|  |
| --- |
| Пермский филиал федерального государственного автономного образовательного учреждения высшего образования  «Национальный исследовательский университет  «Высшая школа экономики»  *Факультет экономики, менеджмента и бизнес-информатики* |
|  |
| Сидоров Арсений Алексеевич  **РАЗРАБОТКА ПРОГРАММЫ ДЛЯ СИСТЕМАТИЗАЦИИ ХРАНИМЫХ АУДИОФАЙЛОВ**  *Курсовая работа*  студента образовательной программы «Программная инженерия»  по направлению подготовки *09.03.04 Программная инженерия*   |  |  | | --- | --- | |  | Доцент кафедры информационных технологий в бизнесе  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  В.О. Кушев | |

Пермь, 2020 год

Аннотация

Программная реализация алгоритма сжатия данных

Автор: Сидоров Арсений Алексеевич, студент второго курса образовательной программы «Программная инженерия».

Руководитель: Вадим Олегович Кушев, доцент кафедры информационных технологий в бизнесе.

Данная курсовая работа посвящена разработке программы для систематизации хранимых аудиофайлов.

Работа включает 28 страниц формата А4, из них 28 в основной части.

Основная часть работы включает в себя 19 иллюстраций и 1 таблицу.

Библиографический список состоит из 6 публикаций.

Оглавление

[Аннотация 2](#_Toc39857539)

[Оглавление 3](#_Toc39857540)

[Введение 5](#_Toc39857541)

[Глава 1 Анализ предметной области 7](#_Toc39857542)

[1.1 Анализ назначения программы 7](#_Toc39857543)

[1.2 Обзор существующих решений 7](#_Toc39857544)

[1.3 Формулировка требований 9](#_Toc39857545)

[1.4 Описание автоматизируемых бизнес-процессов предметной области 10](#_Toc39857546)

[1.5 Диаграмма прецедентов 12](#_Toc39857549)

[Глава 2 Проектирование приложения 14](#_Toc39857551)

[2.1 Проектирование базы данных 14](#_Toc39857552)

[2.2 Проектирование интерфейса 15](#_Toc39857554)

[Глава 3 Реализация программы 20](#_Toc39857564)

[3.1 Создание базы данных 21](#_Toc39857565)

[3.2 Описание запросов 22](#_Toc39857567)

[3.3 Описание программы 24](#_Toc39857570)

[3.4 Тестирование 27](#_Toc39857575)

[3.5 Код программы 27](#_Toc39857576)

[Заключение 28](#_Toc39857577)

[Список литературы 29](#_Toc39857578)

[ПРИЛОЖЕНИЕ А. Техническое задание 30](#_Toc39857579)

[ПРИЛОЖЕНИЕ Б. Руководство пользователя 37](#_Toc39857580)

[ПРИЛОЖЕНИЕ В. Тестирование 41](#_Toc39857586)

Введение

В наше время музыка несомненно оказывает огромное влияние на человека. Несмотря на активное развитие музыкальных сервисов, огромное количество человек, особенно меломанов, предпочитают хранить свои музыкальные композиции на персональном компьютере и слушать их каждый день. Однако, возникает вопрос: как же организовать весь этот большой список композиций разных форматов, жанров и авторов? Ответ прост: организовать весь процесс поиска и сортировки музыкальных файлов в специальной программе. Результатом выполнения данной курсовой работы как раз будет являться система для организации медиатеки на компьютере. Она позволит получить удобный доступ ко всей музыке и в случае надобности организовать все файлы по заданным критериям.

Помимо проблемы разобраться в иерархии папок с музыкой возникает еще одна, подборка плейлиста. Для того, чтобы, например, подобрать нужную музыку и скопировать ее на внешний носитель, чтобы потом проиграть содержимое на каком-нибудь празднике, необходимо вручную заходить в проводник и искать композиции, при этом тратя огромное количество времени. Планируемая для создания система позволит сократить время на этот процесс и предоставит функцию автоматического перенесения подборки на флэш-накопитель или диск.

*Объект* исследования ­– систематизация аудиофайлов. *Предмет* исследования – систематизация хранимых аудиофайлов с помощью приложения.

*Задачами* выполнения данной курсовой работы являются:

1. Провести анализ предметной области.
2. Сформулировать требования к приложению.
3. Изучить работу с SQL и выбрать СУБД.
4. Разработать приложение.
5. Протестировать приложение.

