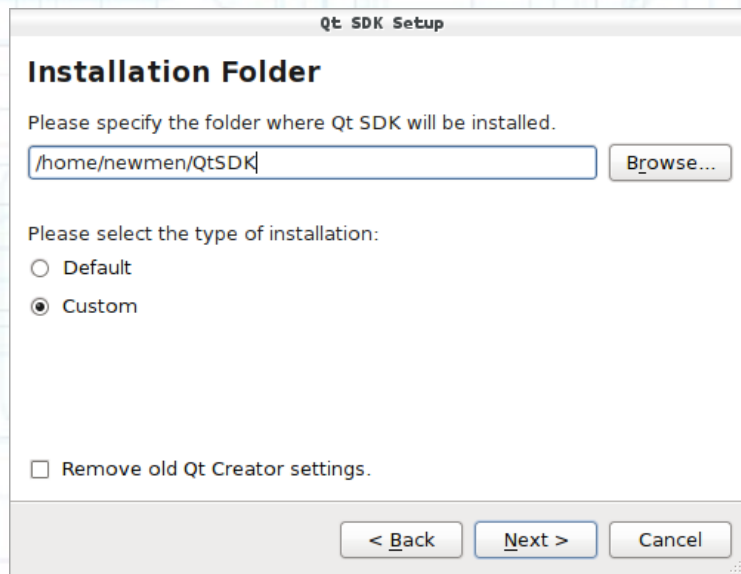
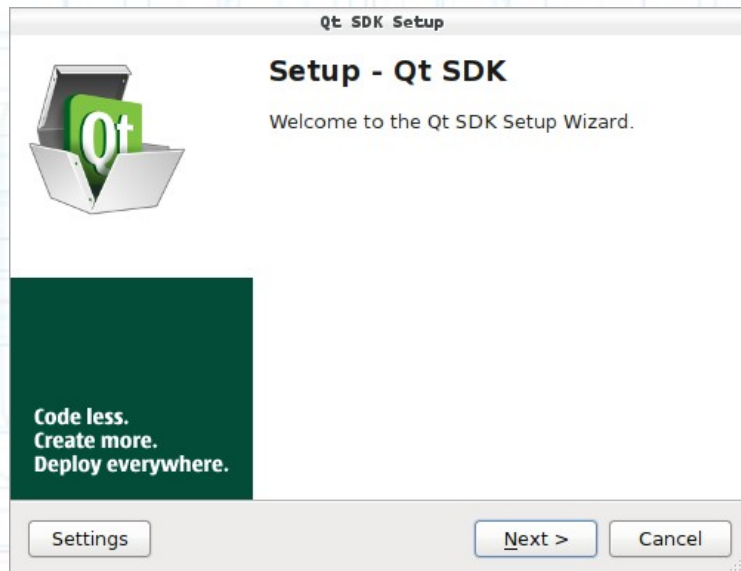


