Федеральное государственное бюджетное образовательное учреждение высшего образования

«Санкт-Петербургский государственный университет телекоммуникаций им. проф. М. А. Бонч-Бруевича»

(СПбГУТ)

Кафедра Программной инженерии и вычислительной техники

**Лабораторные работы №6 по Объектно-ориентированному программированию**

**«Создание проекта с графическим интерфейсом с использованием библиотеки Qt»**

**Работу выполнил:**

Студент 2-го курса

Группа ИКПИ-73

Сударев Андрей Владимирович

**Приняла:**

Петрова Ольга Борисовна

Санкт-Петербург

2018 уч. г.

**Цель работы**

В настоящей лабораторной работе необходимо разработать приложение, обладающее графическим интерфейсом с использованием средств библиотеки Qt. В приложении должно быть предусмотрено меню и панель инструментов. Ниже сформулирован ряд задач. Предлагаемые задачи достаточно просты, что позволяет сосредоточить внимание на вопросах организации интерфейса.

В настоящей лабораторной работе предлагаются две группы задач. в первой группе задач требуется разработать одну из разновидностей калькулятора. Ко второй группе задач относятся задачи, связанные с манипуляцией с содержимым текстового однострочного редактора.

**Задача 1**

На форме находятся три компонента типа LineEdit. Два из них служат для хранения операндов, третий – фиксации результата вычислений. Обеспечить выполнение четырех обычных арифметических операций. Организовать меню и панель инструментов.

**Задача 15**

На форме имеются два компонента LineEdit, вертикально скомпонованные. Обеспечить выполнение следующих операций:

* clear(strEdtLower)
* clear(strEdtUpper).
* clearAll().
* copy(strEdtLower, strEdtUpper).
* concat(strEdtLower, strEdtUpper).
* move(strEdtLower, strEdtUpper)

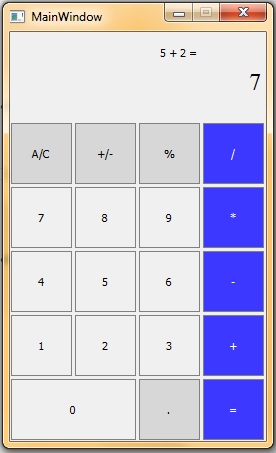
**Таблица идентификаторов**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Calculator v1 | | | | |
| Номер | Идентификатор | Метод класса | Описание метода | Назначение |
| 1 | MainWindow | digits\_numbers() | метод ввода числа | Классы |
| on\_pushButton\_point\_clicked() | метод ввода дробной части |
|  |  | operations() | метод операций |
| on\_pushButton\_clean\_clicked() | метод очистки поля ввода (кнопка AC) |
| math\_operations() | метод задания математической операции |
| on\_pushButton\_result\_clicked() | метод для вывода результата (кнопка =) |
| 2 | QMainWindow | - | - |
| - | - |
| - | - |
| - | - |
| - | - |
| Calculator v2 | | | | |
| 3 | MainWindow | on\_pushButton\_Plus\_clicked() | сложение | Классы |
| on\_pushButton\_Minus\_clicked() | вычитание |
| on\_pushButton\_Div\_clicked() | деление |
| on\_pushButton\_Mul\_clicked() | умножение |
| 4 | QMainWindow | - | - |
| - | - |
| - | - |
| TextEditor | | | | |
| 5 | MainWindow | on\_pushButton\_4\_clicked() | copy | Классы |
| on\_pushButton\_5\_clicked() | move |
| on\_pushButton\_6\_clicked() | concat |
| 6 | QMainWindow | - | - |

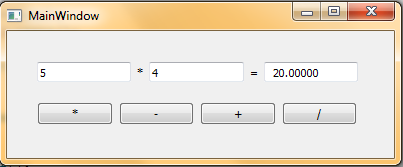
**Пример работы программы**

**Calculator**

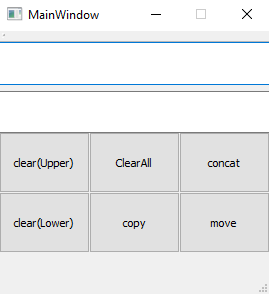
**Вариант 1**

****

**Вариант 2**

****

**TextEditor**

****

**Программа на С++**

**Calculator**

**Вариант 1**

**Main.cpp**

#include "mainwindow.h" // заголовочный файл с классом MainWindow

#include <QApplication> // класс QApplication библиотеки Qt,

// управляющего всеми окнами и виджитами

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

{

QApplication a(argc, argv);

MainWindow w; // создание виджета

w.setFixedSize(256,409);

w.show(); // показ виджета

return a.exec(); // программа работает пока не будет завершена

}

**mainwindow.h**

#ifndef MAINWINDOW\_H

#define MAINWINDOW\_H

#include <QMainWindow>

// установление класса в пространстве имени Ui

namespace **Ui** {

class **MainWindow**; // класс главного окна

}

//определение класса MainWindow - наследника QMainWindow.

