

Учреждение образования  
«Брестский государственный технический университет»  
Кафедра ИИТ

Лабораторная работа №2  
По дисциплине: «ООТПиСП»  
Тема: «Основы Qt»

Выполнил:  
Студент 3 курса  
Группы ПО-7  
Комиссаров А.Е.  
Проверила:  
Хацкевич М.В.

**Цель:** развить понимание структуры приложения на Qt и получить опыт использования стандартной документации Qt.

**Задание:** разработка законченного оконного приложения — текстового редактора, работающего с несколькими документами с возможностями открытия и сохранения, отмены действий, работы с буфером обмена и другими.

**Ход выполнения задания:**

### Файл main.cpp

```
#include "texteditor.h"

#include <QApplication>
#include <QIcon>

int main(int argc, char *argv[])
{
    QApplication a(argc, argv);
    a.setApplicationName("Text Editor");
    a.setApplicationVersion("0.1");
    a.setOrganizationName("ExampleSoft");
    a.setOrganizationDomain("example.com");
    TextEditor w;
    w.show();
    return a.exec();
}
```

### Файл texteditor.h

```
#ifndef TEXTEDITOR_H
#define TEXTEDITOR_H

#include <QMainWindow>
#include <QString>

QT_BEGIN_NAMESPACE
namespace Ui { class TextEditor; }
QT_END_NAMESPACE

class TextEditor : public QMainWindow
{
    Q_OBJECT

public:
    TextEditor(const QString &fileName = QString(), QWidget *parent = nullptr);
    ~TextEditor();

private slots:
    bool saveFile();

    bool saveFileAs();

    void on_actionNew_triggered();

    void on_actionClose_triggered();
}
```

```

void on_actionExit_triggered();

void documentModified();

void on_textEdit_textChanged();

void on_actionSelectFont_triggered();

void on_actionAboutQt_triggered();

void on_actionCut_triggered();

void on_actionCopy_triggered();

void on_actionPaste_triggered();

void on_actionUndo_triggered();

void on_actionRedo_triggered();

void on_textEdit_copyAvailable(bool b);

void on_textEdit_redoAvailable(bool b);

void on_textEdit_undoAvailable(bool b);

void on_actionAbout_triggered();

void on_actionOpen_triggered();

void on_actionSave_triggered();

void on_actionSaveAs_triggered();

private:
    void loadFile(const QString &fileName);
    void setFileName(const QString &fileName);
    Ui::TextEditor *ui;
    QString m_fileName;
protected:
    void closeEvent(QCloseEvent *e);
};
#endif // TEXTEDITOR_H

```

## Файл texteditor.cpp

```

#include "texteditor.h"
#include "../ui_texteditor.h"
#include <QTextEdit>
#include <QMessageBox>
#include <QCloseEvent>
#include <QIcon>
#include <QFontDialog>
#include <QFont>
#include <QFile>
#include <QString>
#include <QFileInfo>
#include <QDir>
#include <QFileDialog>
#include <QTextStream>
#include <iostream>

```

```

#include <stdio.h>

TextEditor::TextEditor(const QString &fileName, QWidget *parent)
    : QMainWindow(parent)
    , ui(new Ui::TextEditor)
{
    ui->setupUi(this);
    this->setAttribute(Qt::WA_DeleteOnClose);
    this->setWindowTitle("Text Editor[*]");
    ui->actionCopy->setEnabled(false);
    ui->actionCut->setEnabled(false);
    ui->actionRedo->setEnabled(false);
    ui->actionUndo->setEnabled(false);
    loadFile(fileName);
}

void TextEditor::loadFile(const QString &fileName){
    if (fileName.isEmpty()){
        setFileName(QString());
        return;
    }
    QString fileContents;
    QFile file(fileName);
    if (!file.open(QIODevice::ReadOnly | QIODevice::Text))
        return;

    while (!file.atEnd()) {
        QByteArray line = file.readLine();
        fileContents = fileContents + line + "\n";
    }
    ui->textEdit->setText(fileContents);
    file.close();
    setFileName(fileName);
    this->setWindowModified(false);
}

void TextEditor::setFileName(const QString &fileName){
    this->m_fileName = fileName;
    this->setWindowTitle(QString("%1[*] - %2")
        .arg(m_fileName.isNull()?"untitled":QFileInfo(m_fileName).fileName())
        .arg(QApplication::applicationName()));
}

bool TextEditor::saveFileAs(){
    QString fileName = QFileDialog::getSaveFileName(this, "Save document",
m_fileName.isNull()?QDir::currentPath():m_fileName, "Text documents (*.txt)");
    if(fileName.isNull()){return false;
    }else{setFileName(fileName);
        return saveFile();}
}

bool TextEditor::saveFile(){
    if(this->m_fileName.isNull()){
        return saveFileAs();
    }else{
        QFile file(m_fileName);
        if (!file.open(QIODevice::WriteOnly | QIODevice::Text)){
            QMessageBox msgBox;
            msgBox.setIcon(QMessageBox::Warning);
            msgBox.setText("File could not be saved properly, try again.");
            msgBox.exec();
            setFileName(QString());
            return false;
        }
    }
}

