Министерство образования и науки Российской Федерации

Севастопольский государственный университет

Кафедра ИС

Отчет

по лабораторной работе № 8

«Организация графического интерфейса.

Библиотека Qt»

Выполнил

ст. гр. ИТб22

Серегин А.В.

Проверил:

Тимофеев И.С.

Севастополь

2015

1. Цель работы

Приобретение практических навыков в написании объектно-ориентированных программ с использованием библиотеки Qt. Освоение особенностей отладки объектно-ориентированных программ с графическим интерфейсом.

2. Варианты заданий

Вариант 5:

Написать программу реализующую записную книжку. Описать методы добавления, удаления и сортировки данных записной книги.

3. Текст программы

MainWindow.h

#ifndef MAINWINDOW\_H

#define MAINWINDOW\_H

#include <QDialog>

#include <QWidget>

#include <QTableWidget>

#include <QLineEdit>

#include <QPushButton>

#include "contactedit.h"

class MyWindow : public QDialog

{

Q\_OBJECT

public:

MyWindow(QWidget \*parent = 0);

~MyWindow();

private:

QTableWidget\* mainTable\_;

QLineEdit\* contact\_;

QLineEdit\* phone\_;

QPushButton\* exit\_;

QPushButton\* add\_;

QPushButton\* edit\_;

QPushButton\* del\_;

ContactEdit\* editWidget\_;

quint64 contacts\_;

private slots:

void addTextChanged();

void addContact();

void editBegin();

void editEnd();

void changeContact();

void selectedContact();

void deleteContact();

};

#endif // MAINWINDOW\_H

Contactedit.h

#ifndef CONTACTEDIT\_H

#define CONTACTEDIT\_H

#include <QString>

#include <QWidget>

#include <QDialog>

#include <QLineEdit>

#include <QPushButton>

class ContactEdit : public QDialog

{

Q\_OBJECT

public:

ContactEdit(const QString& \_name, const QString& phone, const int& \_row, QWidget \*parent = 0);

QString getName() const;

QString getPhone() const;

inline int getRow() const;

~ContactEdit();

private:

int row\_;

QLineEdit\* name\_;

QLineEdit\* phone\_;

QPushButton\* ok\_;

QPushButton\* cancle\_;

private slots:

void addTextChanged();

void ok();

signals:

void okClicked();

};

#endif // CONTACTEDIT\_H

MainWindow.cpp

#include "mainwindow.h"

#include "contactedit.h"

#include <QDebug>

#include <QString>

#include <QList>

#include <QVBoxLayout>

#include <QHBoxLayout>

#include <QSlider>

#include <QLineEdit>

#include <QSpacerItem>

#include <QHeaderView>

#include <QLabel>

#include <QLineEdit>

MyWindow::MyWindow(QWidget \*parent)

: QDialog(parent)

{

contacts\_ = 0;

editWidget\_ = nullptr;

this->setMinimumHeight(350);

this->setMinimumWidth(500);

QLabel\* header = new QLabel("Contact / phone");

contact\_ = new QLineEdit;

contact\_->setMaximumWidth(150);

phone\_ = new QLineEdit;

phone\_->setMaximumWidth(150);

phone\_->setInputMask("+0-000-000-00-00");

add\_ = new QPushButton("&Добавить");

add\_->setEnabled(false);

edit\_ = new QPushButton("Изменить");

edit\_->setEnabled(false);

exit\_ = new QPushButton("&Выход");

del\_ = new QPushButton("Удалить");

del\_->setEnabled(false);

//QLabel\* lbl = new QLabel(tr("%1").arg(phone\_->text().length()));

mainTable\_ = new QTableWidget(contacts\_,2);

mainTable\_->horizontalHeader()->setStretchLastSection(true);

mainTable\_->setHorizontalHeaderItem(0,new QTableWidgetItem("Контакт"));

mainTable\_->setHorizontalHeaderItem(1,new QTableWidgetItem("Телефон"));

mainTable\_->setSortingEnabled(true);

mainTable\_->setSelectionBehavior(QAbstractItemView::SelectRows);

