

Долгосрочное задание

# Программирование пользовательского интерфейса

Библиотека Qt языка программирования C++

## КОНЦЕПТУАЛЬНЫЙ ОБЗОР ОСНОВНОГО РЕЗУЛЬТАТА

Результатом выполнения долгосрочного задания должно являться решение (solution) созданное в среде разработки Microsoft Visual Studio 2017 (!) состоящее из набора проектов, каждый из которых реализует отдельный модуль разработанного на языке программирования C++ программного средства, использующего для реализации графического интерфейса библиотеку Qt (версии 5.3 или выше).

Эскиз главного окна графического интерфейса разрабатываемого программного средства представлен на рисунке 1.

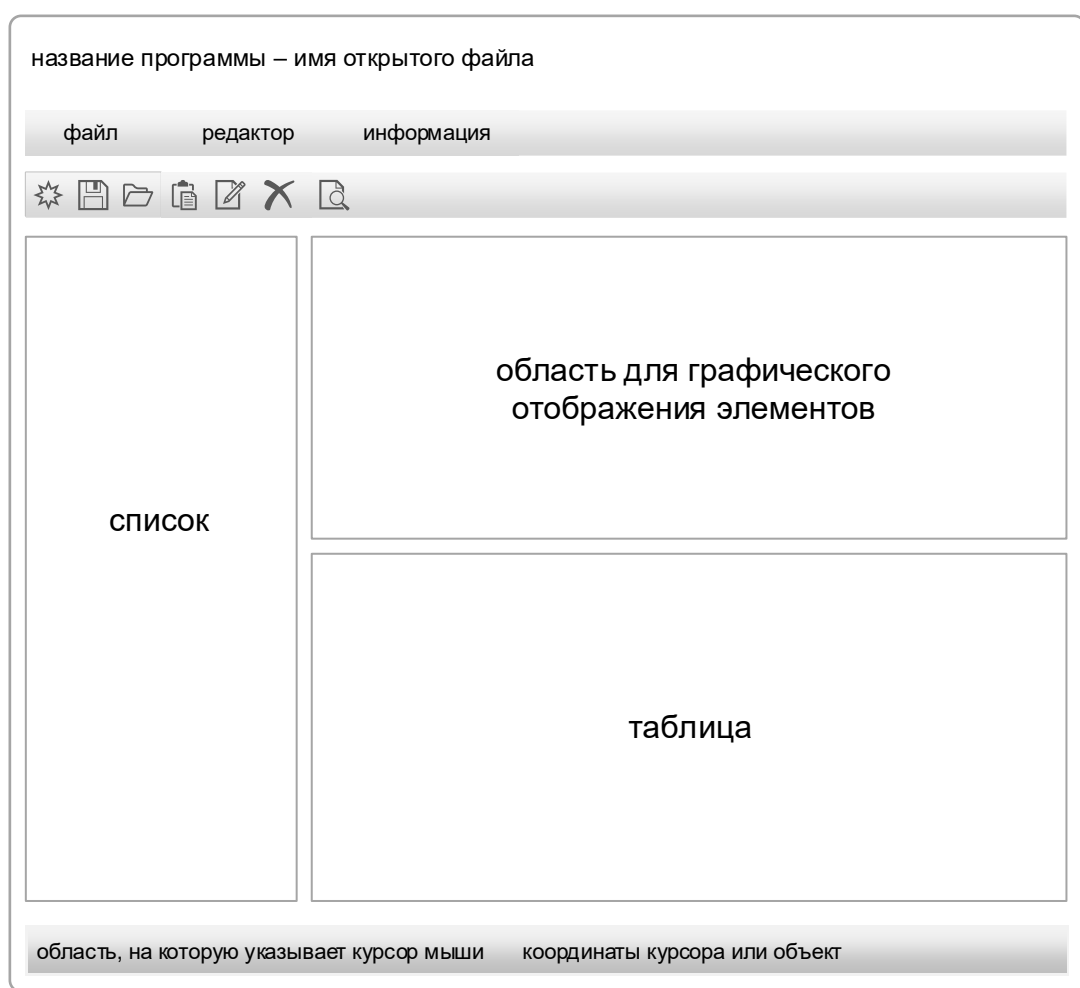


Рисунок 1 – Эскиз главного окна разрабатываемого графического интерфейса

Главное окно должно содержать (перечисление по рисунку сверху-вниз, слева-направо): главное меню, панель управления, список обрабатываемых

объектов, область для графического отображения обрабатываемых объектов, таблицу параметров обрабатываемых объектов, панель статуса.

Разрабатываемое приложение должно иметь возможность обработки (добавления, редактирования, удаления, отображения) информации объектах, характеризующихся некоторым (произвольным) набором параметров. Должны быть реализованы возможности создания, сохранения и загрузки обрабатываемого набора в(из) файл(а).

