**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**

**09.03.02 «Информационные системы и технологии»,**

**ЛАБОРАТОРНАЯ РАБОТА №1**

**Дисциплина:** Технология кроссплатформенного программирования

**Тема:** Стек и очередь

**Выполнил: студент группы ДЦисБ-2-2**

**Матевосян В.А.**

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

(Дата) (Подпись)

**Проверил: Северинов Н.А.**

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

(Дата) (Подпись)

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

**Москва**

**2017 Содержание**

1. Задание……………………..…………………………………...3
2. Листинг программы……………………………………………4
3. Скриншоты программы………………………………………..8

**Задание**

Реализовать программу, производящую операции над элементами стека и очереди. Разработать графический пользовательский интерфейс с помощью фреймворка QT.

**Листинг программы**

Листинг queue.h

#ifndef QUEUE\_H

#define QUEUE\_H

struct Element2

{

Element2( int x, Element2\* y): data(x), prev(y) {}

Element2( int x): data(x){}

Element2\* prev;

int data;

};

class Queue

{

private:

Element2\* last;

Element2\* first;

public:

Queue(): last(0), first(0) {}

void **Push**(int x)

{

if(last != 0)

{

last = new Element2(x, last);

if ((\*first).prev==0) (\*first).prev = last;

}

else

{

last = new Element2(x);

first = new Element2(x);

}

}

int **Pop**()

{

if(last != 0)

{

int el = (\*first).data;

first = (\*first).prev;

return el;

}

return -1;

}

};

#endif // QUEUE\_H

Листинг stack.h

#ifndef STACK\_H

#define STACK\_H

struct Element

{

Element( int x, Element\* y): data(x), next(y) {}

Element( int x): data(x){}

Element\* next;

int data;

};

class Stack

{

private:

Element\* first;

public:

Stack(): first(0) {}

void **Push**(int x)

{

if(first != 0)

first = new Element(x, first);

else

first = new Element(x);

}

int **Pop**()

{

if(first != 0)

{

int el = (\*first).data;

first = (\*first).next;

return el;

}

return -1;

}

};

#endif // STACK\_H

Листинг main.cpp

#include "widget.h"

#include <QApplication>

#include <QLabel>

int main(int argc, char \*argv[])

{

QApplication a(argc, argv);

Widget w;

w.show();

return a.exec();

}

Листинг widget.cpp

#include "widget.h"

#include "ui\_widget.h"

#include <string>

#include <QString>

#include <QListWidget>

#include "stack.h"

#include "queue.h"

Stack NewStack;

Queue NewQueue;

Widget::**Widget**(QWidget \*parent) :

QWidget(parent),

ui(new Ui::Widget)

{

ui->setupUi(this);

}

Widget::~***Widget***()

{

delete ui;

}

void Widget::**on\_pushButton\_clicked**()

{

if (ui->StackBox->isChecked()&& !ui->QueueBox->isChecked()){

if (ui->textEdit->toPlainText().size() != 0){

ui->listWidget->insertItem(0, ui->textEdit->toPlainText());

NewStack.Push(ui->textEdit->toPlainText().toInt());

ui->textEdit->clear();

ui->textEdit->setFocus();

}

}

else if (ui->QueueBox->isChecked() && !ui->StackBox->isChecked()){

if (ui->textEdit->toPlainText().size() != 0){

ui->listWidget\_2->addItem(ui->textEdit->toPlainText());

NewQueue.Push(ui->textEdit->toPlainText().toInt());

ui->textEdit->clear();

ui->textEdit->setFocus();

}

}

else if (ui->QueueBox->isChecked() && ui->StackBox->isChecked()){

if (ui->textEdit->toPlainText().size() != 0){

ui->listWidget\_2->addItem(ui->textEdit->toPlainText());

NewQueue.Push(ui->textEdit->toPlainText().toInt());

ui->listWidget->insertItem(0, ui->textEdit->toPlainText());

NewStack.Push(ui->textEdit->toPlainText().toInt());

ui->textEdit->clear();

ui->textEdit->setFocus();

}

}

}

void Widget::**on\_PopButton\_clicked**()

{

if ( ui->StackBox->isChecked()){

if(ui->listWidget->count() > 0)

{

ui->label->setText(QString::number (NewStack.Pop(), 10));

delete ui->listWidget->item(0);

}

}

if ( ui->QueueBox->isChecked()){

if(ui->listWidget\_2->count() > 0) //Queue

{

ui->label\_2->setText(QString::number (NewQueue.Pop(), 10));

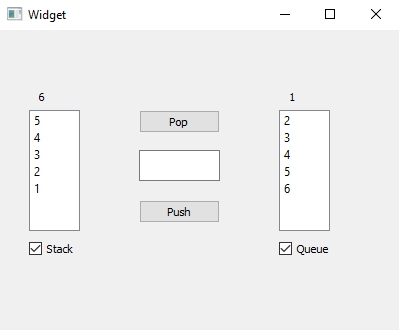
delete ui->listWidget\_2->item(0);

}

}

}

**Скриншоты программы**



Рисунок(1). Графический интерфейс программы