

Федеральное агентство связи

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**

**«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ТЕЛЕКОММУНИКАЦИЙ ИМ. ПРОФ. М. А. БОНЧ-БРУЕВИЧА» (СПбГУТ)**

Факультет информационных технологий и программной инженерии Кафедра: Программная инженерия. Разработка программного обеспечения и приложений искусственного интеллекта в киберфизических системах

ЛАБОРАТОРНАЯ РАБОТА №6

по дисциплине «Объектно-ориентированное программирование»

**Тема: Создание проекта с графическим
интерфейсом с использованием библиотеки Qt**

Выполнил: студент 2-го курса группы ИКПИ-42 Терещенко Максим Андреевич

Преподаватель: Петрова Ольга Борисовна

Санкт-Петербург 2025

Постановка задачи

В данной лабораторной работе необходимо разработать приложение с графическим интерфейсом с использованием библиотеки Qt. В приложении требуется реализовать:

1. **Калькулятор** с четырьмя основными арифметическими операциями (сложение, вычитание, умножение, деление) и выводом результатов в таблицу `QTableWidget` (задача 4).
 - Колонки таблицы: Номер эксперимента, Операнд 1, Операнд 2, Операция, Результат
 - Ввод данных осуществляется через компонент `QLineEdit`.
2. **Текстовый редактор** с двумя `QLineEdit`, расположенными вертикально, и следующими операциями (Задача 14):
 - `reverse(strEdtLower)` — переворот нижней строки
 - `reverse(strEdtUpper)` — переворот верхней строки
 - `reverseAll()` — переворот обеих строк
 - `swap(strEdtLower, strEdtUpper)` — обмен значениями строк
 - `concat(strEdtLower, strEdtUpper)` — объединение строк в порядке нижняя+верхняя
 - `concat(strEdtUpper, strEdtLower)` — объединение строк в порядке верхняя+нижняя

Анализ задачи

Алгоритмическая часть

Калькулятор:

1. Ввод данных в `QLineEdit`, разделённых пробелом.
2. Разделение строки на два операнда.
3. Выбор операции: сложение, вычитание, умножение, деление.
4. Выполнение операции.
5. Добавление результата в `QTableWidget` с автоматическим увеличением номера эксперимента.
6. Проверка деления на ноль для операции деления.

Текстовый редактор (Задача 14):

1. `reverse(strEdtLower)` — переворот строки `strEdtLower`.
2. `reverse(strEdtUpper)` — переворот строки `strEdtUpper`.

- 3. **reverseAll()** — одновременно переворачиваются обе строки.
- 4. **swap(strEdtLower, strEdtUpper)** — строки меняются местами.
- 5. **concat(strEdtLower, strEdtUpper)** — объединение строк в порядке нижняя+верхняя.
- 6. **concat(strEdtUpper, strEdtLower)** — объединение строк в порядке верхняя+нижняя.

Таблица идентификаторов

Таблица идентификаторов

Идентификатор	Описание
lineEdit	Приватное поле класса MainWindow, указатель на виджет QLineEdit для ввода двух чисел калькулятора.
textEdit	Приватное поле класса MainWindow, указатель на виджет QTextEdit (не используется в коде).
strEdtLower	Приватное поле класса MainWindow, указатель на виджет QLineEdit для ввода нижней строки текстового редактора.
strEdtUpper	Приватное поле класса MainWindow, указатель на виджет QLineEdit для ввода верхней строки текстового редактора.
tableWidget	Приватное поле класса MainWindow, указатель на виджет QTableWidget для отображения результатов операций калькулятора.
experimentNumber	Приватное поле класса MainWindow, целое число для нумерации экспериментов в таблице (по умолчанию 1).
MainWindow(QWidget *parent)	Конструктор класса MainWindow, инициализирует интерфейс и подключает обработчики событий.
~MainWindow()	Деструктор класса MainWindow.
on_addButton_clicked()	Слот, выполняющий операцию сложения двух чисел из lineEdit и добавляющий результат в tableWidget.

Идентификатор	Описание
on_subtractButton_clicked()	Слот, выполняющий операцию вычитания двух чисел из lineEdit и добавляющий результат в tableWidget.
on_multiplyButton_clicked()	Слот, выполняющий операцию умножения двух чисел из lineEdit и добавляющий результат в tableWidget.
on_divideButton_clicked()	Слот, выполняющий операцию деления двух чисел из lineEdit с проверкой деления на ноль и добавляющий результат в tableWidget.
on_reverseLower_clicked()	Слот, выполняющий реверс строки в strEdtLower.
on_reverseUpper_clicked()	Слот, выполняющий реверс строки в strEdtUpper.
on_reverseAll_clicked()	Слот, выполняющий реверс строк в strEdtLower и strEdtUpper одновременно.
on_swap_clicked()	Слот, обменивающий содержимое строк strEdtLower и strEdtUpper.
on_concatLowerUpper_clicked()	Слот, конкатенирующий строку из strEdtUpper к строке в strEdtLower.
on_concatUpperLower_clicked()	Слот, конкатенирующий строку из strEdtLower к строке в strEdtUpper.

