

Лабораторная работа № 10

Закрепление навыков программирования в объектно-ориентированном стиле. Визуальные компоненты. Знакомство с QT.

Цель работы: приобретение практических навыков создания приложений на языке C++.

Задание: Для выполнения лабораторных работ с использованием операционной системы Microsoft Windows необходимо установить следующее программное обеспечение: Tortoise Mercurial актуальной версии; Qt SDK или иной дистрибутив Qt согласованный с преподавателем. При изучении дисциплины необходимо обеспечить доступ в интернет для осуществления доступа к системе контроля версий. Лабораторные работы выполняются на объектно-ориентированном языке программирования C++. Изменения, вносимые студентами в свои проекты, следует периодически отдавать на сервер системы контроля версий для осуществления контроля над выполнением заданий со стороны преподавателя. Для успешной сдачи лабораторной работы следует выполнить следующий перечень заданий.

Задание 1.

Создать репозиторий под контролем git на открытой площадке gitlab.com
Задать имя репозитория <год>_<группа>_<имя студента в транслите>_<номер варианта по журналу>.

Задание 2.

Выполнить проектирование задачи в соответствии с вариантом (табл. 1). Для реализации поставленной задачи необходимо спроектировать, реализовать и использовать шаблон «умные указатели». Соответственно это учесть при проектировании программного обеспечения.

Задача 3.

Выполнить реализацию в соответствии с вариантом задачи (табл. 1), используя среду разработки QT. Реализация должна быть кроссплатформенной и выполнена на основе графических окон.

№ варианта	Задание
1	Создать приложение файловый менеджер, с возможностью копировать файлы из 1 директории в другую. Выделять для копирования можно сразу несколько файлов.
2	Создать приложения для просмотра графического материала, поиск файлов с разрешениями jpeg, bmp, png, сохранение ссылок на них в файл, для быстрого доступа при повторном запуске приложения.
3	Создать графический редактор типа Painter, отрисовка стандартных графических примитивов, выбор цвета, толщины нажима. Сохранение в файл, загрузка из файла.
4	Создать приложение векторный редактор, предусмотреть сложение векторов двумя способами.
5	Создать приложение для поиска плагиата в тексте, на основе своей выборки.
6	Создать приложение для контроля изменений в указанной директории, все изменения сохраняются в файл.
7	Создать приложение расстановки мебели в помещении, которое представляет собой вид сверху 2d сцены. Сохранять и загружать мебель и комнаты из файла.
8	Создать приложение фитнес-инструктор.
9	Создать приложение виртуальный журнал преподавателя.
10	Создать приложение инженерный калькулятор.

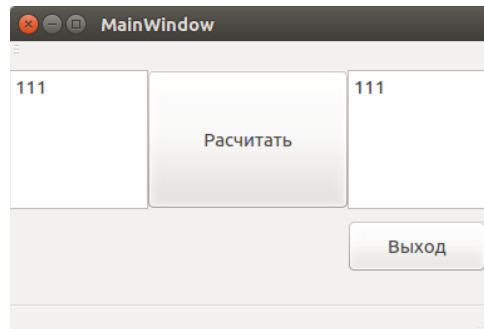
Примечание:

Для реализации шаблона «умный указатель» требуется разработать класс пример которого дан ниже (это пример, а не ваша реализация):

```
template <typename T>
class smart_ptr {
T * obj;
public:
smart_ptr(T *obj)
: obj(obj){ }
~smart_ptr() {
delete obj;
}
T* operator->() { return obj; }
T& operator* () { return *obj; }
}
```

Пример формы:

Данное приложение копирует текст левого окна, в правое, перед этим показывает содержимое окна в DialogBox. Кнопка «Заккрыть» посылает генерирует сигнал closeForm1.