Введение в Qt



Установка

<http://qt.nokia.com/downloads/>



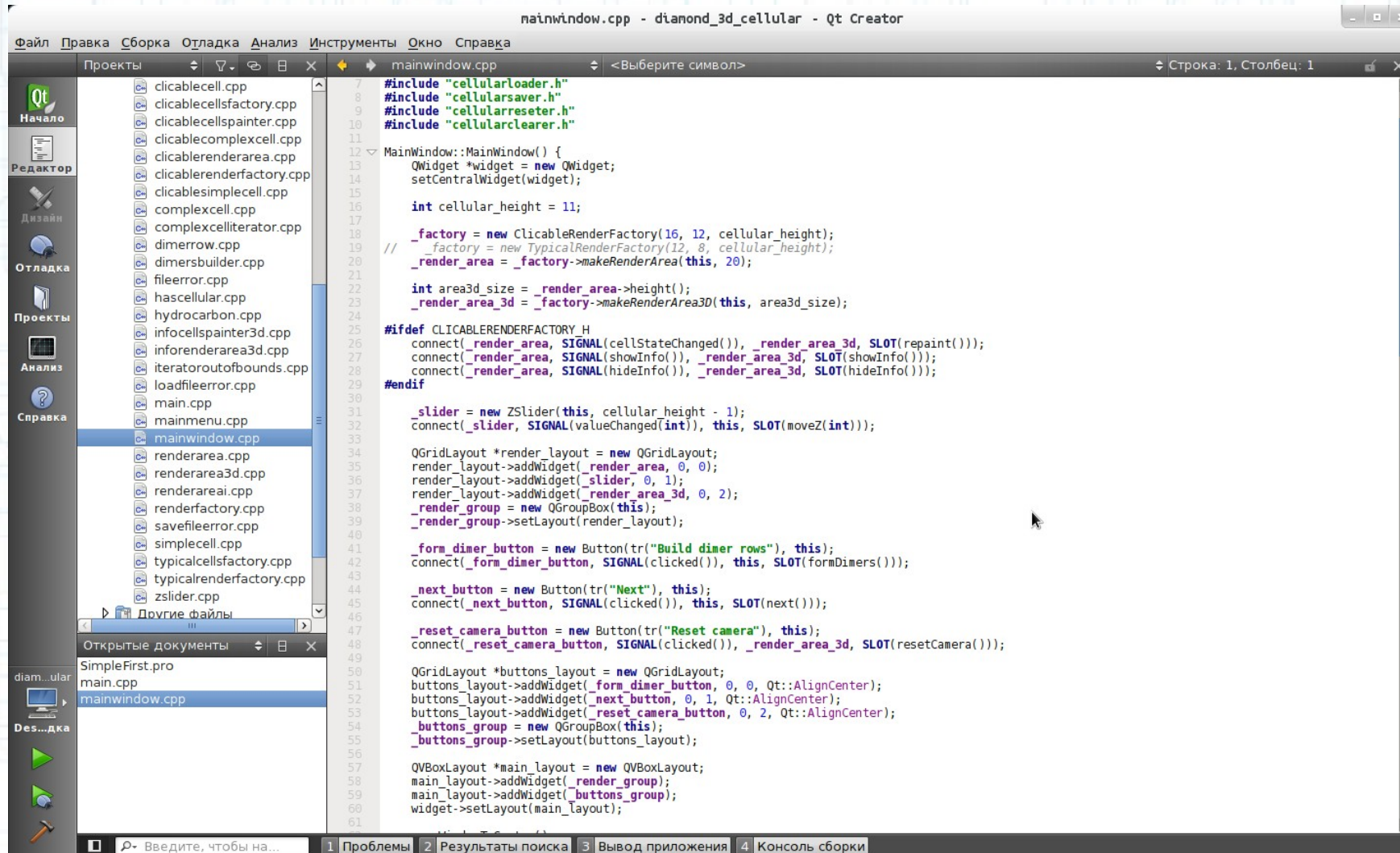
Убираете галочки

- Harmattan
- Symbian
- Mobility
- Simulator
- Device

Установливаете

- Qt Designer

Qt Creator



The screenshot displays the Qt Creator IDE interface. The title bar reads "mainwindow.cpp - diamond_3d_cellular - Qt Creator". The menu bar includes "Файл", "Правка", "Сборка", "Отладка", "Анализ", "Инструменты", "Окно", and "Справка". The left sidebar contains icons for "Начало", "Редактор", "Дизайн", "Отладка", "Проекты", "Анализ", and "Справка". The "Проекты" pane shows a list of source files, with "mainwindow.cpp" selected. The main editor window shows the following C++ code:

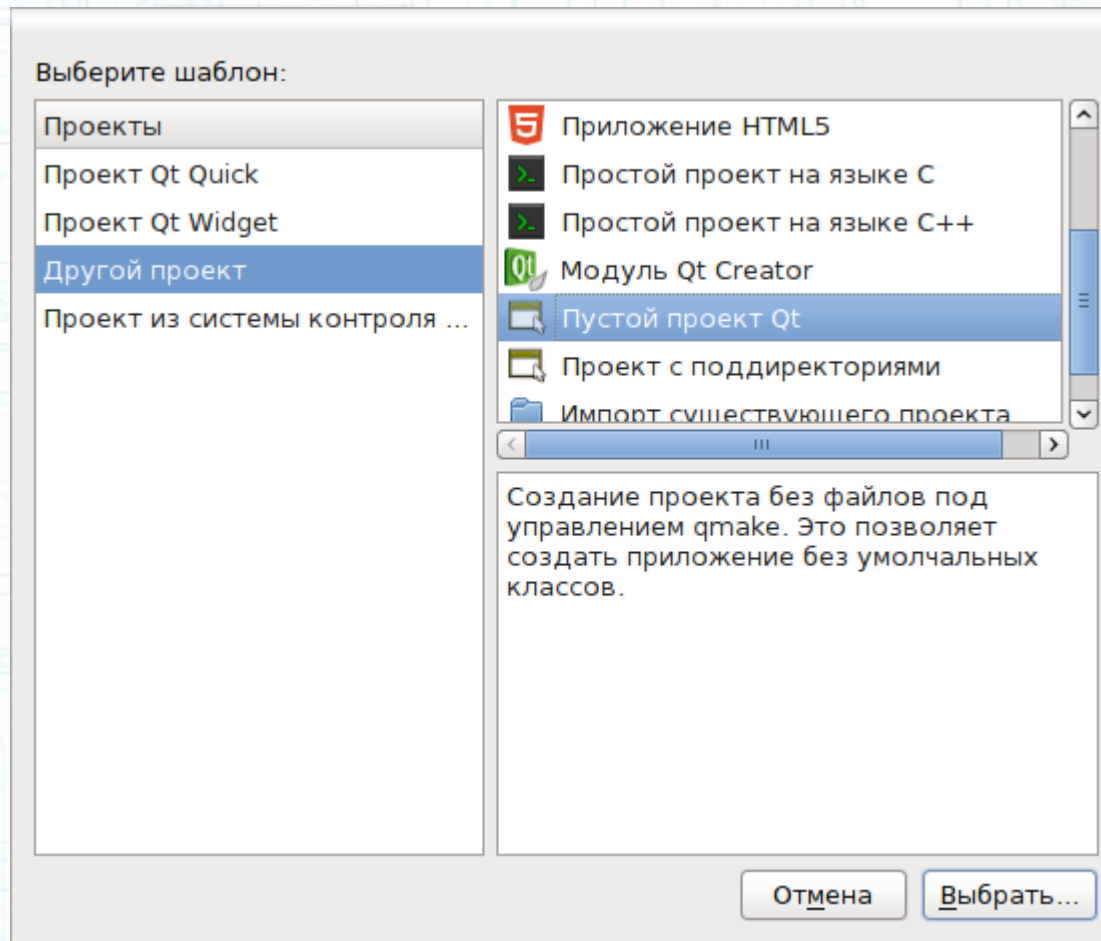
```
7 #include "cellularloader.h"
8 #include "cellularsaver.h"
9 #include "cellularreseter.h"
10 #include "cellularclearer.h"
11
12 MainWindow::MainWindow() {
13     QWidget *widget = new QWidget;
14     setCentralWidget(widget);
15
16     int cellular_height = 11;
17
18     _factory = new ClicableRenderFactory(16, 12, cellular_height);
19     // _factory = new TypicalRenderFactory(12, 8, cellular_height);
20     _render_area = _factory->makeRenderArea(this, 20);
21
22     int area3d_size = _render_area->height();
23     _render_area_3d = _factory->makeRenderArea3D(this, area3d_size);
24
25 #ifndef CLICABLERENDERFACTORY_H
26     connect(_render_area, SIGNAL(cellStateChanged()), _render_area_3d, SLOT(repaint()));
27     connect(_render_area, SIGNAL(showInfo()), _render_area_3d, SLOT(showInfo()));
28     connect(_render_area, SIGNAL(hideInfo()), _render_area_3d, SLOT(hideInfo()));
29 #endif
30
31     _slider = new ZSlider(this, cellular_height - 1);
32     connect(_slider, SIGNAL(valueChanged(int)), this, SLOT(moveZ(int)));
33
34     QGridLayout *render_layout = new QGridLayout;
35     render_layout->addWidget(_render_area, 0, 0);
36     render_layout->addWidget(_slider, 0, 1);
37     render_layout->addWidget(_render_area_3d, 0, 2);
38     _render_group = new QGroupBox(this);
39     _render_group->setLayout(render_layout);
40
41     _form_dimer_button = new Button(tr("Build dimer rows"), this);
42     connect(_form_dimer_button, SIGNAL(clicked()), this, SLOT(formDimers()));
43
44     _next_button = new Button(tr("Next"), this);
45     connect(_next_button, SIGNAL(clicked()), this, SLOT(next()));
46
47     _reset_camera_button = new Button(tr("Reset camera"), this);
48     connect(_reset_camera_button, SIGNAL(clicked()), _render_area_3d, SLOT(resetCamera()));
49
50     QGridLayout *buttons_layout = new QGridLayout;
51     buttons_layout->addWidget(_form_dimer_button, 0, 0, Qt::AlignCenter);
52     buttons_layout->addWidget(_next_button, 0, 1, Qt::AlignCenter);
53     buttons_layout->addWidget(_reset_camera_button, 0, 2, Qt::AlignCenter);
54     _buttons_group = new QGroupBox(this);
55     _buttons_group->setLayout(buttons_layout);
56
57     QVBoxLayout *main_layout = new QVBoxLayout;
58     main_layout->addWidget(_render_group);
59     main_layout->addWidget(_buttons_group);
60     widget->setLayout(main_layout);
61
```

The status bar at the bottom shows "1 Проблемы", "2 Результаты поиска", "3 Вывод приложения", and "4 Консоль сборки".