```

```

        }else{
            QTextStream out(&file);
            out << ui->textEdit->toPlainText();
            file.close();
            this->setWindowModified(false);
            return true;
        }
    }
}

TextEditor::~TextEditor()
{
    delete ui;
}

void TextEditor::on_actionNew_triggered()
{
    TextEditor *new_editor = new TextEditor;
    new_editor->show();
}

void TextEditor::on_actionClose_triggered()
{
    this->close();
}

void TextEditor::on_actionExit_triggered()
{
    qApp->closeAllWindows();
}

void TextEditor::documentModified(){
    this->setWindowModified(true);
}

void TextEditor::closeEvent(QCloseEvent *e){
    if (this->isWindowModified() == true){
        QMessageBox msgBox;
        msgBox.setIcon(QMessageBox::Warning);
        msgBox.setText("The document has been modified.");
        msgBox.setInformativeText("This document has been modified, do you want to save your changes?");
        msgBox.setStandardButtons(QMessageBox::Yes | QMessageBox::No | QMessageBox::Cancel);
        msgBox.setDefaultButton(QMessageBox::Cancel);
        int ret = msgBox.exec();
        switch(ret){
            case QMessageBox::Yes :
                saveFile();
                break;
            case QMessageBox::No :
                e->accept();
                break;
            case QMessageBox::Cancel :
                e->ignore();
                break;
            default :
                e->ignore();
                break;
        }
    }else{
        e->accept();
    }
}

```

```
void TextEditor::on_textEdit_textChanged()
{
    this->documentModified();
}

void TextEditor::on_actionSelectFont_triggered()
{
    this->setFont(QFontDialog::getFont(0, this->font()));
}

void TextEditor::on_actionAboutQt_triggered()
{
    qApp->aboutQt();
}

void TextEditor::on_actionCut_triggered()
{
    ui->textEdit->cut();
}

void TextEditor::on_actionCopy_triggered()
{
    ui->textEdit->copy();
}

void TextEditor::on_actionPaste_triggered()
{
    ui->textEdit->paste();
}

void TextEditor::on_actionUndo_triggered()
{
    ui->textEdit->undo();
}

void TextEditor::on_actionRedo_triggered()
{
    ui->textEdit->redo();
}

void TextEditor::on_textEdit_copyAvailable(bool b)
{
    ui->actionCopy->setEnabled(b);
    ui->actionCut->setEnabled(b);
}

void TextEditor::on_textEdit_redoAvailable(bool b)
{
    ui->actionRedo->setEnabled(b);
}
void TextEditor::on_textEdit_undoAvailable(bool b)
{
    ui->actionUndo->setEnabled(b);
}
```

```

void TextEditor::on_actionAbout_triggered()
{
    QMessageBox msgBox;
    msgBox.setText("Laboratornaya rabota 2 Qt.");
    msgBox.exec();
}

void TextEditor::on_actionOpen_triggered()
{
    QString fileName = QFileDialog::getOpenFileName(this, "Open document", QDir::currentPath(), "Text
documents (*.txt)");
    if (fileName.isNull()){return;}
    if (!fileName.isNull() && !this->isWindowModified()){
        loadFile(fileName);
    } else {
        TextEditor *new_editor = new TextEditor(fileName);
        new_editor->show();
    }
}

void TextEditor::on_actionSave_triggered()
{
    saveFile();
}

void TextEditor::on_actionSaveAs_triggered()
{
    saveFileAs();
}