Список тестов, которые проводились

Тесты калькулятора:

№	Входные данные	Действие	Ожидаемый результат
1	5 3	Сложение	Результат: 8
2	10 2	Вычитание	Результат: 8
3	4 6	Умножение	Результат: 24
4	12 4	Деление	Результат: 3

№	Входные данные	Действие	Ожидаемый результат
5	7 0	Деление	Сообщение об ошибке (деление на ноль)

Тесты текстового редактора:

№	strEdtLower	strEdtUpper	Операция	Ожидаемый результат
1	hello	world	Reverse Lower	olleh , world
2	hello	world	Reverse Upper	hello , dlrow
3	abc	xyz	Reverse All	cba , zyx
4	first	second	Swap	second , first
5	one	two	Concat Lower+Upper	onetwo , two
6	one	two	Concat Upper+Lower	one , twoone

Копии экранов выполнения

Тестирование калькулятора:

lab6_OOP

Add Subtract Multiply Divide

Calculator Input (two numbers separated by space):

7 0

	№	Operand 1	Operand 2	Operation	Result
1	1	5	3	+	8
2	2	10	2	-	8
3	3	4	6	*	24
4	4	12	4	/	3
5	5	7	1	/	7

Text Editor Operations:

Reverse Lower Reverse Upper Reverse All Swap Concat Lower+Upper Concat Upper+Lower

Тестирование работы со строками:

Тест 1-2:

Text Editor Operations:

olleh

dlrow

Reverse Lower Reverse Upper Reverse All Swap Concat Lower+Upper Concat Upper+Lower

Тест 3:

Text Editor Operations:

cba

zyx

Reverse Lower

Reverse Upper

Reverse All

Swap

Concat Lower+Upper

Concat Upper+Lower

Тест 4:

Text Editor Operations:

second

first

Reverse Lower

Reverse Upper

Reverse All

Swap

Concat Lower+Upper

Concat Upper+Lower

Тест 5:

Text Editor Operations:

onetwo

two

Reverse Lower

Reverse Upper

Reverse All

Swap

Concat Lower+Upper

Concat Upper+Lower

Тест 6:

Text Editor Operations:

one

twoone

Reverse Lower

Reverse Upper

Reverse All

Swap

Concat Lower+Upper

Concat Upper+Lower

Выводы

В ходе выполнения лабораторной работы были достигнуты следующие результаты:

1. Реализован калькулятор с поддержкой четырех арифметических операций и выводом результатов в таблицу с номерами экспериментов.
2. Реализованы все операции задачи 14
3. Проведены контрольные расчеты и тесты, результаты полностью соответствуют ожидаемым.

Приложение

mainwindow.h:


```

#ifndef MAINWINDOW_H
#define MAINWINDOW_H

#include <QMainWindow>
#include <QLineEdit>
#include <QTextEdit>
#include <QTableWidget>

class MainWindow : public QMainWindow
{
    Q_OBJECT

public:
    MainWindow(QWidget *parent = nullptr);
    ~MainWindow();

private slots:
    // Калькулятор
    void on_addButton_clicked();
    void on_subtractButton_clicked();
    void on_multiplyButton_clicked();
    void on_divideButton_clicked();

    // Текстовый редактор
    void on_reverseLower_clicked();
    void on_reverseUpper_clicked();
    void on_reverseAll_clicked();
    void on_swap_clicked();
    void on_concatLowerUpper_clicked();
    void on_concatUpperLower_clicked();

private:
    // Члены класса для виджетов
    QLineEdit *lineEdit;
    QTextEdit *textEdit;
    QLineEdit *strEdtLower;
    QLineEdit *strEdtUpper;
    QTableWidget *tableWidget;
    int experimentNumber = 1;
};

```

```
#endif // MAINWINDOW_H
```

main.cpp:

```

#ifndef MAINWINDOW_H
#define MAINWINDOW_H

#include <QMainWindow>
#include <QLineEdit>
#include <QTextEdit>
#include <QTableWidget>

class MainWindow : public QMainWindow
{
    Q_OBJECT

public:
    MainWindow(QWidget *parent = nullptr);
    ~MainWindow();

private slots:
    // Калькулятор
    void on_addButton_clicked();
    void on_subtractButton_clicked();
    void on_multiplyButton_clicked();
    void on_divideButton_clicked();

    // Текстовый редактор
    void on_reverseLower_clicked();
    void on_reverseUpper_clicked();
    void on_reverseAll_clicked();
    void on_swap_clicked();
    void on_concatLowerUpper_clicked();
    void on_concatUpperLower_clicked();

private:
    // Члены класса для виджетов
    QLineEdit *lineEdit;
    QTextEdit *textEdit;
    QLineEdit *strEdtLower;
    QLineEdit *strEdtUpper;
    QTableWidget *tableWidget;
    int experimentNumber = 1;
};

