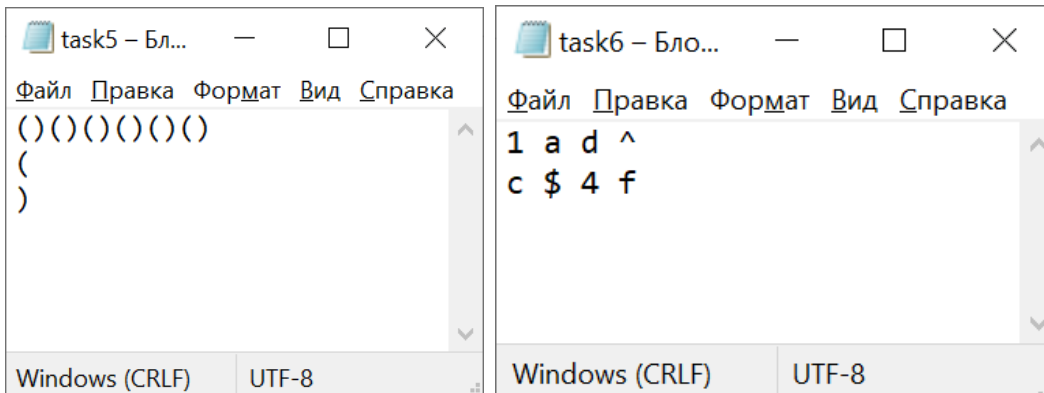
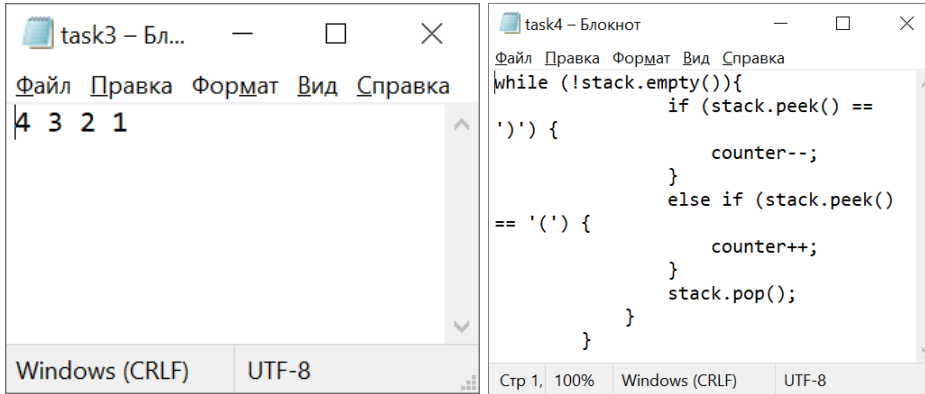
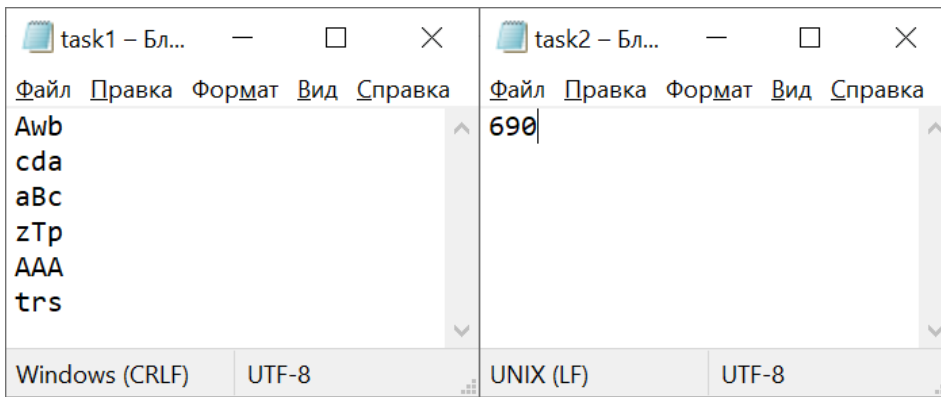
$$\langle \text{Цифра} \rangle ::= 0 \mid 1 \mid 2 \mid 3 \mid 4 \mid 5 \mid 6 \mid 7 \mid 8 \mid 9$$
где буквами обозначены функции:
M – определение максимума, **N** – определение минимума.
Используя **стек**, вычислить значение заданного выражения.
11. Дан текстовый файл. Используя **стек**, проверить, является ли содержимое текстового файла правильной записью формулы вида:

