Министерство цифрового развития, связи и массовых коммуникаций

Ордена Трудового Красного Знамени

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Московский технический университет связи и информатики»



Отчет по лабораторной работе №4

по дисциплине «Структура и алгоритмы обработки данных»

по теме «Реализация стека/дека»

Выполнила: студент группы

БВТ1902

Соколова А.Ю.

Москва

2021 г

Оглавление

[Цель работы 3](#_Toc72690649)

[Задание 1 3](#_Toc72690650)

[Код программы 3](#_Toc72690651)

[Задание 2 5](#_Toc72690652)

[Код программы 5](#_Toc72690653)

[Задание 3 7](#_Toc72690654)

[Код программы 7](#_Toc72690655)

[Задание 4 8](#_Toc72690656)

[Код программы 8](#_Toc72690657)

[Задание 5 9](#_Toc72690658)

[Код программы 9](#_Toc72690659)

[Задание 6 10](#_Toc72690660)

[Код программы 10](#_Toc72690661)

[Задание 7 12](#_Toc72690662)

[Код программы 12](#_Toc72690663)

[Задание 8 13](#_Toc72690664)

[Код программы 13](#_Toc72690665)

[Задание 9 14](#_Toc72690666)

[Код программы 14](#_Toc72690667)

[Задание 10 18](#_Toc72690668)

[Код программы 18](#_Toc72690669)

[Задание 11 21](#_Toc72690670)

[Код программы 21](#_Toc72690671)

[Общий код программы 23](#_Toc72690672)

[Входные данные 26](#_Toc72690673)

[Снимки экрана работы программ 30](#_Toc72690674)

[Вывод 32](#_Toc72690675)

## Цель работы

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.

## Задание 1

Отсортировать строки файла, содержащие названия книг, в алфавитном порядке с использованием двух деков.

## Код программы

public class Tasks {

static Scanner open(String path) {

try {

return new Scanner(new FileInputStream(path));

} catch (Exception e) {

throw new Error(e.getMessage());

}

}

static FileInputStream openFile(String path) {

try {

return new FileInputStream(path);

} catch (Exception e) {

throw new Error(e.getMessage());

}

}

static void task1() {

System.out.println(" # Task 1: Книги по алфавиту");

Scanner file = open("files/task1"); // Файл с книгами

Deque<String> first = new Deque<>();

Deque<String> second = new Deque<>();

while (file.hasNextLine()) {

String text = file.nextLine();

if (first.isEmpty()) {

first.pushBack(text);

} else {

second.pushBack(text);

// Сортировка

if (second.end().compareTo(first.end()) >= 0) {

first.pushBack(second.end());

second.popBack();

} else { //если нашли то, что надо положить в середину или начало

while (!first.isEmpty() && second.end().compareTo(first.end()) < 0) {

// перекладываем в начало второго дека, пока не найдем нужную позицию

second.pushFront(first.end());

first.popBack();

}

first.pushBack(second.end());

second.popBack();

while (!second.isEmpty()) { //возвращаем книги обратно в первый дек

first.pushBack(second.begin());

second.popFront();

}

}

}

}

file.close();

while (!first.isEmpty()) {

System.out.println(first.begin());

first.popFront();

}

System.out.println();

}

## Задание 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.

## Код программы

static void task2() {

System.out.println(" # Task 2: Расшифрованное сообщение");

Scanner file = open("files/task2"); // Файл с книгами

String msg = file.nextLine();

file.close();

Deque<Character> st = new Deque<>();

for (int i = 0; i < msg.length(); i++) {

st.pushBack(msg.charAt(i));

}

st.pushFront(st.end());

st.popBack();

st.pushFront(st.end());

st.popBack();

System.out.print("Сообщение: ");

while (!st.isEmpty()) {

System.out.print(st.begin());

st.popFront();

}

System.out.println();

System.out.println();

}

static void hanoi(int n, Stack<String> start, Stack<String> middle, Stack<String> end) {

if (n == 1) {

end.pushBack(start.end());

start.popBack();

} else {

hanoi(n - 1, start, end, middle);

end.pushBack(start.end());

start.popBack();

hanoi(n-1,middle, start, end);

}

}

## Задание 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.

## Код программы

static void task3() {

System.out.println(" # Task 3: Ханойская башня");

Stack<String> first = new Stack<>();

Stack<String> second = new Stack<>();

Stack<String> third = new Stack<>();

