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»

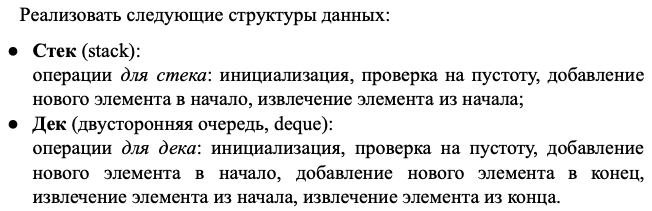
по дисциплине «Структуры и алгоритмы обработки данных»

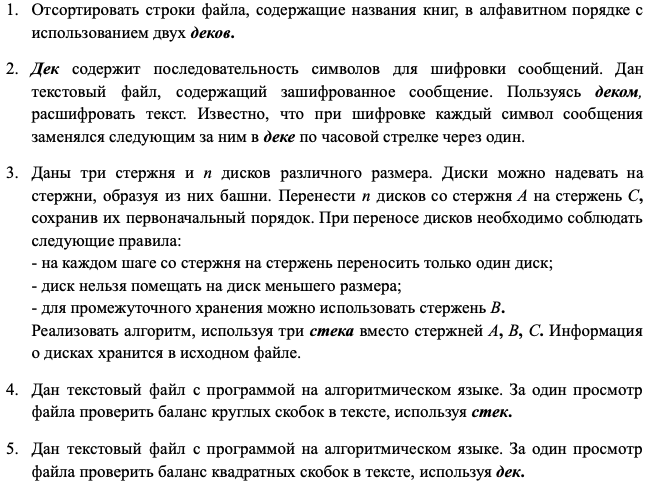
Выполнил: студент группы БВТ1902

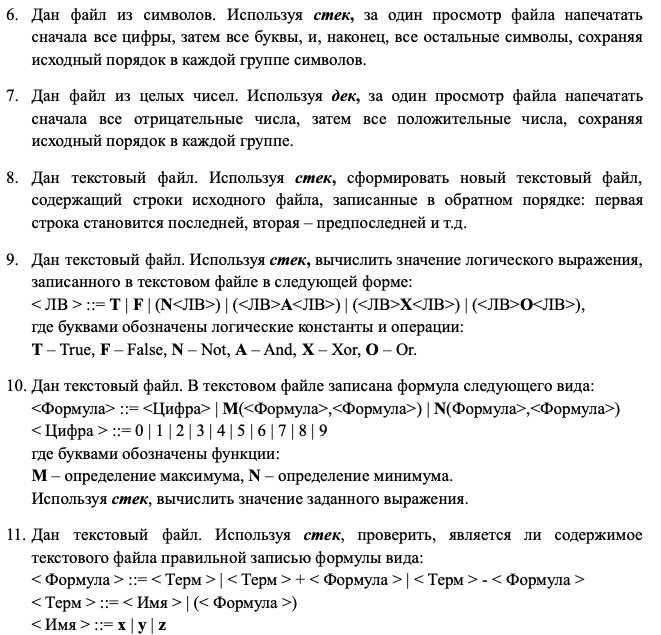
Холмовский Олег Васильевич

Москва

2021

Задание





Код программы

Стек:

*class* Stack {

constructor() {

*this*.items = [];

*this*.count = 0;

}

push(element) {

*this*.items[*this*.count] = element;

*this*.count += 1;

*return* *this*.count - 1;

}

pop() {

*if* (*this*.count == 0) *return* *undefined*;

let deleteItem = *this*.items[*this*.count - 1];

*this*.count -= 1;

*return* deleteItem;

}

peek() {

*return* *this*.items[*this*.count - 1];

}

isEmpty() {

*return* *this*.count == 0;

}

size() {

*return* *this*.count;

}

print() {

let str = '';

*for* (let i = 0; i < *this*.count; i++) {

str += *this*.items[i] + ' ';

}

*return* str;

}

clear() {

*this*.items = [];

*this*.count = 0;

*return* *this*.items;

}

}

*module*.*exports* = { Stack };

Дек:

*class* Dequeue {

constructor() {

*this*.items = {};

*this*.count = 0;

*this*.lowestCount = 0;

}

addFront(element) {

*if* (*this*.isEmpty()) {

*this*.addBack(element);

} *else* *if* (*this*.lowestCount > 0) {

*this*.lowestCount--;