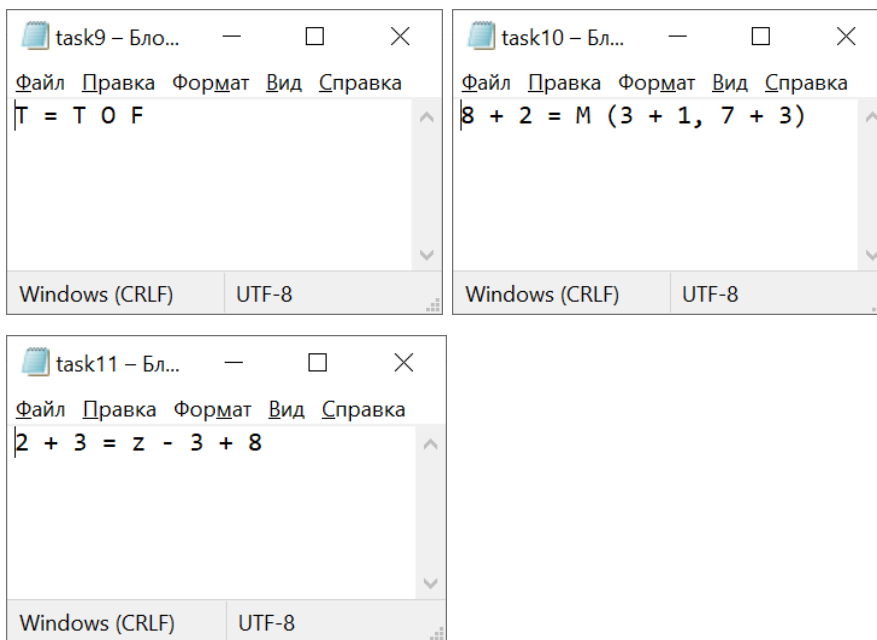
$$\langle \text{Формула} \rangle ::= \langle \text{Терм} \rangle \mid \langle \text{Терм} \rangle + \langle \text{Формула} \rangle \mid \langle \text{Терм} \rangle - \langle \text{Формула} \rangle$$

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

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

Документы





Листинг

```
import java.io.*;
import java.util.*;

public class Lab4 {

    public static void task1() throws Exception {
        FileReader fr = new FileReader("task1.txt");
        Scanner scan = new Scanner(fr);
        Deque<String> sorted = new ArrayDeque<>();
        Deque<String> alph = new ArrayDeque<>();

        List<String> books = new ArrayList<>();
        while (scan.hasNextLine()) {
            books.add(scan.nextLine().toLowerCase());
        }

        String [] mass = new String[books.size()];
        for(int i = 0; i < mass.length; i++){
            mass[i] = books.get(i);
        }

        int len = mass.length;

        if (len == 0){
            System.out.println(sorted);
        }
        else if(len == 1){
            sorted.push(mass[0]);
            System.out.println(sorted);
        }

        sorted.push(mass[0]);
```

```

        for (int i = 1; i < len; i++){
            while ((sorted.size() > 0) && checkLetters(sorted.getFirst(),
mass[i])) {
                alph.addLast(sorted.removeFirst());
            }
            sorted.addFirst(mass[i]);
            while (!alph.isEmpty()) {
                sorted.addFirst(alph.removeLast());
            }
        }
        fr.close();
        scan.close();

        System.out.println(sorted);
    }

    public static boolean checkLetters(String book, String book2){
        for (int i = 0; i < Math.min(book2.length(), book.length()); i++)
        {
            if (book.charAt(i) != book2.charAt(i))
            {
                return book.charAt(i) <= book2.charAt(i);
            }
        }
        return book.length() <= book2.length();
    }

    public static void task2() throws Exception {
        FileReader fr = new FileReader("task2.txt");
        FileWriter fw = new FileWriter("task2.txt", true);
        Deque<Character> deque = new ArrayDeque<>();
        Scanner scan = new Scanner(fr);

        deque.push('2');
        deque.push('0');
        deque.push('4');
        deque.push('8');
        deque.push('6');
        deque.push('1');
        deque.push('9');
        deque.push('5');
        deque.push('7');
        deque.push('3');

        String str = scan.nextLine(); //690
        fw.write("\n");

        for(int i = 0; i < str.length();){
            while (deque.getLast() != str.charAt(i)){
                deque.push(deque.pollLast());
            }
            deque.push(deque.pollLast());
            deque.push(deque.pollLast());
            fw.write(deque.getLast());
            i++;
        }
        scan.close();
    }