Scanner file = open("files/task3");

while (file.hasNextLine()) {

first.pushBack(file.nextLine());

}

file.close();

hanoi(first.size(), first, second,third);

System.out.print("На третьем столбике: ");

while(!third.isEmpty())

{

System.out.print(third.end() + " ");

third.popBack();

}

System.out.println();

System.out.println();

}

## Задание 4

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

## Код программы

static void task4() {

System.out.println(" # Task 4: Баланс круглых скобок");

Scanner file = open("files/task4");

Stack<Character> st = new Stack<>();

while(file.hasNextLine()) {

String text = file.nextLine();

for (int i = 0; i < text.length(); i++) {

st.pushBack(text.charAt(i));

}

}

file.close();

int cnt = 0;

while (!st.isEmpty()) {

if (st.end().equals('(')) {

cnt++;

} else if (st.end() == ')') {

cnt--;

}

st.popBack();

}

System.out.println("Баланс скобок: " + (cnt == 0));

System.out.println();

}

## Задание 5

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

## Код программы

static void task5() {

System.out.println(" # Task 5: Баланс квадратных скобок");

Scanner file = open("files/task5");

Deque<Character> st = new Deque<>();

while(file.hasNextLine()) {

String text = file.nextLine();

for (int i = 0; i < text.length(); i++) {

st.pushBack(text.charAt(i));

}

}

file.close();

int cnt = 0;

while (!st.isEmpty()) {

if (st.end().equals('[')) {

cnt++;

} else if (st.end() == ']') {

cnt--;

}

st.popBack();

}

System.out.println("Баланс скобок: " + (cnt == 0));

System.out.println();

}

## Задание 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

## Код программы

static void task6() {

System.out.println(" # Task 6: Цифры, буквы, символы");

Scanner file = open("files/task6");

Stack<Character> nums = new Stack<>();

Stack<Character> letters = new Stack<>();

Stack<Character> other = new Stack<>();

while(file.hasNextLine()) {

String text = file.nextLine();

for (int i = 0; i < text.length(); i++) {

char ch = text.charAt(i);

if (Character.isDigit(ch)) {

nums.pushBack(ch);

} else if (Character.isLetter(ch)) {

letters.pushBack(ch);

} else if (ch != '\n'){

other.pushBack(ch);

}

}

}

file.close();

Stack<Character> result = new Stack<>(); // Сосздаем строку из символов в массиве

while (!other.isEmpty()) {

result.pushBack(other.popBack());

}

while (!letters.isEmpty()) {

result.pushBack(letters.popBack());

}

while (!nums.isEmpty()) {

result.pushBack(nums.popBack());

}

String resultStr = "";

while (!result.isEmpty()) {

resultStr += result.popBack();

}

System.out.println("Полученная строка: " + resultStr);

System.out.println();

}

## Задание 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.

## Код программы

static void task7() {

System.out.println(" # Task 7: Отрицательные и положительные");

Scanner file = open("files/task7");

Deque<Integer> pos = new Deque<>();

Deque<Integer> neg = new Deque<>();

while(file.hasNext()) {

try {

String text = file.next();

int x = Integer.parseInt(text);

if (x >= 0) {

pos.pushBack(x);

} else {

neg.pushBack(x);

}

} catch (Exception e) {}

}

file.close();

String resultStr = "";

while (!neg.isEmpty()) {

resultStr += neg.popFront() + " ";

}

while (!pos.isEmpty()) {

resultStr += pos.popFront() + " ";

}

System.out.println("Результат: " + resultStr);

System.out.println();

}

## Задание 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.

## Код программы

static void task8() {

System.out.println(" # Task 8: ");

try {

Scanner file = open("files/task8");

Stack<String> lines = new Stack<>();

while(file.hasNextLine()) {

String text = file.nextLine();

lines.pushBack(text);

}

file.close();

FileWriter fileO = new FileWriter("files/task8output.txt");

while (!lines.isEmpty()) {

fileO.write(lines.popBack() + "\n");

}

fileO.close();

System.out.println("Файл files/task8output.txt сохранен");

System.out.println();

} catch (IOException e) {

e.printStackTrace();

}

}

## Задание 9

Дан текстовый файл. Используя стек, вычислить значение логического выражения, записанного в текстовом файле в следующей форме: < ЛВ > ::= T | F | (N) | (A) | (X) | (O), где буквами обозначены логические константы и операции: T – True, F – False, N – Not, A – And, X – Xor, O – Or.