*this*.items[*this*.lowestCount] = element;

} *else* {

*for* (let index = *this*.count; index > 0; index--) {

*this*.items[index] = *this*.items[index - 1];

}

*this*.count++;

*this*.items[0] = element;

}

}

addBack(element) {

*this*.items[*this*.count] = element;

*this*.count++;

}

removeFront() {

*if* (*this*.isEmpty()) {

*return* *undefined*;

}

let result = *this*.items[*this*.lowestCount];

*delete* *this*.items[*this*.lowestCount];

*this*.lowestCount++;

*return* result;

}

removeBack() {

*if* (*this*.isEmpty()) {

*return* *undefined*;

}

let result = *this*.items[*this*.count - 1];

*delete* *this*.items[*this*.count - 1];

*this*.count--;

*return* result;

}

peekFront() {

*if* (*this*.isEmpty()) {

*return* *undefined*;

}

*return* *this*.items[*this*.lowestCount];

}

peekBack() {

*if* (*this*.isEmpty()) {

*return* *undefined*;

}

*return* *this*.items[*this*.count - 1];

}

isEmpty() {

*return* *this*.count - *this*.lowestCount == 0;

}

size() {

*return* *this*.count - *this*.lowestCount;

}

clear() {

*this*.items = {};

*this*.count = 0;

*this*.lowestCount = 0;

}

print() {

*if* (*this*.isEmpty()) {

*return* '';

}

let string = `${*this*.items[*this*.lowestCount]}`;

*for* (let index = *this*.lowestCount + 1; index < *this*.count; index++) {

string = `${string},${*this*.items[index]}`;

}

*return* string;

}

}

*module*.*exports* = { Dequeue };

Задания:

const { Dequeue } = require('./dequeue');

const { Stack } = require('./stack');

const arrayTask1 = [

'Гадюка',

'Винни-Пух',

'В гостях у Чебурашки',

'Алые паруса',

'Три зятя',

'Евгений Онегин',

'Добрыня. Былины 2',

'Баба и курица',

];

const task1 = (array) => {

let result = [];

const a = *new* Dequeue();

const b = *new* Dequeue();

*for* (let i = 0; i < array.length; i++) {

*if* (a.size() === 0) {

a.addBack(array[i]);

*continue*;

}

*if* (

array[i].charAt(0).toLowerCase() > a.peekBack().charAt(0).toLowerCase()

) {

a.addBack(array[i]);

*continue*;

}

*if* (

array[i].charAt(0).toLowerCase() < a.peekFront().charAt(0).toLowerCase()

) {

a.addFront(array[i]);

*continue*;

}

*while* (

a.peekBack().charAt(0).toLowerCase() > array[i].charAt(0).toLowerCase()

) {

let x = a.removeBack();

b.addFront(x);

}

a.addBack(array[i]);

*while* (b.size() > 0) {

let x = b.removeFront();

a.addBack(x);

}

}

*while* (a.size() > 0) {

let x = a.removeFront();

result.push(x);

}

*return* result;

};

console.log('Задание 1:');

console.log(task1(arrayTask1));

const strTask2 = 'б скъх пв иу';

const dequeueTask2 = *new* Dequeue();

const str2Task2 = 'абвгдеёжзийклмнопрстуфхцчшщъыьэюя';

*for* (let i = 0; i < str2Task2.length; i++) {

dequeueTask2.addBack(str2Task2.charAt(i));

}

const task2 = (str, dequeue) => {

let result = '';

*for* (i = 0; i < str.length; i++) {

*if* (str[i] === ' ') {

result += ' ';

*continue*;

}

*while* (result.length <= i) {

*if* (dequeue.peekBack() === str[i]) {

let x = dequeue.removeBack();

dequeue.addFront(x);

x = dequeue.removeBack();

dequeue.addFront(x);

result += dequeue.removeBack();

dequeue.addBack(result[i]);

} *else* {

let x = dequeue.removeBack();

dequeue.addFront(x);

}

}

}

*return* result;

};

console.log('Задание 2:');

console.log(`Было: ${strTask2}`);

console.log(`Стало: ${task2(strTask2, dequeueTask2)}`);

const a = *new* Stack();

const b = *new* Stack();

const c = *new* Stack();

i = 5;

*for* (i = 5; i != 0; i--) {

a.push(i);