Сегодня будет рассмотрено

- Создание простого приложения
- Наследование от QWidget
- Главное окно
- Дизайнер форм
- Рисование на QWidget

Создание простого приложения



Создание простого приложения

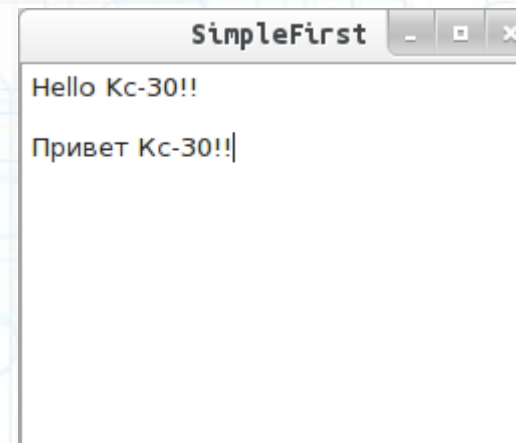
```
// main.cpp

#include <QApplication>
#include <QTextEdit>

int main(int argc, char **argv) {
    QApplication app(argc, argv);

    QTextEdit textEdit;
    textEdit.show();

    return app.exec();
}
```



Создание простого приложения (с кнопкой)

```
#include <QtGui>

int main(int argc, char **argv) {
    QApplication app(argc, argv);

    QTextEdit textEdit;
    QPushButton qButton("Close window");

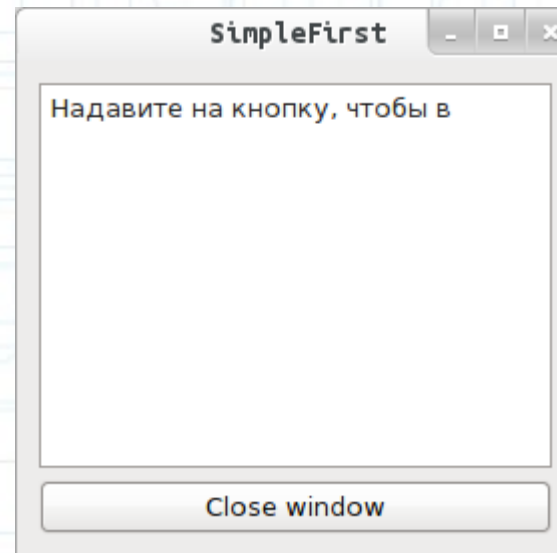
    QObject::connect(&qButton, SIGNAL(clicked()), &app, SLOT(quit()));

    QVBoxLayout layout;
    layout.addWidget(&textEdit);
    layout.addWidget(&qButton);

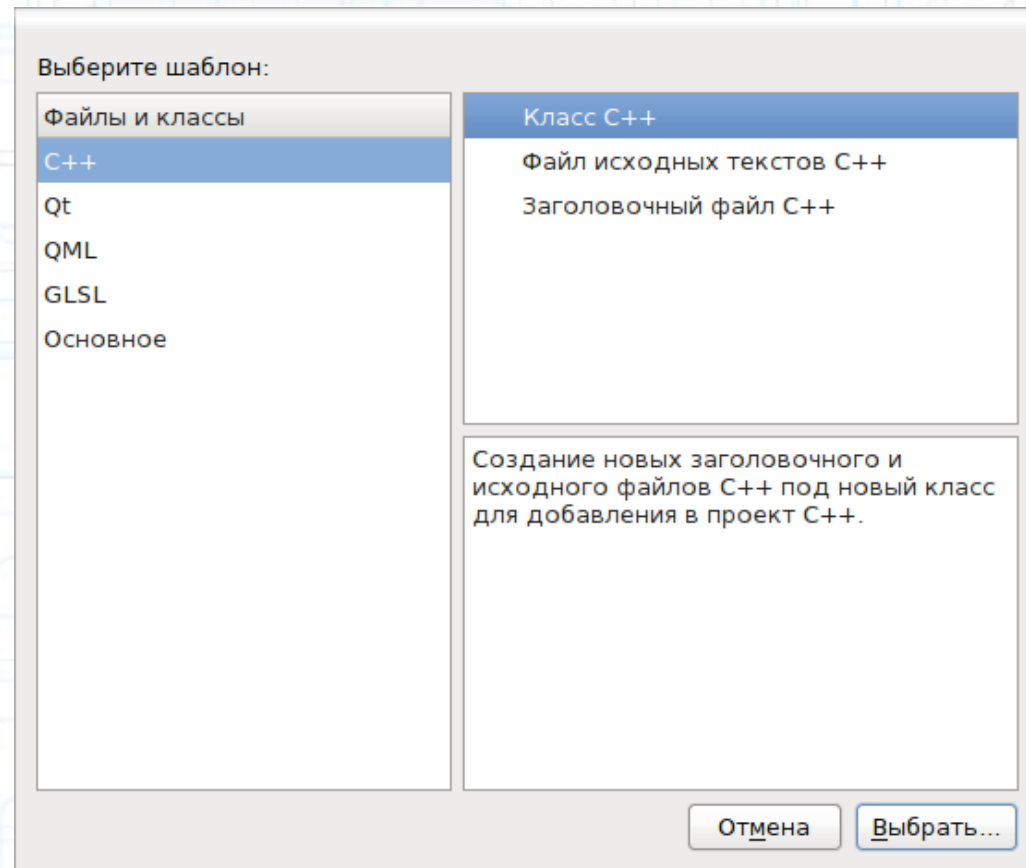
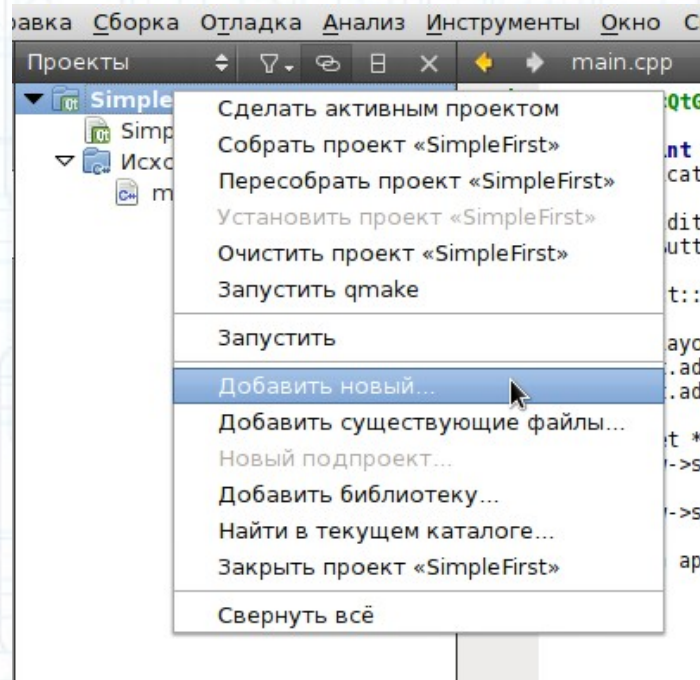
    QWidget *window = new QWidget;
    window->setLayout(&layout);

    window->show();

    return app.exec();
}
```



