

Федеральное государственное автономное  
образовательное учреждение  
высшего образования  
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт космических и информационных технологий

институт

Вычислительная техника

кафедра

УТВЕРЖДАЮ

Заведующий кафедрой

О. В. Непомнящий

подпись

инициалы, фамилия

2020г.

**БАКАЛАВРСКАЯ РАБОТА**

09.03.01 – «Информатика и вычислительная техника»

код - наименование направления

Управление данными приложения «Пазлы-Живопись»

тема

Руководитель

подпись, дата

канд. техн. наук, доцент

должность, ученая степень

А. И. Постников

инициалы, фамилия

Консультант

подпись, дата

ст. преподаватель

должность, ученая степень

В. С. Васильев

инициалы, фамилия

Выпускник

подпись, дата

К. А. Зимин

инициалы, фамилия

Нормоконтролер

подпись, дата

канд. техн. наук, доцент

А. И. Постников

инициалы, фамилия

Красноярск 2020

Федеральное государственное автономное  
образовательное учреждение  
высшего образования  
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт космических и информационных технологий

---

институт

Вычислительная техника

---

кафедра

УТВЕРЖДАЮ

Заведующий кафедрой

\_\_\_\_\_ О. В. Непомнящий

подпись

инициалы, фамилия

« \_\_\_\_\_ » \_\_\_\_\_ 2020г.

**ЗАДАНИЕ  
НА ВЫПУСКНУЮ КВАЛИФИКАЦИОННУЮ РАБОТУ**

**в форме**

**бакалаврской работы**

---

бакалаврской работы, дипломного проекта, дипломной работы, магистерской  
диссертации



## РЕФЕРАТ

Выпускная квалификационная работа по теме: «Управление данными приложения «Пазлы-Живопись»». Пояснительная записка содержит 51 страницу текстового документа, 35 иллюстраций, 7 таблиц, 1 приложение, и 13 использованных источников.

ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ, ANDROID, ПАЗЛЫ, БИБЛИОТЕКА ИЗОБРАЖЕНИЙ, JSON, АЛГОРИТМ.

Цель работы: создание приложения для управления данными Android-приложения «Пазлы-Живопись», разработанного выпускником ИКИТ СФУ Брюхановой У. А для студентов Сибирского государственного института искусств имени Дмитрия Хворостовского.

В результате выполнения ВКР было разработано приложение для ОС Windows, позволяющее формировать библиотеку изображений в JSON-файле.

В выпускную квалификационную работу входит введение, 3 главы и заключение.

Во введении ставится цель и производится ее декомпозиция на задачи.

В первой главе анализируется проблема и ее причины, формируются основные требования к разрабатываемому приложению и выбираются инструменты разработки.

Во второй главе разрабатывается интерфейс и алгоритм работы приложения.

В третьей главе описана реализация основных элементов приложения, составлена инструкция для программиста и проведено тестирование.

В заключении подводятся итоги по выполненной работе.

## СОДЕРЖАНИЕ

Введение.....	3
1 Анализ задания на выпускную квалификационную работу.....	5
1.1 Текущее решение .....	5
1.1.1 Текущий способ работы с библиотекой .....	5
1.1.2 Формат хранения данных.....	6
1.2 Основание для разработки .....	8
1.2.1 Функциональные требования .....	8
1.2.2 Нефункциональные требования .....	24
1.3 Выбор инструментов .....	24
1.4 Выводы по главе .....	24
2 Проектирование .....	25
2.1 Интерфейс приложения.....	25
2.2 Общий алгоритм .....	31
2.3 Алгоритм добавления файла изображения .....	36
2.4 Алгоритм слияния библиотек.....	37
2.5 Выводы по главе .....	38
3 Реализация и тестирование .....	39
3.1 Конфигурационный файл.....	39
3.2 Инструкция для программиста .....	41
3.3 Тестирование .....	42
3.4 Выводы по главе .....	43
Заключение .....	45
Список использованных источников .....	46
Приложение А .....	48

## ВВЕДЕНИЕ

Пазлы являются классической игрой, направленной на развитие логического и пространственного мышления у детей. В настоящее время существует тенденция реализации такого рода игр в виде приложений для мобильных платформ, так как это позволяет родителям экономить деньги, а также обеспечивает сохранность игры (элементы виртуального пазла нельзя порвать или потерять).

Студенты Сибирского государственного института искусств имени Дмитрия Хворостовского в рамках дисциплин «История отечественного искусства и культуры», «История зарубежного искусства и культуры» заучивают огромное количество картин, при этом ряд изображений очень похожи, например иконы 12 – 13 веков. Преподавателями СГИИ было предложено разработать приложение, напоминающее игру Пазлы, но обладающее рядом отличий. Данное приложение было разработано выпускником ИКИТ СФУ Брюхановой У. А. [6].

Приложение было одобрено преподавателями СГИИ и используется в учебном процессе студентами. Данное приложение имеет существенный недостаток – отсутствие удобного способа формировать и обновлять библиотеку изображений.

Целью работы является создание приложения для ОС Windows, позволяющего создавать и обновлять библиотеку изображений для Android-приложения «Пазлы-Живопись», разработанного в рамках ВКР Брюхановой У. А. Для обслуживания библиотеки пользователь не должен обладать глубокими познаниями в сфере информационных технологий и программирования. Для достижения цели в работе решаются следующие задачи:

- проанализировать текущий способ создания и обновления библиотеки изображений;

- сформулировать требования к разрабатываемому приложению;
- разработать приложение для ОС Windows, позволяющее облегчить и автоматизировать данный процесс.

# 1 Анализ задания на выпускную квалификационную работу

## 1.1 Текущее решение

### 1.1.1 Текущий способ работы с библиотекой

На данный момент работа с библиотекой происходит в полностью ручном режиме. Преподаватель передает администратору данные библиотеки, администратор загружает изображения на веб-сервер и вручную создает JSON-файл. Данный процесс представлен на рисунке 1.1.

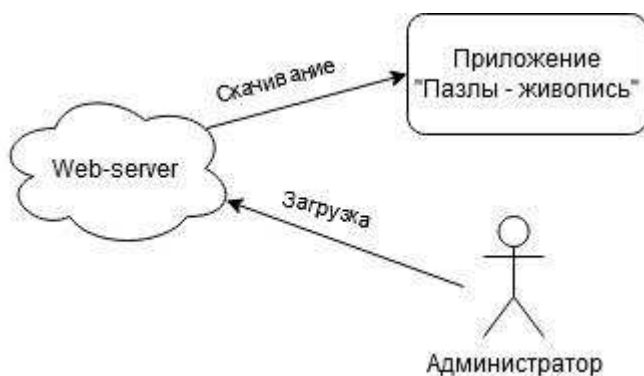


Рисунок 1.1 – Текущий способ работы с библиотекой

Данный подход создает ряд проблем:

- JSON - строго типизированный формат [10], после создания файла он должен пройти валидацию для исключения ошибок при формировании, это тоже приходится делать вручную с помощью онлайн сервисов или локальных инструментов;
- валидировать необходимо не только формат, но и данные, например дату последнего обновление записи, что так же делается вручную;
- отсутствует контроль логических ошибок в структуре библиотеки, ввиду чего могут возникать ошибки в работе приложения;
- для обслуживания библиотеки администратор должен обладать определенным уровнем познаний в сфере информационных технологий и программирования;
- для обслуживания библиотеки требуется большой объем работы при ручном формировании файла JSON.



## 1.1.2 Формат хранения данных

Библиотека изображений для приложения «Пазлы-живопись» хранится на веб-сервере и скачивается приложением при установке или обновлении базы данных. Корневой каталог библиотеки содержит файл «levels.json» и следующие подкаталоги:

- «authors» – содержит портреты авторов;
- «eras» – содержит изображения эпох;
- «arts» – содержит изображения предметов искусства.

Информация о картинах, авторах и эпохах хранится в файле «levels.json», размещаясь по трем разделам: arts, authors, eras.

В разделе «eras» описаны существующие в библиотеке эпохи. Каждая запись содержит три поля, описанных в таблице 1.1.

Таблица 1.1 – Описание полей раздела «eras»

Имя поля	Описание	Тип данных
eraName	Название эпохи	string
eraImagePath	Путь к изображению эпохи на веб-сервере	string
lastUpdate	Дата последнего обновления записи	string

В разделе «authors» описаны существующие в библиотеке авторы. Каждая запись содержит четыре поля, описанных в таблице 1.2.

Таблица 1.2 – Описание полей раздела «authors»

Имя поля	Описание	Тип данных
authorName	Имя автора	string
authorImagePath	Путь к изображению автора на веб-сервере	string
authorInfo	Информация об авторе	string
lastUpdate	Дата последнего обновления записи	string

В разделе «arts» описаны существующие в библиотеке произведения искусства. Каждая запись содержит семь полей, описанных в таблице 1.3.

Таблица 1.3 – Описание полей раздела «arts»

Имя поля	Описание	Тип данных
artEra	Эпоха произведения искусства	string
artPath	Путь к изображению произведения искусства на веб-сервере	string
artInfo	Информация о произведении искусства	string
artAuthors	Авторы произведения искусства. Значением данного поля является массив из одного или более элементов	array
artName	Путь к изображению автора на веб-сервере	string
isDomestic	Информация об авторе	bool
lastUpdate	Дата последнего обновления записи	string

В разделе «arts» значение поля «artEra» соответствует значению поля «eraName» существующей записи раздела «eras», элементы массива значения поля «artAuthor» соответствуют значениям полей «authorName» существующих записей раздела «authors». Структура данных json файла приведена на рисунке 1.2.

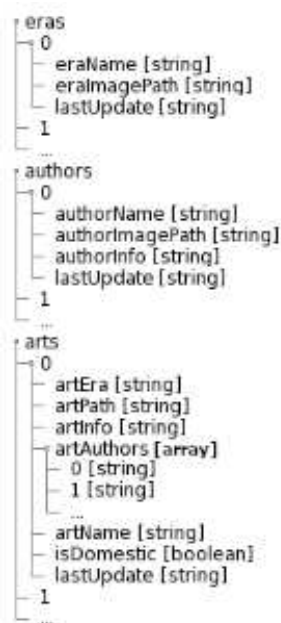


Рисунок 1.2 – Структура данных файла json

## **1.2 Основание для разработки**

Для предотвращения проблем, описанных в п. 1.1.1 разрабатывается приложение для ОС Windows, позволяющее упростить и автоматизировать работу с библиотекой изображений.

Работа с библиотекой подразумевает две роли пользователей:

- преподаватель – наполняет библиотеку произведениями искусства;
- администратор – объединяет библиотеки нескольких преподавателей в одну и подготавливает её для загрузки на веб-сервер.

Для удобства работы с библиотекой необходимо предусмотреть такие возможности, как фильтрация и сортировка списка записей. А так же возможность создавать записи по шаблону для добавления произведений искусства с одинаковыми параметрами (например, эпоха и категория).

Для предотвращения ошибок, связанных с некорректными именами файлов, необходимо разработать алгоритм переименования файлов в допустимый алфавит:

- буквы латинского алфавита [a..z];
- цифры [0..9];
- символы [-, \_].

Необходимо составить словарь замены для транслитерации русских символов в латинский алфавит и замены пробельных символов на символ подчеркивания. Остальные символы, не входящие в допустимый алфавит, должны быть удалены из имени файла.

### **1.2.1 Функциональные требования**

Для формирования функциональных требований составлена диаграмма вариантов использования [4], представленная на рисунке 1.3 и текстовое описание прецедентов.



Рисунок 1.3 – Диаграмма вариантов использования

Текстовое описание прецедентов:

**Название прецедента:** создание библиотеки.

**Действующее лицо:** преподаватель или администратор.

**Цель:** создать библиотеку.

**Предусловия:** пользователь запустил приложение.

**Главная последовательность:**

- a) пользователь нажимает в *начальном окне* кнопку «Создать»;
- b) приложение отображает *диалоговое окно сохранения файла*;
- c) пользователь вводит имя файла библиотеки, выбирает путь и нажимает кнопку «Сохранить»;
- d) приложение создает по заданному пути файл библиотеки и одноименный каталог с файловой структурой, пользователю показывается окно выбора раздела созданной библиотеки.

**Постусловия:** по указанному пути создана библиотека.

**Название прецедента:** открытие библиотеки.

**Действующее лицо:** преподаватель или администратор.

**Цель:** открыть библиотеку.

**Предусловия:** пользователь запустил приложение.

**Главная последовательность:**

а) пользователь нажимает в начальном окне кнопку «Открыть»;

б) приложение отображает диалоговое окно открытия файла;

с) пользователь переходит к расположению файла библиотеки, выделяет его и нажимает кнопку «Открыть»;

д) приложение открывает выбранный файл и отображает *окно выбора раздела*.

**Постусловия:** открыта выбранная библиотека.

**Альтернативная последовательность** (открытие последнего ранее открытого файла):

пользователь нажимает в начальном окне кнопку «Открыть [имя\_файла]»;

приложение открывает последний открытый файл библиотеки и отображает *окно выбора раздела*.

**Постусловия:** открыта последняя используемая библиотека.

**Название прецедента:** добавление автора.

**Действующее лицо:** преподаватель.

**Цель:** добавить в библиотеку автора.

**Предусловия:** преподаватель запустил приложение и создал либо открыл библиотеку.

**Главная последовательность:**

а) преподаватель нажимает в *окне выбора раздела* кнопку «Авторы»;

b) приложение отображает преподавателю *окно просмотра списка авторов*, содержащее *список имеющихся в библиотеке авторов*, а также кнопки *«Добавить»*, *«Изменить»*, *«Удалить»* и *«Назад»*;

c) преподаватель нажимает кнопку *«Добавить»*;

d) приложение отображает преподавателю *окно добавления автора*, содержащее *область просмотра изображения (портрета)*, *поля ввода для имени автора и информации*, а также кнопки *«Добавить»* и *«Отмена»*;

e) преподаватель двойным кликом по области изображения вызывает диалоговое окно открытия файла изображения и выбирает портрет автора, затем вводит имя автора и информацию о нем и нажимает кнопку *«Добавить»*;

f) приложение добавляет автора в библиотеку и отображает преподавателю *окно просмотра списка авторов*, в списке присутствует добавленный автор.

