Министерство цифрового развития, связи и массовых коммуникаций Государственное образовательного учреждение высшего образования

Ордена Трудового Красного Знамени

«Московский технический университет связи и информатики»

Лабораторная работа № 4

«Реализация стека/дека»

Выполнил студент

группы БВТ1902

Шульпина Полина

Москва

2021

**Задание**

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.

Оформить отчет о лабораторной работе в ipynb или pdf-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 *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.

* 1. Дан текстовый файл с программой на алгоритмическом языке. За один просмотр файла проверить баланс круглых скобок в тексте, используя *стек*.
  2. Дан текстовый файл с программой на алгоритмическом языке. За один просмотр файла проверить баланс квадратных скобок в тексте, используя *дек*.

1. 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.
2. 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.
3. 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.
4. Дан текстовый файл. Используя *стек*, вычислить значение логического выражения, записанного в текстовом файле в следующей форме:  
   < ЛВ > ::= T | F | (N<ЛВ>) | (<ЛВ>A<ЛВ>) | (<ЛВ>X<ЛВ>) | (<ЛВ>O<ЛВ>),  
   где буквами обозначены логические константы и операции:

T – True, F – False, N – Not, A – And, X – Xor, O – Or.

1. 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

# Выполнение

Листинг программы:

Класс «Main»:

package com.company;  
  
  
import javax.script.ScriptException;  
import java.lang.reflect.Array;  
import java.util.Collection;  
import java.util.Deque;  
import java.util.Iterator;  
import java.util.LinkedList;  
  
public class Main {  
  
 public static void main(String[] args) throws ScriptException {  
 //task1  
 String str1="A C B D D D SD ";  
 new task1(str1);  
 System.*out*.println();  
  
 //task2  
 String str = "152342356789(";  
 Deque dec = new LinkedList();  
 dec.addLast('1');  
 dec.addLast('2');  
 dec.addLast('я');  
 dec.addLast('л');  
 dec.addLast('3');  
 dec.addLast('4');  
 dec.addLast('ю');  
 dec.addLast('б');  
 dec.addLast('5');  
 dec.addLast('6');  
 dec.addLast(' ');  
 dec.addLast('с');  
 dec.addLast('7');  
 dec.addLast('8');  
 dec.addLast('и');  
 dec.addLast('а');  
 dec.addLast('9');  
 dec.addLast('(');  
 dec.addLast('о');  
 dec.addLast('д');  
  
 new task2(dec, str);  
 //task3  
 System.*out*.print("task 3: ");  
 new task3(4, 'A','B','C');  
 //task4  
 String str4 = "((((1151()))()))";  
 new task4(str4);  
 //task 5  
 String str5 = "[]";  
 new task5(str5);  
 //task6  
 String str6 = "111qqqwww111--1q-\_+";  
 new task6(str6);  
 System.*out*.println();  
 //task7  
 int[] arr7 = {1, -15, 16, 71};  
 new task7(arr7);  
 System.*out*.println();  
 //task8  
 String str8 = "15 q1 1521 qwt";  
 new task8(str8);  
 System.*out*.println();  
 //task9  
 String str9="(F+(T\*F+(F+T)))X(NT)";  
 new task9(str9);  
 //task10  
 String str10="N(9,(M(1,N(1,2))))";  
 new task10(str10);  
 //task11  
 String str11="x+(y+z+(z+y)))";  
 new task11(str11);  
 }  
}

Класс «task1»:

package com.company;  
  
import java.util.Deque;  
import java.util.LinkedList;  
  
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;  
 } else {  
 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]);  
 }  
  
 for(String str = " "; !a.isEmpty(); str = "") {  
 int i;  
 for(i = 0; i < a.size(); ++i) {  
 if (*cpv*(str, a.peekFirst().toString())) {  
 str = a.peekFirst().toString();  
 }  
  
 a.addLast(a.removeFirst());  
 }  
  
 for(i = 0; i < a.size(); ++i) {  
 if (*cpv2*(str, a.peekFirst().toString())) {  
 a.removeFirst();  
 break;  
 }  
  
 a.addLast(a.removeFirst());  
 }  
  
 b.addLast(str);  
 }  
  
 System.*out*.print("task 1: ");  
  
 while(!b.isEmpty()) {  
 System.*out*.print(b.removeFirst() + " ");  
 }  
  
 }  
}

Класс «task2»:

package com.company;  
  
import java.util.Deque;  
  