}

console.log('Задание 3:');

console.log('Было:');

console.log(`a: ${a.print()}`);

console.log(`b: ${b.print()}`);

console.log(`c: ${c.print()}`);

const task3 = (n, a, c, b) => {

*if* (n === 2) {

let x = a.pop();

b.push(x);

x = a.pop();

c.push(x);

x = b.pop();

c.push(x);

}

*if* (n > 2) {

task3(n - 1, a, b, c);

x = a.pop();

c.push(x);

task3(n - 1, b, c, a);

}

};

task3(5, a, c, b);

console.log('Стало:');

console.log(`a: ${a.print()}`);

console.log(`b: ${b.print()}`);

console.log(`c: ${c.print()}`);

const strTask4 = '()(()()))()';

const task4 = (str) => {

const stack = *new* Stack();

*for* (i = 0; i < str.length; i++) {

*if* (str[i] === '(' || str[i] === ')') {

*if* (stack.isEmpty()) {

*if* (str[i] === ')') {

*return* false;

}

stack.push(str[i]);

*continue*;

}

*if* (str[i] === ')' && stack.peek() === '(') {

stack.pop();

*continue*;

}

stack.push(str[i]);

}

*if* (!stack.isEmpty()) {

*return* false;

}

*return* true;

}

};

console.log('Задание 4:');

console.log(`Строка: ${strTask4}`);

console.log(task4(strTask4));

const strTask5 = '[[]]';

const task5 = (str) => {

const dequeue = *new* Dequeue();

*for* (i = 0; i < str.length; i++) {

*if* (str[i] == '[' || str[i] == ']') {

*if* (dequeue.size() === 0) {

*if* (str[i] == ']') {

*return* false;

}

dequeue.addBack(str[i]);

*continue*;

}

*if* (str[i] == ']' && dequeue.peekBack() == '[') {

dequeue.removeBack();

*continue*;

}

dequeue.addBack(str[i]);

}

}

*if* (dequeue.size() > 0) {

*return* false;

}

*return* true;

};

console.log('Задание 5:');

console.log(`Строка: ${strTask5}`);

console.log(task5(strTask5));

const strTask6 = '3a$sd!z54%xc';

const task6 = (str) => {

stack1 = *new* Stack();

stack2 = *new* Stack();

stack3 = *new* Stack();

*for* (i = str.length - 1; i >= 0; i--) {

let x = str[i].charAt(0);

*if* (!!x.trim() && x > -1) {

stack1.push(str[i]);

*continue*;

}

*if* (x.match(/[a-z]/i)) {

stack2.push(str[i]);

*continue*;

}

stack3.push(str[i]);

}

let arr1 = [];

let arr2 = [];

let arr3 = [];

*while* (!stack1.isEmpty() || !stack2.isEmpty() || !stack3.isEmpty()) {

*if* (!stack1.isEmpty()) {

arr1.push(stack1.pop());

}

*if* (!stack2.isEmpty()) {

arr2.push(stack2.pop());

}

*if* (!stack3.isEmpty()) {

arr3.push(stack3.pop());

}

}

*return* arr1.concat(arr2, arr3).join('');

};

console.log('Задание 6:');

console.log(`Было: ${strTask6}`);

console.log(`Стало: ${task6(strTask6)}`);

const arrTask7 = [1, 8, 55, -1, -11, 52, -111, -55, 2, 34];

const task7 = (arr) => {

result = [];

dequeue = *new* Dequeue();

*for* (num *of* arr) {

*if* (num > 0) {

dequeue.addFront(num);

*continue*;

}

result.push(num);

}

*while* (!dequeue.isEmpty()) {

result.push(dequeue.removeBack());

x = dequeue.peekBack();

}

*return* result;

};

console.log('Задание 7:');

console.log(`Было: ${arrTask7}`);

console.log(`Стало: ${task7(arrTask7)}`);

const strTask8 = `У лукоморья дуб зелёный;

Златая цепь на дубе том:

И днём и ночью кот учёный

Всё ходит по цепи кругом;`;

const task8 = (str) => {

let result = '';

str = str.split('\n');

stack = *new* Stack();

*for* (line *of* str) {

stack.push(line);

}

*while* (!stack.isEmpty()) {

result += stack.pop() + '\n';

}

*return* result;

};

console.log('Задание 8:');

console.log(`Было: ${strTask8} \n`);

console.log(`Стало: ${task8(strTask8)}`);

let strTask9 = '(F+(T\*F+(F+T)))X(NT)';

const task9 = (Str) => {

let str1 = '';

let stack = *new* Stack();