**Постусловия:** в библиотеку добавлен автор.

**Альтернативная последовательность** (добавление нескольких авторов подряд):

a) преподаватель нажимает в *окне выбора раздела* кнопку *«Авторы»*;

b) приложение отображает преподавателю *окно просмотра списка авторов*, содержащее *список имеющихся в библиотеке авторов*, а также кнопки *«Добавить»*, *«Изменить»*, *«Удалить»* и *«Назад»*;

c) преподаватель нажимает кнопку *«Добавить»*;

d) приложение отображает преподавателю *окно добавления автора*, содержащее *область просмотра изображения (портрета)*, *поля ввода для имени автора и информации*, а также кнопки *«Добавить»* и *«Отмена»*;

e) преподаватель двойным кликом по области изображения вызывает диалоговое окно открытия файла изображения и выбирает портрет автора, затем вводит имя автора и информацию о нем, активирует флажок, находящийся левее кнопки *«Добавить»* и нажимает кнопку *«Добавить»*;

f) приложение добавляет автора в библиотеку и очищает формы ввода *окна добавления автора*;

g) преподаватель повторяет действия начиная с пункта «е», пока не добавит всех авторов, при добавлении последнего автора преподаватель деактивирует флажок левее кнопки «*добавить*» перед её нажатием;

h) приложение отображает преподавателю *окно просмотра списка авторов*, в списке присутствуют все добавленные авторы.

**Постусловия:** в библиотеку добавлено несколько авторов.

**Название прецедента:** добавление эпохи.

**Действующее лицо:** преподаватель.

**Цель:** добавить в библиотеку эпоху.

**Предусловия:** преподаватель запустил приложение и создал либо открыл библиотеку.

**Главная последовательность:**

a) преподаватель нажимает в *окне выбора раздела* кнопку «*Эпохи*»;

b) приложение отображает преподавателю *окно просмотра списка эпох*, содержащее *список имеющихся в библиотеке эпох*, а также кнопки «*Добавить*», «*Изменить*», «*Удалить*» и «*Назад*»;

c) преподаватель нажимает кнопку «*Добавить*»;

d) приложение отображает преподавателю *окно добавления эпохи*, содержащее *область просмотра изображения и поле ввода для названия эпохи*, а также кнопки «*Добавить*» и «*Отмена*»;

e) преподаватель двойным кликом по области изображения вызывает диалоговое окно открытия файла изображения и выбирает изображение эпохи, затем вводит название эпохи и нажимает кнопку «*Добавить*»;

f) приложение добавляет эпоху в библиотеку и отображает преподавателю *окно просмотра списка эпох*, в списке присутствует добавленная эпоха.

**Постусловия:** в библиотеку добавлена эпоха.

**Альтернативная последовательность** (добавление нескольких эпох подряд):

- a) преподаватель нажимает в *окне выбора раздела* кнопку «*Эпохи*»;
- b) приложение отображает преподавателю *окно просмотра списка эпох*, содержащее *список имеющихся в библиотеке эпох*, а также кнопки «*Добавить*», «*Изменить*», «*Удалить*» и «*Назад*»;
- c) преподаватель нажимает кнопку «*Добавить*»;
- d) приложение отображает преподавателю *окно добавления эпохи*, содержащее область просмотра изображения и поле ввода для названия эпохи, а также кнопки «*Добавить*» и «*Отмена*»;
- e) преподаватель двойным кликом по области изображения вызывает диалоговое окно открытия файла изображения и выбирает изображение эпохи, затем вводит название эпохи, активирует флажок, находящийся левее кнопки «*Добавить*» и нажимает кнопку «*Добавить*»;
- f) приложение добавляет эпоху в библиотеку и очищает формы ввода *окна добавления эпохи*;
- g) преподаватель повторяет действия, начиная с пункта «e», пока не добавит все эпохи, при добавлении последней эпохи преподаватель деактивирует флажок левее кнопки «*Добавить*» перед её нажатием;
- h) приложение отображает преподавателю *окно просмотра списка эпох*, в списке присутствуют все добавленные эпохи.

**Постусловия:** в библиотеку добавлено несколько эпох.

**Название прецедента:** добавление произведения искусства.

**Действующее лицо:** преподаватель.

**Цель:** добавить в библиотеку произведение искусства.

**Предусловия:** преподаватель запустил приложение и создал либо открыл библиотеку; в библиотеке существуют автор и эпоха добавляемого произведения искусства.

**Главная последовательность:**



а) преподаватель нажимает в окне выбора раздела кнопку «Предметы искусства»;

б) приложение отображает преподавателю окно просмотра списка произведений искусства, содержащее список имеющихся в библиотеке произведений искусства, а также кнопки «Добавить», «Изменить», «Удалить» и «Назад»;

с) преподаватель нажимает кнопку «Добавить»;

д) приложение отображает преподавателю *окно добавления произведения искусства*, содержащее область просмотра изображения, поля ввода для названия и информации произведения искусства, выпадающие списки для выбора эпохи и категории, *поле списка авторов*, а также кнопки «Добавить» и «Отмена»;

е) преподаватель двойным кликом по области изображения вызывает диалоговое окно открытия файла изображения и выбирает изображение произведения искусства, затем вводит название и описание произведения искусства и нажимает кнопку «Добавить», находящуюся справа от поля списка авторов;

ф) приложение отображает окно выбора авторов, которое содержит строку поиска и список существующих в библиотеке авторов;

г) преподаватель отмечает одного или нескольких авторов двойным кликом мыши и нажимает кнопку «Выбрать»

h) приложение закрывает окно выбора авторов и добавляет выбранных авторов в список;

і) преподаватель нажимает кнопку «Добавить»;

ј) приложение добавляет произведение искусства в библиотеку и отображает преподавателю окно просмотра списка произведений искусства, в списке присутствует добавленное произведение искусства.

**Постусловия:** в библиотеку добавлено произведение искусства.

**Альтернативная последовательность** (добавление нескольких произведений искусства подряд):

а) преподаватель нажимает в *окне выбора раздела* кнопку «*Предметы искусства*»;

б) приложение отображает преподавателю *окно просмотра списка произведений искусства*, содержащее список имеющихся в библиотеке произведений искусства, а также кнопки «*Добавить*», «*Изменить*», «*Удалить*» и «*Назад*»;

с) преподаватель нажимает кнопку «*Добавить*»;

д) приложение отображает преподавателю *окно добавления произведения искусства*, содержащее область просмотра изображения, поля ввода для названия и информации произведения искусства, выпадающие списки для выбора эпохи и категории, поле списка авторов, а также кнопки «*Добавить*» и «*Отмена*»;

е) преподаватель двойным кликом по области изображения вызывает диалоговое окно открытия файла изображения и выбирает изображение произведения искусства, затем вводит название и описание произведения искусства и нажимает кнопку «*Добавить*», находящуюся справа от поля списка авторов;

ф) приложение отображает *окно выбора авторов*, которое содержит строку поиска и список существующих в библиотеке авторов;

г) преподаватель отмечает одного или нескольких авторов двойным кликом мыши и нажимает кнопку «*Выбрать*»;

h) приложение закрывает *окно выбора авторов* и добавляет выбранных авторов в список;

и) преподаватель активирует флажок, находящийся левее кнопки «*Добавить*» и нажимает кнопку «*Добавить*»;

j) приложение добавляет произведение искусства в библиотеку и очищает формы ввода *окна добавления произведения искусства*;

к) преподаватель повторяет действия, начиная с пункта «е», пока не добавит все произведения искусства, при добавлении последнего произведения преподаватель деактивирует флажок левее кнопки *«Добавить»* перед её нажатием;

л) приложение отображает преподавателю *окно просмотра списка произведений искусства*, в списке присутствуют все добавленные произведения искусства.

**Постусловия:** в библиотеку добавлено несколько произведений искусства.

**Альтернативная последовательность** (добавление нескольких произведений искусства с однотипными данными):

а) преподаватель нажимает в *окне выбора раздела* кнопку *«Предметы искусства»*;

б) приложение отображает преподавателю *окно просмотра списка произведений искусства*, содержащее список имеющихся в библиотеке произведений искусства, а также кнопки *«Добавить»*, *«Изменить»*, *«Удалить»* и *«Назад»*;

с) преподаватель нажимает кнопку *«Добавить»*;

д) приложение отображает преподавателю *окно добавления произведения искусства*, содержащее область просмотра изображения, поля ввода для названия и информации произведения искусства, выпадающие списки для выбора эпохи и категории, поле списка авторов, а также кнопки *«Добавить»* и *«Отмена»*;

е) преподаватель вводит данные, которые будут повторяться для следующих произведений искусства (например: категория и эпоха), нажимает кнопку *«шаблон»* в нижней части экрана и выбирает пункт *«Сохранить»*;

ф) приложение записывает введенные данные как шаблон в конфигурационный файл;

г) преподаватель двойным кликом по области изображения вызывает диалоговое окно открытия файла изображения и выбирает изображение произведения искусства, затем вводит данные, не содержащиеся в шаблоне, активирует флажок, находящийся левее кнопки «Добавить» и нажимает кнопку «Добавить»;

h) приложение добавляет произведение искусства в библиотеку и очищает формы ввода *окна добавления произведения искусства*, кроме тех, что сохранены в шаблоне;

и) преподаватель повторяет действия, начиная с пункта «г», пока не добавит все произведения искусства, при добавлении последнего произведения преподаватель деактивирует флажок левее кнопки «Добавить» перед её нажатием;

j) приложение отображает преподавателю *окно просмотра списка произведений искусства*, в списке присутствуют все добавленные произведения искусства.

**Постусловия:** в библиотеку добавлено несколько произведений искусства.

**Название прецедента:** удаление произведения искусства.

**Действующее лицо:** преподаватель.

**Цель:** удалить из библиотеки произведение искусства.

**Предусловия:** преподаватель запустил приложение и открыл библиотеку; в библиотеке существует хотя бы одно произведение искусства.

**Главная последовательность:**

а) преподаватель нажимает в *окне выбора раздела* кнопку «Предметы искусства»;

б) приложение отображает преподавателю *окно просмотра списка произведений искусства*, содержащее список имеющихся в библиотеке произведений искусства, а также кнопки «Добавить», «Изменить», «Удалить» и «Назад»;

с) преподаватель выделяет требуемое произведение искусства (или несколько произведений искусства с зажатой клавишей «*ctrl*») и нажимает кнопку «*Удалить*»;

д) приложение удаляет выбранные произведения искусства из библиотеки.

**Постусловия:** из библиотеки удалено одно или несколько произведений искусства.

**Название прецедента:** удаление автора.

**Действующее лицо:** преподаватель.

**Цель:** удалить из библиотеки автора.

**Предусловия:** преподаватель запустил приложение и открыл библиотеку; в библиотеке существует хотя бы один автор.

**Главная последовательность:**

а) преподаватель нажимает в *окне выбора раздела* кнопку «*Авторы*»;

б) приложение отображает преподавателю *окно просмотра списка авторов*, содержащее *список имеющихся в библиотеке авторов*, а также кнопки «*Добавить*», «*Изменить*», «*Удалить*» и «*Назад*»;

с) преподаватель выделяет требуемого автора (или несколько авторов с зажатой клавишей «*ctrl*») и нажимает кнопку «*Удалить*»;

д) приложение удаляет выбранных авторов из библиотеки, очищает поля в произведениях искусства, если данный автор был выбран;

е) если удаленный автор был выбран в существующих произведениях искусства, в верхней части окна выбора раздела появляется сообщение о наличии ошибок и кнопка для их просмотра.

**Постусловия:** из библиотеки удален автор.

**Название прецедента:** удаление эпохи.

**Действующее лицо:** преподаватель.

**Цель:** удалить из библиотеки эпоху.

**Предусловия:** преподаватель запустил приложение и открыл библиотеку; в библиотеке существует хотя бы одна эпоха.

**Главная последовательность:**

- а) преподаватель нажимает в *окне выбора раздела* кнопку «*Эпохи*»;
- б) приложение отображает преподавателю *окно просмотра списка эпох*, содержащее *список имеющихся в библиотеке эпох*, а также кнопки «*Добавить*», «*Изменить*», «*Удалить*» и «*Назад*»;
- с) преподаватель выделяет требуемую эпоху (или несколько эпох, с зажатой клавишей «*ctrl*») и нажимает кнопку «*Удалить*»;
- д) приложение удаляет выбранные эпохи из библиотеки, очищает поля в произведениях искусства, если данная эпоха был выбрана;
- е) если удаленная эпоха была выбрана в существующих произведениях искусства, в верхней части окна выбора раздела появляется сообщение о наличии ошибок и кнопка для их просмотра.

**Постусловия:** из библиотеки удалена эпоха.

**Название прецедента:** изменение произведения искусства.

**Действующее лицо:** преподаватель.

**Цель:** изменить имеющееся в библиотеке произведение искусства.

**Предусловия:** преподаватель запустил приложение и открыл библиотеку; в библиотеке существует хотя бы одно произведение искусства.

**Главная последовательность:**

- а) преподаватель нажимает в *окне выбора раздела* кнопку «*Предметы искусства*»;
- б) приложение отображает преподавателю *окно просмотра списка произведений искусства*, содержащее *список имеющихся в библиотеке произведений искусства*, а также кнопки «*Добавить*», «*Изменить*», «*Удалить*» и «*Назад*»;
- с) преподаватель выделяет необходимое произведение искусства левой кнопкой мыши и нажимает кнопку «*Изменить*»;

d) приложение отображает преподавателю *окно редактирования произведения искусства*, содержащее *область просмотра изображения*, *поля ввода для названия и информации произведения искусства*, *выпадающие списки для выбора эпохи и категории*, *поле списка авторов*, а также кнопки «Изменить» и «Отмена»;

e) преподаватель изменяет необходимые данные и нажимает кнопку «Изменить»;

f) приложение сохраняет изменения и отображает преподавателю *окно просмотра списка произведений искусства*.