```

## Файл texteditor.qrc

```

<!DOCTYPE RCC>
<RCC version="1.0">
    <qresource prefix="icons">
        <file>images/copy.png</file>
        <file>images/cut.png</file>
        <file>images/new.png</file>
        <file>images/open.png</file>
        <file>images/paste.png</file>
        <file>images/save.png</file>
    </qresource>
</RCC>

```

## Файл texteditor.ui

```

<?xml version="1.0" encoding="UTF-8"?>
<ui version="4.0">
    <class>TextEditor</class>
    <widget class="QMainWindow" name="TextEditor">
        <property name="geometry">
            <rect>
                <x>0</x>
                <y>0</y>
                <width>800</width>
                <height>600</height>
            </rect>
        </property>
        <property name="minimumSize">

```

```

<size>
  <width>800</width>
  <height>600</height>
</size>
</property>
<property name="windowTitle">
  <string>TextEditor</string>
</property>
<property name="windowIcon">
  <iconset>
    <normalon>:/icons/images/new.png</normalon>
  </iconset>
</property>
<widget class="QWidget" name="centralwidget">
  <widget class="QTextEdit" name="textEdit">
    <property name="geometry">
      <rect>
        <x>0</x>
        <y>0</y>
        <width>800</width>
        <height>550</height>
      </rect>
    </property>
  </widget>
</widget>
<widget class="QMenuBar" name="menubar">
  <property name="geometry">
    <rect>
      <x>0</x>
      <y>0</y>
      <width>800</width>
      <height>21</height>
    </rect>
  </property>
  <widget class="QMenu" name="menuFile">
    <property name="title">
      <string>File</string>
    </property>
    <addaction name="actionOpen"/>
    <addaction name="actionNew"/>
    <addaction name="separator"/>
    <addaction name="actionSave"/>
    <addaction name="actionSaveAs"/>
    <addaction name="actionClose"/>
    <addaction name="actionExit"/>
  </widget>
  <widget class="QMenu" name="menuView">
    <property name="title">
      <string>View</string>
    </property>
    <addaction name="actionSelectFont"/>
  </widget>
  <widget class="QMenu" name="menuAbout">
    <property name="title">
      <string>About...</string>
    </property>
    <addaction name="actionAbout"/>
    <addaction name="actionAboutQt"/>
  </widget>
  <widget class="QMenu" name="menuEdit">
    <property name="title">
      <string>Edit</string>
    </property>

```



```
<addaction name="actionOpen"/>
<addaction name="actionUndo"/>
<addaction name="actionRedo"/>
<addaction name="actionCut"/>
<addaction name="actionCopy"/>
<addaction name="actionPaste"/>
</widget>
<addaction name="menuFile"/>
<addaction name="menuEdit"/>
<addaction name="menuView"/>
<addaction name="menuAbout"/>
</widget>
<widget class="QStatusBar" name="statusbar"/>
<action name="actionNew">
<property name="icon">
<iconset resource="texteditor.qrc">
<normaloff>:/icons/images/new.png</normaloff>:/icons/images/new.png</iconset>
</property>
<property name="text">
<string>New File</string>
</property>
<property name="shortcut">
<string>Ctrl+N</string>
</property>
</action>
<action name="actionClose">
<property name="text">
<string>Close</string>
</property>
</action>
<action name="actionExit">
<property name="text">
<string>Exit</string>
</property>
</action>
<action name="actionSelectFont">
<property name="text">
<string>Select Font...</string>
</property>
<property name="toolTip">
<string>Select the display font</string>
</property>
</action>
<action name="actionAbout">
<property name="text">
<string>About</string>
</property>
</action>
<action name="actionAboutQt">
<property name="text">
<string>About Qt</string>
</property>
</action>
<action name="actionCut">
<property name="icon">
<iconset resource="texteditor.qrc">
<normaloff>:/icons/images/cut.png</normaloff>:/icons/images/cut.png</iconset>
</property>
<property name="text">
<string>Cut</string>
</property>
<property name="shortcut">
<string>Ctrl+X</string>
```

```
</property>
</action>
<action name="actionCopy">
  <property name="icon">
    <iconset resource="texteditor.qrc">
      <normaloff>:/icons/images/copy.png</normaloff>:/icons/images/copy.png</iconset>
    </property>
    <property name="text">
      <string>Copy</string>
    </property>
    <property name="shortcut">
      <string>Ctrl+C</string>
    </property>
  </action>
  <action name="actionPaste">
    <property name="icon">
      <iconset resource="texteditor.qrc">
        <normaloff>:/icons/images/paste.png</normaloff>:/icons/images/paste.png</iconset>
      </property>
      <property name="text">
        <string>Paste</string>
      </property>
      <property name="shortcut">
        <string>Ctrl+V</string>
      </property>
    </action>
    <action name="actionUndo">
      <property name="text">
        <string>Undo</string>
      </property>
      <property name="shortcut">
        <string>Ctrl+Z</string>
      </property>
    </action>
    <action name="actionRedo">
      <property name="text">
        <string>Redo</string>
      </property>
      <property name="shortcut">
        <string>Ctrl+Y</string>
      </property>
    </action>
    <action name="actionOpen">
      <property name="icon">
        <iconset resource="texteditor.qrc">
          <normaloff>:/icons/images/open.png</normaloff>:/icons/images/open.png</iconset>
        </property>
        <property name="text">
          <string>Open</string>
        </property>
        <property name="toolTip">
          <string>Open a document</string>
        </property>
        <property name="shortcut">
          <string>Ctrl+O</string>
        </property>
      </action>
      <action name="actionSave">
        <property name="icon">
          <iconset resource="texteditor.qrc">
            <normaloff>:/icons/images/save.png</normaloff>:/icons/images/save.png</iconset>
          </property>
          <property name="text">
```

```

    <string>Save</string>
  </property>
  <property name="shortcut">
    <string>Ctrl+S</string>
  </property>
</action>
<action name="actionSaveAs">
  <property name="text">
    <string>Save as</string>
  </property>
  <property name="shortcut">
    <string>Ctrl+Shift+S</string>
  </property>
</action>
</widget>
<resources>
  <include location="texteditor.qrc"/>
</resources>
<connections/>
</ui>