```

```

        fr.close();
        fw.close();

    }

    public static void task3(int n, Stack<Integer> a, Stack<Integer> b,
Stack<Integer> c, String left, String mid, String right)
    {
        if (n == 1)
        {
            b.push(a.pop());
            System.out.println("Переносим диск 1 с " + left + " на " + mid);
            return;
        }
        task3(n - 1, a, c, b, left, right, mid);
        b.push(a.pop());
        System.out.println("Переносим диск " + n + " с " + left + " на " + mid);
        task3(n - 1, c, b, a, right, mid, left);
    }

    public static boolean task4() throws Exception {
        FileReader fr = new FileReader("task4.txt");
        Stack<Character> stack = new Stack<>();
        Scanner scan = new Scanner(fr);

        int counter = 0;

        while (scan.hasNext()) {
            String str = scan.nextLine();

            for (int i = 0; i < str.length(); i++){
                stack.push(str.charAt(i));
            }

            while (!stack.empty()){
                if (stack.peek() == ')') {
                    counter--;
                }
                else if (stack.peek() == '(') {
                    counter++;
                }
                stack.pop();
            }
        }

        fr.close();

        return counter==0;
    }

    public static boolean task5() throws Exception {
        FileReader fr = new FileReader("task5.txt");
        Deque<Character> deque = new ArrayDeque<>();
        Scanner scan = new Scanner(fr);

        int counter = 0;

        while (scan.hasNext()) {
            String str = scan.nextLine();

```

```

        deque.push(str.charAt(i));
    }

    while(!deque.isEmpty()){
        if (deque.peek() == ')') {
            counter--;
        }
        else if (deque.peek() == '(') {
            counter++;
        }
        deque.pop();
    }
}

fr.close();

return counter==0;
}

public static void task6() throws Exception {
    FileReader fr = new FileReader("task6.txt");
    Stack<Character> stack = new Stack<>();
    Stack<Character> stack2 = new Stack<>();
    StringBuilder sb = new StringBuilder();
    Scanner scan = new Scanner(fr);

    String str = "";

    while (scan.hasNext()) {
        str += scan.nextLine();
    }
    str = str.replaceAll(" ", "");

    for (int i = 0; i < str.length(); i++) {
        if (str.charAt(i) >= 48 && str.charAt(i) <= 57)
        {
            sb.append(str.charAt(i));
        }
        else
        {
            if ((str.charAt(i) >= 65 && str.charAt(i) <= 90) ||
(str.charAt(i) >= 97 && str.charAt(i) <= 122))
            {
                stack.push(str.charAt(i));
            }
            else
            {
                stack2.push(str.charAt(i));
            }
        }
    }

    int index = sb.length();
    while (!stack.isEmpty())
    {
        sb.insert(index, stack.pop());
    }
}