**Постусловия:** изменены данные произведения искусства.

**Название прецедента:** изменение автора.

**Действующее лицо:** преподаватель.

**Цель:** изменить имеющегося в библиотеке автора.

**Предусловия:** преподаватель запустил приложение и открыл библиотеку; в библиотеке существует хотя бы один автор.

**Главная последовательность:**

a) преподаватель нажимает в *окне выбора раздела* кнопку «Авторы»;

b) приложение отображает преподавателю *окно просмотра списка авторов*, содержащее *список имеющихся в библиотеке авторов*, а также кнопки «Добавить», «Изменить», «Удалить» и «Назад»;

c) преподаватель выделяет необходимое произведение искусства *левой кнопкой мыши* и нажимает кнопку «Изменить»;

d) приложение отображает преподавателю *окно редактирования автора*, содержащее *область просмотра изображения*, *поля ввода для названия и информации об авторе*, а также кнопки «Изменить» и «Отмена»;

e) преподаватель изменяет необходимые данные и нажимает кнопку «Изменить»;

f) приложение сохраняет изменения и отображает преподавателю *окно просмотра списка авторов*;

г) если измененный автор был выбран в существующих произведениях искусства, приложение обновляет данные в соответствующих полях.

**Постусловия:** изменены данные автора.

**Название прецедента:** изменение эпохи.

**Действующее лицо:** преподаватель.

**Цель:** изменить имеющуюся в библиотеке эпоху.

**Предусловия:** преподаватель запустил приложение и открыл библиотеку; в библиотеке существует хотя бы одна эпоха.

**Главная последовательность:**

а) преподаватель нажимает в *окне выбора раздела* кнопку «*Эпохи*»;

б) приложение отображает преподавателю *окно просмотра списка эпох*, содержащее *список имеющихся в библиотеке эпох*, а также кнопки «*Добавить*», «*Изменить*», «*Удалить*» и «*Назад*»;

с) преподаватель выделяет необходимое произведение искусства левой кнопкой мыши и нажимает кнопку «*Изменить*»;

д) приложение отображает преподавателю *окно редактирования эпохи*, содержащее *область просмотра изображения*, *поле ввода для названия эпохи*, а также кнопки «*Изменить*» и «*Отмена*»;

е) преподаватель изменяет необходимые данные и нажимает кнопку «*Изменить*»;

ф) приложение сохраняет изменения и отображает преподавателю *окно просмотра списка эпох*;

г) если измененная эпоха была выбрана в существующих произведениях искусства, приложение обновляет данные в соответствующих полях.

**Постусловия:** изменены данные эпохи.

**Название прецедента:** просмотр ошибок.

**Действующее лицо:** преподаватель или администратор.

**Цель:** просмотреть имеющиеся в библиотеке ошибки.



**Предусловия:** преподаватель запустил приложение и открыл библиотеку; в библиотеке имеются ошибки.

**Главная последовательность:**

- a) пользователь нажимает в окне выбора раздела кнопку «Подробнее», расположенную в верхней части окна, справа от надписи «Найдены ошибки»;
- b) приложение отображает преподавателю окно ошибок, содержащее список найденных в библиотеке ошибок, а также кнопку «Закрыть»;
- c) преподаватель просматривает ошибки и нажимает кнопку «Закрыть»;
- d) приложение закрывает окно ошибок.

**Постусловия:** пользователь получил сведения об имеющихся ошибках.

**Название прецедента:** объединение библиотек.

**Действующее лицо:** администратор.

**Цель:** объединить две библиотеки.

**Предусловия:** администратор запустил приложение и создал либо открыл библиотеку.

**Главная последовательность:**

- a) администратор нажимает в *окне выбора раздела* кнопку «Объединить», находящуюся в правой нижней части окна;
- b) приложение отображает *диалоговое окно открытия файла*;
- c) администратор переходит к расположению файла библиотеки, которую необходимо добавить, выделяет его и нажимает кнопку «Открыть»;
- d) приложение копирует данные из добавляемой библиотеки в открытую и выводит диалоговое окно со списком добавленных данных;
- e) администратор просматривает данные и нажимает кнопку «Закрыть»;
- f) приложение отображает *окно выбора раздела*.

**Постусловия:** объединены две библиотеки.

**Название прецедента:** экспорт библиотеки.

**Действующее лицо:** администратор.

**Цель:** подготовить библиотеку к загрузке на веб-сервер.

**Предусловия:** администратор запустил приложение и открыл библиотеку; в библиотеке отсутствуют ошибки.

**Главная последовательность:**

а) администратор в *окне выбора раздела* нажимает правой кнопкой мыши на кнопку «*Сохранить*» и выбирает пункт «*Экспорт в JSON*»;

б) приложение отображает *окно экспорта*, содержащее поле ввода *префикса пути для файлов* и кнопки «*Экспортировать*» и «*Отмена*»;

с) администратор вводит необходимый *префикс пути* и нажимает кнопку «*Экспортировать*»;

д) приложение отображает диалоговое окно сохранения файла .json;

е) администратор вводит имя файла, выбирает путь и нажимает кнопку «*Сохранить*»;

ф) приложение создает в выбранном расположении файл .json с заданным именем, копирует в данное расположение файловую структуру библиотеки и выводит диалоговое окно с сообщением об успешном экспорте.

**Постусловия:** данные подготовлены для загрузки на веб-сервер.

Приложение должно отвечать следующим требованиям:

- автоматическое формирование файла JSON;
- обработка ошибок незаполненных полей;
- обработка ошибок использования несуществующих записей;
- обработка ошибок отсутствия файлов;
- возможность редактирования существующей библиотеки;
- возможность объединения нескольких библиотек;
- переименование файлов в допустимый алфавит;
- возможность добавления записей по шаблону;
- возможность удаления записей;
- сохранение настроек пользователя (шаблонов, путей, размеров регулируемых изображений).

## **1.2.2 Нефункциональные требования**

Приложение работает под ОС Windows.

Для использования приложения не требуется обладать глубокими познаниями в сфере информационных технологий и программировании.

## **1.3 Выбор инструментов**

Приложению необходима библиотека для валидации формата JSON и простой способ создания пользовательского интерфейса. Для этих задач неплохо подходит современный фреймворк Qt [2]. Основным языком программирования для фреймворка Qt является C++

## **1.4 Выводы по главе**

На основе текущего способа работы с библиотекой сделаны выводы:

- обработка библиотеки в ручном режиме является трудоемким процессом и может повлечь за собой появление ошибок;
- поскольку задача является узкоспециализированной, готовые инструменты отсутствуют;
- формирование библиотеки в ручном режиме требует от администратора относительно глубоких познаний в сфере ИТ и программировании;
- составлена и описана диаграмма вариантов использования;
- сформулированы требования к разрабатываемому приложению;
- выбраны инструменты разработки.

## 2 Проектирование

### 2.1 Интерфейс приложения

Структурная схема интерфейса приложения представлена на рисунке 2.1.

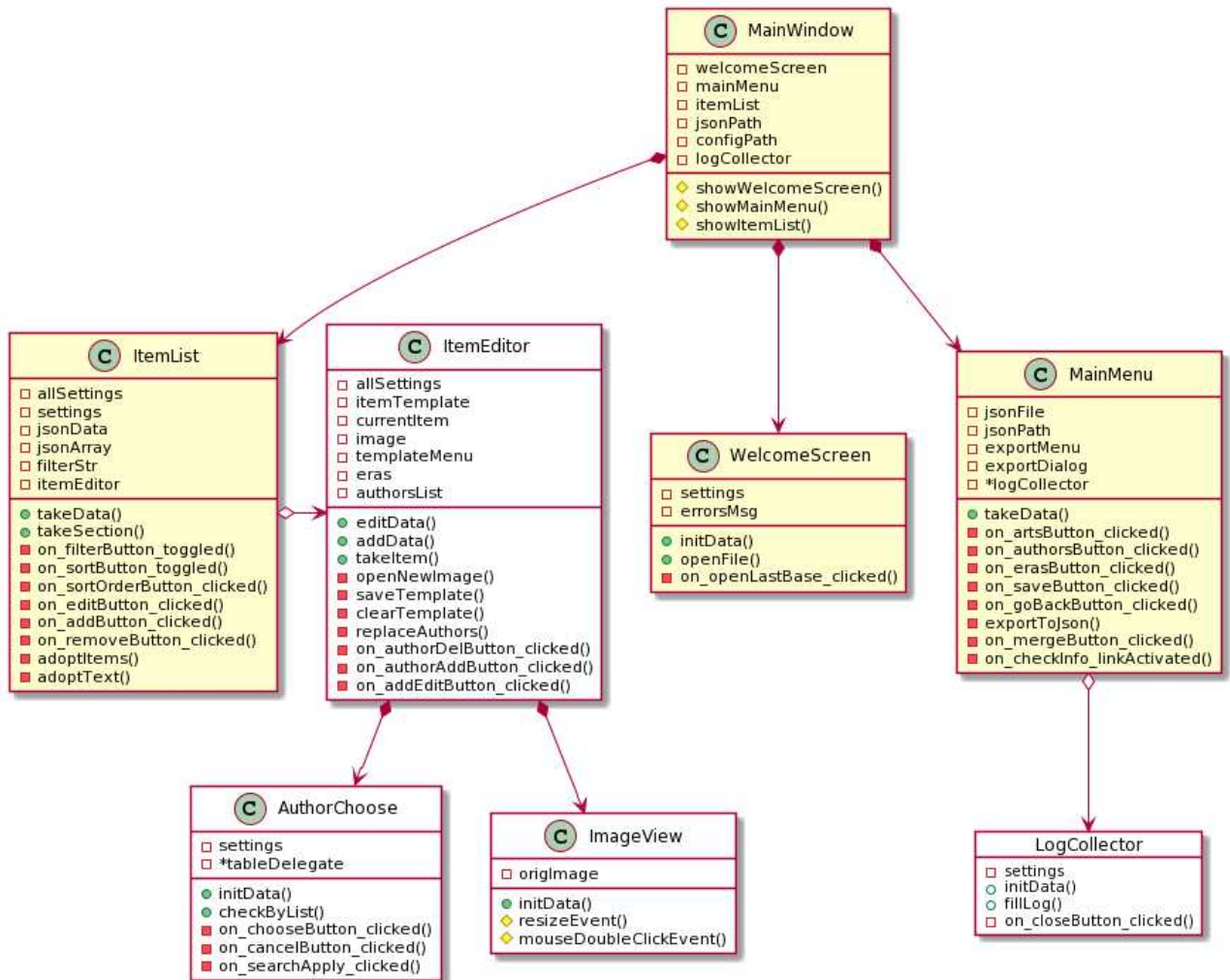


Рисунок 2.1 – Структурная схема интерфейса приложения

MainWindow – основное окно приложения, не имеет графического оформления, служит родительским окном, в котором рисуются дочерние виджеты.

WelcomeScreen – виджет начального окна, открываемого при старте приложения. Имеет кнопки «Открыть» и «Создать», по нажатию на которые открывается диалог открытия/сохранения файла библиотеки, а так же кнопку «Открыть [имя файла]», по нажатию на которую открывается последний использованный файл библиотеки. Макет приведен на рисунке 2.2.



Рисунок 2.2 – Виджет начального окна

MainMenu – виджет окна выбора раздела, кнопки соответствуют разделам в файле библиотеки. При нажатии одной из кнопок происходит открытие ItemList для соответствующего раздела. Так же имеет кнопку «Сохранить» для записи изменений в файл библиотеки, кнопку «Объединить» для добавления данных из другой библиотеки и кнопку «Назад» для возврата в WelcomeScreen. В верхней левой части находится информация о библиотеке и имеющихся ошибках. Макет представлен на рисунке 2.3.

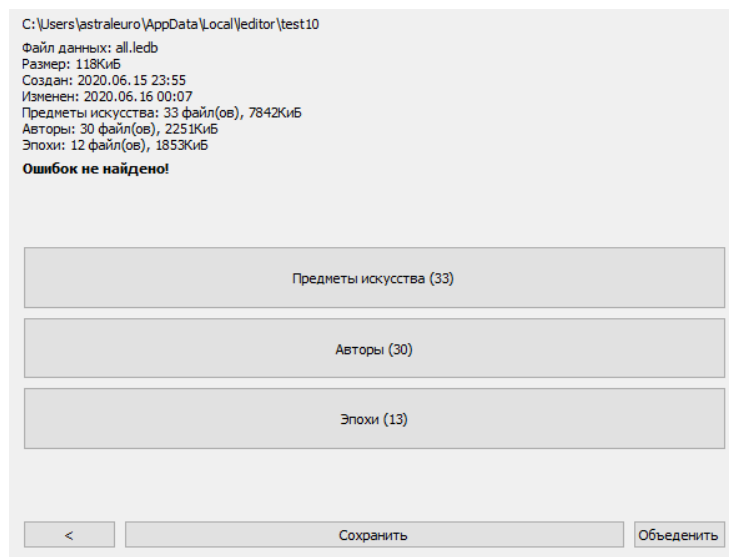


Рисунок 2.3 – Виджет окна выбора раздела

ItemList – виджет окна списка записей, имеет кнопки «Фильтр» и «Сортировка», открывающие настройки фильтрации и сортировки записей. В табличном блоке ниже отображается список записей соответствующего раздела. Каждая строка соответствует одной записи в библиотеке, в первом

столбце отображается пиктограмма предпросмотра изображения, в правом – основная информация записи. Записи поддерживают множественный выбор с зажатой клавишей «ctrl», для удаления нескольких записей. В нижней части окна располагаются кнопки «Добавить», «Изменить» и «Удалить» для добавления, изменения и удаления записей соответственно, а так же кнопку «Назад» для возврата в MainMenu. Макет представлен на рисунке 2.4.