Наследование от QWidget



Наследование от QWidget

Введите имя класса
Названия файла исходных текстов и заголовочного файла будут получены из имени класса

⇒ Подробнее
Итог

Имя класса:

Базовый класс:

Тип класса:

Заголовочный файл:

Файл исходников:

Путь:

⇒ Подробнее
Итог

Управление проектом

Добавить в проект:

Добавить под контроль версий:

Добавляемые файлы
/home/newmen/c++/Qt/SimpleFirst:
notepad.h
notepad.cpp

Наследование от QWidget

```
// notepad.h

#ifndef NOTEPAD_H
#define NOTEPAD_H

#include <QtGui>

class Notepad : public QWidget
{
    Q_OBJECT
public:
    Notepad(QApplication *app);
    ~Notepad();

private slots:
    void quit();

private:
    QApplication *_app;

    QTextEdit *_textEdit;
    QPushButton *_qButton;
};

#endif // NOTEPAD_H
```

Наследование от QWidget

```
// notepad.cpp

#include "notepad.h"

Notepad::Notepad(QApplication *app) : _app(app) {
    _textEdit = new QTextEdit;
    _qButton = new QPushButton("Close window");

    connect(_qButton, SIGNAL(clicked()), this, SLOT(quit()));

    QVBoxLayout *layout = new QVBoxLayout;
    layout->addWidget(_textEdit);
    layout->addWidget(_qButton);

    setLayout(layout);

    setWindowTitle("Simple Notepad");
}

Notepad::~Notepad() {
    delete _textEdit;
    delete _qButton;
}

void Notepad::quit() {
    _app->quit();
}
```