```



```

        index = sb.length();
        while (!stack2.isEmpty())
        {
            sb.insert(index, stack2.pop());
        }
        fr.close();
        System.out.println(sb.toString());
    }

    public static void task7() throws Exception {
        FileReader fr = new FileReader("task7.txt");
        Deque<String> deque = new ArrayDeque<>();
        Scanner scan = new Scanner(fr);

        String str = "";

        while (scan.hasNext()) {
            str += scan.nextLine();
        }

        String[] nums = str.split(" ");

        for(int i = nums.length - 1; i >= 0; i--){
            if(Integer.parseInt(nums[i]) < 0){
                deque.push(nums[i]);
                nums[i] = null;
            }
        }

        for(int i = 0; i < nums.length; i++){
            if(nums[i] != null){
                deque.addLast(nums[i]);
            }
        }

        System.out.println(deque);

        fr.close();
    }

    public static void task8() throws Exception {
        FileReader fr = new FileReader("task8.txt");
        FileWriter fw = new FileWriter("task8_reversed.txt");
        Stack<String> stack = new Stack<>();
        Scanner scan = new Scanner(fr);

        while (scan.hasNext()) {
            stack.push(scan.nextLine());
        }
        while(!stack.empty()){
            fw.write(stack.pop() + "\n");
        }
        scan.close();
        fr.close();
        fw.close();
    }
}

```

```

public static boolean task9() throws Exception {
    FileReader fr = new FileReader("task9.txt");
    Stack<String> stack = new Stack<>();
    Scanner scan = new Scanner(fr);
    String str = scan.nextLine();

    int index = str.indexOf("=");
    String left = str.substring(0, index).trim();
    stack.push(left);
    String[] right = str.substring(index + 1).trim().split(" ");

    int result = 0;
    int first = 0;
    int second = 0;

    if(right.length == 3){
        if(right[0].equals("T")) {
            first = 1;
        }
        else if(right[0].equals("F")){
            first= 0;
        }
        if(right[2].equals("T")) {
            second = 1;
        }
        else if(right[2].equals("F")){
            second = 0;
        }

        switch (right[1]){
            case "A":
                result = first & second;
                break;
            case "X":
                result = first ^ second;
                break;
            case "O":
                result = first | second;
                break;
        }
    }
    else if(right.length == 2){
        if(right[0].equals("N") && right[1].equals("T")){
            result = 0;
        }
        else if(right[0].equals("N") && right[1].equals("F")){
            result = 1;
        }
    }
    else if(right.length == 1){
        if(right[0].equals("T")){
            result = 1;
        }
        else if(right[0].equals("F")){
            result = 0;
        }
    }
}

```

```

        if(result == 1){
            stack.push("T");
        }
        else if(result == 0){
            stack.push("F");
        }

        fr.close();
        scan.close();

        return stack.pop().equals(stack.pop());
    }

    public static boolean task10() throws Exception {
        FileReader fr = new FileReader("task10.txt");
        Stack<Integer> stack = new Stack<>();
        Scanner scan = new Scanner(fr);
        String str = scan.nextLine();

        int index = str.indexOf("=");
        String[] left = str.substring(0, index).trim().split(" ");
        stack.push(claculate(left));
        if(str.charAt(index + 2) == 'M' || str.charAt(index + 2) == 'N'){
            String right = str.substring(index+5, str.length() - 1);
            int ind = right.indexOf(",");
            String[] first = right.substring(0, ind).split(" ");
            String[] second = right.substring(ind + 2).split(" ");
            if(str.charAt(index + 2) == 'M') {
                stack.push(Math.max(claculate(first), claculate(second)));
            }
            if(str.charAt(index + 2) == 'N') {
                stack.push(Math.min(claculate(first), claculate(second)));
            }
        }
        else {
            String right = str.substring(index + 1).trim();
            stack.push(Integer.parseInt(right));
        }
        int r = stack.pop();
        int l = stack.pop();

        fr.close();
        scan.close();
        return r==l;
    }

    public static int claculate(String[] str){

        if (str.length == 1) {
            return Integer.parseInt(str[0]);
        }

        if (str.length != 3) {
            return 0;
        }
        switch (str[1]) {
            case "+":
                return Integer.parseInt((str[0])) + Integer.parseInt(str[2]);