class **MainWindow** : public QMainWindow

{

Q\_OBJECT

public:

explicit **MainWindow**(QWidget \*parent = nullptr); // конструктор

~***MainWindow***(); // деструктор

private:

Ui::MainWindow \*ui; // указатель на объект-построитель интерфейса

private slots:

void **digits\_numbers**(); // метод ввода числа

void **on\_pushButton\_point\_clicked**(); // метод ввода дробной части

void **operations**(); // метод операций

void **on\_pushButton\_clean\_clicked**(); // метод очистки поля ввода (кнопка AC)

void **math\_operations**(); // метод задания математической операции

void **on\_pushButton\_result\_clicked**(); // метод для вывода результата (кнопка =)

};

#endif // MAINWINDOW\_H

**mainwindow.cpp**

#include "mainwindow.h"

#include "ui\_mainwindow.h"

#include <QString>

double num\_first; //переменная для первого числа

MainWindow::**MainWindow**(QWidget \*parent) :

// Ещё до выполнения конструктора будет вызван

// конструктор родительского класса

// А поле ui будет инициализировано выражением new UI::MainWindow.

// Оператор new выделяет память под объект в куче (как malloc), затем

// вызывает конструктор для инициализации объекта.

QMainWindow(parent),

ui(new Ui::MainWindow)

{

ui->setupUi(this);

//указываем, какой слот будет выполнен при нажатии определенной кнопки

connect(ui->pushButton\_0,SIGNAL(clicked()),this, SLOT(digits\_numbers()));

connect(ui->pushButton\_1,SIGNAL(clicked()),this, SLOT(digits\_numbers()));

connect(ui->pushButton\_2,SIGNAL(clicked()),this, SLOT(digits\_numbers()));

connect(ui->pushButton\_3,SIGNAL(clicked()),this, SLOT(digits\_numbers()));

connect(ui->pushButton\_4,SIGNAL(clicked()),this, SLOT(digits\_numbers()));

connect(ui->pushButton\_5,SIGNAL(clicked()),this, SLOT(digits\_numbers()));

connect(ui->pushButton\_6,SIGNAL(clicked()),this, SLOT(digits\_numbers()));

connect(ui->pushButton\_7,SIGNAL(clicked()),this, SLOT(digits\_numbers()));

connect(ui->pushButton\_8,SIGNAL(clicked()),this, SLOT(digits\_numbers()));

connect(ui->pushButton\_9,SIGNAL(clicked()),this, SLOT(digits\_numbers()));

connect(ui->pushButton\_sign,SIGNAL(clicked()),this, SLOT(operations()));

connect(ui->pushButton\_persent,SIGNAL(clicked()),this, SLOT(operations()));

connect(ui->pushButton\_sum,SIGNAL(clicked()),this, SLOT(math\_operations()));

connect(ui->pushButton\_dif,SIGNAL(clicked()),this, SLOT(math\_operations()));

connect(ui->pushButton\_div,SIGNAL(clicked()),this, SLOT(math\_operations()));

connect(ui->pushButton\_mul,SIGNAL(clicked()),this, SLOT(math\_operations()));

//указываем, что кнопки действий тригерные

ui->pushButton\_sum->setCheckable(true);

ui->pushButton\_dif->setCheckable(true);

ui->pushButton\_mul->setCheckable(true);

ui->pushButton\_div->setCheckable(true);

ui->pushButton\_result->setCheckable(true);

}

MainWindow::~***MainWindow***()//деструктор

{

delete ui;

}

void MainWindow::**digits\_numbers**()

{

QPushButton \*button = (QPushButton \*)sender(); // указатель на нажатую кнопку

double all\_numbers;

QString new\_label;

//если число дробное

if (ui->result\_show->text().contains(".")&&button->text() == "0")

{

new\_label = ui->result\_show->text() + button->text();

}

else

{

all\_numbers = (ui->result\_show->text()+button->text()).toDouble();

new\_label = QString::number(all\_numbers, 'g', 10);

}

if(new\_label.size() < 10)// проверка на лимит

ui->result\_show->setText(new\_label);// вывод на экран числа

else

QApplication::beep();

}

void MainWindow::**on\_pushButton\_point\_clicked**()

{

if(!(ui->result\_show->text().contains('.'))){

ui->result\_show->setText(ui->result\_show->text() + ".");}

else

QApplication::beep();

}

void MainWindow::**operations**()

{

QPushButton \*button = (QPushButton \*)sender();

double all\_numbers;

QString new\_label;

if(button->text() == "+/-")