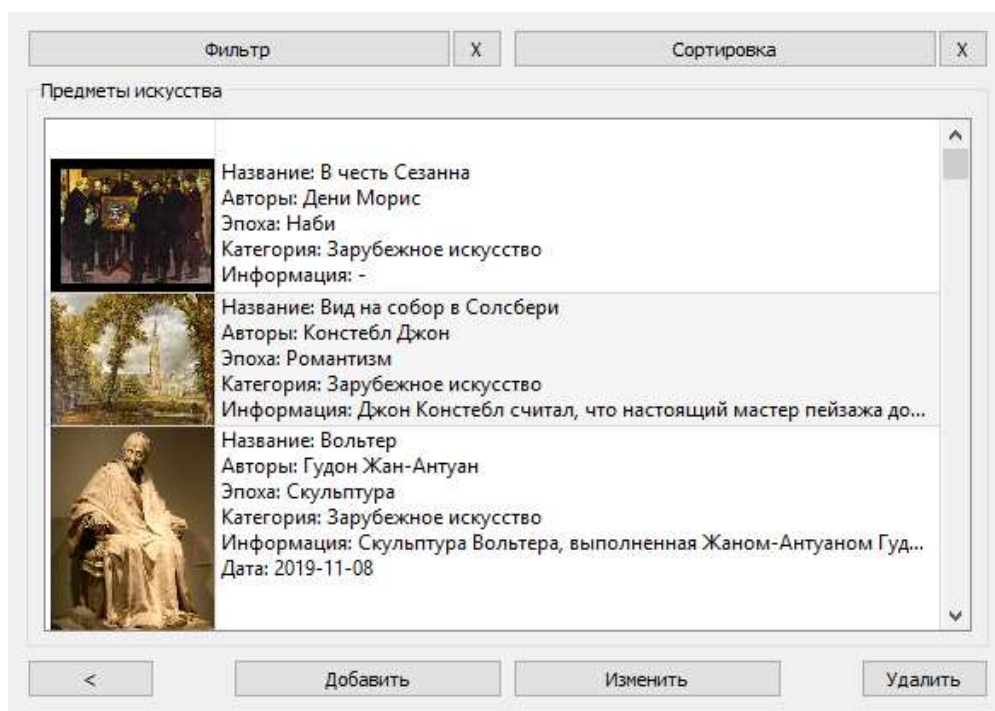


Рисунок 2.4 – Виджет окна просмотра списка записей

ItemEditor – окно редактирования записи. Окно для произведений искусства содержит область просмотра изображения, по двойному клику мышью по которому открывается QFileDialog для выбора файла изображения. Поле «Наименование» является текстовым полем ввода. Поля «Категория» и «Эпоха» выпадающими списками. В поле «Категория» для выбора доступны значения «Зарубежное искусство» и «Отечественное искусство». В поле «Эпоха» для выбора доступны существующие библиотеке эпохи. Поле «Авторы» представляет собой список элементов, значения которых – существующие в библиотеке авторы. Кнопка «-» справа от списка служит для удаления выделенного автора из списка, кнопка «...» справа от списка – для

открытия окна выбора авторов. Поле «Информация» является многострочным текстовым полем ввода. Макет представлен на рисунке 2.5.

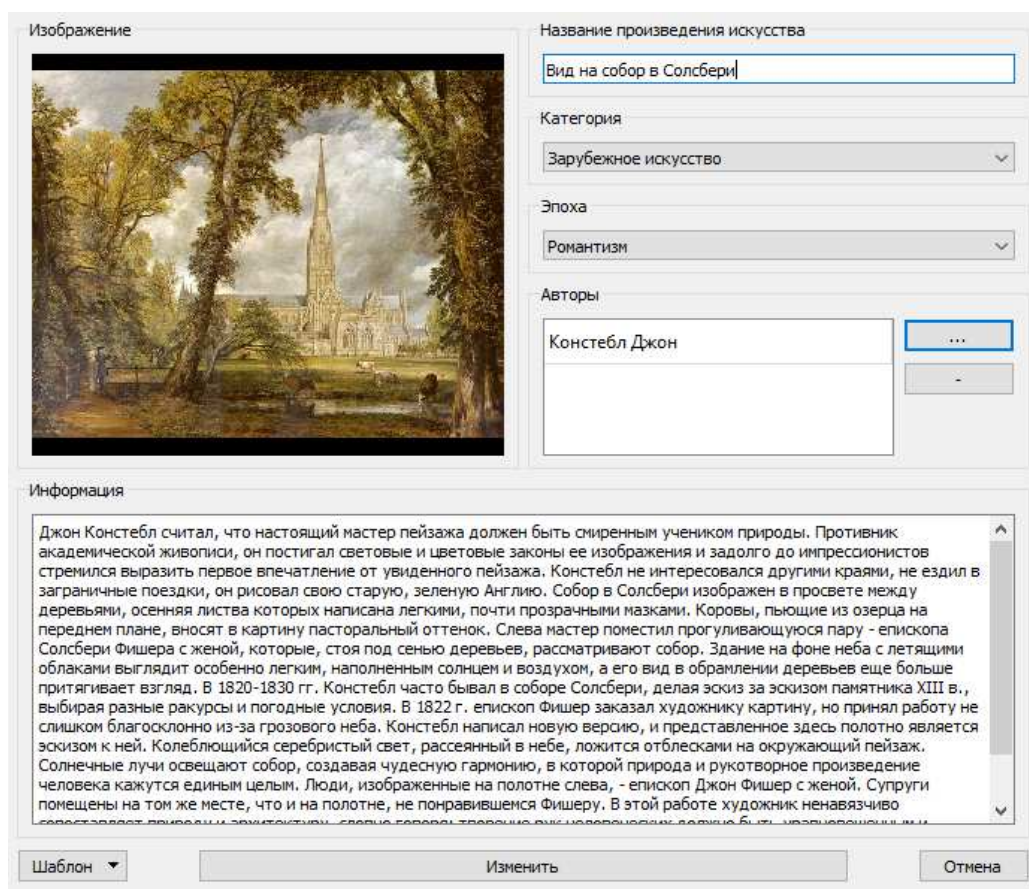


Рисунок 2.5 – Виджет окна редактирования произведения искусства

Окно редактирования записи для авторов и эпох содержит область просмотра изображения, поля «Наименование» и «Информация», которые являются текстовыми полями ввода. Макеты представлены на рисунках 2.6 и 2.7.



Рисунок 2.6 – Виджет окна редактирования автора

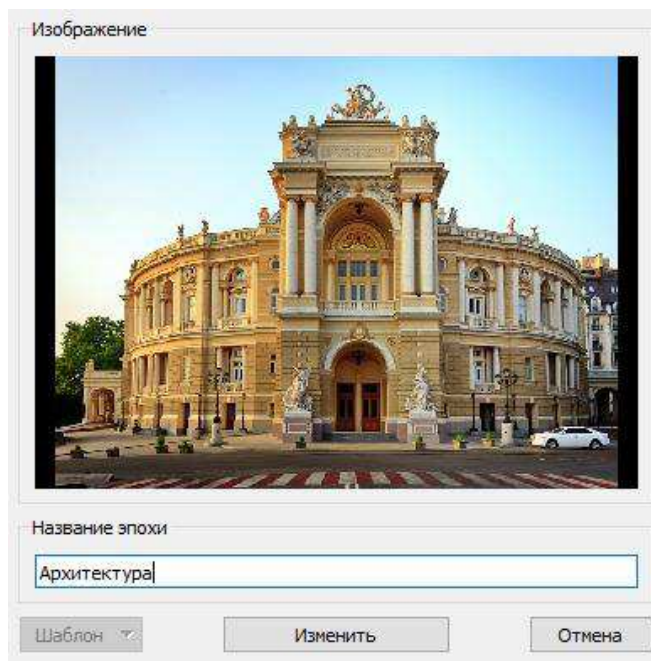


Рисунок 2.7 – Виджет окна редактирования эпохи

AuthorChoose – окно выбора авторов, содержит строку поиска и список доступных в библиотеке авторов. Двойной клик по строке автора добавляет/удаляет автора в списке авторов произведения искусства. Знак «+» в правой части строки свидетельствует о том, что автор выбран в данный момент. Выбор подтверждается кнопкой «Выбрать» в нижней части окна. Макет представлен на рисунке 2.8.

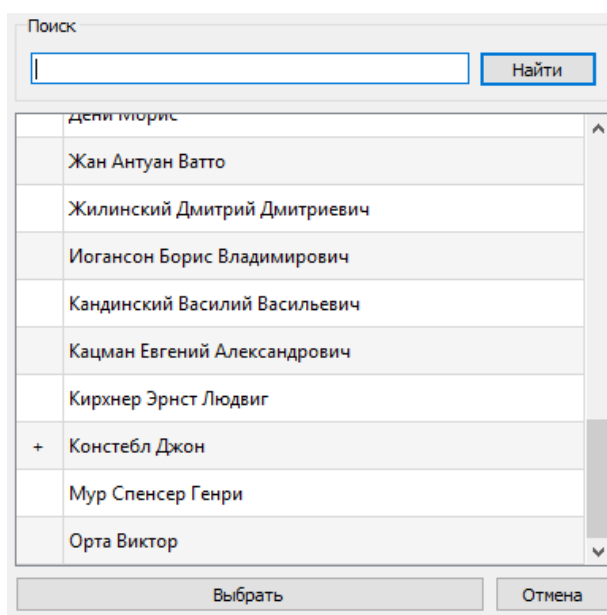


Рисунок 2.8 – Окно выбора авторов



LogCollector – виджет отображения ошибок. Имеет область просмотра ошибок и кнопку «Заккрыть». В области просмотра находится список ошибок, возникающих при использовании несуществующей категории, автора или эпохи, при отсутствии описания, а так же при отсутствии файлов, используемых в библиотеке. Макет представлен на рисунке 2.9.

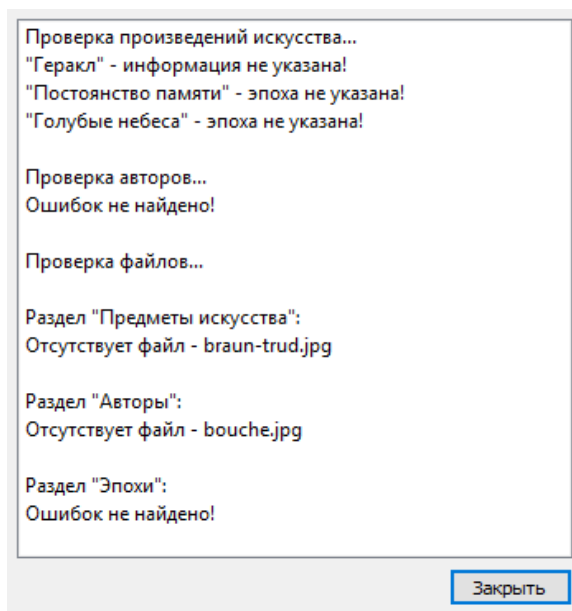


Рисунок 2.9 – Окно отображения ошибок

JsonExport – диалог подготовки библиотеки к загрузке на веб-сервер. Содержит поле ввода для префикса пути к файлам, а так же кнопки «Экспортировать» и «Отмена». После нажатия кнопки «Экспортировать» открывается диалог сохранения файла библиотеки и файловой структуры. Макет представлен на рисунке 2.10.

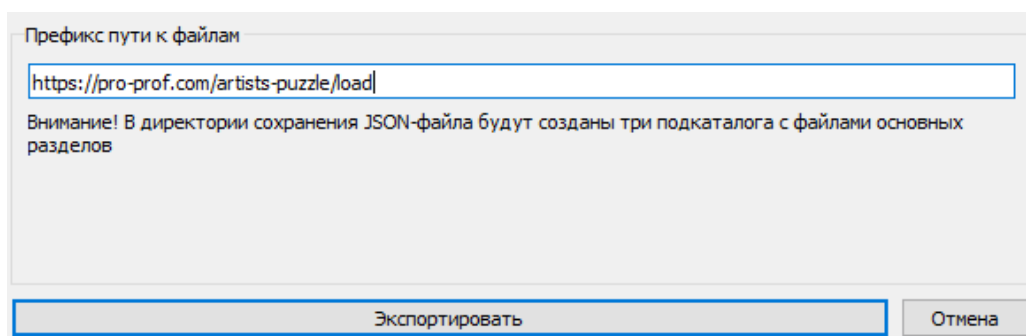


Рисунок 2.10 – Диалог экспорта библиотеки

## 2.2 Общий алгоритм

Для описания алгоритма работы приложения, на основании вариантов использования составлены диаграммы последовательности [3].

Диаграммы последовательности для процесса создания и открытия библиотеки приведены на рисунках 2.11 – 2.13.