```

```

        case "-":
            return Integer.parseInt(str[0]) - Integer.parseInt(str[2]);
        case "/":
            return Integer.parseInt(str[0]) / Integer.parseInt(str[2]);
        case "*":
            return Integer.parseInt(str[0]) * Integer.parseInt(str[2]);
        default:
            return 0;
    }
}

public static boolean task11() throws Exception {
    FileReader fr = new FileReader("task11.txt");
    Stack<Boolean> stack = new Stack<>();
    Scanner scan = new Scanner(fr);
    String str = scan.nextLine();

    int index = str.indexOf("=");
    String[] left = str.substring(0, index).trim().split(" ");
    String[] right = str.substring(index + 1).trim().split(" ");
    stack.push(check(left));
    stack.push(check(right));

    fr.close();
    scan.close();

    return stack.pop() == stack.pop();
}

public static boolean check(String[] str){
    for(int i = 0; i < str.length; i++){
        if (i % 2 == 0){
            if (!str[i].matches("[x-zX-Z[\\d]]")){
                return false;
            }
        }
        if (i % 2 != 0){
            if(!str[i].equals("*") && !str[i].equals("/") &&
!str[i].equals("-") && !str[i].equals("+")){
                return false;
            }
        }
    }
    return true;
}

public static void main(String[] args) throws Exception {
    System.out.println("Задание 1: ");
    task1();
    System.out.println("Задание 2: \nРезультат в файле");
    task2();

    Stack<Integer> a = new Stack<>();
    Stack<Integer> b = new Stack<>();
    Stack<Integer> c = new Stack<>();
    String left = "A";