{

all\_numbers = (ui->result\_show->text()).toDouble();

all\_numbers = all\_numbers \* -1;

new\_label = QString::number(all\_numbers, 'g', 10);

ui->result\_show->setText(new\_label);

}

else

if(button->text() == "%")

{

all\_numbers = (ui->result\_show->text()).toDouble();

all\_numbers = all\_numbers \* 0.01;

new\_label = QString::number(all\_numbers, 'g', 10);

ui->result\_show->setText(new\_label);

}

}

void MainWindow::**math\_operations**()

{

QPushButton \*button = (QPushButton \*)sender();

QString number1;

num\_first = ui->result\_show->text().toDouble();

number1 = QString::number(num\_first, 'g', 10);

ui->label->setText(number1);

if(ui->pushButton\_sum->isChecked())

{

ui->label->setText(ui->label->text() + " + ");

}

else

if(ui->pushButton\_dif->isChecked())

{

ui->label->setText(ui->label->text() + " - ");

}

else

if(ui->pushButton\_div->isChecked())

{

ui->label->setText(ui->label->text() + " / ");

}

else

if(ui->pushButton\_mul->isChecked())

{

ui->label->setText(ui->label->text() + " \* ");

}

ui->result\_show->setText("");

button->setChecked(true);

}

void MainWindow::**on\_pushButton\_clean\_clicked**()//при нажатии на AC

{

ui->pushButton\_sum->setChecked(false);

ui->pushButton\_dif->setChecked(false);

ui->pushButton\_div->setChecked(false);

ui->pushButton\_mul->setChecked(false);

ui->result\_show->setText("0");

ui->label->setText(" ");

}

void MainWindow::**on\_pushButton\_result\_clicked**()

{

double labelNumber, num\_second;

QString new\_label;

num\_second = ui->result\_show->text().toDouble();

QString number2;

number2 = QString::number(num\_second, 'g', 10);

ui->label->setText(ui->label->text() + number2 + " = ");

if(ui->pushButton\_sum->isChecked())

{

labelNumber = num\_first + num\_second;

new\_label = QString::number(labelNumber, 'g', 15);

if(new\_label.size() < 15)// проверка на лимит

ui->result\_show->setText(new\_label);

else

ui->result\_show->setText("Error! Limit exceeded!");

ui->pushButton\_sum->setChecked(false);

}

else

if(ui->pushButton\_dif->isChecked())

{

labelNumber = num\_first - num\_second;

new\_label = QString::number(labelNumber, 'g', 15);

if(new\_label.size() < 15)// проверка на лимит

ui->result\_show->setText(new\_label);

else

ui->result\_show->setText("Error! Limit exceeded!");

ui->pushButton\_dif->setChecked(false);

}

else

if(ui->pushButton\_mul->isChecked())

{

labelNumber = num\_first \* num\_second;

new\_label = QString::number(labelNumber, 'g', 15);

if(new\_label.size() < 15)// проверка на лимит

ui->result\_show->setText(new\_label);

else

ui->result\_show->setText("Error! Limit exceeded!");

ui->pushButton\_mul->setChecked(false);

}

else

if(ui->pushButton\_div->isChecked())

{

if (num\_second == 0)

{

ui->result\_show->setText("Error");

}

else

{

labelNumber = num\_first / num\_second;

new\_label = QString::number(labelNumber, 'g', 15);

if(new\_label.size() < 15)// проверка на лимит

ui->result\_show->setText(new\_label);

else

ui->result\_show->setText("Error! Limit exceeded!");

ui->pushButton\_div->setChecked(false);

}

}

}

**Вариант 2**

**Main.cpp**

#include "mainwindow.h"

#include <QApplication>

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

{

QApplication a(argc, argv);

MainWindow w;

w.show();

return a.exec();

}

**mainwindow.h**

#ifndef MAINWINDOW\_H

#define MAINWINDOW\_H

#include <QMainWindow>

namespace **Ui** {

class **MainWindow**;

}

class **MainWindow** : public QMainWindow

{

Q\_OBJECT

public:

explicit **MainWindow**(QWidget \*parent = nullptr);

~***MainWindow***();

private slots:

void **on\_pushButton\_Plus\_clicked**();

void **on\_pushButton\_Minus\_clicked**();

void **on\_pushButton\_Div\_clicked**();

void **on\_pushButton\_Mul\_clicked**();

private:

Ui::MainWindow \*ui;

};

#endif // MAINWINDOW\_H

**mainwindow.cpp**

#include "mainwindow.h"

#include "ui\_mainwindow.h"

MainWindow::**MainWindow**(QWidget \*parent) :

QMainWindow(parent),

ui(new Ui::MainWindow)

{

ui->setupUi(this);

}

MainWindow::~***MainWindow***()

{

delete ui;

}

void MainWindow::**on\_pushButton\_Plus\_clicked**()