*Целью* выполнения курсовой работы является разработка программного продукта, позволяющего автоматизировать работу с аудиофайлами.

К теоретической значимости работы можно отнести исследование работы баз данных MsSQL с работой с медиатекой. Данное исследование отобразит, как эффективно и рационально создать таблицы, в которых будет храниться информация о композициях.

Практическая значимость работы – программа, выполняющая автоматический анализ иерархии папок с музыкальными композициями, сортировку по папкам в случае надобности и выгрузку необходимого плейлиста на внешний носитель.

Работа будет состоять из трех глав:

1. Анализ предметной области
2. Проектирование приложения
3. Разработка приложения

В первой главе изучается информация о предметной области, построение UML диаграммы. На этапе проектирования создается модель данных с таблицами, а также построение проекта непосредственно самого приложения, его интерфейс, функциональные возможности, способы ввода-вывода и редактирования. Третья глава содержит реализацию спроектированного приложения и последующее его тестирование.

Глава 1 Анализ предметной области

В этой главе описывается назначение программы и рассказывается о существующих решениях, которые выполняют похожие задачи. В конце главы формулируются требования к разрабатываемой системе.

1.1 Анализ назначения программы

В настоящее время существует множество решений, позволяющих взаимодействовать с музыкальными и другими видами медиа файлов. Однако большинство из них фокусируются именно на воспроизведении этих файлов в удобном пользовательском формате. И также существуют люди, которым требуется простая программа, способная хранить информацию о песнях и позволять редактировать её. К примеру, такой информации является альбом или год записи. Однако программа, просто хранящая информацию, не придется особо полезной, если этой информацией нельзя воспользоваться. Было бы лучше, если с помощью данных песни можно было отобрать необходимые и скопировать их на внешний носитель, скажем, чтобы составить плейлист для вечеринки. Другие известные программы не имеют таких возможностей, так как пользователи пользуются ими для других целей. Программа будет фокусироваться именно на хранении и редактировании информации о песне, а не на воспроизведении.

1.2 Обзор существующих решений

На настоящий момент существует достаточно большое количество решений, выполняющих похожую работу. Но подавляющее большинство из них являются аудиоплеерами, а систематизация файлов — это второстепенная функция, которая не всегда способна удовлетворить запросы пользователя. Одним из примеров является AIMP [2] — бесплатный аудиопроигрыватель с закрытым исходным кодом. Он был выпущен в 2006 году, но обновляется по сей день (на момент написания текста последняя версия была выпущена 28 ноября 2019). С годами он утратил свою популярность, но некоторое количество людей все ещё используют его. За другой пример можно взять ITunes [1]. Эта программа является нечто большим, чем просто проигрывателем файлов, по сути, воспроизведение музыки это только часть его задач. С помощью данной программы можно просматривать и запускать не только аудиофайлы, но и фильмы, а также книги. Помимо сказанного, ITunes выполняет синхронизирование устройств компании Apple с компьютером, их обновление и настройку. Программа является одной из самых популярных среди всех медиаплееров, так как имеется практически у каждого пользователя техники Apple, но несмотря на это, пользоваться ей не очень удобно. В перспективе компания планирует разбить приложение, чтобы пользователи сами могли выбирать, чем они хотят пользоваться, а также весь сервис. Еще одна схожая с разрабатываемой программа – VLC Player [3]. Она, как и ITunes, способна воспроизводить не только аудио, но и видеофайлы, но еще обладает уникальной функцией использования ее в качестве сервера для потокового воспроизведения по сети.

Программа, планируемая для разработки в курсовой работе, имеет фокус на решении проблемы с доступом и работой с музыкой, что нельзя сказать про программы, перечисленные сверху. Приложение будет помогать навести порядок в директориях с музыкой, но не будет предлагать этот функционал для последующего прослушивания внутри во встроенном плеере.

В таблице, приведенной ниже, указаны функции, имеющиеся у перечисленных решений и самой программы:

**Таблица 1 – Функциональные возможности программ.**



Таблица выше дала нам понять, что по проставленным критериям все известные решения имеют низкий рейтинг по проставленным критериям. Из этого можно сделать очевидный вывод, что программы не предназначены для манипулирования, детального просмотра и редактирования аудиофайлов, а используются в основном для воспроизведения этих файлов.

Так как распространенные существующие решения не выполняют поставленные мной задачи, приложение, разрабатываемое в курсовой работе, должно полностью покрывать эти критерии.

