

Лабораторная работа №3. (лекции 78)

Цель: научиться создавать приложение основанное на архитектуре модельпредставление Qt. Понять как модель взаимодействует с остальным приложением.

Qt Metaobject system, Qt resource system, Qt SignalSlot mechanism, Qt model/view architecture, Qt Creator.

QApplication, QDataStream, QAbstractItemModel, QTreeView, QModelIndex, QHeaderView, QItemDelegate

Продолжительность: 4 часа

Задача:


Разработать приложение для хранения информации о музыкальной коллекции. Приложение должно использовать деревянную модель для хранения треков, сгруппированный по альбому и исполнителю. Данная программа сложнее, чем примеры на лекциях, поэтому лучше потратить время на изучения каждого этапа создания приложения и прочитайте документацию, чтобы понимать что делается и почему.

При выполнении ЛР рекомендуется использовать т.н. подход *от частного к общему (снизу вверх)*. Т.е. сначала создаются небольшие модули из которых потом собирается все приложение. Недостаток этого подхода в том, что нет общей картины при разработке и сложно отследить насколько работа закончена. Это компенсируется возможностью легко

протестировать каждый компонент отдельно без взаимодействия с другими частями приложения. Тестирование очень важная часть разработки, когда речь идет об архитектуре модельпредставление и тем более о пользовательских моделях, где очень легко сделать ошибку.

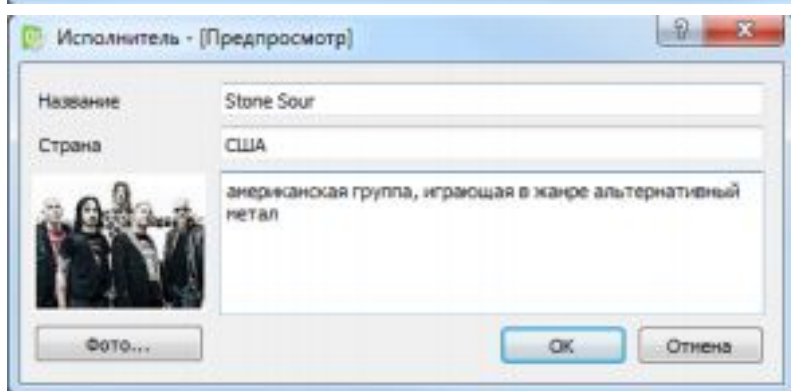
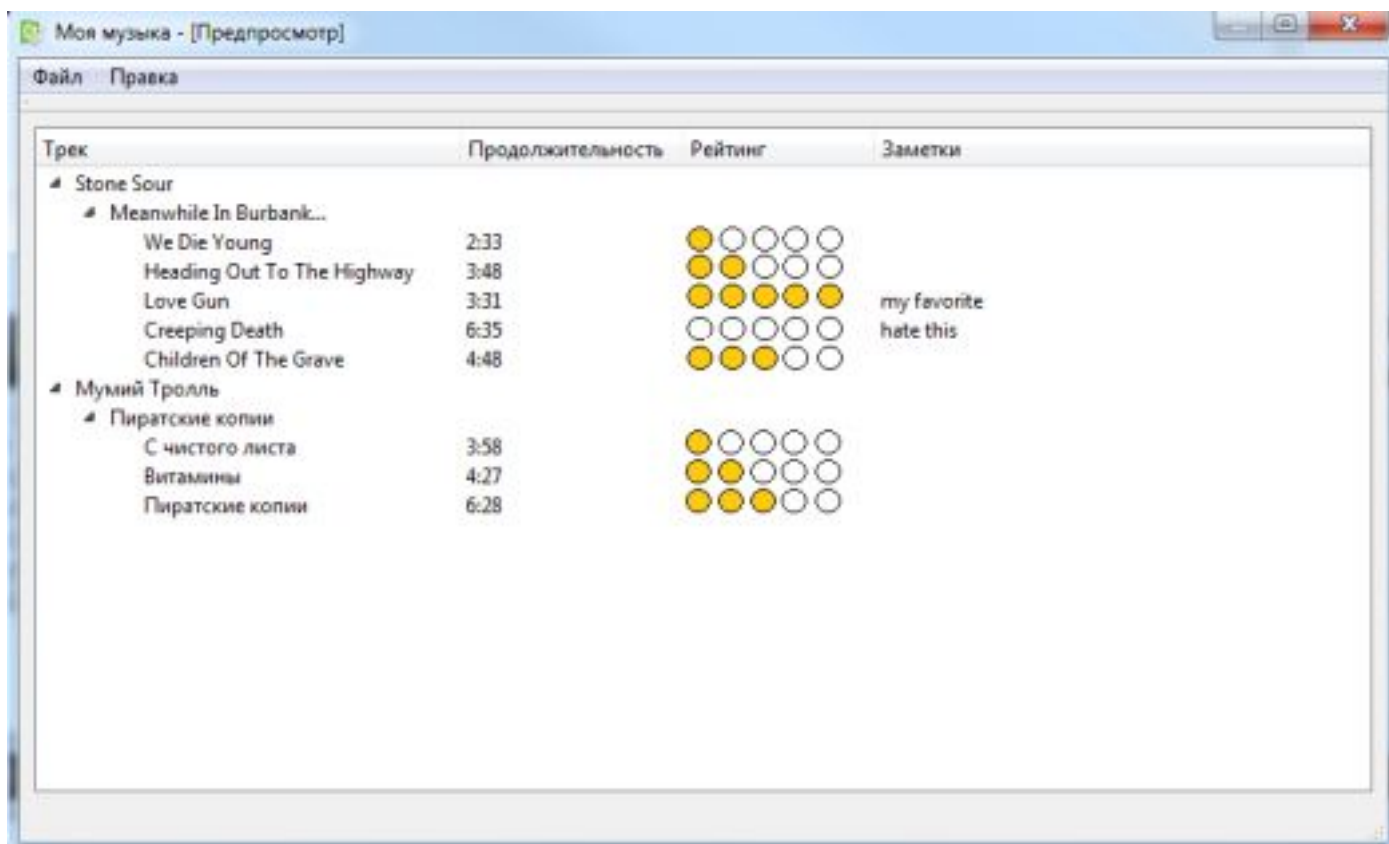
Выполнение этой работы подразумевает, что вы уже свободно владеете основами создания Qt приложений. Инструкция больше уделяет внимания задаче, которую нужно сделать, а не тому как ее выполнить.