## Код программы

static void task9() {

System.out.println(" # Task 9: ");

String text = "";

String formula = "";

Stack<Character> first = new Stack<>();

char buf1;

char buf2;

char cond1;

char cond2;

Scanner file = open("files/task9");

while (file.hasNextLine())

{

text = file.nextLine();

formula=text;

}

file.close();

for (int i=0; i< formula.length();)

{

if (formula.charAt(i) != ')')

{

if (formula.charAt(i) != '(')

{

first.pushBack(formula.charAt(i));

}

i++;

}

else if (!first.isEmpty())

{

buf1=first.end();

first.popBack();

char var = first.end();

first.pushBack(buf1);

switch (var)

{

case 'N':

{

char cond = first.end();

i++;

for (int t=0; t<2;t++)

{

first.popBack();

}

if (cond == 'T') {

first.pushBack('F');

} else {

first.pushBack('T');

}

break;

}

case 'X':

{

buf1=first.end();

first.popBack();

buf2=first.end();

first.popBack();

cond1=first.end();

first.pushBack(buf2);

first.pushBack(buf1);

cond2= first.end();

i++;

for(int t = 0; t < 3; t++)

{

first.popBack();

}

first.pushBack((cond1==cond2)? 'F': 'T');

break;

}

case 'A':

{

buf1=first.end();

first.popBack();

buf2=first.end();

first.popBack();

cond1=first.end();

first.pushBack(buf2);

first.pushBack(buf1);

cond2= first.end();

i++;

for(int t = 0; t < 3; t++)

{

first.popBack();

}

first.pushBack((cond1==cond2 && cond2=='T')? 'T':'F');

break;

}

case 'O':

{

buf1=first.end();

first.popBack();

buf2=first.end();

first.popBack();

cond1=first.end();

first.pushBack(buf2);

first.pushBack(buf1);

cond2= first.end();

i++;

for(int t = 0; t < 3; t++)

{

first.popBack();

}

first.pushBack((cond1==cond2 && cond2=='F')? 'F':'T');

break;

}

}

}

}

System.out.println("Выражение: " + formula);

System.out.println("Результат: " + first.end());

System.out.println();

}

## Задание 10

Дан текстовый файл. В текстовом файле записана 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.

## Код программы

public static void computeTask10(String expr, Stack<Integer> st, int deep) throws Exception {

int idx;

// String d = " ".repeat(deep);

// System.out.println(d + "EXPR: " + expr);

int num = -1;

boolean isNumber = true;

try {

num = Integer.parseInt(expr);

} catch (Exception e) {

isNumber = false;

}

if (isNumber) {

st.pushBack(num);

} else if (isInBrackets(expr)) {

computeTask10(expr.substring(1, expr.length() - 1), st, deep + 1);

} else if (expr.startsWith("M(") && expr.endsWith(")")) {

computeTask10(expr.substring(2, expr.length() - 1), st, deep + 1);

st.pushBack(Math.max(st.popBack(), st.popBack()));

} else if (expr.startsWith("N(") && expr.endsWith(")")) {

computeTask10(expr.substring(2, expr.length() - 1), st, deep + 1);

st.pushBack(Math.min(st.popBack(), st.popBack()));

} else if ((idx = indexWithBrackets(expr,',')) != -1) {

computeTask10(expr.substring(0, idx), st, deep + 1);

computeTask10(expr.substring(idx + 1), st, deep + 1);

} else {

throw new Exception("Illegar expression");

}

}

public static int indexWithBrackets(String s, char ch) {

int bracket = 0;

for (int i = 0; i < s.length(); i++) {

if (s.charAt(i) == '(') { bracket += 1;}

else if (s.charAt(i) == ')') { bracket -= 1; }

else if (bracket == 0) { // not in brackets

if (s.charAt(i) == ch) {

return i;

}

}

}

return -1;

}

public static boolean isInBrackets(String s) throws Exception {

if (!(s.startsWith("(") && s.endsWith(")"))) {

return false;

}

s = s.substring(1);

int bracket = 1;

for (int i = 0; i < s.length(); i++) {

if (bracket == 0) { return false; }

if (s.charAt(i) == '(') { bracket += 1;}

else if (s.charAt(i) == ')') { bracket -= 1; }

if (bracket < 0) { throw new Exception("Illegar expression"); }

}

return bracket == 0;