```

## Результат работы программы:

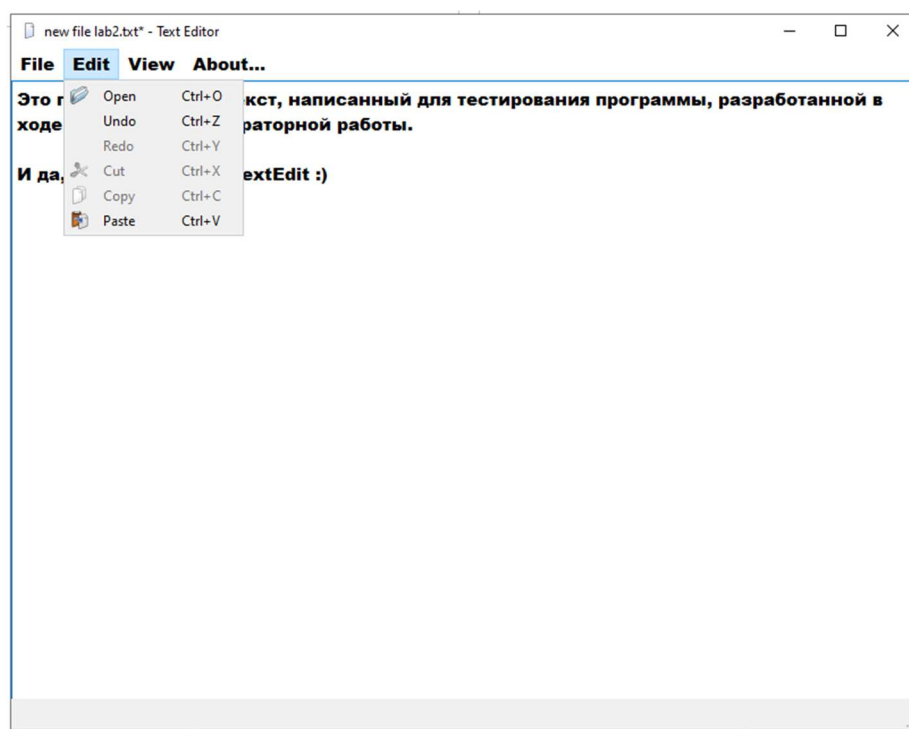


Рис. 1 – Результат работы программы

**Вывод:** я развил понимание структуры приложения на Qt и получил опыт использования стандартной документации Qt.