{

QString s;

double x1, x2, x3;

ui->label->setText("+");

x1=ui->lineEdit->text().toDouble();

x2=ui->lineEdit\_2->text().toDouble();

x3=x1+x2;

s.sprintf("%10.5f", x3);

ui->lineEdit\_3->setText(s);

}

void MainWindow::**on\_pushButton\_Minus\_clicked**()

{

QString s;

double x1, x2, x3;

ui->label->setText("-");

x1=ui->lineEdit->text().toDouble();

x2=ui->lineEdit\_2->text().toDouble();

x3=x1-x2;

s.sprintf("%10.5f", x3);

ui->lineEdit\_3->setText(s);

}

void MainWindow::**on\_pushButton\_Div\_clicked**()

{

QString s;

double x1, x2, x3;

x1=ui->lineEdit->text().toDouble();

x2=ui->lineEdit\_2->text().toDouble();

if (x2 == 0)

s = "division on zero";

else

{

x3=x1/x2;

s.sprintf("%10.5f", x3);

}

ui->lineEdit\_3->setText(s);}

void MainWindow::**on\_pushButton\_Mul\_clicked**()

{

QString s;

double x1, x2, x3;

ui->label->setText("\*");

x1=ui->lineEdit->text().toDouble();

x2=ui->lineEdit\_2->text().toDouble();

x3=x1\*x2;

s.sprintf("%10.5f", x3);

ui->lineEdit\_3->setText(s);

}

**TextEditor**

**Main.cpp**

#include "mainwindow.h"

#include <QApplication>

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

{

QApplication a(argc, argv);

MainWindow w;

w.show();

return a.exec();

}

**mainwindow.h**

#ifndef MAINWINDOW\_H

#define MAINWINDOW\_H

#include <QMainWindow>

// установление класса в пространстве имени Ui

namespace **Ui** {

class **MainWindow**; // класс главного окна

}

//определение класса MainWindow - наследника QMainWindow.

class **MainWindow** : public QMainWindow

{

Q\_OBJECT

public:

explicit **MainWindow**(QWidget \*parent = nullptr);//конструктор

~***MainWindow***();//деструктор

private slots:

void **on\_clearAll\_clicked**();

void **on\_clear\_up\_clicked**();

void **on\_clear\_lo\_clicked**();

void **on\_copy\_clicked**();

void **on\_concat\_clicked**();

void on\_move\_clicked();

private:

Ui::MainWindow \*ui;// указатель на объект-построитель интерфейса

};

#endif // MAINWINDOW\_H

**mainwindow.cpp**

#include "mainwindow.h"

#include "ui\_mainwindow.h"

MainWindow::**MainWindow**(QWidget \*parent) :

QMainWindow(parent),

ui(new Ui::MainWindow)

{

ui->setupUi(this);

}

MainWindow::~***MainWindow***()

{

delete ui;

}

void MainWindow::**on\_pushButton\_4\_clicked**()

{

QString str1,str2;

str1 = ui->lineEdit->text();

str2 = ui->lineEdit\_2->text();

str1 = str2;

ui->lineEdit->setText(str1);

}

void MainWindow::**on\_pushButton\_5\_clicked**()

{

QString str1,str2;

str1 = ui->lineEdit->text();

str2 = ui->lineEdit\_2->text();\

str1 = str2;

ui->lineEdit->setText(str1);

ui->lineEdit\_2->setText("");

}

void MainWindow::**on\_pushButton\_6\_clicked**()

{

QString str1,str2,str3;

str1 = ui->lineEdit->text();

str2 = ui->lineEdit\_2->text();

str3 = str2+str1;

ui->lineEdit\_2->setText(str3);

}

MainWindow::**MainWindow**(QWidget \*parent) :

QMainWindow(parent),

ui(new Ui::MainWindow)

{

ui->setupUi(this);

}

MainWindow::~***MainWindow***()

{

delete ui;

}

void MainWindow::**on\_clearAll\_clicked**()

{

ui->lineEdit\_lo->setText("");

ui->lineEdit\_up->setText("");

}

void MainWindow::**on\_clear\_up\_clicked**()

{

ui->lineEdit\_up->setText("");

}

void MainWindow::**on\_clear\_lo\_clicked**()

{

ui->lineEdit\_lo->setText("");

}

void MainWindow::**on\_copy\_clicked**()

{

QString s = ui->lineEdit\_up->text();

ui->lineEdit\_lo->setText(s);

}

void MainWindow::**on\_concat\_clicked**()

{

QString s;

s = ui->lineEdit\_lo->text() + ui->lineEdit\_up->text();

ui->lineEdit\_lo->setText(s);

}

void MainWindow::**on\_move\_clicked**()

{

QString s = ui->lineEdit\_up->text();

ui->lineEdit\_lo->setText(s);

ui->lineEdit\_up->setText("");

}