}

public static void task10() {

System.out.println(" # Task 10");

Scanner file = open("files/task10");

String formula = file.nextLine();

file.close();

Stack<Integer> st = new Stack<>();

try {

computeTask10(formula, st, 0);

int result = st.popBack();

System.out.println("Выражение: " + formula);

System.out.println("Результат: " + result);

System.out.println();

} catch (Exception e) {

e.printStackTrace();

}

}

## Задание 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

## Код программы

public static void computeTask11(String expr, Stack<Integer> st, int deep) throws Exception {

// String d = " ".repeat(deep);

// System.out.println(d + "EXPR: " + expr);

if (expr.equals("x")) {

st.pushBack(1);

} else if (expr.equals("y")) {

st.pushBack(2);

} else if (expr.equals("z")) {

st.pushBack(3);

} else if (isInBrackets(expr)) {

computeTask11(expr.substring(1, expr.length() - 1), st, deep + 1);

} else if ((indexWithBrackets(expr, '+') != -1) || (indexWithBrackets(expr, '-') != -1)) {

int idx = Math.min(

maxIfM1(indexWithBrackets(expr, '+')),

maxIfM1(indexWithBrackets(expr, '-'))

);

computeTask11(expr.substring(0, idx), st, deep + 1);

computeTask11(expr.substring(idx + 1), st, deep + 1);

if (expr.charAt(idx) == '+') {

st.pushBack(st.popBack() + st.popBack());

} else {

st.pushBack(- st.popBack() + st.popBack());

}

} else {

throw new Exception("Illegar expression");

}

}

public static int maxIfM1(int x) {

if (x == -1) { return Integer.MAX\_VALUE; }

else { return x; }

}

public static void task11() {

System.out.println(" # Task 11");

Scanner file = open("files/task11");

String formula = file.nextLine();

file.close();

try {

Stack<Integer> st = new Stack<>();

computeTask11(formula, st, 0);

int result = st.popBack();

System.out.println("Выражение: " + formula);

System.out.println("Результат: " + "корректно");

System.out.println();

} catch (Exception e) {

System.out.println("Выражение: " + formula);

System.out.println("Результат: " + "НЕкорректно");

System.out.println();

}

}

}

## Общий код программы

public class Stack<T> {

public static final int N = 1000;

Object[] data;

int last;

public Stack(int capacity) {

this.data = new Object[capacity];

this.last = 0;

}

public Stack() {

this(N);

}

public boolean isEmpty() {

return last == 0;

}

public void pushBack(T value) {

if (last == data.length) {

throw new Error("Stack overflow");

}

this.data[last] = value;

last += 1;

}

public T popBack() {

if (last == 0) {

return null;

} else {

T el = (T) this.data[last-1];

this.data[last-1] = null; // Удаляем объект

last -= 1;

return el;

}

}

public T end() {

if (last == 0) return null;

else return (T) data[last-1];

}

public int size() {

return last;

}

public class Deque<T> extends Stack<T> {

public void pushFront(T value) {

if (last == data.length) {

throw new Error("Overflow");

}

for (int i = last; i > 0; i--) {

data[i] = data[i - 1];

}

this.data[0] = value;

last += 1;

}

public T popFront() {

if (last == 0) {

return null;

} else {

T el = (T) this.data[0];

for (int i = 0; i < last-1; i++) {

data[i] = data[i + 1];

}

this.data[last-1] = null; // Удаляем объект

last -= 1;

return el;

}

}

public T begin() {

if (last == 0) {

return null;

} else {

return (T) data[0];

}

}

}

public class Main {

public static void main(String[] args) {

Tasks.task1();

Tasks.task2();

Tasks.task3();

Tasks.task4();

Tasks.task5();

Tasks.task6();

Tasks.task7();

Tasks.task8();

Tasks.task9();

Tasks.task10();

Tasks.task11();