*for* (let i = 0; i < Str.length; i++) {

stack.push(Str[i]);

}

*for* (let i = 0; i < Str.length; i++) {

*if* (stack.peek() === 'T') str1 = 'true ' + str1;

*if* (stack.peek() === 'F') str1 = 'false ' + str1;

*if* (stack.peek() === 'N') str1 = '! ' + str1;

*if* (stack.peek() === 'A' || stack.peek() === '\*') str1 = '&& ' + str1;

*if* (stack.peek() === 'X') str1 = '!= ' + str1;

*if* (stack.peek() === 'O' || stack.peek() === '+') str1 = '|| ' + str1;

*if* (stack.peek() === '(') str1 = '( ' + str1;

*if* (stack.peek() === ')') str1 = ')' + str1;

stack.pop();

}

*return* eval(str1);

};

console.log('Задание 9:');

console.log(`Строка: ${strTask9}`);

console.log(task9(strTask9));

let strTask10 = 'M(9,(M(1,N(1,2))))';

const task10 = (Str) => {

let str1 = '';

let stack = *new* Stack();

*for* (let i = 0; i < Str.length; i++) {

stack.push(Str[i]);

}

*for* (let i = 0; i < Str.length; i++) {

*if* (stack.peek() === '0') str1 = '0' + str1;

*if* (stack.peek() === '1') str1 = '1' + str1;

*if* (stack.peek() === '2') str1 = '2' + str1;

*if* (stack.peek() === '3') str1 = '3' + str1;

*if* (stack.peek() === '4') str1 = '4' + str1;

*if* (stack.peek() === '5') str1 = '5' + str1;

*if* (stack.peek() === '6') str1 = '6' + str1;

*if* (stack.peek() === '7') str1 = '7' + str1;

*if* (stack.peek() === '8') str1 = '8' + str1;

*if* (stack.peek() === '9') str1 = '9' + str1;

*if* (stack.peek() === 'M') str1 = 'Math.max' + str1;

*if* (stack.peek() === 'N') str1 = 'Math.min' + str1;

*if* (stack.peek() === ',' || stack.peek() === '.') str1 = ',' + str1;

*if* (stack.peek() === '(') str1 = '(' + str1;

*if* (stack.peek() === ')') str1 = ')' + str1;

stack.pop();

}

*return* eval(str1);

};

console.log('Задание 10:');

console.log(`Строка: ${strTask10}`);

console.log(task10(strTask10));

let strTask11 = 'x+(y+z+(z+y))';

const task11 = (Str) => {

let x = 1,

y = 1,

z = 1;

let stack = *new* Stack();

let str = '';

*for* (let i = 0; i < Str.length; i++) {

stack.push(Str[i]);

}

*for* (let i = 0; i < Str.length; i++) {

str = stack.pop() + str;

}

*try* {

eval(str);

} *catch* (err) {

*return* false;

}

*return* true;

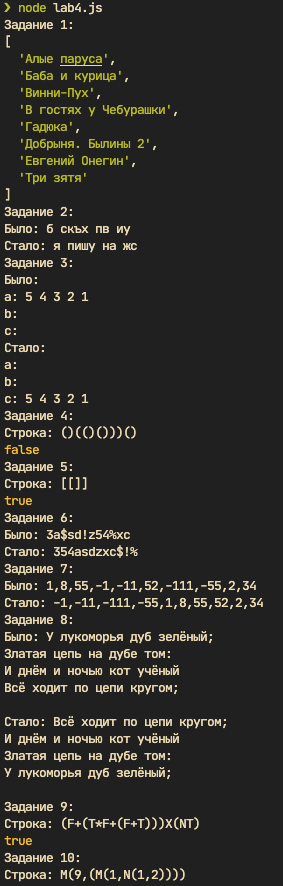
};

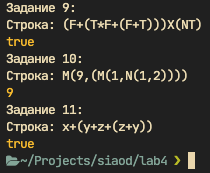
console.log('Задание 11:');

console.log(`Строка: ${strTask11}`);

console.log(task11(strTask11));

Cнимки экрана работы программы





Вывод

В ходе выполнения лабораторной работы, мной были получены навыки реализации таких структур данных как стек и дек, а так же решения задач с их использованием.