**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**



**МОСКОВСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**

**ВЫСШАЯ ШКОЛА ПЕЧАТИ И МЕДИАИНДУСТРИИ**

***Институт Принтмедиа и информационных технологий***

***Кафедра Информатики и информационных технологий***

**направление подготовки**

**09.03.01 «Информатика и вычислительная техника»**

**ЛАБОРАТОРНАЯ РАБОТА № 1**

**Дисциплина:** Технология кроссплатформенного программирования

**Тема:** Визуализация операций над стеком и очередью с помощью библиотеки DraculaGraph

**Выполнила:** студентка группы ДЦисБ-3-3

Рыбакова Д.Н.

**Проверила:**  Алпатова М.В. ***\_\_\_\_\_\_\_\_\_\_\_***

(Фамилия И.О.) **(Оценка)**

**Дата, подпись** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  ***\_\_\_\_\_\_\_\_\_\_\_***

(Дата) (Подпись)

**Замечания: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Москва**

**2018**

Оглавление

[Задание на лабораторную работу 3](#_Toc528861221)

[Листинг программы 4](#_Toc528861222)

[Скриншоты выполнения программы 7](#_Toc528861223)

Задание на лабораторную работу

Необходимо написать клиентское приложение на языке JavaScript, которое позволяет добавлять и удалять элементы из стека и очереди. После выполнения каждой операции над стеком или очередью, визуализация, выполненная при помощи библиотеки DraculaGraph, должна автоматически обновляться.

Листинг программы

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<script type="text/javascript" src="raphael-min.js"></script>

<script type="text/javascript" src="dracula\_graph.js"></script>

<script type="text/javascript" src="dracula\_graffle.js"></script>

<script type="text/javascript" src="jquery-1.4.2.min.js"></script>

<style> </style>

</head>

<body>

<center><h2>СТЕК</h2>

<div id = "form">

<form>

<p><input type="text" id="str" >

<input type="button" id = "addBut" value="Добавить" onclick="st.push();" >

<input type="button" id = "delBut" value="Удалить" onclick="st.pop();"> </p>

</form>

</div>

<div width ="700" height="500" id="canvas" ></div> </center>

<script type = "text/javascript">

var stackArr = [];

var g = new Graph();

var w = 700;

var h = 500;

var renderer, layouter;

var Stack = function(size){ //функция для стека

this.Size = size;

var last = -1;

this.push = function(){ //функция добавления элемента в стек

if(last < size - 1){

var str = document.getElementById("str").value;

if(str == '')

alert("Ошибка! Введите элемент."); //ошибка, если ничего не введено

else {

temp = ++last;

stackArr.push({temp, str}); //заполняем массив стека

g.addNode(temp,{label:str}); //добавляем узел

if(temp!=0)

g.addEdge(temp,temp-1); //соединяем узлы

layouter.layout();

renderer.draw();

document.getElementById("str").value = "";

}

}

else alert("Стек полон!"); //ошибка, если введено больше 10 элементов

}

this.pop = function(){ //функция извлечения последнего элемента из стека

if(last >= 0){

var temp = last--;

var value = stackArr[temp].Value;

stackArr.pop({temp, value}); //удаляем элемент из массива

g.nodes[temp].hide(); //скрываем связь

g.removeNode(temp); //удаляем узел

renderer.draw();

}

else alert("Невозможно удалить элемент, поскольку стек пуст.");

}

return this;

}

var st = new Stack(10);

renderer = new Graph.Renderer.Raphael('canvas', g, w, h);

layouter = new Graph.Layout.Spring(g);

</script>

<hr>

<center><h2>ОЧЕРЕДЬ</h2>

<div id = "form1">

<form>

<p><input type="text" id="str1" >

<input type="button" id = "addBut1" value="Добавить" onclick="qu.push();" >

<input type="button" id = "delBut1" value="Удалить" onclick="qu.pop();"> </p>

</form>

</div>

<div width ="700" height="500" id="canvas1" ></div> </center>

<script type = "text/javascript">

var QueueArr = [];

var g1 = new Graph();

var w = 700;

var h = 500;

var renderer1, layouter1;

var Queue = function(size){ //функция для очереди

this.Size = size;

var last = 0;

var first = 0;

this.push = function(){ //функция добавления элемента в очередь

if ((last + 1)%size == first) alert ("Очередь заполнена!");

else{

var str = document.getElementById("str1").value;

if(str == '') alert("Ошибка! Введите элемент.");

else {

temp = last++;