```

```

String mid = "B";
String right = "C";

FileReader fr = new FileReader("task3.txt");
Scanner scan = new Scanner(fr);
String[] tower = scan.nextLine().split(" ");
for(int i =0; i<tower.length;i++){
    a.push(Integer.parseInt(tower[i]));
}

int n = tower.length;
System.out.println("Задание 3: ");
task3(n, a, b, c, left, mid, right);

System.out.println("Задание 4:");
if(task4()){
    System.out.println("Количество скобок одинаковое");
}
else {
    System.out.println("Количество скобок различное");
}

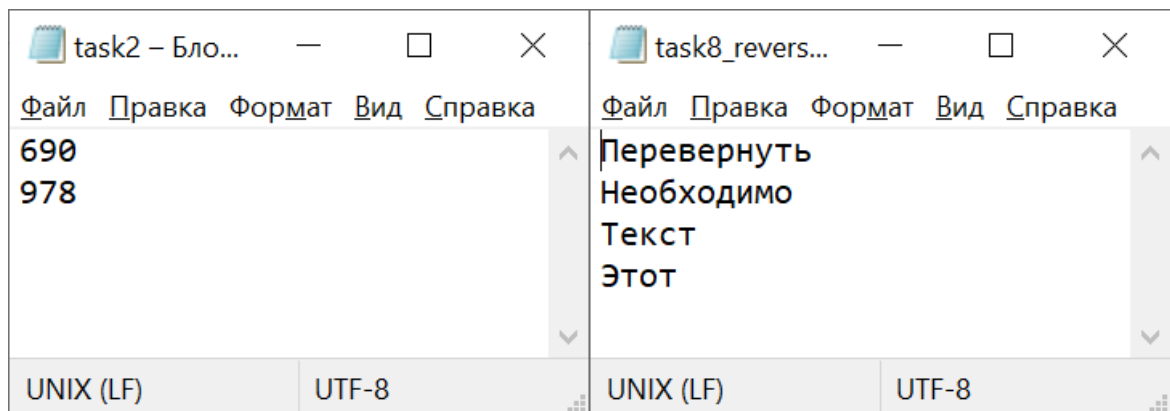
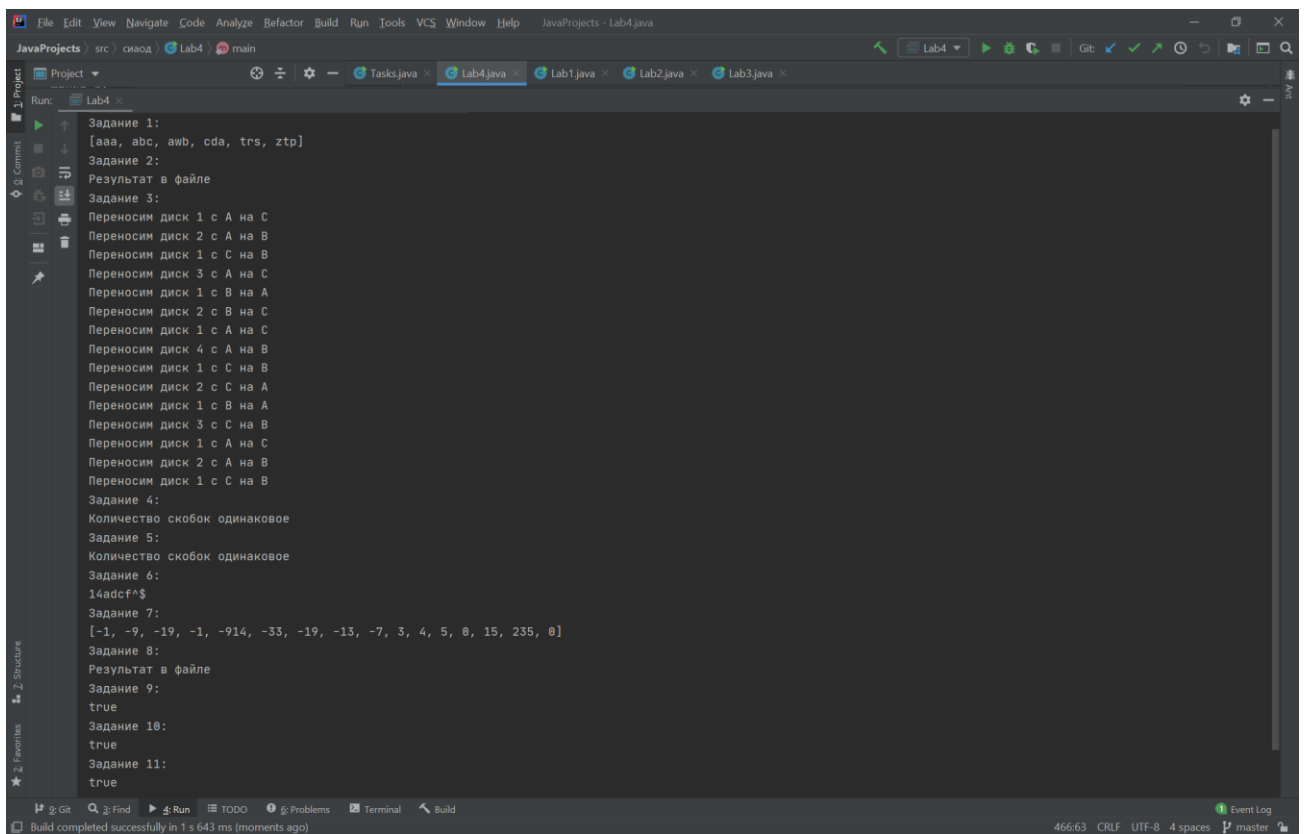
System.out.println("Задание 5:");
if(task5()){
    System.out.println("Количество скобок одинаковое");
}
else {
    System.out.println("Количество скобок различное");
}

System.out.println("Задание 6:");
task6();
System.out.println("Задание 7:");
task7();
System.out.println("Задание 8:\nРезультат в файле");
task8();
System.out.println("Задание 9:");
System.out.println(task9());
System.out.println("Задание 10:");
System.out.println(task10());
System.out.println("Задание 11:");
System.out.println(task11());

fr.close();
scan.close();
}
}

```

Результат выполнения работы



Вывод

В ходе данной лабораторной работы мы познакомились со стеком и деком, а также реализовали их на языке программирования java.