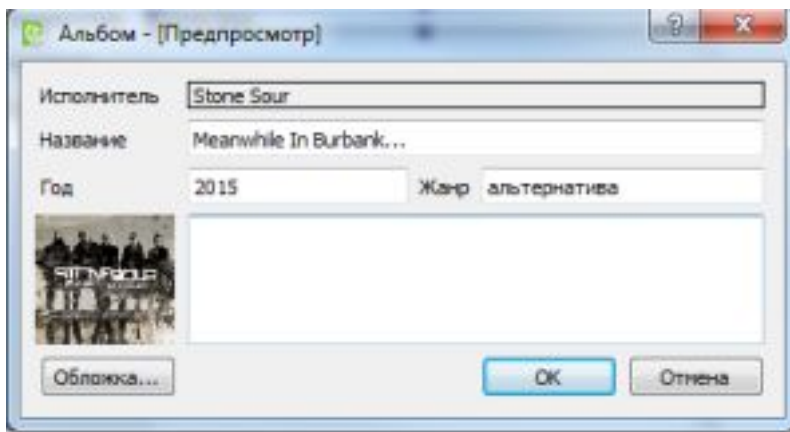
При использовании подхода *от частного к общему* вам нужно разработать общий план всего проекта. Для выполнения этой ЛР план может быть таким:

1. Разработать структуры для хранения информации о песнях, альбомах и исполнителях с методами записи и чтения их из постоянного хранилища (например жесткий диск) а. классы `Artist`, `Album` и `Song` для хранения данных
b. сохранение и чтение всех данных в файл через `QDataStream`
2. Реализовать модель данных, которая "оборачивает" структуры в интерфейсы архитектуры модельпредставление Qt
a. класс `MusicModel` реализующий интерфейс `QAbstractItemModel` для иерархической модели
b. модель поддерживает редактирование данных, вставку и удаление строк (исполнителей, альбомов, песен)
3. Реализовать делегаты для отрисовки и редактирования данных
a. рейтинг песни отображается и редактируется мышью с помощью пользовательского делегата.
b. Внешний вид рейтинга должен быть таким 
4. Реализовать диалог для удобного управления данными в модели



5. Реализовать главное окно приложения с меню, панелью инструментов и представлением, которое объединяет все части вместе
 - а. представление QTreeView для отображения данных модели
 - б. действие для загрузки и сохранения данных в файл(меню, панель инструментов)
 - в. действие для добавления/удаления элементов коллекции
 - г. пункты меню и кнопки панели инструментов активны когда нужно (например нельзя добавить трек к исполнителю).





2

НИЯУ МИФИ. Кафедра 33. Беляков А.К.
Курс "Системное прикладное программное обеспечение"
Лабораторная работа №3

Структура данных для музыкальной коллекции

Исполнитель описывается следующими свойствами (характеристиками, атрибутами)

Атрибут (переменная) Тип данных

имя строка

фото изображение

страна строка

комментарий строка

альбомы список элементов типа "Альбом" Альбом имеет характеристики:

Атрибут (переменная) Тип данных

Имя строка

год выхода целое

обложка изображение

жанр строка

комментарий строка

Песни список элементов типа "Трек" Исполнитель Исполнитель

Трек имеет характеристики

Атрибут (переменная) Тип данных

название строка

продолжительность время

рейтинг целое

комментарий строка

Требования к приложению.

1. Для хранения данных о музыкальной коллекции реализованы классы `Artist`, `Album` и `Song`, являющиеся частью иерархической структуры согласно заданию.
2. Музыкальная коллекция сохраняется и загружается из файла с помощью `QDataStream`.
3. Реализована иерархическая модель (класс `MusicModel`) на основе `QAbstractItemModel`, обеспечивающая доступ к данным в объектах классов `Artist`, `Album` и `Song`.
4. Модель поддерживает редактирование данных о треке (название, длительность, комментарий) с помощью стандартных делегатов
5. Для отображения и редактирования рейтинга трека используется собственный делегат (класс `RatingDelegate`) на основе класса `QStyledItemDelegate`.
6. Главное окно приложения содержит представление `QTreeView`. Отображение модели `MusicModel` в представлении соответствует структуре коллекции.
7. Реализованы диалоги для отображения и редактирования всех свойств элементов коллекции: исполнитель и альбом. Диалог открывается по контекстному меню "Свойства" у элемента дерева.
8. При закрытии диалога свойств элемента коллекции по кнопке "ОК" измененные данные сохраняются в модель и отображаются в представлении.
9. Приложение позволяет создавать, удалять и редактировать элементы коллекции через меню и кнопки панели инструментов. Пункты меню приложения и кнопки панели управления активны когда ожидается.
10. При добавлении нового элемента коллекции открывается соответствующий диалог свойств элемента для ввода данных об элементе
11. Модель корректно обрабатывает операцию добавления нового элемента в коллекцию и новый элемент появляется в представлении
12. Модель корректно обрабатывает операцию удаления элемента из коллекции и новый элемент исчезает из представления
13. Заголовок главного окна приложения и признак изменения данных коллекции работают как ожидалось.

Оценка

Максимальная оценка за выполнение работы 25 баллов. Оценка выставляется по результатам проверки работоспособности программы по пунктам требований по правилу: • 1 балл за пункты 2,4,9,13

- 2 балла за каждый из пунктов 6,7,8,10,11,12
- 3 балла за каждый из пунктов 1,3,5

При выполнении работы на 25 баллов самостоятельно, без использования методического материала, оценка увеличивается на 20% (т.е. можно получить 30 баллов). Если выполнение работы с помощью методического материала займет более чем неделю, оценка уменьшается на 20% (т.е. можно получить не более 20 баллов).