Наследование от QWidget

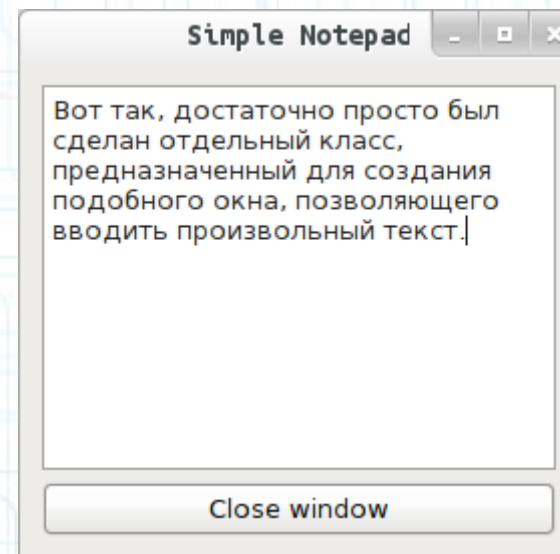
```
// main.cpp

#include <QApplication>
#include "notepad.h"

int main(int argc, char **argv) {
    QApplication app(argc, argv);

    Notepad notepad(&app);
    notepad.show();

    return app.exec();
}
```



Главное окно

```
#ifndef NOTEPAD_H
#define NOTEPAD_H

#include <QtGui>

class Notepad : public QMainWindow
{
    Q_OBJECT
public:
    Notepad(QApplication *app);
    ~Notepad();

private slots:
    void open();
    void save();
    void quit();

private:
    QApplication *_app;
    QTextEdit *_textEdit;

    QAction *_openAction;
    QAction *_saveAction;
    QAction *_quitAction;
    QMenu *_fileMenu;
};

#endif // NOTEPAD_H
```


Главное окно

```
#include "notepad.h"

Notepad::Notepad(QApplication *app) : _app(app) {
    _openAction = new QAction("&Open", this);
    _saveAction = new QAction("&Save", this);
    _quitAction = new QAction("E&xit", this);

    connect(_openAction, SIGNAL(triggered()), this, SLOT(open()));
    connect(_saveAction, SIGNAL(triggered()), this, SLOT(save()));
    connect(_quitAction, SIGNAL(triggered()), this, SLOT(quit()));

    _fileMenu = menuBar()->addMenu("&File");
    _fileMenu->addAction(_openAction);
    _fileMenu->addAction(_saveAction);
    _fileMenu->addSeparator();
    _fileMenu->addAction(_quitAction);

    _textEdit = new QTextEdit;
    setCentralWidget(_textEdit);
    setWindowTitle("Notepad");
}

Notepad::~~Notepad() {
    delete _textEdit;
    delete _openAction;
    delete _saveAction;
    delete _quitAction;
}
```


Главное окно

```
void Notepad::open() {
    QString fileName = QFileDialog::getOpenFileName(this, "Open File",
        "", "Text Files (*.txt);;C++ Files (*.cpp *.h)");

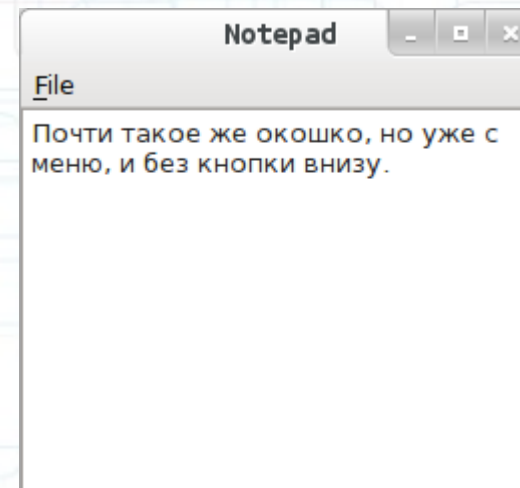
    if (fileName != "") {
        QFile file(fileName);
        if (!file.open(QIODevice::ReadOnly)) {
            QMessageBox::critical(this, "Error", "Could not open file");
            return;
        }
        QString contents = file.readAll().constData();
        _textEdit->setPlainText(contents);
        file.close();
    }
}
```

