**Защита лабораторной работы**

#include "mainwindow.h"

#include "ui\_mainwindow.h"

#include "secondwindow.h"

#include <QMessageBox>

#include "QStandardItemModel"

#include "QStandardItem"

#include <QApplication>

#include <QtSql>

#include <QDebug>

#include <QPrinter>

#include <QPrintDialog>

#include <QPrinterInfo>

QStandardItemModel \*model = new QStandardItemModel;

QStandardItem \*item;

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

: QMainWindow(*parent*)

, ui(new Ui::MainWindow)

{

ui->setupUi(this);

QStringList horizontalHeader;

horizontalHeader.append("Фамилия");

horizontalHeader.append("Имя");

horizontalHeader.append("Отчество");

horizontalHeader.append("Год рождения");

horizontalHeader.append("Пол");

horizontalHeader.append("Гражданство РФ");

model->setHorizontalHeaderLabels(horizontalHeader);

ui->tableView->*setModel*(*model*);

ui->tableView->verticalHeader()->hide();

ui->tableView->resizeRowsToContents();

ui->tableView->resizeColumnsToContents();

//Выделение строки ui->tableView->selectRow(0);

//bd

QSqlDatabase sdb = QSqlDatabase::addDatabase("QSQLITE");

sdb.setDatabaseName("db\_name.sqlite");

if (!sdb.open()) {

qDebug() << sdb.lastError().text();

}

QSqlQuery query;

query.exec("CREATE TABLE mytable ( id INTEGER PRIMARY KEY AUTOINCREMENT NULL, familia text, name text, otch text, gr text, pol text, graj text); ");

//query.exec("INSERT INTO mytable (id, familia, name, otch, gr, pol, graj) VALUES (NULL, '1234', '123', '123', '123', true, true); ");

query.exec("SELECT \* FROM mytable");

while (query.next()) {

QString textfrombd = query.value(1).toString();

int i = ui->tableView->model()->*rowCount*();

item = new QStandardItem(query.value(0).toString());

model->setItem(i, 0, *item*);

item = new QStandardItem(query.value(1).toString());

model->setItem(i, 1, *item*);

item = new QStandardItem(query.value(2).toString());

model->setItem(i, 2, *item*);

item = new QStandardItem(query.value(3).toString());

model->setItem(i, 3, *item*);

item = new QStandardItem(query.value(4).toString());

model->setItem(i, 4, *item*);

item = new QStandardItem(query.value(5).toString());

model->setItem(i, 5, *item*);

ui->tableView->*setModel*(*model*);

}

sdb.close();

}

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

{

delete ui;

}

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

{

secondwindow window;

window.setModal(true);

if (window.*exec*())

{

int i = ui->tableView->model()->*rowCount*();

item = new QStandardItem(QString(window.value()[0]));

model->setItem(i, 0, *item*);

item = new QStandardItem(QString(window.value()[1]));

model->setItem(i, 1, *item*);

item = new QStandardItem(QString(window.value()[2]));

model->setItem(i, 2, *item*);

item = new QStandardItem(QString(window.value()[3]));

model->setItem(i, 3, *item*);

item = new QStandardItem(QString(window.value()[4]));

model->setItem(i, 4, *item*);

item = new QStandardItem(QString(window.value()[5]));

model->setItem(i, 5, *item*);

ui->tableView->*setModel*(*model*);

}

}

void MainWindow::**on\_pushButton\_3\_clicked**()

{

secondwindow window;

window.setModal(true);

QString familia = ui->tableView->model()->*data*(ui->tableView->model()->*index*(ui->tableView->currentIndex().row(),0)).toString();

QString name = ui->tableView->model()->*data*(ui->tableView->model()->*index*(ui->tableView->currentIndex().row(),1)).toString();

QString otch = ui->tableView->model()->*data*(ui->tableView->model()->*index*(ui->tableView->currentIndex().row(),2)).toString();

QString gr = ui->tableView->model()->*data*(ui->tableView->model()->*index*(ui->tableView->currentIndex().row(),3)).toString();

bool pol;

bool graj;

if (ui->tableView->model()->*data*(ui->tableView->model()->*index*(ui->tableView->currentIndex().row(),4)).toString()=="М"){

pol=true;

}

else{

pol=false;

}

if (ui->tableView->model()->*data*(ui->tableView->model()->*index*(ui->tableView->currentIndex().row(),5)).toString()=="Да"){

graj = true;

}

else{

graj = false;

}

window.setchanges(familia, name, otch, gr, pol, graj);

if (window.*exec*())

{

int i = ui->tableView->currentIndex().row();

item = new QStandardItem(QString(window.value()[0]));

model->setItem(i, 0, *item*);

item = new QStandardItem(QString(window.value()[1]));

model->setItem(i, 1, *item*);

item = new QStandardItem(QString(window.value()[2]));

model->setItem(i, 2, *item*);

item = new QStandardItem(QString(window.value()[3]));

model->setItem(i, 3, *item*);

item = new QStandardItem(QString(window.value()[4]));

model->setItem(i, 4, *item*);

item = new QStandardItem(QString(window.value()[5]));

model->setItem(i, 5, *item*);

ui->tableView->*setModel*(*model*);

}

}

void MainWindow::**on\_pushButton\_2\_clicked**()

