

Содержание

[1](#_gjdgxs) Постановка задачи 3

[1.1](#_30j0zll) Цель 3

[1.2](#_1fob9te) Задачи 3

[1.3](#_3znysh7) Задание 3

[1.4](#_2et92p0) Общее ограничение на лабораторную работу 3

[1.5](#_tyjcwt) Требования к отчёту 3

[2](#_3dy6vkm) Диаграммы 4

[3](#_1t3h5sf) Исходный код 5

[4](#_2s8eyo1) ЛИСТИНГ ВЫполнения программы 8

[5](#_17dp8vu) Вывод 9

1. Постановка задачи
   1. Цель

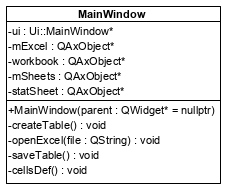
Познакомится с основами работы c ActiveX приложениями в Qt.

* 1. Задачи

1. изучить основы работы с QAxObject;
2. изучить основы работы с Excel из других приложений.
   1. Задание
3. создать QTableView;
4. все данные из QTableWidget которые введёт пользователь должны быть экспортированы в файл xlsx.
   1. Общее ограничение на лабораторную работу

* использовать разбитие на файлы основной программы (в main.cpp только функция main.cpp);
* документировать весь код используя аннотации Doxygen.
  1. Требования к отчёту
* цели лабораторной работы;
* задачи лабораторной работы;
* выполняемое задание;
* диаграмма(ы) классов;
* диаграммы деятельности (для методов и/или функций больше 10 строк кода (исключая форматирования и пустые строки));
* исходный код программы с комментариями (2 пробела вместо 4, 10pt, моноширинный шрифт);
* листинг выполнения программы;
* исходный код в виде файлов для запуска на произвольном ПК.

1. Диаграммы

**

*Рисунок 1 – Диаграмма классов*

1. Исходный код

В листингах 2.1-2.5 представлен исходный код программы.

*Листинг 2.1 – Код программы*

/\*\*

\* @file main.cpp

\* @author Oreshnikov Egor

\*/

#include "mainwindow.h"

#include <QApplication>

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

{

QApplication a(argc, argv);

MainWindow w;

w.show();

return a.exec();

}

*Листинг 2.2 – Заголовочный файл mainwindow.h*

/\*\*

\* @file mainwindow.h

\* @brief window with table

\* @author Oreshnikov Egor

\*/

#ifndef MAINWINDOW\_H

#define MAINWINDOW\_H

#include <QMainWindow>

#include <ActiveQt/QAxObject>

QT\_BEGIN\_NAMESPACE

namespace Ui { class MainWindow; }

QT\_END\_NAMESPACE

class MainWindow : public QMainWindow

{

Q\_OBJECT

public:

MainWindow(QWidget \*parent = nullptr);

~MainWindow();

private:

Ui::MainWindow \*ui;

QAxObject \*mExcel;

QAxObject \*workbook;

QAxObject \*mSheets;

QAxObject \*StatSheet;

private slots:

void createTable();

void openExcel(QString file);

void saveTable();

void cellsDef();

};

#endif // MAINWINDOW\_H

*Листинг 2.3 – Заголовочный файл mainwindow.cpp*

/\*\*

\* @file mainwindow.cpp

\* @brief window with table

\* @author Oreshnikov Egor

\*/

#include "mainwindow.h"

#include "ui\_mainwindow.h"

#include <QFileDialog>

#include <QUrl>

#include <QMessageBox>

MainWindow::MainWindow(QWidget \*parent)

: QMainWindow(parent)

, ui(new Ui::MainWindow)

{

ui->setupUi(this);

connect(ui->create, &QPushButton::clicked,

this, &MainWindow::createTable);

connect(ui->save, &QPushButton::clicked,

this, &MainWindow::saveTable);

}

MainWindow::~MainWindow()

{

delete ui;

}

void MainWindow::createTable()

{

ui->tableWidget->setColumnCount(ui->col\_edit->displayText().toInt());

ui->tableWidget->setRowCount(ui->row\_edit->displayText().toInt());

ui->tableWidget->resizeRowsToContents();

ui->tableWidget->resizeColumnsToContents();

}

void MainWindow::openExcel(QString file)

{

mExcel = new QAxObject("Excel.Application", this);

QAxObject \*workbooks = mExcel->querySubObject( "Workbooks" );

workbook = workbooks->querySubObject( "Open(const QString&)", QUrl::fromLocalFile(file) );

mSheets = workbook->querySubObject( "Sheets" );

int count = mSheets->dynamicCall("Count()").toInt();

QString name;

for (int i=1; i<=count; i++){

QAxObject\* sheet = mSheets->querySubObject( "Item( int )", i );

name = sheet->dynamicCall("Name()").toString();

}

StatSheet = mSheets->querySubObject( "Item(const QVariant&)", QVariant(name) );

StatSheet->setProperty("Name", "My table");

}

void MainWindow::saveTable()

{

QString file = QFileDialog::getOpenFileName(this, tr("Open files"),

QString(),

tr("Excel Files (\*.xlsx \*.xls)"));

this->openExcel(file);

this->cellsDef();

workbook->dynamicCall("Save()");

workbook->dynamicCall("Close()");

mExcel->dynamicCall("Quit()");

QMessageBox::information(NULL,QObject::tr("Information"), tr("Таблица сохранена успешно"));

}

void MainWindow::cellsDef()

{

QAxObject\* Cell1 = StatSheet->querySubObject("Cells(QVariant&,QVariant&)", 1, 1);

QAxObject\* Cell2 = StatSheet->querySubObject("Cells(QVariant&,QVariant&)", ui->row\_edit->text().toInt(), ui->col\_edit->text().toInt());

QAxObject\* range = StatSheet->querySubObject("Range(const QVariant&,const QVariant&)", Cell1->asVariant(), Cell2->asVariant() );

QList<QVariant> cellsList;

QList<QVariant> rowsList;

for (int i = 0; i < ui->row\_edit->text().toInt(); i++)

{

cellsList.clear();

for (int j = 0; j < ui->col\_edit->text().toInt(); j++){

QVariant myData;

QModelIndex myIndex;

myIndex = ui->tableWidget->model()->index(i, j, QModelIndex());

myData = ui->tableWidget->model()->data(myIndex, Qt::DisplayRole);

cellsList << myData;

}

rowsList << QVariant(cellsList);

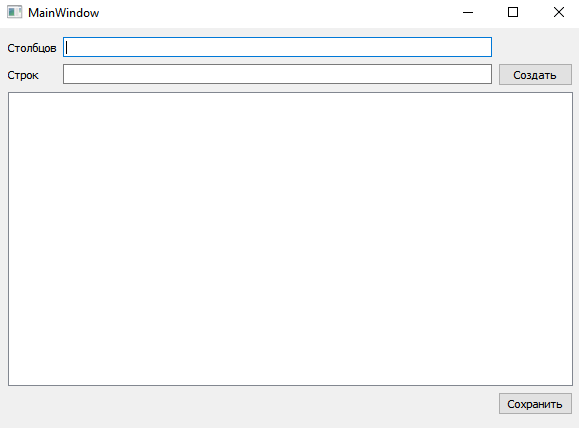
}

range->setProperty("Value", QVariant(rowsList) );

}

1. ЛИСТИНГ ВЫполнения программы

На рисунке 2 представлен интерфейс работы программы.



*Рисунок 2 – Интерфейс программы*

1. Вывод

В ходе выполнения лабораторной работы были изучены принципы работы с приложением MS Excel через Qt. Была написана программа, которая создает таблицу и записывает ее в файл Excel.