Главное окно

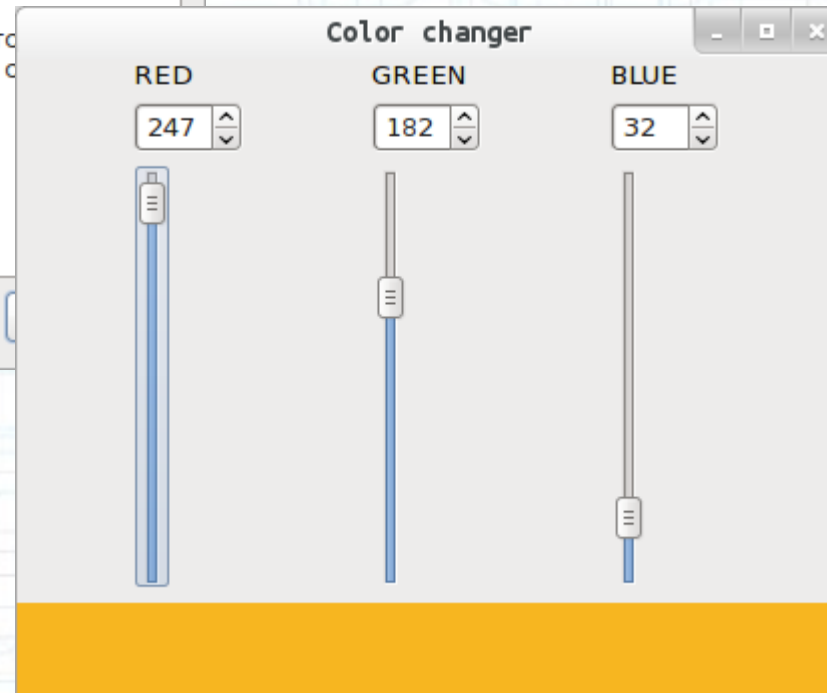
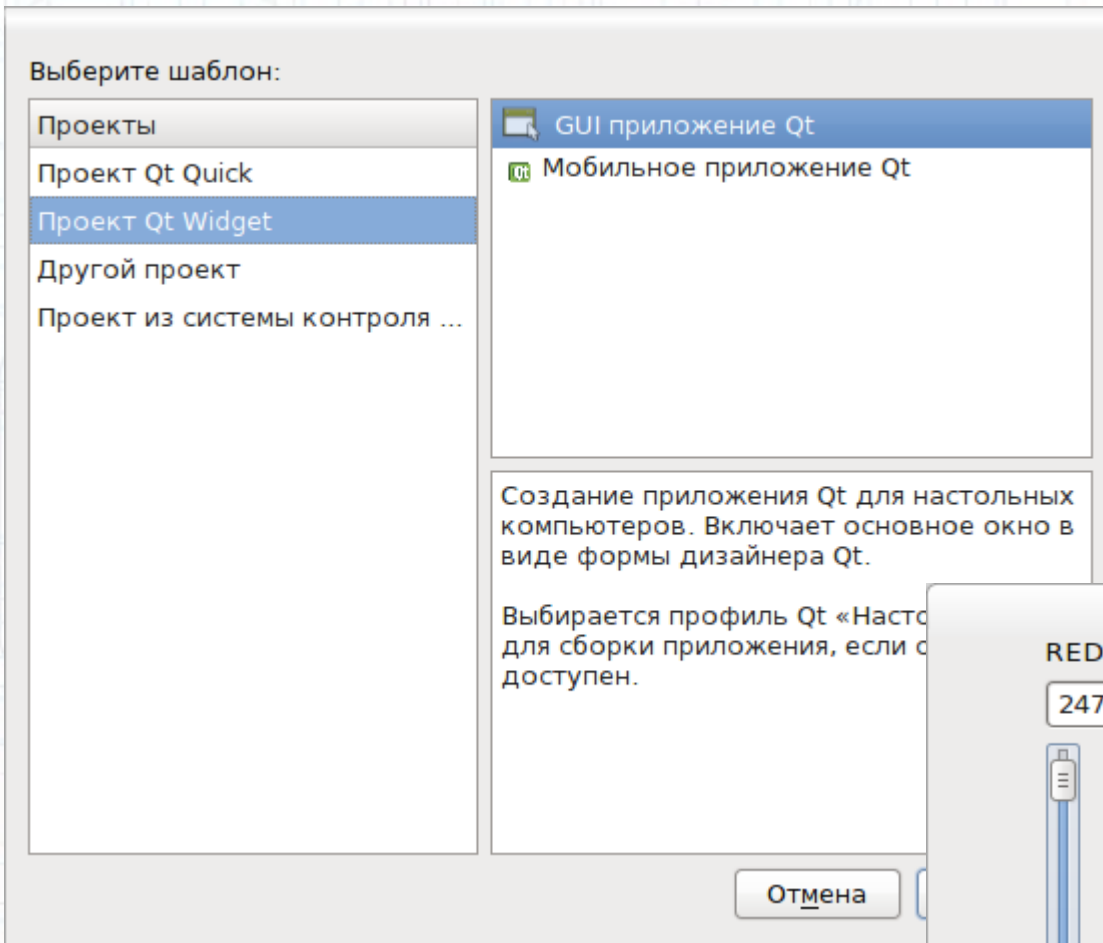
```
void Notepad::save() {
    QString fileName = QFileDialog::getSaveFileName(this, "Save File",
        "", "Text Files (*.txt);;C++ Files (*.cpp *.h)");

    if (fileName != "") {
        QFile file(fileName);
        if (!file.open(QIODevice::WriteOnly)) {
            // error message
        } else {
            QTextStream stream(&file);