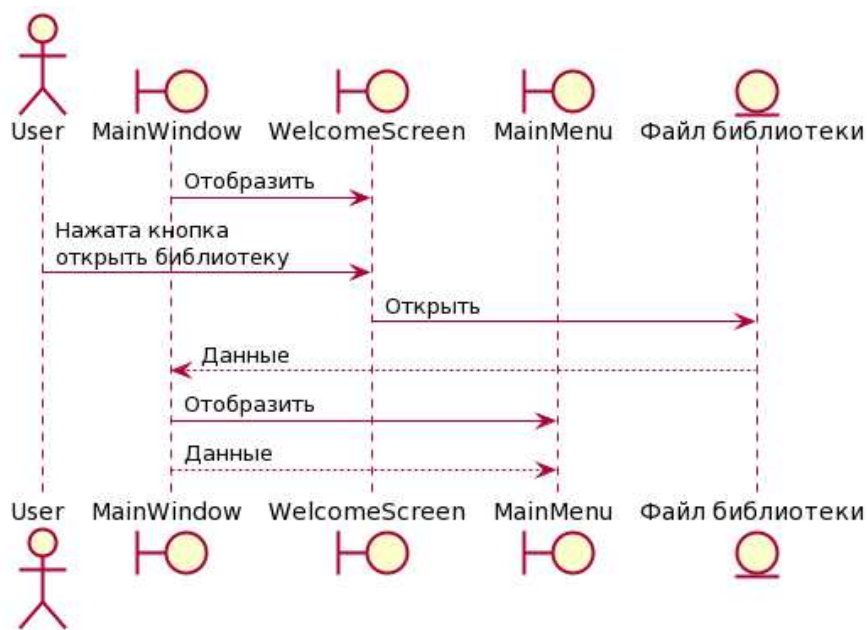


Рисунок 2.11 – Диаграмма последовательности создания библиотеки

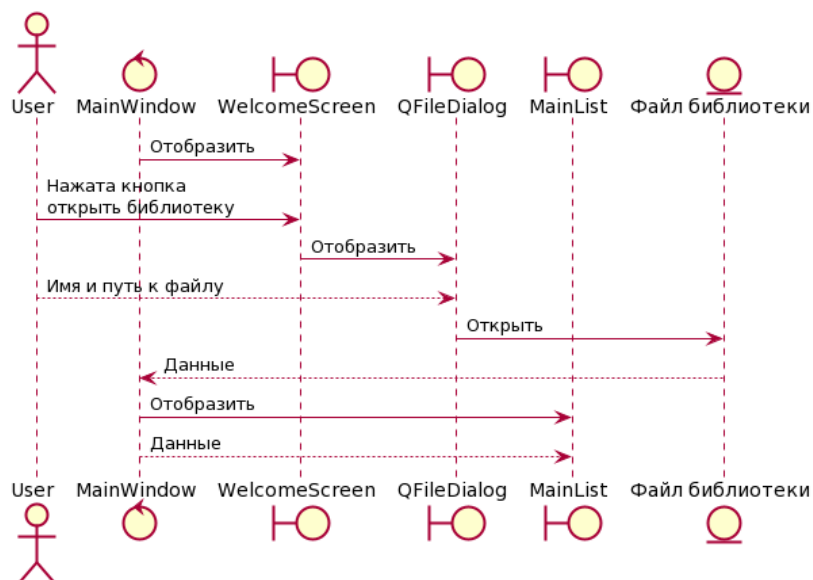


Рисунок 2.12 – Диаграмма последовательности открытия библиотеки

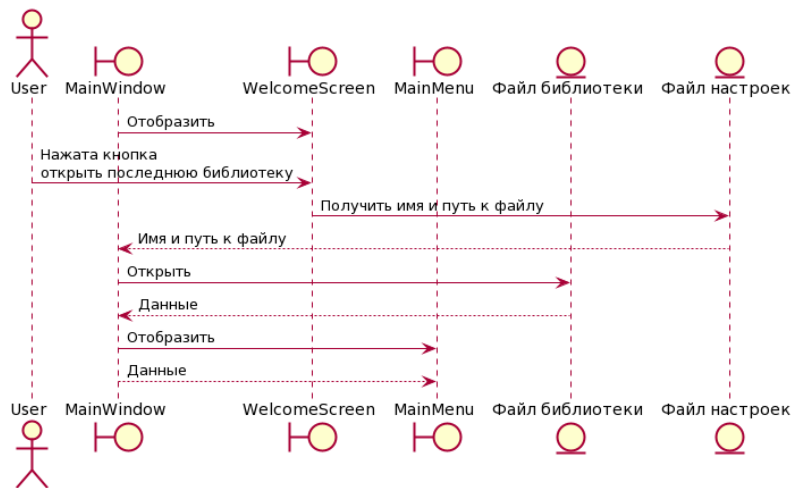


Рисунок 2.13 – Диаграмма последовательности открытия последней библиотеки

Процесс добавления автора и эпохи принципиально не отличается, поэтому представлен только процесс добавления автора. Диаграмма приведена на рисунке 2.14.

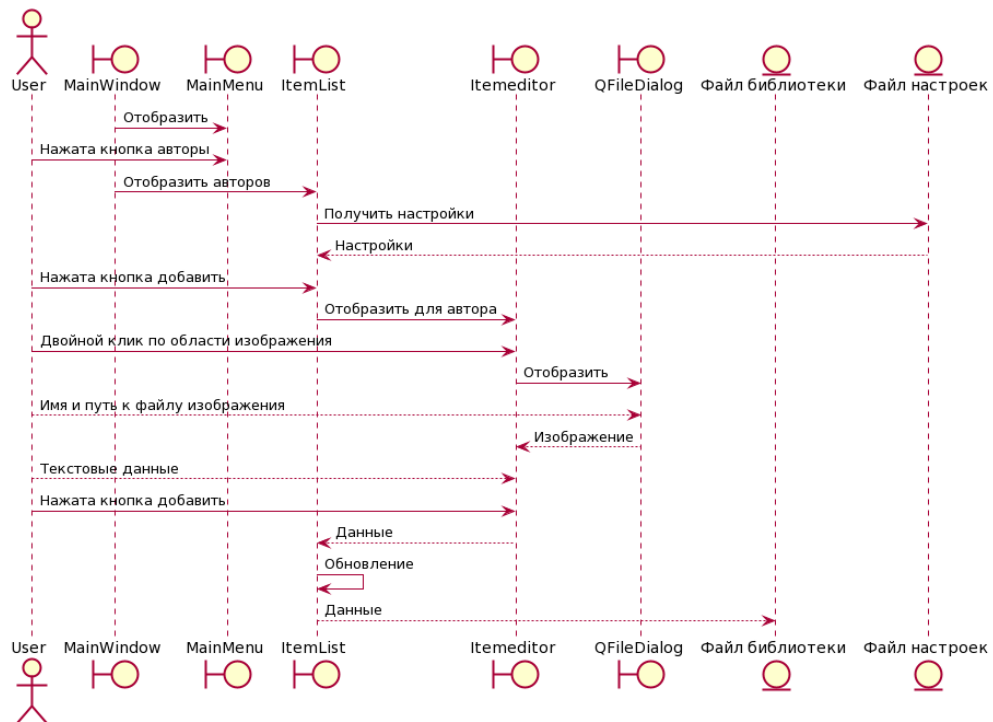


Рисунок 2.14 – Диаграмма последовательности добавления автора

Диаграмма последовательности для процесса добавления произведения искусства представлена на рисунке 2.15.

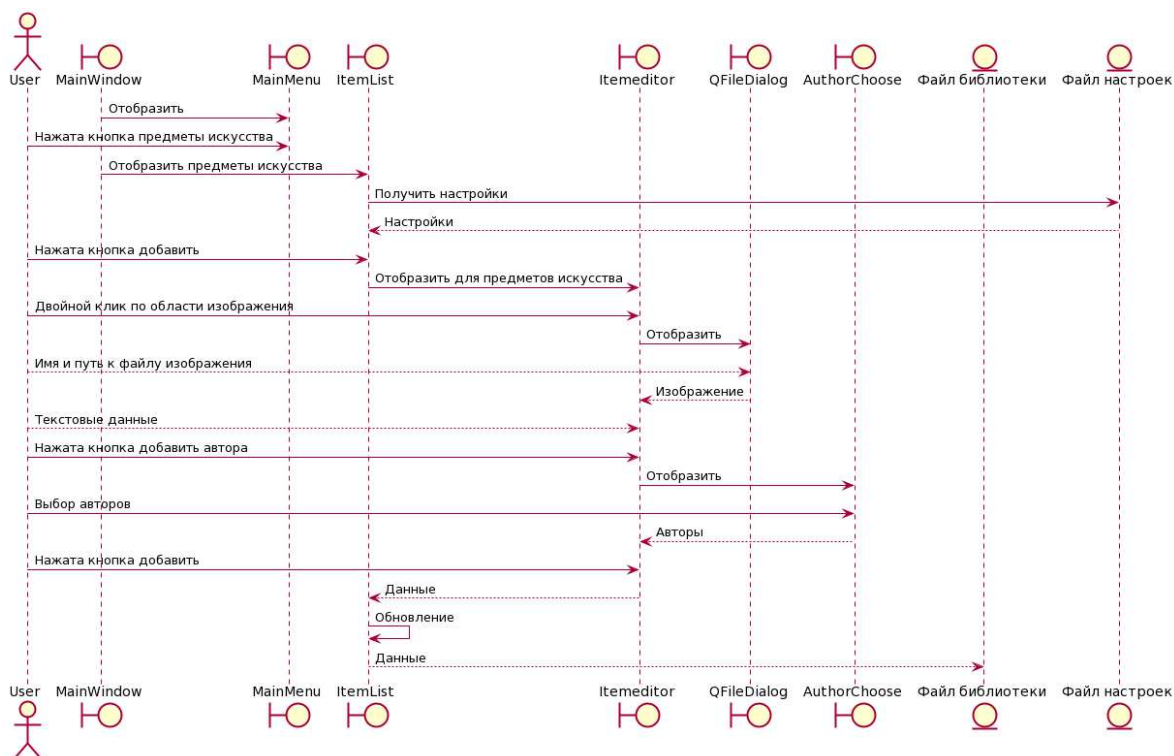


Рисунок 2.15 – Диаграмма последовательности добавления произведения искусства

Процесс удаления автора и эпохи принципиально не отличается, поэтому представлен только процесс удаления автора. Диаграмма приведена на рисунке 2.16.

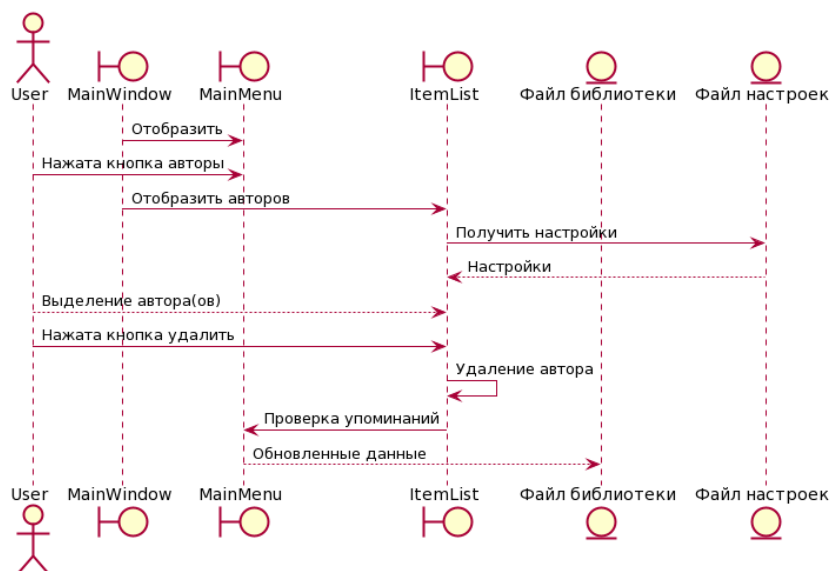


Рисунок 2.16 – Диаграмма последовательности удаления автора

Диаграмма последовательности процесса удаления произведения искусства представлена на рисунке 2.17.

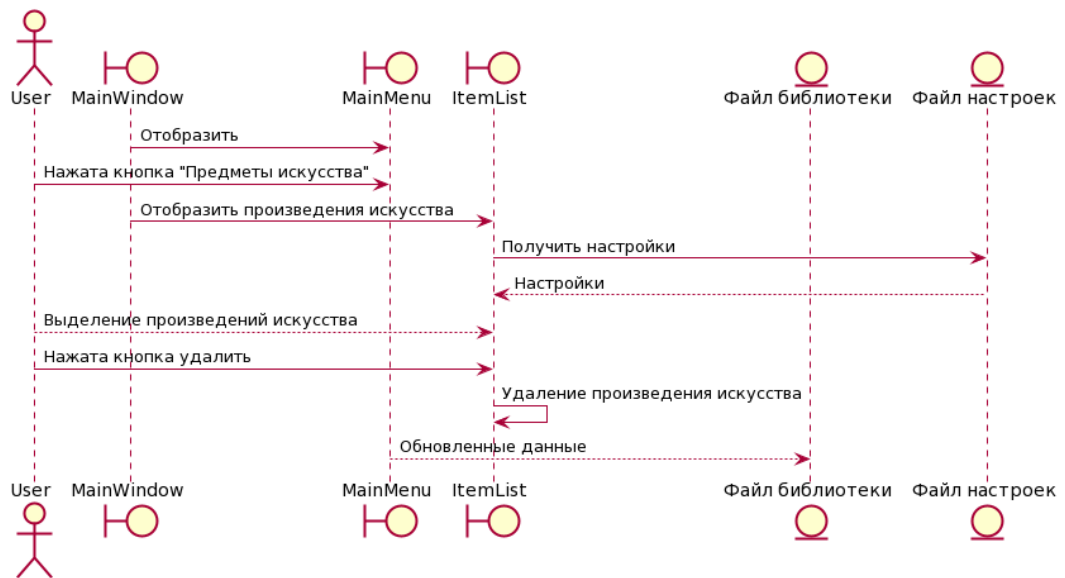


Рисунок 2.17 – Диаграмма последовательности удаления произведения искусства

Процесс изменения автора и эпохи принципиально не отличается, поэтому представлен только процесс изменения автора. Диаграммы последовательности процессов изменения автора и произведения искусства приведены на рисунках 2.18 и 2.19 соответственно.