## **ЗАДАЧИ ПРОЕКТА**

Для достижения цели учебного проекта необходимо решить следующие задачи:

1. Разработка и реализация диалогового окна для добавления и редактирования параметров обрабатываемых объектов.
2. Разработка и реализация главного окна программного средства, без наполнения рабочей области.
3. Разработка и реализация модели для хранения данных об обрабатываемых объектах, для её последующего использования в объектах, обозначенных на рисунке 1 как «список» и «таблица».
4. Разработка и реализация модели для отображения информации об обрабатываемых объектах в графическом виде.
5. Объединения всех разработанных составляющих в единое программное средство с графическим интерфейсом.

Каждая задача решается в рамках отдельного проекта в среде разработки Microsoft Visual Studio. Все проекты объединяются в одно решение согласно предоставленному шаблону (см. git-репозиторий).

Для корректной компиляции проектов шаблона необходимо наличие в операционной (кроме, очевидно, установленной библиотеки Qt) следующих переменных окружения:

`qtdir` – корневой каталог библиотеки Qt (содержит директории `bin`, `include`, `lib`, `plugins` ...);

`objdir` – каталог, внешний по отношению к каталогу репозитория, в котором будут размещаться промежуточные результаты сборки проектов.

Для корректной работы откомпилированных программных средств, в переменной окружения `path` должны содержаться пути к каталогам, содержащим dll-библиотеки модулей библиотеки Qt и путь к каталогу `%qtdir%\plugins\` (обратите внимание на символ “\”).

В настройки проектов шаблона допускается вносить любые изменения за исключением изменения путей для размещения результатов компиляции. Для ссылки на каталоги библиотеки Qt следует обязательно использовать переменную `qtdir`.

*Примечание:* Для решения поставленных задач в качестве справочного материала рекомендуется использовать книгу Шлее М. «Qt 5.X. Профессиональное программирование на C++» того издания, которое сможете найти. Далее по каждой задаче будут приведены ссылки на соответствующие главы книги, которые рекомендуется изучить.

## ДИАЛОГОВОЕ ОКНО

Эскиз диалогового окна для создания и редактирования параметров обрабатываемого объекта представлен на рисунке 2. Открытие диалогового окна должно производиться в модальном режиме. Изменение размеров окна допускается только по горизонтали. По вертикали размер должен быть зафиксирован. Прямоугольные области текстовых меток и элементов ввода данных не должны накладываться одна на другую. Текстовые метки форматируются по левому краю области. Элементы ввода данных – по ширине. Размеры кнопок устанавливаются по умолчанию в зависимости от заданного текста и иконки (специально задавать не нужно). Расположение кнопок форматируется по правому краю. Заголовок окна должен содержать описание выполняемого в данный момент действия (создание/изменение чего-

либо). Для диалогового окна задаётся иконка для отображения в элементах управления операционной системы.



Рисунок 2 – Эскиз диалогового окна для создания и редактирования параметров обрабатываемого объекта

Функция разворачивания окна на весь экран отключается. Так же отключаются любые другие функции, использование которых не предполагается замыслом. Пример недопустимых ошибок в размещении графических элементов представлен на рисунке 3.

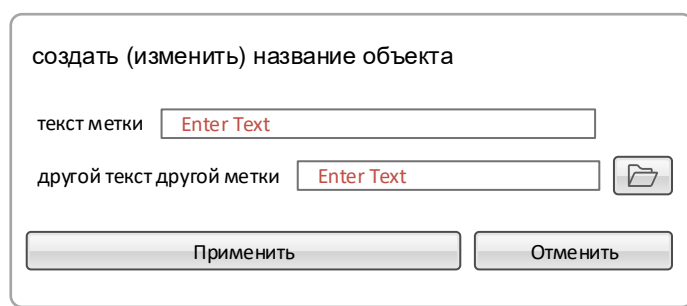


Рисунок 3 – Распространённые ошибки, допускаемые при размещении графических элементов диалогового окна

Отдельный проект для разработки диалогового окна (dialog) кроме собственно файлов, содержащих реализацию соответствующего функционала, должен содержать код создания и отображения диалогового окна в момент

запуска приложения. Реализация диалогового окна в исходном коде должна представлять собой два файла с одинаковым именем, совпадающим с названием реализуемого класса, и расширениями `h` – заголовочный и `cpp` – файл-реализация соответственно. Заголовочный файл размещается в каталоге `include` решения `matter`. Файл-реализация помещается в каталог соответствующего проекта.

*Примечание:* Для решения рассмотренной задачи рекомендуется (в режиме справочника) изучить главы 2, 5–8, 10, 11, 32 книги Шлее М. «Qt 5.X. Профессиональное программирование на C++».

## ГЛАВНОЕ ОКНО

В рамках данного задания необходимо разработать и реализовать главное окно без реализации виджетов рабочей (центральной) части.

### Главное меню и панель управления

Главное меню формируется из трёх пунктов: «файл», «редактор», «информация». Содержимое пунктов главного меню представлено на рисунке 4.