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) {  
 for(; str1.length() < i + 1; dec.add(dec.pop())) {  
 if (*cpv*(Character.*toString*(str.charAt(i)), dec.peekLast().toString())) {  
 dec.add(dec.pop());  
 dec.add(dec.pop());  
 str1 = str1 + dec.peekLast();  
 }  
 }  
 }  
  
 System.*out*.println("task 2: " + str1);  
 }  
}

Класс «task3»:

package com.company;  
  
public class task3 {  
 public task3(int topN, char from, char inter, char to) {  
 if (topN == 1) {  
 System.*out*.println("Диск 1 из " + from + " на " + to);  
 } else {  
 new task3(topN - 1, from, to, inter);  
 System.*out*.println("Диск " + topN + " из " + from + " на " + to);  
 new task3(topN - 1, inter, from, to);  
 }  
  
 }  
}

Класс «task4»:

package com.company;  
  
import java.util.Stack;  
  
public class task4 {  
 public task4(String str) {  
 Stack stc = new Stack();  
 Stack stc1 = new Stack();  
  
 for(int i = 0; i < str.length(); ++i) {  
 if (str.charAt(i) == '(') {  
 stc.push(str.charAt(i));  
 } else if (str.charAt(i) == ')') {  
 stc1.push(str.charAt(i));  
 }  
 }  
  
 if (stc.size() == stc1.size()) {  
 System.*out*.println("task 4: true");  
 } else {  
 System.*out*.println("task 4: false");  
 }  
  
 }  
}

Класс «task5»:

package com.company;  
  
import java.util.Deque;  
import java.util.LinkedList;  
  
public class task5 {  
 public task5(String str) {  
 Deque stc = new LinkedList();  
 Deque stc1 = new LinkedList();  
  
 for(int i = 0; i < str.length(); ++i) {  
 if (str.charAt(i) == '[') {  
 stc.push(str.charAt(i));  
 } else if (str.charAt(i) == ']') {  
 stc1.push(str.charAt(i));  
 }  
 }  
  
 if (stc.size() == stc1.size()) {  
 System.*out*.println("task 5: true");  
 } else {  
 System.*out*.println("task 5: false");  
 }  
  
 }  
}

Класс «task6»:

package com.company;  
  
import java.util.Stack;  
  
public class task6 {  
 public task6(String str) {  
 Stack num = new Stack();  
 Stack sim = new Stack();  
 Stack buk = new Stack();  
 Stack per = new Stack();  
  
 for(int i = 0; i < str.length(); ++i) {  
 if (str.charAt(i) < '9' && '0' < str.charAt(i)) {  
 num.push(str.charAt(i));  
 } else if ((str.charAt(i) >= 'Z' || 'A' >= str.charAt(i)) && (str.charAt(i) >= 'z' || 'a' >= str.charAt(i))) {  
 sim.push(str.charAt(i));  
 } else {  
 buk.push(str.charAt(i));  
 }  
 }  
  
 System.*out*.print("task 6: ");  
  
 while(!num.empty()) {  
 per.push(num.pop());  
 }  
  
 while(!per.empty()) {  
 System.*out*.print(per.pop());  
 }  
  
 while(!buk.empty()) {  
 per.push(buk.pop());  
 }  
  
 while(!per.empty()) {  
 System.*out*.print(per.pop());  
 }  
  
 while(!sim.empty()) {  
 per.push(sim.pop());  
 }  
  
 while(!per.empty()) {  
 System.*out*.print(per.pop());  
 }  
  
 }  
}

Класс «task7»:

package com.company;  
  
import java.util.Deque;  
import java.util.LinkedList;  
  
public class task7 {  
 public task7(int[] arr) {  
 Deque deq = new LinkedList();  
  
 int i;  
 for(i = 0; i < arr.length; ++i) {  
 if (arr[i] < 0) {  
 deq.addLast(arr[i]);  
 } else {  
 deq.addFirst(arr[i]);  
 }  
 }  
  
 System.*out*.print("task 7: ");  
  
 for(i = 0; i < arr.length; ++i) {  
 System.*out*.print(" " + deq.removeLast());  
 }  
  
 }  
}

Класс «task8»:

package com.company;  
  
import java.io.PrintStream;  
import java.util.Stack;  
  
public class task8 {  
 public task8(String str) {  
 String[] words = str.split(" ");  
 Stack stc = new Stack();  
 System.*out*.print("task 8: ");  
  
 for(int i = words.length - 1; i >= 0; --i) {  
 PrintStream var10000 = System.*out*;  
 Object var10001 = stc.push(words[i]);  
 var10000.print(var10001 + " ");  
 }  
  
