Отчет по GUI

Лабораторная работа №5

Студентки гр. И-220а

Выполнила:

Кадыгроб Юлия

Вариант -3

**Задание:**

Окно приложения содержит кнопку нажатия, бегунок, два списка, надпись, LCD-индикатор (QLCDNumber). Количество элементов списков и их содержимое – произвольное (при этом списки должны для наглядности быть как минимум наполовину заполнены). При нажатии на кнопку запускается/останавливается следующий процесс: по событию таймера элементы (по одному) из одного списка перемещаются в другой (и наоборот – по очереди). Элементы выбираются случайно. Бегунок отвечает за интервал таймера (в секундах). Надпись должна быть расположена рядом со списками (над) и иметь текст “Осторожно – идет расчет!”. LCD-индикатор должен отображать значение последнего элемента перемещенного из одного списка в другой.

**Код:**

**mainwindow.h**

#ifndef WINDOW\_H

#define WINDOW\_H

#include <QMainWindow>

#include <QListWidget>

#include <QListWidgetItem>

#include <QSlider>

#include <QPushButton>

#include <QLCDNumber>

#include <QLabel>

#include <QString>

QT\_BEGIN\_NAMESPACE

namespace **Ui** { class **Window**; }

QT\_END\_NAMESPACE

class **Window** : public QMainWindow

{

Q\_OBJECT

QListWidget \*list1;

QListWidget \*list2;

QLabel \*text1;

QSlider \*slide1;

QPushButton \*butt1;

QLCDNumber \*lcd1;

int \*chek;

int bottik;

int tik;

public:

**Window**(QWidget \*parent = nullptr);

~***Window***();

public slots:

void **press**();//метод нажатия кнопки

void ***timerEvent***(QTimerEvent \*event);//переопределённый метод, метод таймера

};

#endif // WINDOW\_H

Сначала мы подключаем необходимые нам библиотеки.

Объявляем объекты, переменные и слоты.  
**Mainwindow.cpp**

#include "mainwindow.h"

#include "ui\_mainwindow.h"

#include "random"

#include "QRandomGenerator"

Window::**Window**(QWidget \*parent)

: QMainWindow(*parent*)

{

resize(300, 400);

text1 = new QLabel("Осторожно-расчет!",this);

tik = 0;//переменная для столбца с которым мы работаем

bottik = 0;//переменная для контроля состояния кнопки

list1 = new QListWidget(this);

list2 = new QListWidget(this);

butt1 = new QPushButton("Start", this);

lcd1 = new QLCDNumber(10, this);

slide1 = new QSlider(Qt::Horizontal, this);//размещения бегунка в гориз. положении

slide1 -> setRange(0,5);//значения бегунка

text1 -> move(70,10);

list1 -> move(10, 50);

list1 -> resize(60,100);

for(int i = 0; i < 5; i++)//цикл заполения списка

{

QString b = QString::number(i+1);

list1 -> addItem(b);

}

list2 -> move(130, 50);

list2 -> resize(60,100);

for(int i = 0; i < 5; i++)//заполение второго списка

{

QString b = QString::number(i+1);

list2 -> addItem(b);

}

butt1 -> move(10, 190);

lcd1 -> move(10, 230);

slide1 -> move(10, 270);

connect(butt1, SIGNAL(pressed()),this, SLOT(press()));

}

void Window :: **press**()//метод активации таймера по нажатию кнопки

{

bottik ++;

if(bottik%2 == 0)

{

killTimer(\*chek);//для чётного нажатия(останавливает таймер)

}else

{

int a = slide1->value();

startTimer(a\*1000);//вызывает ***timerEvent***

}

}

void Window::***timerEvent***(QTimerEvent \*event)

{

int r = qrand()%5;

if(tik %2 == 0)

{//перебрасывет цифры из второго в первый

QListWidgetItem \*text2 = list2 ->takeItem(r);

list1 ->insertItem(6,*text2*);

lcd1 ->display(r+1);

tik++;

}else{//перебрасывает из первого во второй

QListWidgetItem \*text1 = list1 ->takeItem(r);

list2 ->insertItem(6,*text1*);

lcd1 ->display(r+1);

tik++;

}

\*chek = event->timerId();//записывает индификатор таймера

}

Window::~***Window***()

{

}

Объявляем наши объекты, прописываем сигналы (связываем объекты класса со слотами), расписываем работу слотов.

**Main.cpp**

#include "mainwindow.h"

#include <QApplication>

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

{

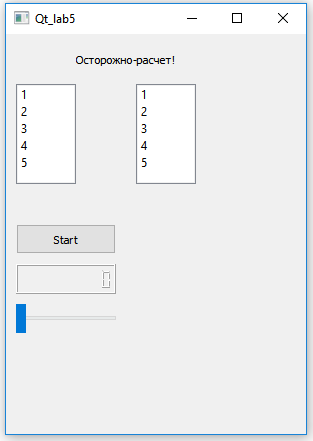
QApplication a(*argc*, *argv*);

Window w;

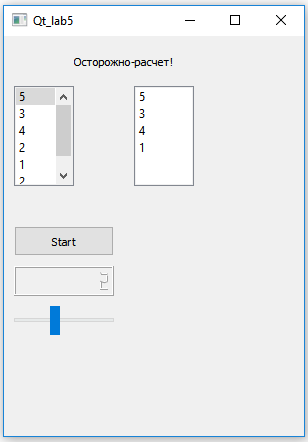
w.show();

return a.exec();

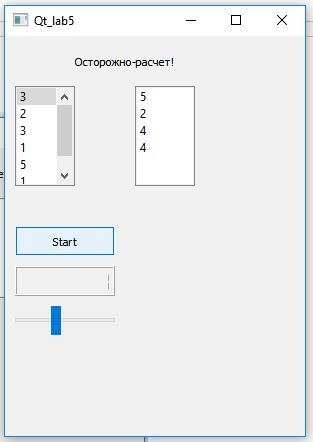
}



Начальное состояние.



Нажали на страт, выбрали бегунок (с какой скоростью будет происходить работа). Можем наблюдать, как рандомные значения перебрасываются из списка в список.



Нажали на старт еще раз, останавливается работа.