{

ui->tableView->model()->removeRow(ui->tableView->currentIndex().row());

}

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

{

secondwindow window;

window.setModal(true);

if (window.*exec*())

{

int i = ui->tableView->currentIndex().row()+1;

item = new QStandardItem(QString(window.value()[0]));

model->insertRow(i, *item*);

item = new QStandardItem(QString(window.value()[1]));

model->setItem(i, 1, *item*);

item = new QStandardItem(QString(window.value()[2]));

model->setItem(i, 2, *item*);

item = new QStandardItem(QString(window.value()[3]));

model->setItem(i, 3, *item*);

item = new QStandardItem(QString(window.value()[4]));

model->setItem(i, 4, *item*);

item = new QStandardItem(QString(window.value()[5]));

model->setItem(i, 5, *item*);

ui->tableView->*setModel*(*model*);

}

}

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

{

QSqlDatabase sdb = QSqlDatabase::addDatabase("QSQLITE");

sdb.setDatabaseName("db\_name.sqlite");

if (!sdb.open()) {

qDebug() << sdb.lastError().text();

}

QSqlQuery query;

int count = ui->tableView->model()->*rowCount*();

for (int i = 0; i < count; i++){

QString familia = ui->tableView->model()->*data*(ui->tableView->model()->*index*(i,0)).toString();

QString name = ui->tableView->model()->*data*(ui->tableView->model()->*index*(i,1)).toString();

QString otch = ui->tableView->model()->*data*(ui->tableView->model()->*index*(i,2)).toString();

QString gr = ui->tableView->model()->*data*(ui->tableView->model()->*index*(i,3)).toString();

QString pol = ui->tableView->model()->*data*(ui->tableView->model()->*index*(i,4)).toString();

QString graj = ui->tableView->model()->*data*(ui->tableView->model()->*index*(i,5)).toString();

query.exec("INSERT INTO mytable (id, familia, name, otch, gr, pol, graj) VALUES (NULL, '"+familia+"', '"+name+"', '"+otch+"', '"+gr+"', '"+pol+"', '"+graj+"'); ");

}

sdb.close();

//quit

QApplication::quit();

}

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

{

QPrinter \* printer = new QPrinter();

QPrintDialog printDialog(*printer*, this);

if (printDialog.*exec*() != QDialog::Accepted) {

return;

}

QWebEnginePage \* page = new QWebEnginePage;

page->setHtml("<html><body>Привет<body/></html>");

connect(

page,

&QWebEnginePage::loadFinished,

[page, printer] (bool ok) {

if (!ok) {

qDebug() << "Загрузка документа провалилась.";

delete page;

delete printer;

return;

}

page->print(printer, [page, printer](bool ok) {

if (ok) {

qDebug() << "Документ напечатан.";

}

else {

qDebug() << "Печать документа провалилась.";

}

delete page;

delete printer;

});

}

);

}

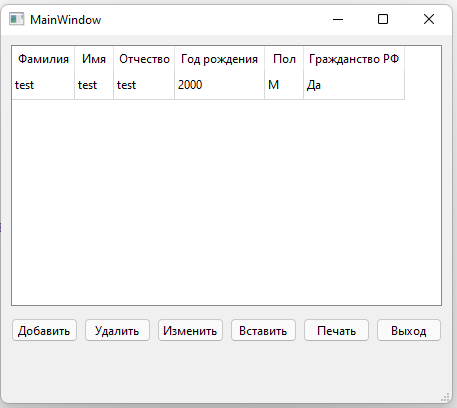


Рис 1. Результат выполнения

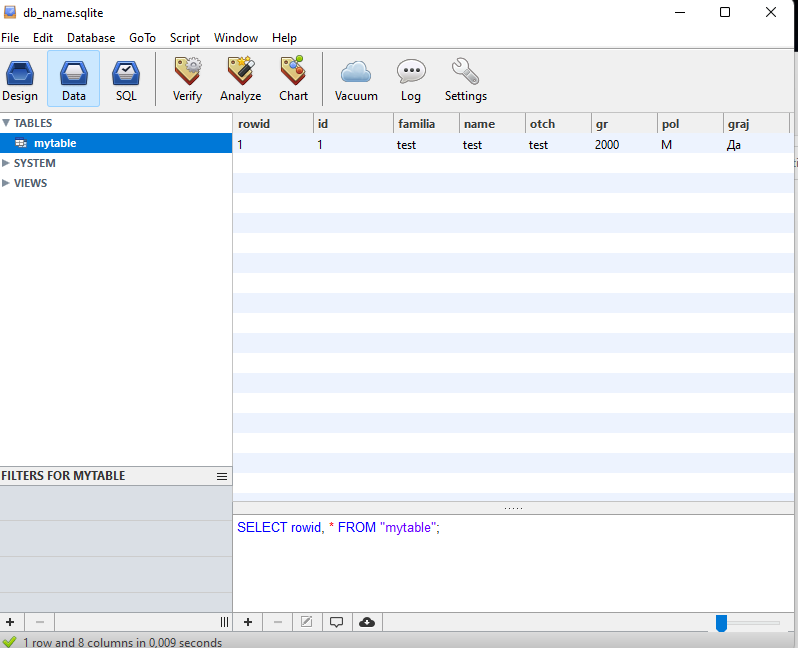


Рис 2. Результат записи в БД