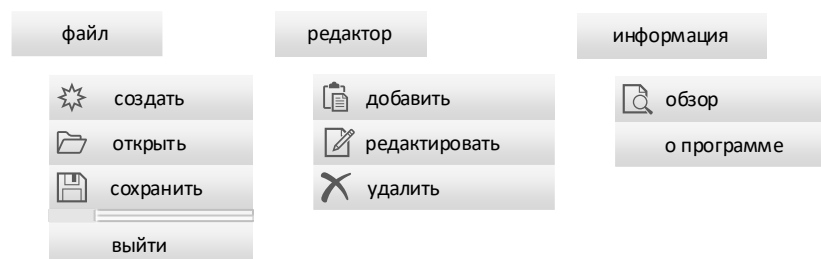


Рисунок 4 – Содержимое пунктов главного меню

Описание функций, активируемых каждым из подпунктов главного меню приведено в таблице 1. Данные функции так же дублируются в панели управления, размещаемой под главным меню (см. рисунок 5). Пункт главного меню и кнопки панели управления должны быть реализованы с помощью единого набора элементов типа `QAction`. Для всех пунктов меню необходим настроить быстрые клавиши доступа к ним.



Рисунок 5 – Эскиз панели управления главного окна

Таблица 1

Функции, активируемые пунктами главного меню приложения

Пункт меню	Описание функции
создать	Создание нового набора обрабатываемых объектов.
открыть	Загрузка набора обрабатываемых объектов, ранее сохранённый в файле. Файл выбирается пользователем с помощью файлового диалога, настроенного для сохранения файла.
сохранить	Сохранение набора обрабатываемых объектов в файле с именем заданным пользователем. Файл выбирается пользователем с помощью файлового диалога, настроенного для открытия файла.
выйти	Завершение работы программного средства. При необходимости (желании разработчика) завершение приложения предваряется подтверждением на закрытие и предложением сохранить несохранённый набор данных.
добавить	Добавление информации о новом объекте в текущий набор. Запускается диалоговое окно, разработанное в рамках задачи 1; при положительном завершении диалога введённые в диалоговом окне пользователем данные помещаются в модели данные и рабочие виджеты приложения.
редактировать	Изменение значения выделенного объекта. Запускается диалоговое окно, разработанное в рамках задачи 1. Параметры выделенного объекта отображаются в элементах ввода данных диалогового окна. При положительном завершении диалога введённые в диалоговом окне пользователем данные помещаются в модели данные и рабочие виджеты приложения.
удалить	Выделенный объект удаляется.
обзор	Просмотр всей информации. Информация объектах текущего набора размещается в формате html и отображается в отдельном диалоговом окне без возможности её изменения. При необходимости для пролистывания списка параметров используется скроллинг.
о программе	Вывод диалогового окна вида About, содержащего информацию о программе и её авторе.

Логика работы всех пунктов меню должна быть реализована таким образом, чтобы действия, которые пользователь в данный момент выполнить не может были не активны (disabled).

### **Остальные составляющие графического интерфейса главного окна**

Центральная область главного окна в рамках решения данной задачи наполняется тройкой пустых виджетов-заглушек, размещённых так, как изображено на эскизе главного окна (рисунок 1). Виджеты-заглушки объединяются в единую область с помощью вспомогательных виджетов типа QSplitter.

В нижней части главного окна размещается панель состояния (статуса) (см. рисунок 6), в которой размещаются две информационные области, отображающие текстовое описание области над которой в данный момент располагается курсор мыши и координаты курсора мыши или объект, над которым расположен курсор мыши, если такой имеется под курсором. С правого края панели статуса размещается стандартный элемент управления для изменения размеров главного окна (уголок).



Схема панели статуса главного окна, состоящая из двух прямоугольных областей. Первая область с текстом "область, на которую указывает курсор мыши" занимает левую часть панели. Вторая область с текстом "координаты курсора или объект" занимает правую часть панели. Справа от второй области находится стандартный элемент управления для изменения размеров окна (уголок).

Рисунок 6 – Эскиз панели состояния главного окна программного средства

Главное окно снабжается иконкой для отображения в элементах управления операционной системы. Заголовок окна содержит название приложения и имя файла обрабатываемого набора объектов.

### **Требования к реализации проекта MS Visual Studio**

Отдельный проект для разработки диалогового окна (mainWindow) кроме собственно файлов, содержащих реализацию соответствующего функционала, должен содержать код создания и отображения главного окна в момент запуска приложения. Реализация диалогового окна в исходном коде должна представлять собой два файла с одинаковым именем, совпадающим



с названием реализуемого класса, и расширениями `h` – заголовочный и `cpp` – файл-реализация соответственно. Заголовочный файл размещается в каталоге `include` решения `matter`. Файл-реализация помещается в каталог соответствующего проекта.

*Примечание:* Для решения рассмотренной задачи рекомендуется (в режиме справочника) изучить главы 6, 32, 34 книги Шлее М. «Qt 5.X. Профессиональное программирование на C++».