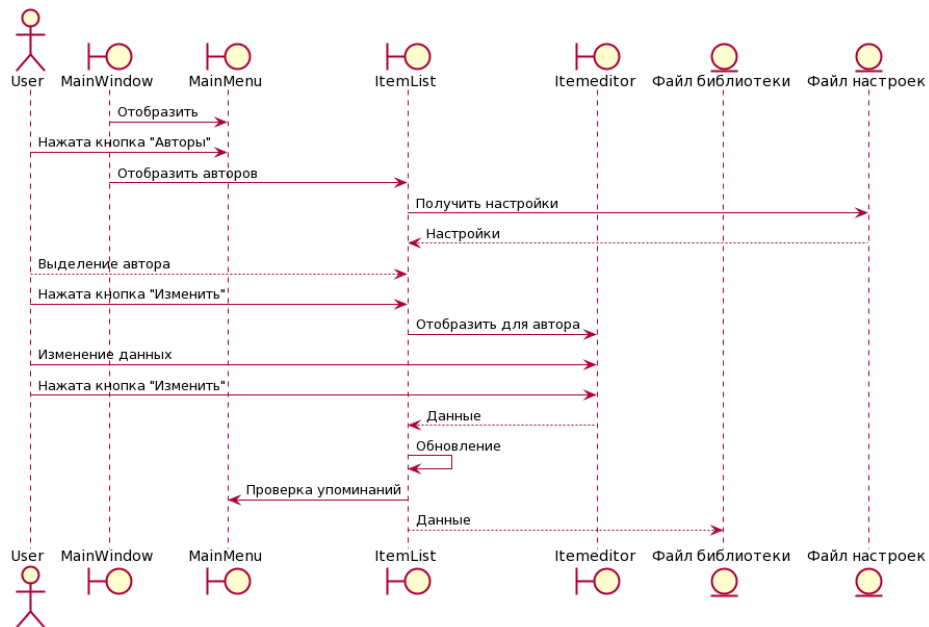


Рисунок 2.18 – Диаграмма последовательности изменения автора

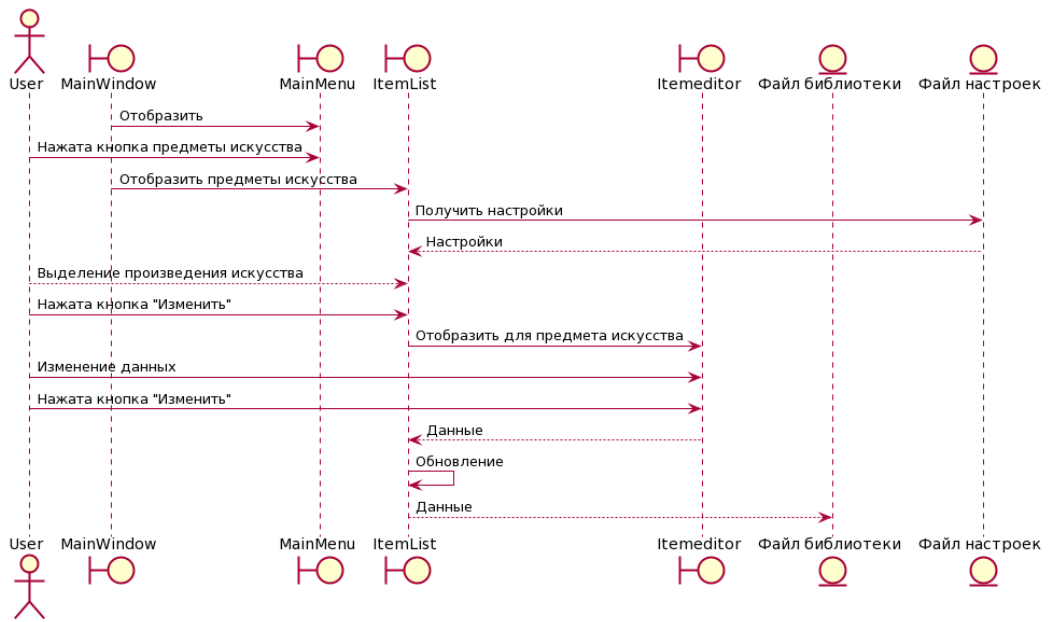


Рисунок 2.19 – Диаграмма последовательности изменения произведения искусства

Диаграмма последовательности процесса просмотра ошибок приведена на рисунке 2.20.

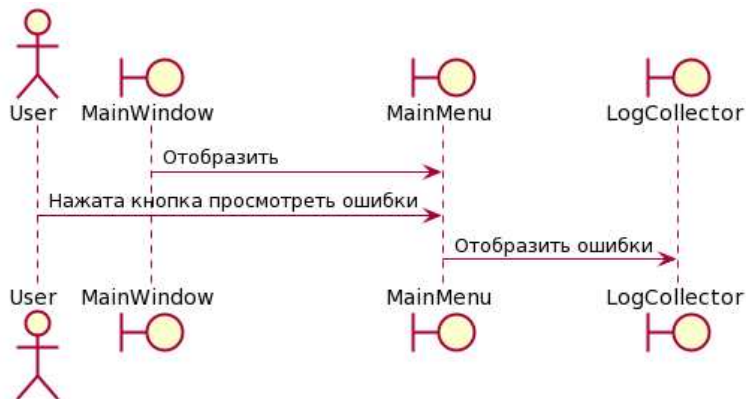


Рисунок 2.20 – Диаграмма последовательности просмотра ошибок

Диаграмма последовательности процесса подготовки библиотеки к загрузке на веб-сервер приведена на рисунке 2.21.

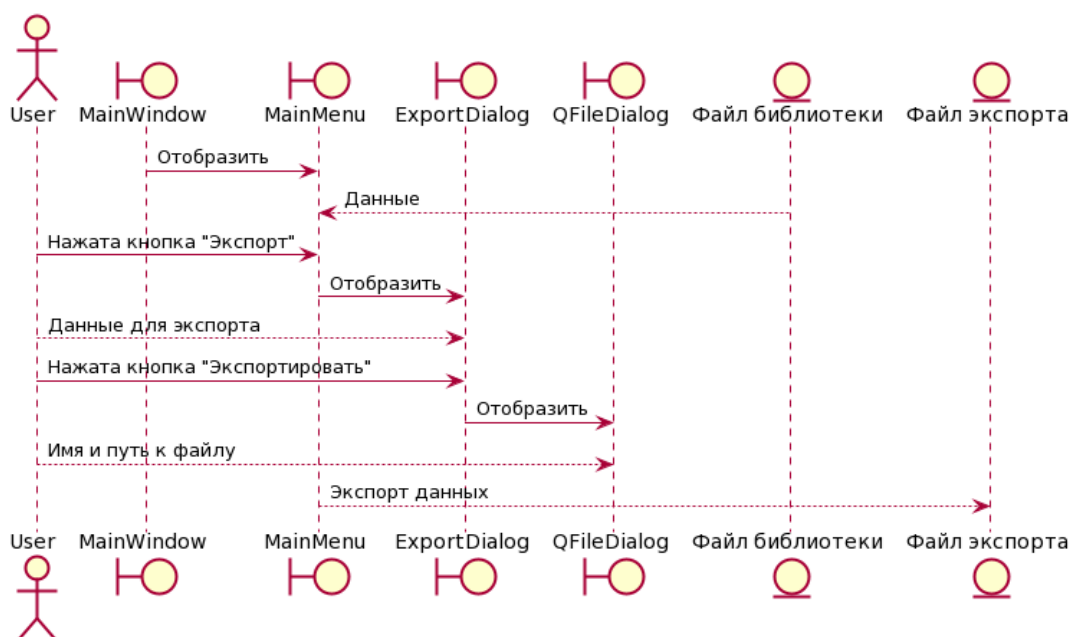


Рисунок 2.21 – Диаграмма последовательности экспорта библиотеки

### 2.3 Алгоритм добавления файла изображения

Для предотвращения ошибок, связанных с некорректными именами файлов, разработан механизм переименования файлов при добавлении в библиотеку, составлен словарь для транслитерации и замены пробельных символов. При добавлении файла в библиотеку, имя файла посимвольно проверяется на соответствие алфавиту: если символ не принадлежит допустимому алфавиту, проверяется наличие символа в словаре замены, если символ присутствует в словаре – производится замена, если нет – символ удаляется. Затем проверяется наличие файла с таким именем, если файл с таким именем существует, в цикле к имени файла добавляется индекс, цикл повторяется, увеличивая индекс, до тех пор, пока не будет найдено имя файла, которого еще не существует в библиотеке. Диаграмма последовательности алгоритма добавления файла в библиотеку приведена на рисунке 2.22.

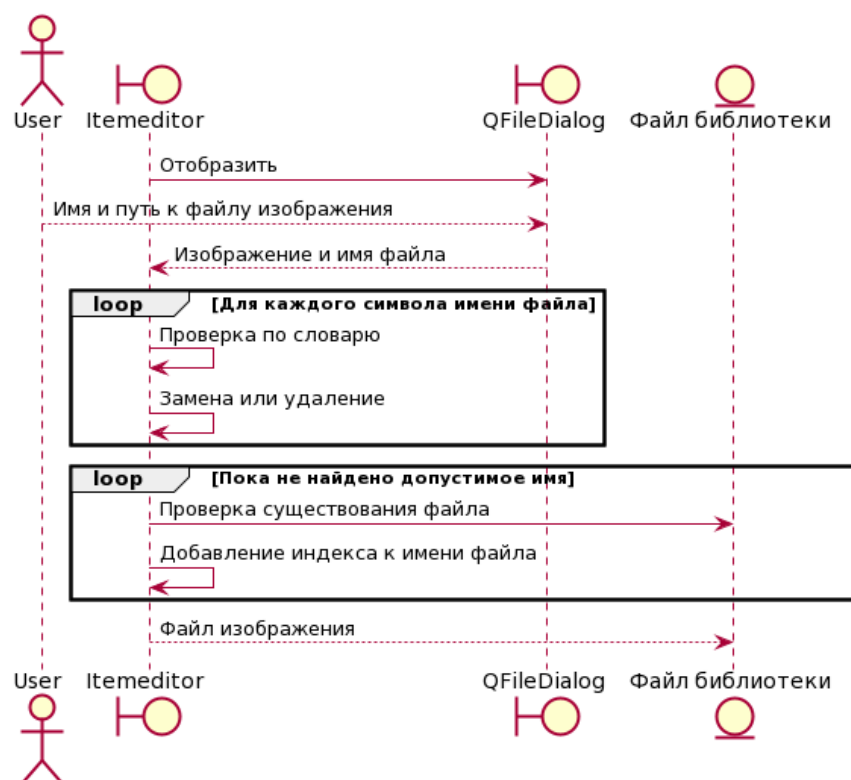


Рисунок 2.22 – Диаграмма последовательности алгоритма добавления файла изображения

## 2.4 Алгоритм слияния библиотек

После открытия файла добавляемой библиотеки запускается перебор записей данной библиотеки. Для каждого элемента ищется совпадение в текущей библиотеке по наименованию записи. Если совпадение найдено, сравнивается дата последнего обновления записей. Если дата последнего обновления во входящей библиотеке новее, запись текущей библиотеки обновляется, а файл изображения заменяется. Если совпадения не были найдены, запись входящей библиотеки дописывается в текущую, файлы изображений копируются в файловую структуру текущей библиотеки. Данная операция повторяется для каждого раздела библиотеки. Диаграмма последовательности алгоритма добавления библиотеки представлена на рисунке 2.23.



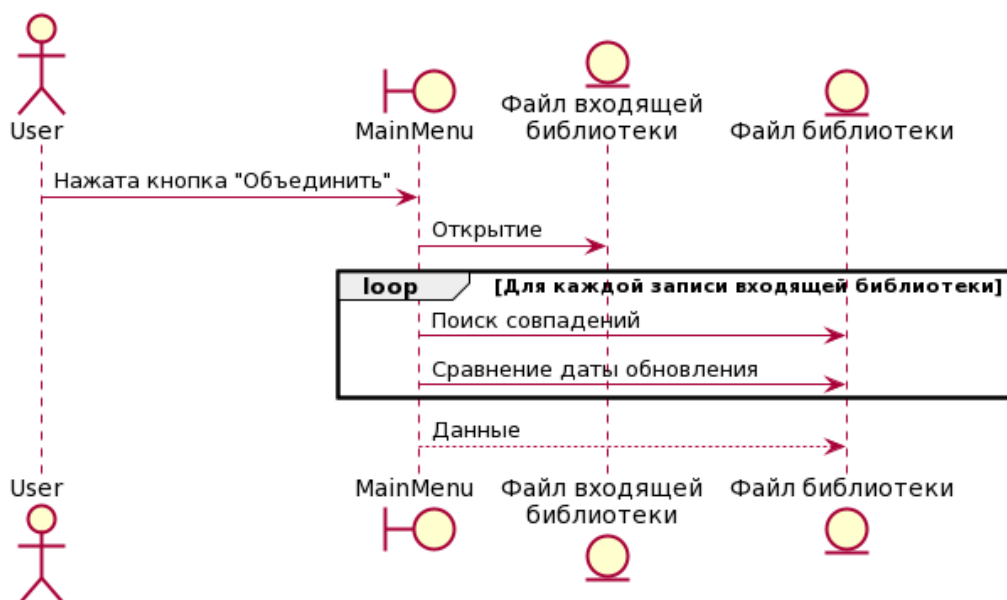


Рисунок 2.23 – Диаграмма последовательности алгоритма слияния библиотек

## 2.5 Выводы по главе

Произведены следующие этапы работ:

- спроектированы и описаны макеты интерфейса для основных окон приложения;
- с помощью диаграмм последовательности, составленных на основе вариантов использования, детализированы алгоритмы приложения;
- разработан алгоритм переименования файлов при добавлении изображений в библиотеку;
- разработан алгоритм слияния нескольких библиотек.

## 3 Реализация и тестирование

### 3.1 Конфигурационный файл

При работе приложения пользовательские настройки, сохраняются в файле настроек «LEditor.conf». Если приложение запущено с ключом «--portable», файл настроек считывается и записывается в каталоге приложения, в противном случае по пути «%LocalAppData%\leditor\LEditor.conf». Если при запуске приложения файл не был найден, он создастся со значениями по умолчанию. В таблицах 3.1 – 3.4 описаны параметры, сохраняемые в конфигурационном файле.

Таблица 3.1 – Секция «P10MainWindow» – настройки главного окна

Параметр	Описание	Значение по умолчанию
mainHeight	Высота окна в пикселях	400
mainWidth	Ширина окна в пикселях	600

Таблица 3.2 – Секция «13WelcomeScreen» – настройки начального окна

Параметр	Описание	Значение по умолчанию
createButton	Имя кнопки создания новой библиотеки	«Создать...»
exitButton	Имя кнопки выхода из приложения	«Выход»
lastButton	Начало имени кнопки открытия последнего файла библиотеки	«Открыть »
lastFile	Путь к последнему использованному файлу библиотеки	null
openButton	Имя кнопки открытия файла библиотеки	«Открыть...»

Таблица 3.3 – Секция «P8MainMenu» – настройки окна выбора раздела

Параметр	Описание	Значение по умолчанию
arts	Имя кнопки режима просмотра списка произведений искусства	«Предметы искусства»
authors	Имя кнопки режима просмотра списка авторов	«Авторы»
eras	Имя кнопки режима просмотра списка эпох	«Эпохи»

Секция «P9ItemList» разделена на три подсекции, отвечающие за отображение виджета в режиме просмотра списка произведений искусства, авторов и эпох. Подсекции имеют схожие параметры, в таблице 3.4 приведены параметры для подсекции «arts».