 }  
}

Класс «task9»:

package com.company;  
  
import java.util.Stack;  
import javax.script.ScriptEngine;  
import javax.script.ScriptEngineManager;  
import javax.script.ScriptException;  
  
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();  
  
 int i;  
 for(i = 0; i < Str.length(); ++i) {  
 stk.add(Str.charAt(i));  
 }  
  
 for(i = 0; i < Str.length(); ++i) {  
 stk2.add(stk.pop());  
 }  
  
 for(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("task 9: " + o);  
 }  
}

Класс «task10»:

package com.company;  
  
import java.util.Stack;  
import javax.script.ScriptEngine;  
import javax.script.ScriptEngineManager;  
import javax.script.ScriptException;  
  
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");  
 StringBuilder str1 = new StringBuilder();  
 Stack stk = new Stack();  
 Stack stk2 = new Stack();  
  
 int i;  
 for(i = 0; i < Str.length(); ++i) {  
 stk.add(Str.charAt(i));  
 }  
  
 for(i = 0; i < Str.length(); ++i) {  
 stk2.add(stk.pop());  
 }  
  
 for(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("task 10: " + o);  
 }  
}

Класс «task11»:

package com.company;  
  
import java.util.Stack;  
import javax.script.ScriptEngine;  
import javax.script.ScriptEngineManager;  
import javax.script.ScriptException;  
  
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");  
 StringBuilder str1 = new StringBuilder();  
 Stack stk = new Stack();  
 Stack stk2 = new Stack();  
  
 int i;  
 for(i = 0; i < Str.length(); ++i) {  
 stk.add(Str.charAt(i));  
 }  
  
 for(i = 0; i < Str.length(); ++i) {  
 stk2.add(stk.pop());  
 }  
  
 for(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 {  
 engine.eval(str1.toString());  
 System.*out*.println("task 11: true");  
 } catch (Throwable var8) {  
 System.*out*.println("task 11: false");  
 }  
  
 }  
}

**Снимки экрана выполнения программы**



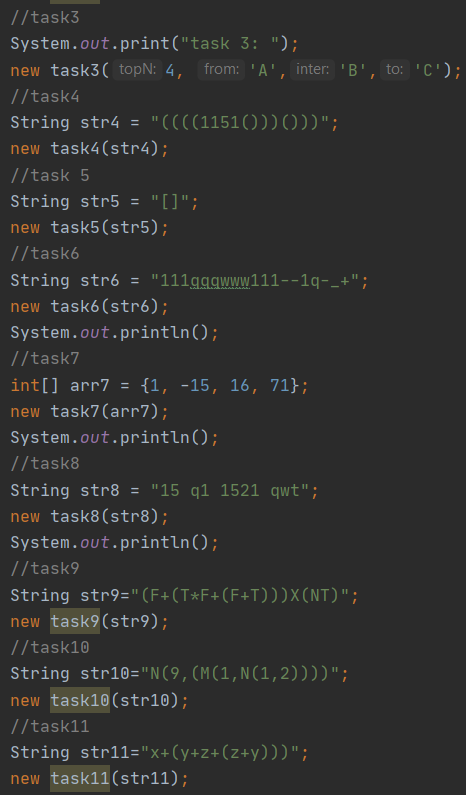


Рис. 1 Входные данные

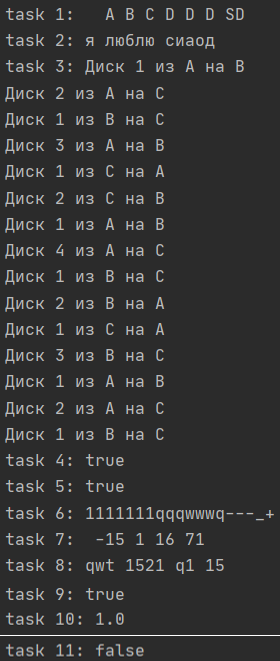


Рис. 2 Выходные данные

# Вывод

В ходе выполнения лабораторной работы, я реализовала структуры данных стек и дек, при помощи них отсортировала строку в алфавитном порядке, расшифровала сообщение, решила задачу на диски и стержни, проверила соответствие скобок, рассортировала буквы, цифры (отрицательные и положительные), решила логическое выражение, нашла максимум и минимум и проверила выражение.