```

```
#endif // MAINWINDOW_H
```

mainwindow.cpp:

```

#include "mainwindow.h"
#include <QVBoxLayout>
#include <QHBoxLayout>
#include <QPushButton>
#include <QLabel>
#include <QToolBar>
#include <QMenuBar>
#include <QTableWidget>
#include <QStringList>
#include <algorithm>

MainWindow::MainWindow(QWidget *parent)
    : QMainWindow(parent), experimentNumber(1)
{
    QWidget *central = new QWidget(this);
    setCentralWidget(central);

    QLineEdit *lineEdit = new QLineEdit(this);
    tableWidget = new QTableWidget(this);
    tableWidget->setColumnCount(5);
    QStringList headers = {"#", "Operand 1", "Operand 2", "Operation", "Result"};
    tableWidget->setHorizontalHeaderLabels(headers);
    tableWidget->setEditTriggers(QAbstractItemView::NoEditTriggers);

    QPushButton *addButton = new QPushButton("Add", this);
    QPushButton *subButton = new QPushButton("Subtract", this);
    QPushButton *mulButton = new QPushButton("Multiply", this);
    QPushButton *divButton = new QPushButton("Divide", this);

    QToolBar *toolBar = addToolBar("Operations");
    toolBar->addWidget(addButton);
    toolBar->addWidget(subButton);
    toolBar->addWidget(mulButton);
    toolBar->addWidget(divButton);

    connect(addButton, &QPushButton::clicked, this, &MainWindow::on_addButton_clicked);
    connect(subButton, &QPushButton::clicked, this, &MainWindow::on_subtractButton_clicked);
    connect(mulButton, &QPushButton::clicked, this, &MainWindow::on_multiplyButton_clicked);
    connect(divButton, &QPushButton::clicked, this, &MainWindow::on_divideButton_clicked);

    strEdtLower = new QLineEdit(this);
    strEdtUpper = new QLineEdit(this);

```

```

QPushButton *reverseLowerBtn = new QPushButton("Reverse Lower", this);
QPushButton *reverseUpperBtn = new QPushButton("Reverse Upper", this);
QPushButton *reverseAllBtn = new QPushButton("Reverse All", this);
QPushButton *swapBtn = new QPushButton("Swap", this);
QPushButton *concatLowerUpperBtn = new QPushButton("Concat Lower+Upper", this);
QPushButton *concatUpperLowerBtn = new QPushButton("Concat Upper+Lower", this);

connect(reverseLowerBtn, &QPushButton::clicked, this, &MainWindow::on_reverseLower_clicked);
connect(reverseUpperBtn, &QPushButton::clicked, this, &MainWindow::on_reverseUpper_clicked);
connect(reverseAllBtn, &QPushButton::clicked, this, &MainWindow::on_reverseAll_clicked);
connect(swapBtn, &QPushButton::clicked, this, &MainWindow::on_swap_clicked);
connect(concatLowerUpperBtn, &QPushButton::clicked, this, &MainWindow::on_concatLowerUpper_clicked);
connect(concatUpperLowerBtn, &QPushButton::clicked, this, &MainWindow::on_concatUpperLower_clicked);

QVBoxLayout *mainLayout = new QVBoxLayout();

// Калькулятор
mainLayout->addWidget(new QLabel("Calculator Input (two numbers separated by space):"));
mainLayout->addWidget(lineEdit);
mainLayout->addWidget(tableWidget);

// Текстовый редактор
mainLayout->addWidget(new QLabel("Text Editor Operations:"));
mainLayout->addWidget(strEdtLower);
mainLayout->addWidget(strEdtUpper);

QHBoxLayout *btnLayout = new QHBoxLayout();
btnLayout->addWidget(reverseLowerBtn);
btnLayout->addWidget(reverseUpperBtn);
btnLayout->addWidget(reverseAllBtn);
btnLayout->addWidget(swapBtn);
btnLayout->addWidget(concatLowerUpperBtn);
btnLayout->addWidget(concatUpperLowerBtn);

mainLayout->addLayout(btnLayout);

central->setLayout(mainLayout);
}

MainWindow::~MainWindow() {}

// ----- Калькулятор -----
void MainWindow::on_addButton_clicked() {