QVBoxLayout\* leftLay = new QVBoxLayout;

leftLay->addWidget(header);

leftLay->addWidget(contact\_);

leftLay->addWidget(phone\_);

leftLay->addWidget(add\_);

leftLay->addWidget(edit\_);

leftLay->addWidget(del\_);

leftLay->addStretch();

//leftLay->addWidget(lbl);

leftLay->addWidget(exit\_);

QHBoxLayout\* mainLay = new QHBoxLayout;

mainLay->addLayout(leftLay);

mainLay->addWidget(mainTable\_);

QObject::connect(exit\_,SIGNAL(clicked()),this,SLOT(close()));

QObject::connect(add\_, SIGNAL(clicked()),this,SLOT(addContact()));

QObject::connect(del\_,SIGNAL(clicked(bool)),this,SLOT(deleteContact()));

QObject::connect(contact\_,SIGNAL(textChanged(QString)),this, SLOT(addTextChanged()));

QObject::connect(phone\_,SIGNAL(textChanged(QString)),this, SLOT(addTextChanged()));

QObject::connect(mainTable\_,SIGNAL(itemSelectionChanged()),this, SLOT(selectedContact()));

QObject::connect(edit\_,SIGNAL(clicked(bool)),this, SLOT(editBegin()));

this->setLayout(mainLay);

}

void MyWindow::addTextChanged()

{

if(contact\_->text().length() && (phone\_->text().length() == 16))

add\_->setEnabled(true);

else

add\_->setEnabled(false);

}

void MyWindow::addContact()

{

QTableWidgetItem\* contact = new QTableWidgetItem(contact\_->text());

QTableWidgetItem\* phone = new QTableWidgetItem(phone\_->text());

contact->setFlags(contact->flags() ^ Qt::ItemIsEditable);

phone->setFlags(phone->flags() ^ Qt::ItemIsEditable);

mainTable\_->setRowCount(contacts\_ + 1);

mainTable\_->setSortingEnabled(false);

mainTable\_->setItem(contacts\_,0,contact);

mainTable\_->setItem(contacts\_,1,phone);

mainTable\_->setSortingEnabled(true);

contact\_->clear();

phone\_->clear();

contacts\_++;

}

void MyWindow::editBegin()

{

this->setEnabled(false);

mainTable\_->setSortingEnabled(false);

int row = mainTable\_->currentRow();

editWidget\_ = new ContactEdit(mainTable\_->item(row,0)->text(),

mainTable\_->item(row,1)->text(), row);

QObject::connect(editWidget\_,SIGNAL(rejected()), this, SLOT(editEnd()));

QObject::connect(editWidget\_,SIGNAL(okClicked()), this, SLOT(changeContact()));

editWidget\_ -> show();

editWidget\_ -> exec();

}

void MyWindow::changeContact()

{

QTableWidgetItem\* contact = new QTableWidgetItem(editWidget\_->getName());

QTableWidgetItem\* phone = new QTableWidgetItem(editWidget\_->getPhone());

contact->setFlags(contact->flags() ^ Qt::ItemIsEditable);

phone->setFlags(phone->flags() ^ Qt::ItemIsEditable);

int row = mainTable\_->currentRow();

mainTable\_->setItem(row, 0, contact);

mainTable\_->setItem(row, 1, phone);

editWidget\_ ->close();

}

void MyWindow::editEnd()

{

mainTable\_->setSortingEnabled(true);

delete editWidget\_;

this->setEnabled(true);

}

MyWindow::~MyWindow()

{

delete mainTable\_;

delete contact\_;

delete phone\_;

delete exit\_;

delete add\_;

delete edit\_;

delete editWidget\_;

delete del\_;