            stream << _textEdit->toPlainText();
            stream.flush();
            file.close();
        }
    }
}

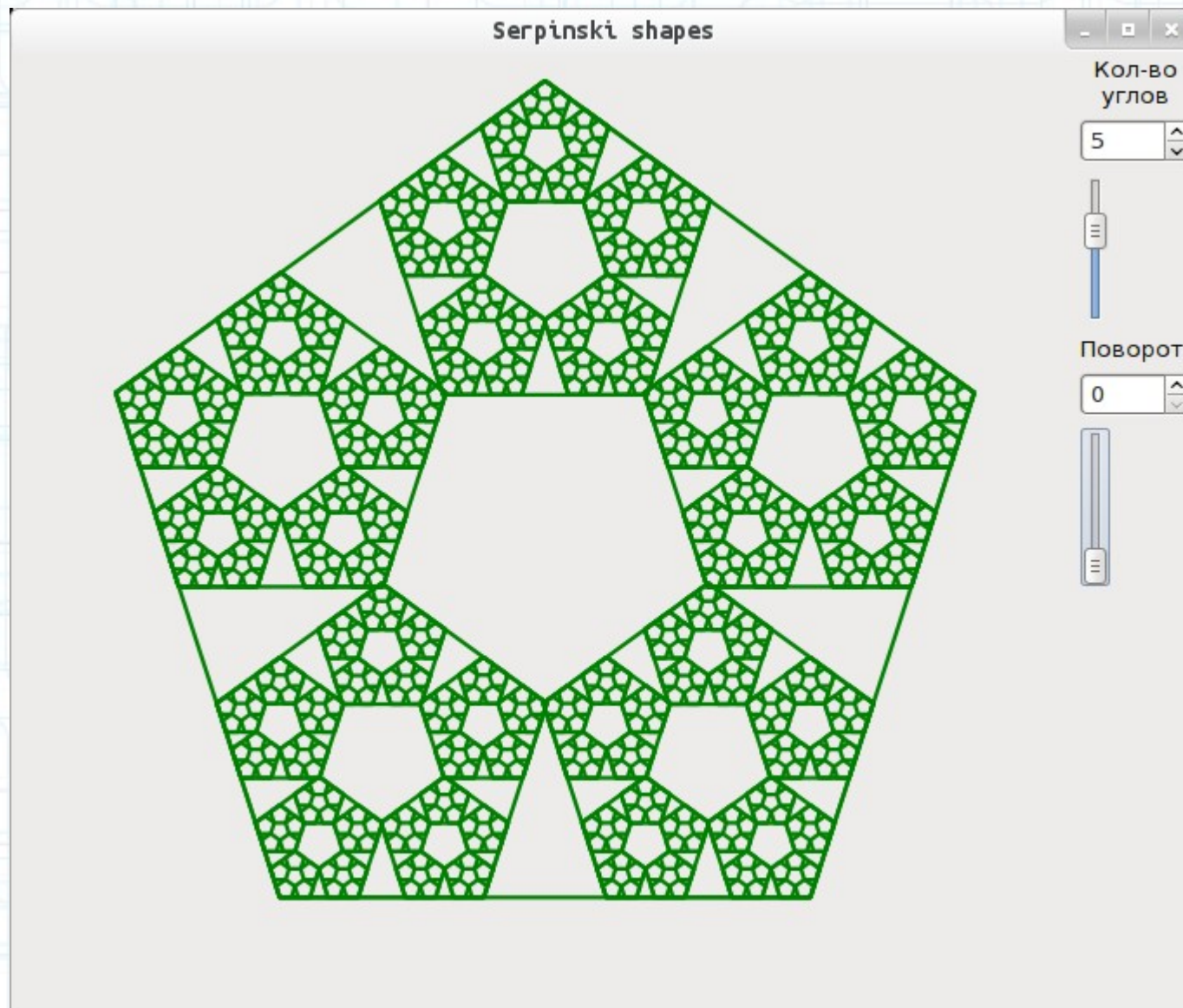
void Notepad::quit() {
    _app->quit();
}
```



Дизайнер форм



Рисование на QWidget



В заключении

Исходные коды проектов:

- https://github.com/newmen/qt_serpinskiShapes
- https://github.com/newmen/qt_cellular

Презентация доступна по адресу:

- <http://newmen.pro/qt/1.pdf>