```

```

QStringList operands = lineEdit->text().split(" ", Qt::SkipEmptyParts);
if (operands.size() != 2) return;

bool ok1, ok2;
double a = operands[0].toDouble(&ok1);
double b = operands[1].toDouble(&ok2);
if (!ok1 || !ok2) return;

double res = a + b;

int row = tableWidget->rowCount();
tableWidget->insertRow(row);
tableWidget->setItem(row, 0, new QTableWidgetItem(QString::number(experimentNumber++)));
tableWidget->setItem(row, 1, new QTableWidgetItem(QString::number(a)));
tableWidget->setItem(row, 2, new QTableWidgetItem(QString::number(b)));
tableWidget->setItem(row, 3, new QTableWidgetItem("+"));
tableWidget->setItem(row, 4, new QTableWidgetItem(QString::number(res)));
}

void MainWindow::on_subtractButton_clicked() {
    QStringList operands = lineEdit->text().split(" ", Qt::SkipEmptyParts);
    if (operands.size() != 2) return;

    bool ok1, ok2;
    double a = operands[0].toDouble(&ok1);
    double b = operands[1].toDouble(&ok2);
    if (!ok1 || !ok2) return;

    int row = tableWidget->rowCount();
    tableWidget->insertRow(row);
    tableWidget->setItem(row, 0, new QTableWidgetItem(QString::number(experimentNumber++)));
    tableWidget->setItem(row, 1, new QTableWidgetItem(QString::number(a)));
    tableWidget->setItem(row, 2, new QTableWidgetItem(QString::number(b)));
    tableWidget->setItem(row, 3, new QTableWidgetItem("-"));
    tableWidget->setItem(row, 4, new QTableWidgetItem(QString::number(a - b)));
}

void MainWindow::on_multiplyButton_clicked() {
    QStringList operands = lineEdit->text().split(" ", Qt::SkipEmptyParts);
    if (operands.size() != 2) return;

    bool ok1, ok2;
    double a = operands[0].toDouble(&ok1);

```

```

double b = operands[1].toDouble(&ok2);
if (!ok1 || !ok2) return;

int row = tableWidget->rowCount();
tableWidget->insertRow(row);
tableWidget->setItem(row, 0, new QTableWidgetItem(QString::number(experimentNumber++)));
tableWidget->setItem(row, 1, new QTableWidgetItem(QString::number(a)));
tableWidget->setItem(row, 2, new QTableWidgetItem(QString::number(b)));
tableWidget->setItem(row, 3, new QTableWidgetItem("*"));
tableWidget->setItem(row, 4, new QTableWidgetItem(QString::number(a * b)));
}

void MainWindow::on_divideButton_clicked() {
    QStringList operands = lineEdit->text().split(" ", Qt::SkipEmptyParts);
    if (operands.size() != 2) return;

    bool ok1, ok2;
    double a = operands[0].toDouble(&ok1);
    double b = operands[1].toDouble(&ok2);
    if (!ok1 || !ok2 || b == 0) return;

    int row = tableWidget->rowCount();
    tableWidget->insertRow(row);
    tableWidget->setItem(row, 0, new QTableWidgetItem(QString::number(experimentNumber++)));
    tableWidget->setItem(row, 1, new QTableWidgetItem(QString::number(a)));
    tableWidget->setItem(row, 2, new QTableWidgetItem(QString::number(b)));
    tableWidget->setItem(row, 3, new QTableWidgetItem("/"));
    tableWidget->setItem(row, 4, new QTableWidgetItem(QString::number(a / b)));
}

// ----- Текстовый редактор -----
void MainWindow::on_reverseLower_clicked() {
    QString text = strEdtLower->text();
    std::reverse(text.begin(), text.end());
    strEdtLower->setText(text);
}

void MainWindow::on_reverseUpper_clicked() {
    QString text = strEdtUpper->text();
    std::reverse(text.begin(), text.end());
    strEdtUpper->setText(text);
}

```



```
void MainWindow::on_reverseAll_clicked() {
    QString lower = strEdtLower->text();
    QString upper = strEdtUpper->text();
    std::reverse(lower.begin(), lower.end());
    std::reverse(upper.begin(), upper.end());
    strEdtLower->setText(lower);
    strEdtUpper->setText(upper);
}
```

```
void MainWindow::on_swap_clicked() {
    QString lower = strEdtLower->text();
    QString upper = strEdtUpper->text();
    strEdtLower->setText(upper);
    strEdtUpper->setText(lower);
}
```

```
void MainWindow::on_concatLowerUpper_clicked() {
    QString lower = strEdtLower->text();
    QString upper = strEdtUpper->text();
    strEdtLower->setText(lower + upper);
}
```

```
void MainWindow::on_concatUpperLower_clicked() {
    QString lower = strEdtLower->text();
    QString upper = strEdtUpper->text();
    strEdtUpper->setText(upper + lower);
}
```