}

void MyWindow::selectedContact()

{

if(!mainTable\_ ->selectedItems().count())

{

edit\_->setEnabled(false);

del\_->setEnabled(false);

return;

}

del\_->setEnabled(true);

if (mainTable\_->selectedItems().count() < 3)

edit\_->setEnabled(true);

else

edit\_->setEnabled(false);

}

void MyWindow::deleteContact()

{

while(mainTable\_->selectedRanges().count())

{

mainTable\_->removeRow(mainTable\_->selectedRanges().at(0).bottomRow());

contacts\_--;

}

del\_->setEnabled(false);

edit\_->setEnabled(false);

}

Contactedit.cpp

#include "contactedit.h"

#include <QVBoxLayout>

#include <QHBoxLayout>

ContactEdit::ContactEdit(const QString& \_name, const QString& \_phone, const int& \_row, QWidget\* parent)

: QDialog(parent)

{

row\_ = \_row;

name\_ = new QLineEdit(\_name);

phone\_ = new QLineEdit(\_phone);

phone\_->setInputMask("+0-000-000-00-00");

ok\_ = new QPushButton("Ок");

ok\_->setEnabled(false);

cancle\_ = new QPushButton("Отмена");

QHBoxLayout\* botLay = new QHBoxLayout;

botLay->addWidget(ok\_);

botLay->addWidget(cancle\_);

QVBoxLayout\* mainLay = new QVBoxLayout;

mainLay->addWidget(name\_);

mainLay->addWidget(phone\_);

mainLay->addLayout(botLay);

this->setLayout(mainLay);

QObject::connect(ok\_, SIGNAL(clicked(bool)), this, SLOT(ok()));

QObject::connect(name\_, SIGNAL(textChanged(QString)), this, SLOT(addTextChanged()));

QObject::connect(phone\_, SIGNAL(textChanged(QString)), this, SLOT(addTextChanged()));

QObject::connect(cancle\_, SIGNAL(clicked()), this, SLOT(close()));

}

QString ContactEdit::getName() const

{

return name\_->text();

}

QString ContactEdit::getPhone() const

{

return phone\_->text();

}

int ContactEdit::getRow() const

{

return row\_;

}

ContactEdit::~ContactEdit()

{

delete name\_;

delete phone\_;

delete ok\_;

delete cancle\_;

}

void ContactEdit::addTextChanged()

{

if(name\_->text().length() && phone\_->text().length() == 16)

ok\_->setEnabled(true);

else

ok\_->setEnabled(false);

}

void ContactEdit::ok()

{

emit okClicked();

}

main.cpp

#include "mainwindow.h"

#include <QApplication>

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

{

QApplication a(argc, argv);

MyWindow\* window = new MyWindow;

window->show();

return a.exec();

}

4. Тестовые примеры.

На рисунке ниже приведен результат выполнения программы.



Рисунок 1 – Главное меню.



Рисунок 2 – Добавление нового элемента.

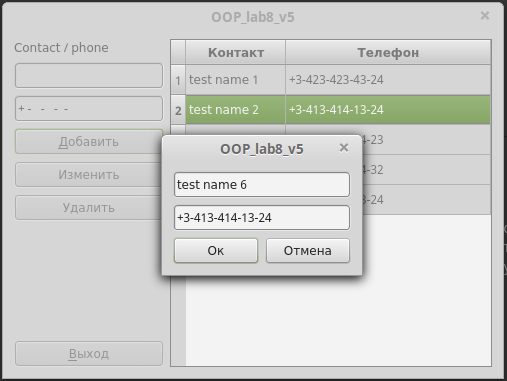


Рисунок 3 – Изменение контакта.



Рисунок 4 – Удаление 4-го контакта.

Выводы

В ходе лабораторной работы были приобретены навыки написания объектно-ориентированных программ с использованием фрамворка Qt. Был реализован графический интерфейс программы «Записная книга».