Таблица 3.4 – Секция «P9ItemList» подсекция «arts»

Параметр	Описание	Значение по умолчанию
filterData	Строка фильтрации	null
filterVisible	Видимость настроек фильтрации	false
imageWidth	Ширина пиктограммы изображения в пикселях	100
itemKeyLabels	Заголовки строк отображаемой информации	Название artName Категория isDomestic Эпоха artEra Авторы artAuthors Информация artInfo Дата lastUpdate
ranks	Наименования категорий	«Отечественное искусство 1 Зарубежное искусство 0»
sortOrder	Направление сортировки	true
sortVisible	Видимость параметров сортировки	false

## 3.2 Инструкция для программиста

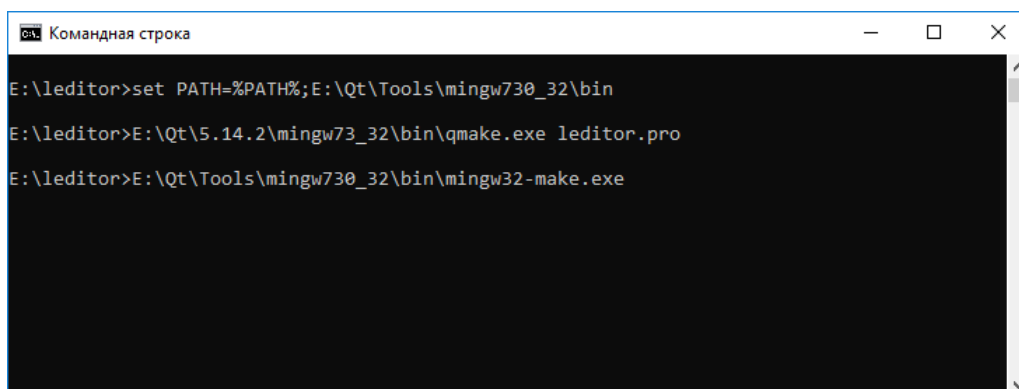
Исходный код написан на языке программирования C++, с использованием библиотеки Qt и выложен в git-репозиторий [11].

Требования для сборки:

- утилита «qmake» из состава Qt, которая помогает облегчить процесс сборки приложения на разных платформах;
- компилятор с поддержкой стандарта C++11 и архитектуры x86 (gcc, mingw);
- библиотека Qt, версии не ниже 5.10 [9].

Сборка под Windows с динамической линковкой библиотек (рисунок 3.1):

- распаковать исходный код и перейти в директорию;
- добавить директорию с компилятором в «переменные окружения»;
- выполнить команду «qmake leditor.pro» для создания Make-файла;
- выполните команду «mingw32-make».



```
cmd - Командная строка
E:\leditor>set PATH=%PATH%;E:\Qt\Tools\mingw730_32\bin
E:\leditor>E:\Qt\5.14.2\mingw73_32\bin\qmake.exe leditor.pro
E:\leditor>E:\Qt\Tools\mingw730_32\bin\mingw32-make.exe
```

Рисунок 3.1 – Сборка проекта под Windows

Зависимости для запуска собранного проекта в Windows без установки

Qt:

- imageformats\qicns.dll;
- imageformats\qico.dll;
- imageformats\qjpeg.dll;
- libgcc\_s\_dw2-1.dll;
- libstdc++-6.dll;
- libwinpthread-1.dll;

- platforms\qwindows.dll;
- qt.conf;
- Qt5Core.dll;
- Qt5Gui.dll;
- Qt5Widgets.dll;
- styles\qwindowsvistastyle.dll.

Данные файлы необходимо взять из установленной версии Qt (на которой собирался проект) и положить в директорию с исполняемым файлом.

Для сборки со статической линковкой библиотек необходимо пересобрать Qt [7], для этого дополнительно потребуется интерпретатор Python [8]. Затем необходимо собрать проект так же, как с динамической линковкой, изменив файл проекта. Пример сборки приведен в приложении А.

### 3.3 Тестирование

Для тестирования с помощью приложения были созданы 2 отдельные библиотеки отечественного и зарубежного искусства (рисунок 3.2).

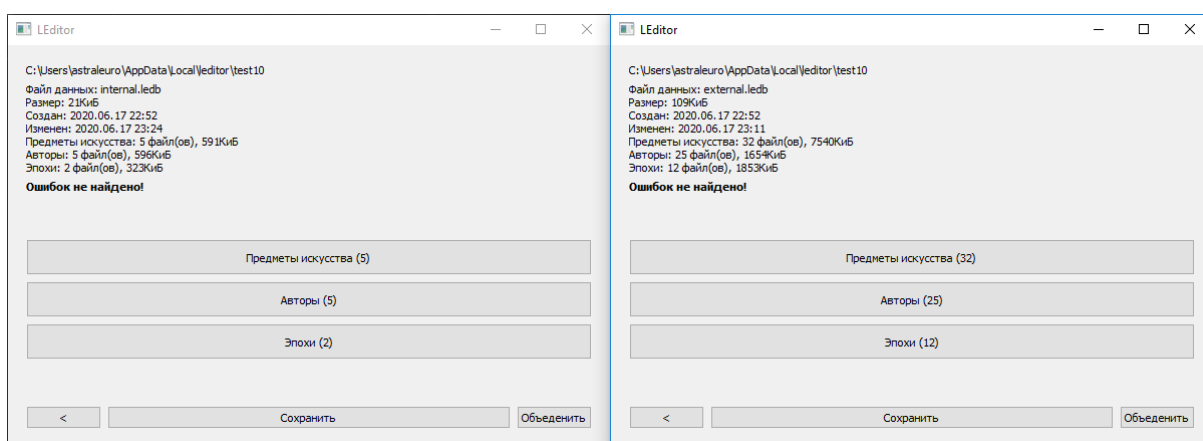


Рисунок 3.2 – Библиотеки отечественного и зарубежного искусства

После чего было выполнено объединение данных библиотек в одну. В библиотеках были повторяющиеся эпохи, объединение прошло корректно, в результирующей библиотеке эпохи не повторяются. Данные о полученной в результате объединения библиотеке приведены на рисунке 3.3.

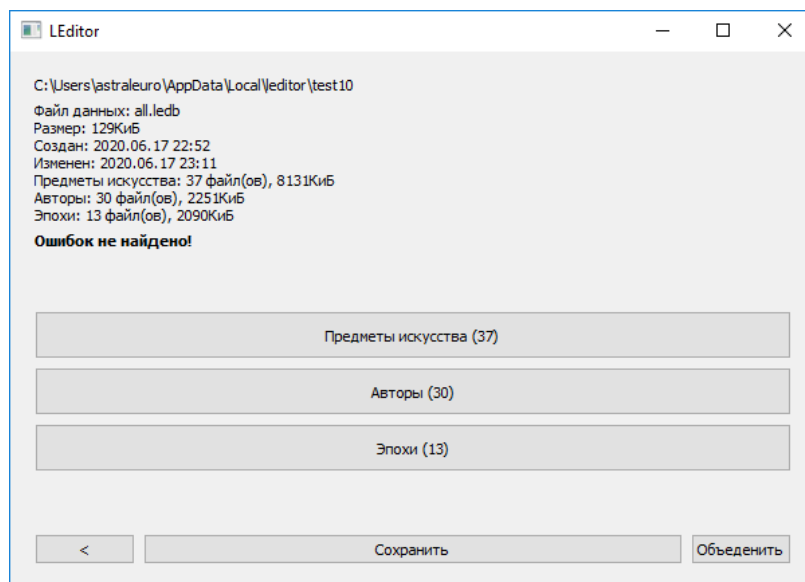


Рисунок 3.3 – Данные библиотеки после объединения

После объединения библиотека была экспортирована и загружена на веб-сервер для проверки работоспособности. Библиотека была успешно загружена приложением «Пазлы-живопись» с веб-сервера.

Процесс работы с библиотекой при помощи приложения приведен в виде диаграммы последовательности на рисунке 3.4.

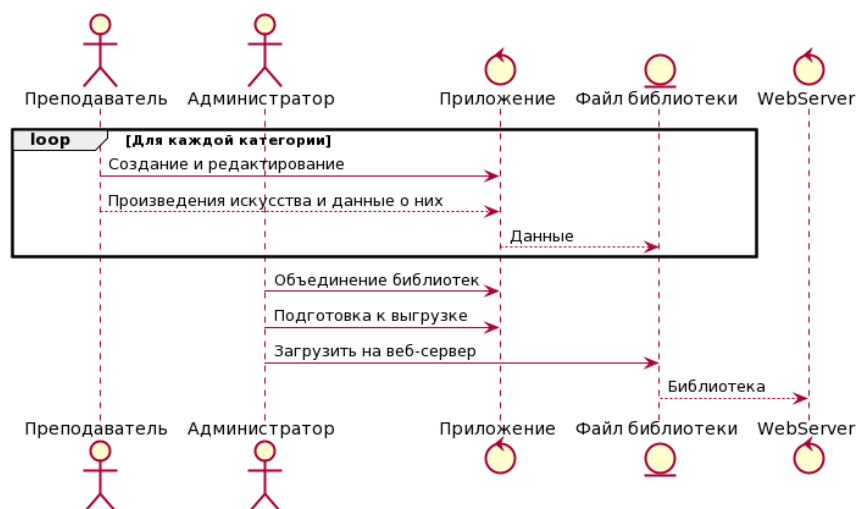


Рисунок 3.4 – Диаграмма последовательности работы с библиотекой с помощью приложения

### 3.4 Выводы по главе

Прделаны следующие этапы работ:

- разработано спроектированное приложение;

- исходный код выложен на git-репозиторий [11];
- составлена инструкция по сборке проекта как с динамической, так и со статической линковкой библиотек;
- проведено тестирование приложения, показавшее, что программное обеспечение соответствует сформулированным требованиям и выполняет функции по облегчению и автоматизации управления данными приложения «Пазлы – живопись».

## ЗАКЛЮЧЕНИЕ

В результате выполнения выпускной квалификационной работы было разработано программное обеспечение для управления данными приложения «Пазлы – живопись», позволяющее автоматизировать и облегчить создание и редактирование библиотеки изображений.

В ходе проделанной работы был проведен анализ текущего способа работы с библиотекой изображений, выявлены его недостатки. Составлена и описана диаграмма вариантов использования, на основании которой сформулированы требования к разрабатываемому программному обеспечению. Были спроектированы макеты интерфейса и алгоритмы работы приложения. Составлена инструкция по сборке проекта из исходного кода.

Для тестирования с помощью приложения были созданы 2 отдельные библиотеки отечественного и зарубежного искусства, после чего было выполнено объединение данных библиотек в одну. Полученная в результате объединения библиотека была экспортирована и загружена веб-сервер для проверки работоспособности.

Программное обеспечение соответствует сформулированным требованиям и выполняет функции по облегчению и автоматизации управления данными приложения «Пазлы – живопись».



## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Введение в JSON [Электронный ресурс] – Режим доступа: <https://www.json.org/json-ru.html>.
2. Документация Qt [Электронный ресурс] – Режим доступа: <http://doc.qt.io>.
3. Основы UML. Диаграммы последовательности [Электронный ресурс] – Режим доступа: <https://pro-prof.com/archives/2769>.
4. Основы UML – диаграммы использования (use-case) [Электронный ресурс] – Режим доступа: <https://pro-prof.com/archives/2594>.
5. СТО 4.2-07-2014 Стандарт организации «Общие требования к построению, изложению и оформлению документов учебной деятельности. – Красноярск : ИПК СФУ, 2014. – 60 с.
6. Брюханова, У. А. Android-приложение для запоминания картин в игровой форме [Электронный ресурс] : выпускная квалификационная работа бакалавра : 09.03.01 / У. А. Брюханова. – Красноярск : СФУ, 2019. – Режим доступа: <http://elib.sfu-kras.ru/handle/2311/125719>.
7. Building a static Qt for Windows using MinGW [Электронный ресурс] – Режим доступа: [https://wiki.qt.io/Building\\_a\\_static\\_Qt\\_for\\_Windows\\_using\\_MinGW](https://wiki.qt.io/Building_a_static_Qt_for_Windows_using_MinGW)
8. Download Python [Электронный ресурс] – Режим доступа: <https://www.python.org/downloads>
9. Download Qt [Электронный ресурс] – Режим доступа: [http://download.qt-project.org/official\\_releases/qt](http://download.qt-project.org/official_releases/qt).
10. JSON – Краткое руководство [Электронный ресурс] – Режим доступа: <https://coderlessons.com/tutorials/java-tekhnologii/vyuchit-json/json-kratkoe-rukovodstvo>.
11. libraryeditor [Электронный ресурс] – Режим доступа: <https://github.com/astraleuro/libraryeditor>.

12. Plant UML [Электронный ресурс] – Режим доступа:  
<http://plantuml.com>.

13. Фаулер, М. UML. Основы / М. Фаулер, К. Скотт; пер. с англ. – СПб.:  
Символ – Плюс, 2002. – 192 с.

## ПРИЛОЖЕНИЕ А

### Сборка проекта со статической линковкой библиотек

Для начала необходимо установить Qt opensource, версии не ниже 5.10, выбор компонентов при установке Qt представлен на рисунке А.1.