Созданный графический интерфейс представляет собой следующее описание

```
#include<QtCore/QVariant>
```

```
#include<QtGui/QAction>
#include<QtGui/QApplication>
#include<QtGui/QPushButtonGroup>
#include<QtGui/QHeaderView>
#include<QtGui/QMainWindow>
#include<QtGui/QMenuBar>
#include<QtGui/QPushButton>
#include<QtGui/QStatusBar>
#include<QtGui/QTextEdit>
#include<QtGui/QToolBar>
#include<QtGui/QWidget>
```

```
QT_BEGIN_NAMESPACE
```

```
classUi_MainWindow
{
public:
QWidget*centralWidget;
QPushButton*pushButton;
QTextEdit*textEdit;
QTextEdit*textEdit_2;
QPushButton*pushButton_2;
QMenuBar*menuBar;
QToolBar*mainToolBar;
QStatusBar*statusBar;

voidsetupUi(QMainWindow*MainWindow)
{
if(MainWindow->objectName().isEmpty())
MainWindow->setObjectName(QString::fromUtf8("MainWindow"));
MainWindow->resize(386,232);
centralWidget=newQWidget(MainWindow);
centralWidget->setObjectName(QString::fromUtf8("centralWidget"));
pushButton=newQPushButton(centralWidget);
pushButton->setObjectName(QString::fromUtf8("pushButton"));
pushButton->setGeometry(QRect(110,10,161,111));
textEdit=newQTextEdit(centralWidget);
textEdit->setObjectName(QString::fromUtf8("textEdit"));
textEdit->setGeometry(QRect(0,10,111,111));
```

```

textEdit->setReadOnly(false);
textEdit_2=newQTextEdit(centralWidget);
textEdit_2->setObjectName(QString::fromUtf8("textEdit_2"));
textEdit_2->setGeometry(QRect(270,10,111,111));
pushButton_2=newQPushButton(centralWidget);
pushButton_2->setObjectName(QString::fromUtf8("pushButton_2"));
pushButton_2->setGeometry(QRect(270,130,111,41));
MainWindow->setCentralWidget(centralWidget);
menuBar=newQMenuBar(MainWindow);
menuBar->setObjectName(QString::fromUtf8("menuBar"));
menuBar->setGeometry(QRect(0,0,386,25));
MainWindow->setMenuBar(menuBar);
mainToolBar=newQToolBar(MainWindow);
mainToolBar->setObjectName(QString::fromUtf8("mainToolBar"));
MainWindow->addToolBar(Qt::TopToolBarArea,mainToolBar);
statusBar=newQStatusBar(MainWindow);
statusBar->setObjectName(QString::fromUtf8("statusBar"));
MainWindow->setStatusBar(statusBar);

retranslateUi(MainWindow);

QMetaObject::connectSlotsByName(MainWindow);
} //setupUi

voidretranslateUi(QMainWindow*MainWindow)
{
MainWindow->
setWindowTitle(QApplication::translate("MainWindow","MainWindow",0,QApplication::
:UnicodeUTF8));
pushButton->setText(QApplication::translate("MainWindow","\
320\240\320\260\321\201\321\207\320\270\321\202\320\260\321\202\321\214
",0,QApplication::UnicodeUTF8));
textEdit->setHtml(QApplication::translate("MainWindow","<
!DOCTYPEHTMLPUBLIC"-//W3C//DTDHTML4.0//EN"\<a href="http://www.w3.org/TR/REC-html40/
strict.dtd">\n"
"<html><head><metaname=\"grichtext\"content=\"1\"/><styletype=\"text/css\">\n"
"p,li{white-space:pre-wrap;}\\n"
"</style></head><bodystyle=\"font-family:'Ubuntu';font-size:11pt;font-
weight:400;font-style:normal;\">\n"
"<pstyle=\"margin-top:0px;margin-bottom:0px;margin-left:0px;margin-right:0px;-
qt-block-indent:0;text-indent:0px;\">111</p></body></
html>",0,QApplication::UnicodeUTF8));
pushButton_2->setText(QApplication::translate("MainWindow","\
320\222\321\213\321\205\320\276\320\264",0,QApplication::UnicodeUTF8));
} //retranslateUi

};

namespaceUi{
classMainWindow:publicUi_MainWindow{};
} //namespaceUi

QT_END_NAMESPACE

#endif //UI_MAINWINDOW_H

```