1.3 Формулировка требований

Исходя из выводов, сделанных выше, обозначены требования к программе. Она должна уметь работать с аудиофайлами, уметь хранить и выводить на экран добавленные композиции, а также выполнять экспорт в другую директорию, в частности на внешний носитель. Также программа должна предусмотреть вывод отчетов в один из видов документов, к примеру Microsoft Excel.

Рассмотрим функциональные требования:

1. Хранение информации о композициях в базе данных

Необходимо создать базу данных, хранящую информацию о каждой добавленной записи. Будет создано несколько таблиц, каждая из которых предназначена для хранения определенного вида информации.

1. Получение информации по определенному критерию

Та информация, которая будет храниться в БД должна быть получена в программе в форме таблицы. Взаимодействие между программой и базой данных будет происходить через ADO.NET.

1. Обновление базы данных

Пользователь должен иметь возможность менять свойства песни в соответствии с его желанием. К примеру, он захочет изменить альбом одного из исполнителей на другой, необходимо отправить соответствующий запрос в базу данных и изменить значение.

1. Экспортирование на внешний носитель

Одна из ключевых возможностей программы – копирование выбранных композиций в другую директорию, в частности на другой носитель. Пользователь должен выбрать из общего списка нужные ему песни, пометить их как песни для экспорта и произвести копирование.

1. Создание отчета в Excel

В программе должна быть предусмотрена возможность создания некоего отчета в формате таблицы Excel. Нужно создать новый файл Excel и скопировать в него все данные запроса, который получил пользователь. В данном файле будет храниться тот самый полученный запрос без каких-либо изменений.

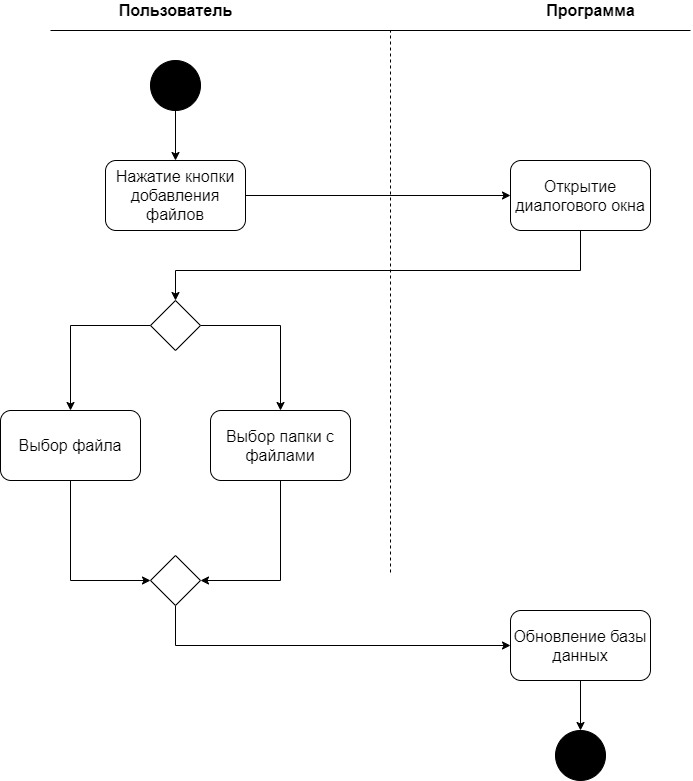
1.4 Описание автоматизируемых бизнес-процессов предметной области

Чтобы описать процессы в программе, необходимо построить диаграмму активностей. Она поможет разобраться, как именно будет работать логика некоторых функций разрабатываемой системы.

Стоит начать с того, что описывать каждую функцию в диаграмме активностей нецелесообразно вследствие простоты большинства функций. Они могут уместиться в один-два блока диаграммы, что не даст никакой полезной информации. Однако самые сложные функции, такие, что их непросто реализовать программно или имеют не один запрос к БД, стоит описать. Таких функций в будущей программе будет две:

1. Добавление новых файлов
2. Экспорт в другую директорию

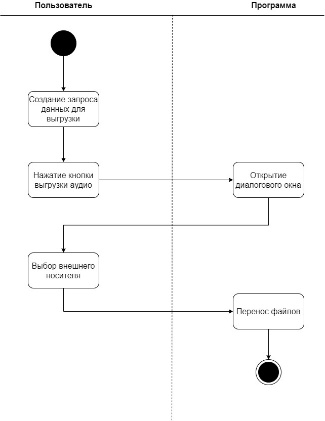
Эти функции были выделены, так как они требуют существенные изменения базы данных и в некоторых случаях директории непосредственно самих музыкальных файлов. Для начала опишем первую функцию в списке:



## Рисунок 1.1 – Диаграмма активностей с описанием добавления файлов.

Рассмотрим подробнее данную диаграмму. Сначала пользователь должен нажать на кнопку добавления файлов. Возможно, будут иметься сразу две кнопки, как для добавления файлов, так и целых папок, но для наглядности процесса в диаграмме имеется только одна кнопка. Далее программа открывает диалоговое окно, в котором пользователь выбирает нужные ему файлы. На диалоговом окне будет стоять ограничение, которое позволит выбирать только нужного формата файлы, чтобы избежать ошибок. Затем пользователь может выбрать файл, несколько файлов или папку. Далее все выбранные файлы заносятся в базу данных через сформированные запросы.

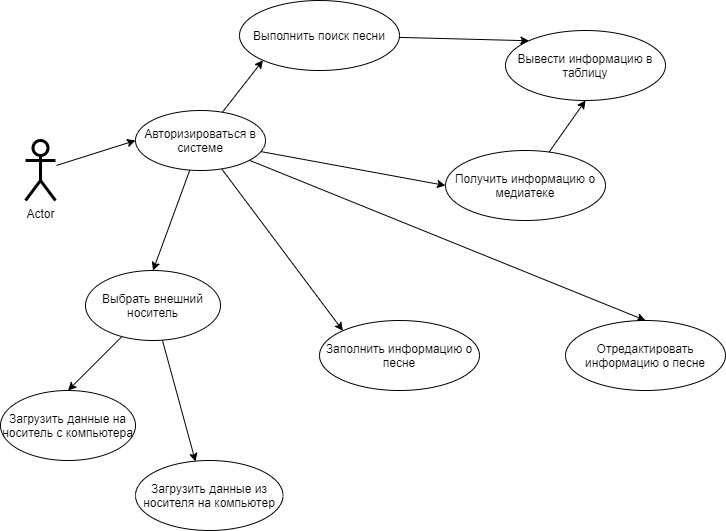
Вторая диаграмма описывает процесс переноса файлов в другую директорию. Сначала пользователь должен выбрать файлы, которые он хочет перенести в другое место. Он сможет это сделать, делая запросы к базе данных и уже в отображаемом ему списке выбрать, что он желает. Таких запросов он должен делать сколько угодно, при этом список добавляемых песен стираться не должен. Так, пользователь получил необходимый ему список, далее он нажимает на кнопку экспорта файлов, открывается диалоговое окно с выбором директории. Выбрав директорию, начнется копирование файлов программой.



## Рисунок 1.2 – Диаграмма активностей с экспортом аудиофайлов.

1.5 Диаграмма прецедентов

Чтобы показать все функции программы, создается диаграмма прецедентов. На рисунке ниже, слева можно увидеть актора – пользователя программы. В каждом овальном блоке диаграммы описана какая-то из функции будущей программы. Например, пользователь может заполнить информацию о песне, но только после входа в программу, в процессе которого происходит подключение к БД.



## Рисунок 1.3 – Диаграмма прецедентов.

Техническое задание

В техническом задании описываются требования заказчика к данной программе, включая основание для разработки и ее назначение. Оно хранится в приложении A.

Итоги главы

В данной главе была поставлена задача для разработки программы. Были выявлены слабые места аналогов, создан список необходимых функций в будущей программе, которых нет в других известных программах. Были также описаны функциональные и нефункциональные требования, которые наводят вектор разработки на правильное направление.

Глава 2 Проектирование приложения

Глава описывает процесс проектирования базы данных, которая используется в программе, а также интерфейс самой программы.