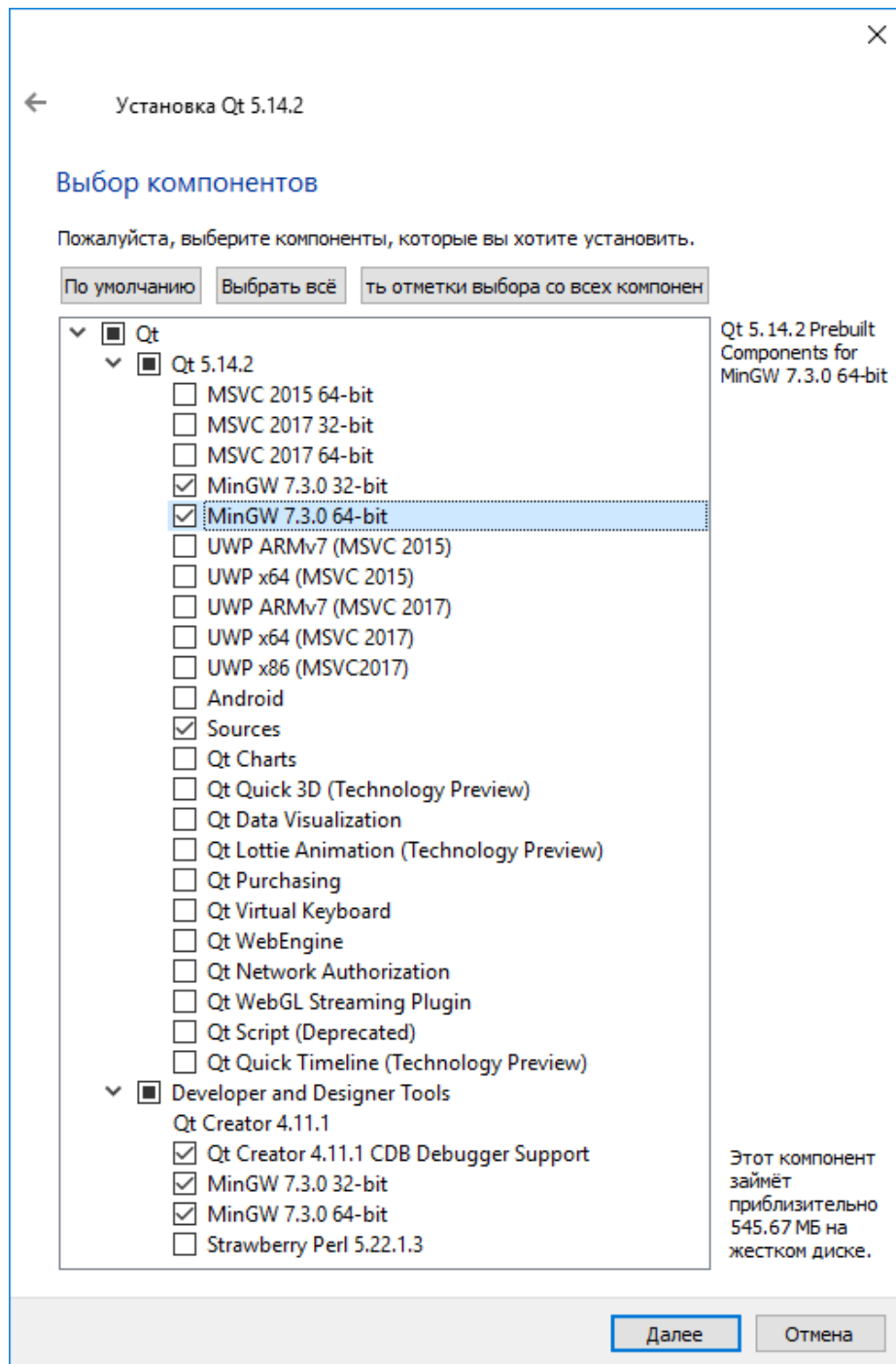


Рисунок А.1 – Выбор компонентов при установке Qt

Далее необходимо установить интерпретатор Python, выбор опций при установке представлен на рисунке А.2.

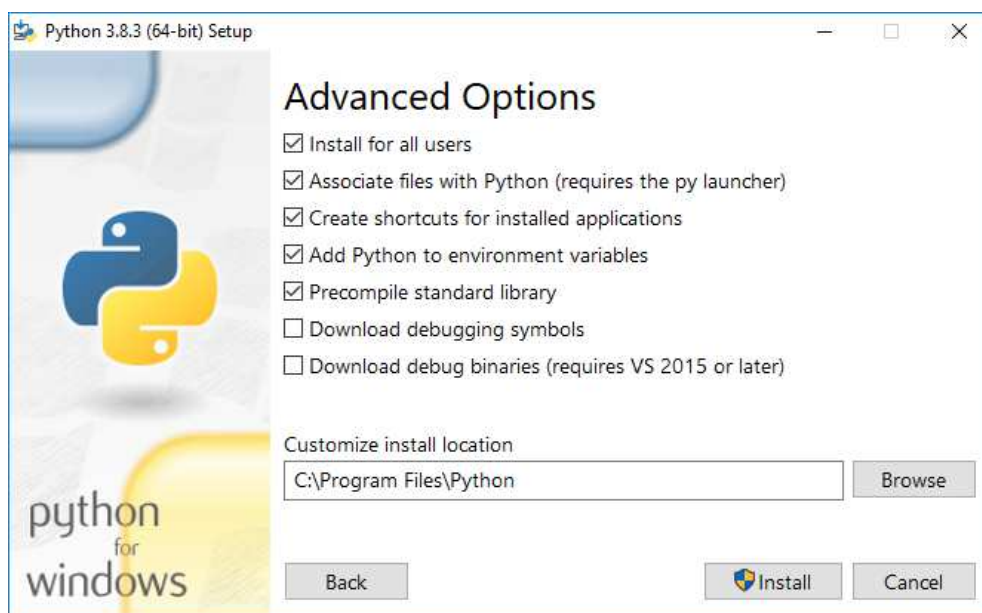


Рисунок А.2 – Выбор опций при установке Python

Затем необходимо изменить «спес»-файл (`..\Qt5.14.2\5.14.2\Src\qtbase\mkspecs\win32-g++\qmake.conf`) для сборки Qt, как показано на рисунке А.3.

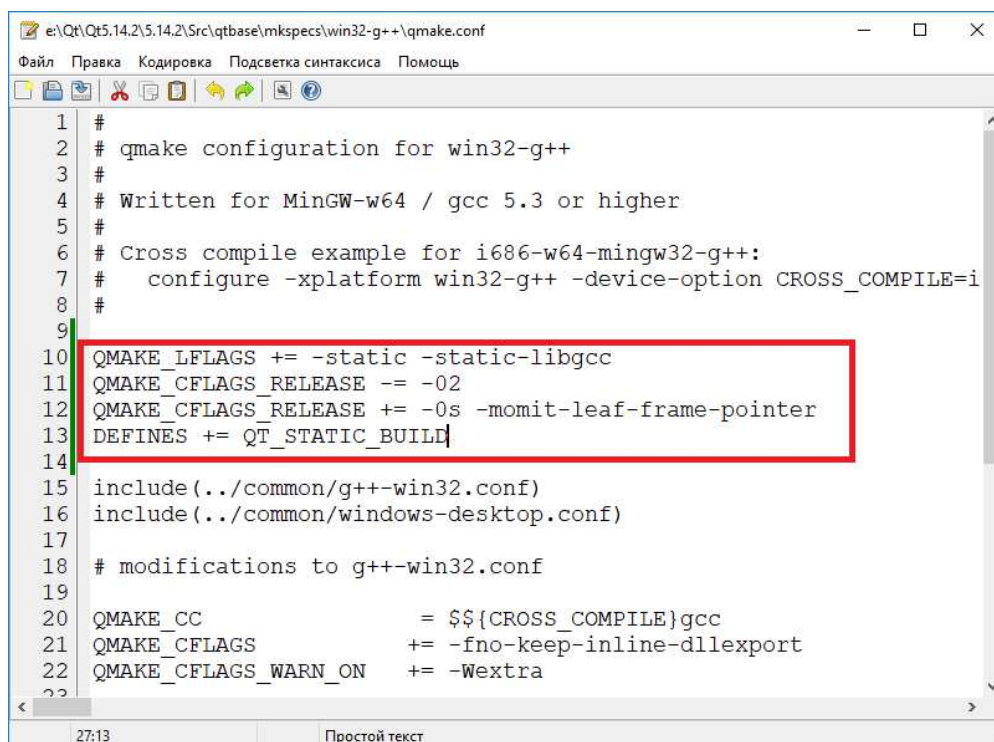


Рисунок А.3 – Изменение файла «qmake.conf»

Для конфигурирования и сборки, с помощью Windows PowerShell выполнить следующий список команд:

- cd E:\Qt\Qt5.14.2\5.14.2\Src;
- \$env:path+=";E:\Qt\Qt5.14.2\5.14.2\mingw73\_32\bin;E:\Qt\Qt5.14.2\Tools\mingw730\_32\bin";
- ./configure -static -debug-and-release -platform win32-g++ -qt-zlib -qt-libpng -qt-libjpeg -qt-freetype -qt-pcre -no-opengl -no-angle -no-openssl -make libs -nomake tools -nomake examples -nomake tests -opensource -confirm-license -prefix "E:\Qt\Qt5.14.2\_Static";
- measure-command {mingw32-make -k -j9};
- mingw32-make -k install.

Затем необходимо в собранной версии Qt изменить «спес»-файл (..\Qt5.14.2\_Static\mkspecs\win32-g++\qmake.conf) для сборки проектов, как показано на рисунке А.4.

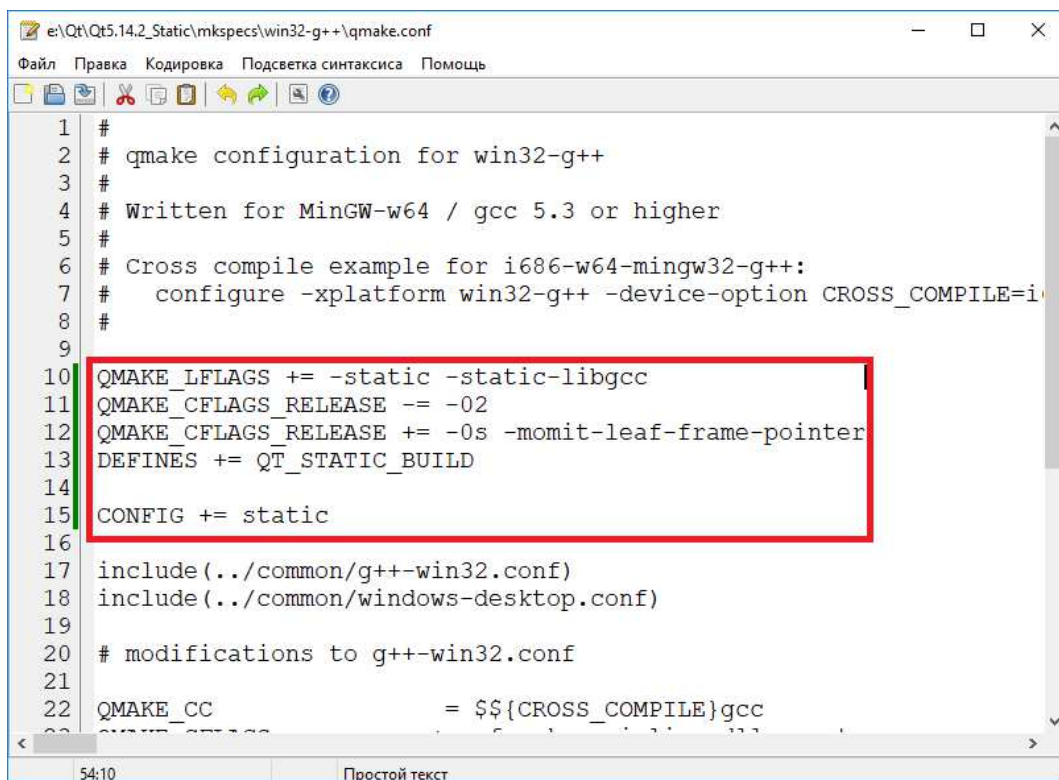
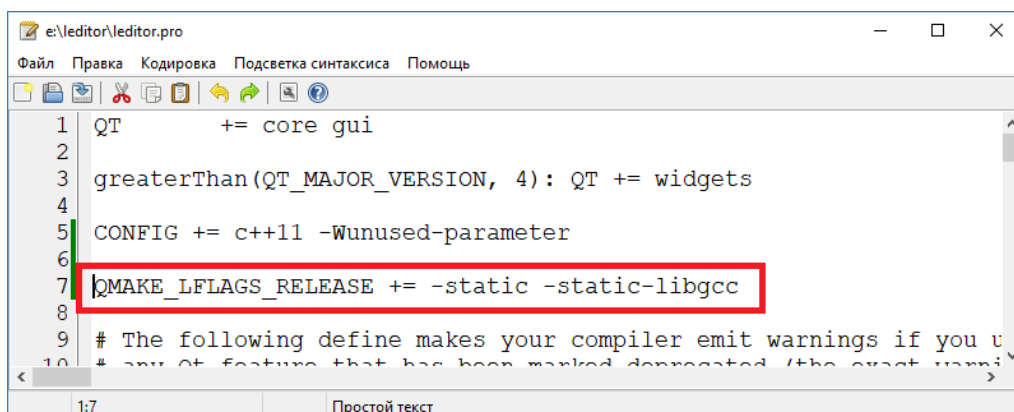


Рисунок А.4 – Изменение файла «qmake.conf»

Для подготовки проекта в файл проекта (leditor.pro) вставить строку «QMAKE\_LFLAGS\_RELEASE += -static -static-libgcc», как показано на рисунке А.5.



```
e:\leditor\leditor.pro
Файл  Правка  Кодировка  Подсветка синтаксиса  Помощь
1  QT      += core gui
2
3  greaterThan(QT_MAJOR_VERSION, 4): QT += widgets
4
5  CONFIG += c++11 -Wunused-parameter
6
7  QMAKE_LFLAGS_RELEASE += -static -static-libgcc
8
9  # The following define makes your compiler emit warnings if you u
10 # any Qt feature that has been marked deprecated (the exact warni
1:7  Простой текст
```

Рисунок А.5 – Изменение файла «leditor.pro»

Далее необходимо с помощью Windows PowerShell перейти в каталог проекта и выполнить следующий список команд:

- \$env:path+="E:\Qt\Qt5.14.2\_Static\bin;E:\Qt\Qt5.14.2\Tools\mingw730\_32\bin";
- qmake leditor.pro;
- mingw32-make -k -j7.

Федеральное государственное автономное  
образовательное учреждение  
высшего образования  
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт космических и информационных технологий  
институт

Вычислительная техника  
кафедра

УТВЕРЖДАЮ  
Заведующий кафедрой

  
подпись

О. В. Непомнящий  
инициалы, фамилия

2020г.

БАКАЛАВРСКАЯ РАБОТА

09.03.01 – «Информатика и вычислительная техника»  
код - наименование направления

Управление данными приложения «Пазлы-Живопись»  
тема

Руководитель

  
подпись, дата

канд. техн. наук, доцент  
должность, ученая степень

А. И. Постников  
инициалы, фамилия

Консультант

  
подпись, дата

ст. преподаватель  
должность, ученая степень

В. С. Васильев  
инициалы, фамилия

Выпускник

  
подпись, дата

К. А. Зимин  
инициалы, фамилия

Нормоконтролер

  
подпись, дата

канд. техн. наук, доцент

А. И. Постников  
инициалы, фамилия

Красноярск 2020