}

}

## Входные данные

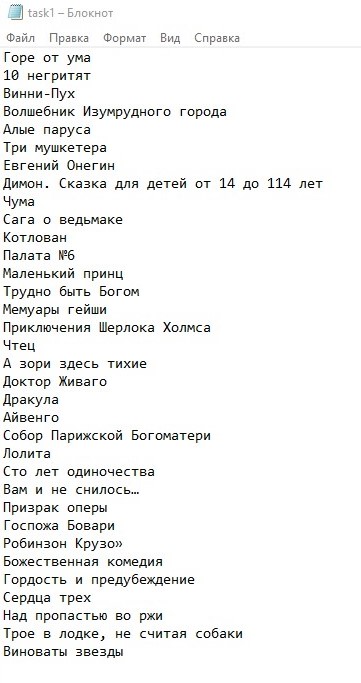


Рисунок 1 – Задача №1

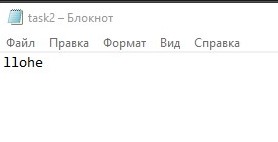


Рисунок 2 – Задача №2

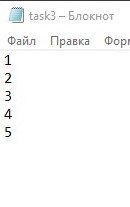


Рисунок 3 – Задача №3

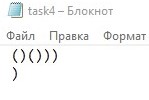


Рисунок 4 – Задача №4

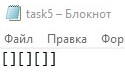


Рисунок 5 – Задача №5

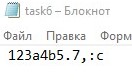


Рисунок 6 – Задача №6

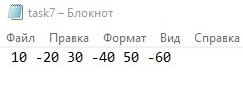


Рисунок 7 – Задача №7

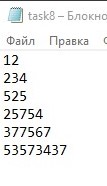


Рисунок 8 – Задача №8

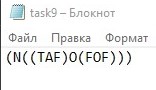


Рисунок 9 – Задача №9

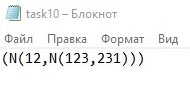


Рисунок 10 – Задача №10

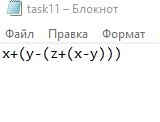


Рисунок 11 – Задача №11

## Снимки экрана работы программ

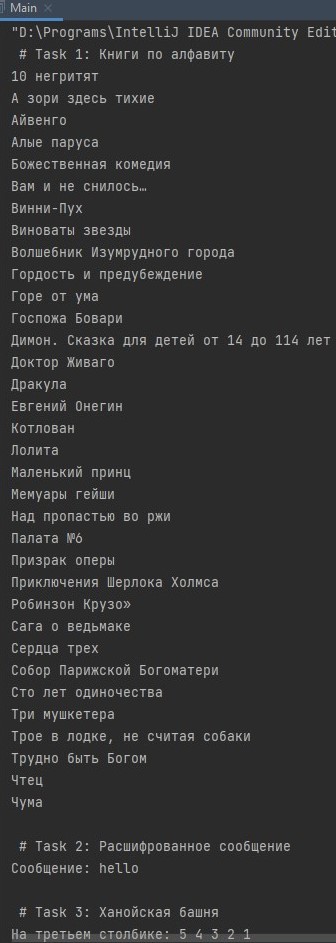


Рисунок 12 – Результат выполнение программы

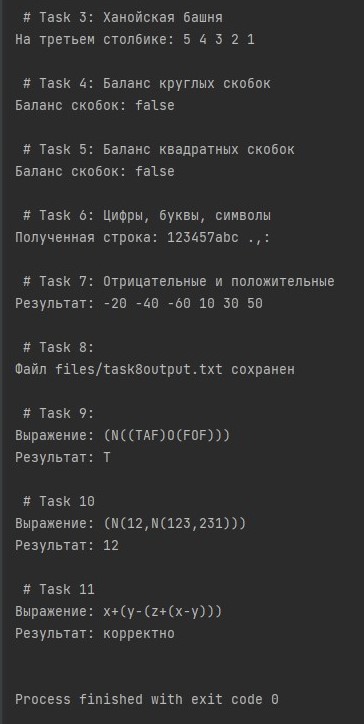


Рисунок 13 – Результат выполнение программы

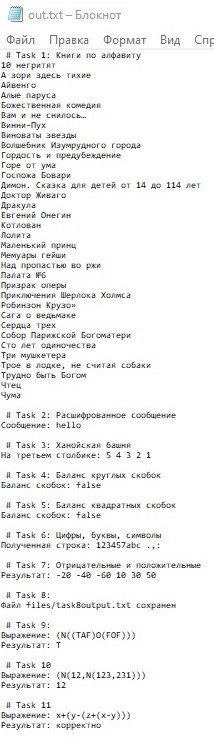


Рисунок 14 – Файл с результатами выполнение программы

## Вывод

Я реализовала структуры данных стек и дек и разработала программу обработки данных, содержащихся в заранее подготовленном txt-файле, в соответствии с заданиями, применив указанную в задании структуру данных.