QueueArr.push({temp, str}); //заполняем массив очереди

g1.addNode(temp, {label : str}); //добавляем новый узел

last%=size;

if(temp!=first){

if(temp ==0 && temp != first)

g1.addEdge(size-1, temp);

else g1.addEdge(temp-1, temp);

}

layouter1.layout();

renderer1.draw();

document.getElementById("str1").value = "";

}

}

}

this.pop = function(){ //функция извлечения элемента из очереди

if(first == last) alert("Ошибка! Очередь пуста.");

else {

var val;

val = first++;

g1.nodes[val].hide();

first%=size;

g1.removeNode(val);

renderer1.draw();

}

}

return this;

}

var qu = new Queue(10);

renderer1 = new Graph.Renderer.Raphael('canvas1', g1, w, h);

layouter1 = new Graph.Layout.Spring(g1);

</script>

</body>

</html>

Скриншоты выполнения программы

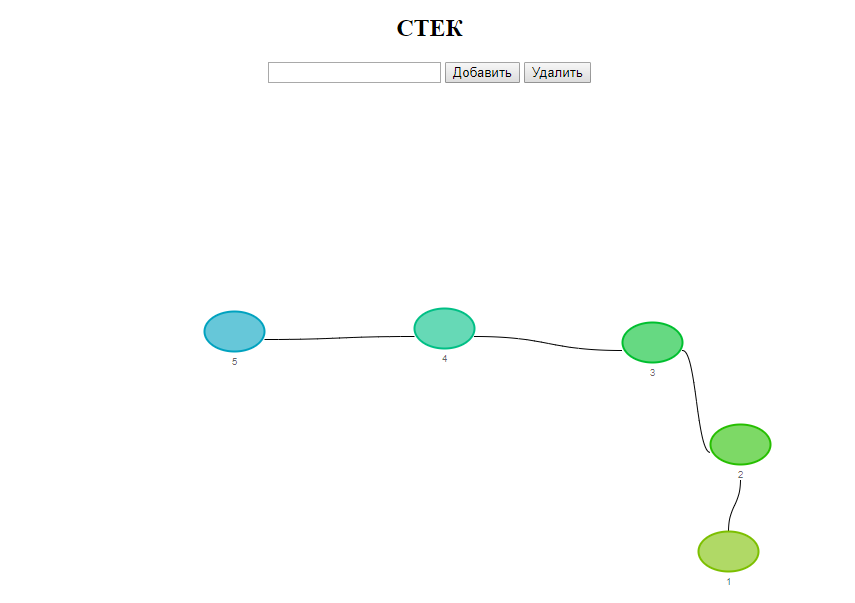


Рис. 1. Результат добавления и удаления элементов из стека (удалены элементы 6 и 7).

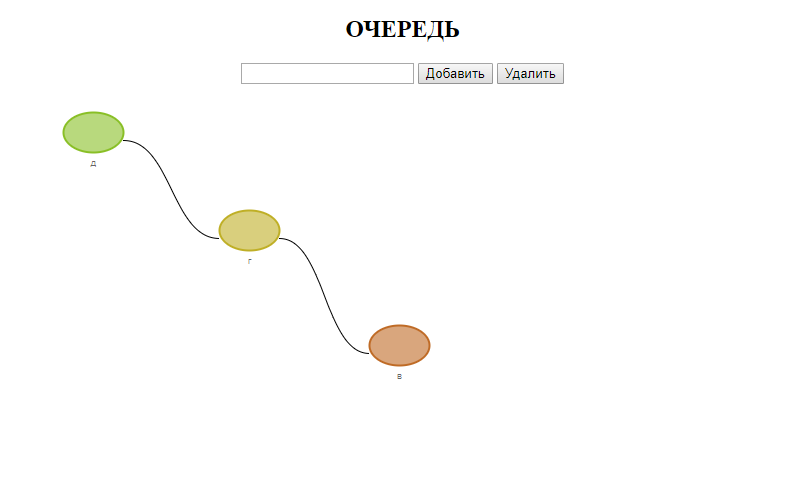


Рис. 2. Результат добавления и удаления элементов из очереди (удалены элементы «а» и «б»).