2.1 Проектирование базы данных

База данных должна хранить информацию о песнях, добавленных пользователем. Всего было создано 8 таблиц:

1. Songs.
2. Countries.
3. Songs\_in\_albums.
4. Genres.
5. Albums.
6. Authors.
7. Formats.
8. Paths.

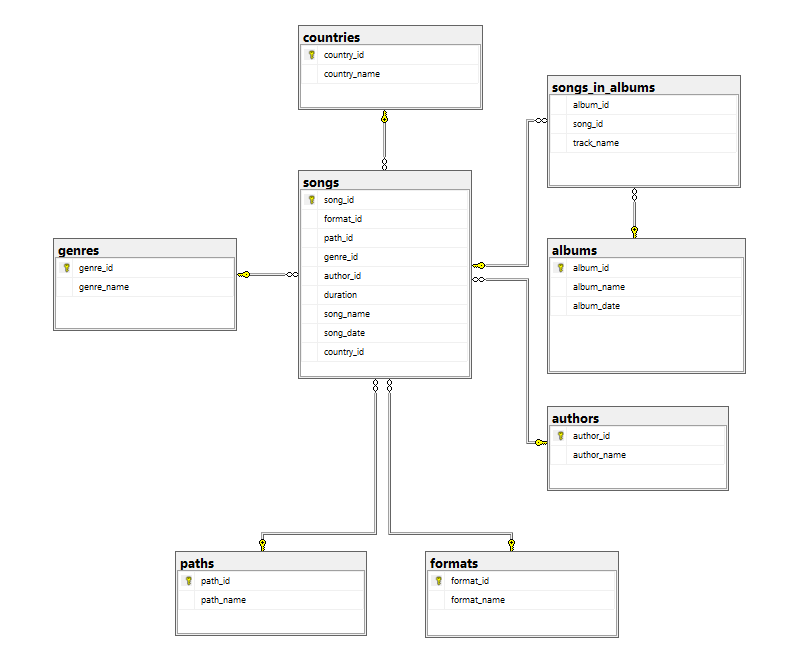
Рассмотрим каждую таблицу поподробнее.

В таблице songs будут храниться непосредственно названия песен, дата записи этой песни исполнителем и длительность. Также будет иметься идентификатор, позволяющий иметь уникальный доступ к каждой песне в таблице. Также таблица содержит идентификаторы формата, расположения, жанра, автора и страны. Доступ к значениям этих строк можно получить из других таблиц, так как эти данные повторяются и хранить в каждой строчке, к примеру, строку расположения песни нецелесообразно.

Таблица форматов содержит допустимые форматы музыкальных композиций. Пользователь не будет иметь возможности добавлять туда новые форматы, но она пригодится для дальнейшего улучшения программы, где будут использованы другие форматы.

Таблицы genres, paths и countries имеют идентичную структуру и были выделены также для нормализации базы данных.

Таблица albums – единственная таблица, которая не связана с таблицей songs напрямую. Она имеет связь только со вспомогательной таблицей songs\_in\_albums, в которой в каждой строчке будет храниться идентификатор песни и альбома этой песни. Сделано это по той причине, что в одном альбоме может находиться несколько песен, как и одна и та же песня может находиться в разных альбомах.

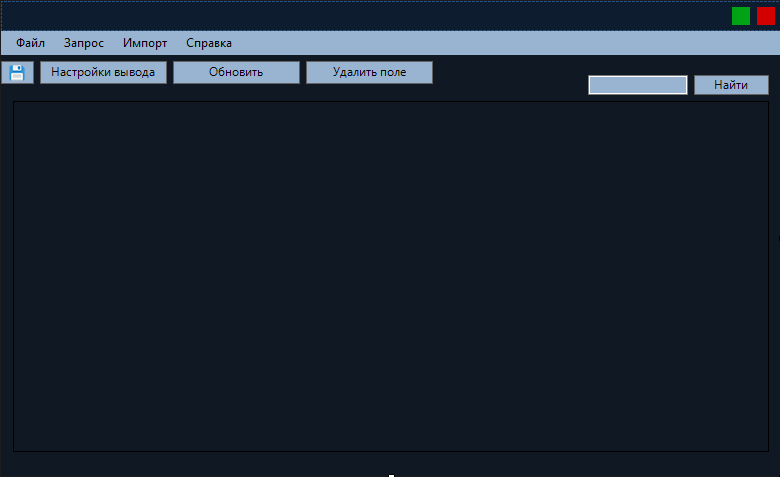


## Рисунок 2.1 – Диаграмма таблиц базы данных.

База данных нормализована до третьей нормальной формы. Это значит, что она нормализована до второй нормальной формы и каждый атрибут, не являющийся ключевым, находится в нетранзитивной зависимости от первичного ключа.

2.2 Проектирование интерфейса

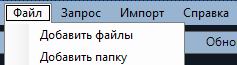
Интерфейс разрабатывается, используя Windows Forms и Metro Forms. Второе, как было сказано в первой главе, добавляет новые элементы, что позволяет более гибко настроить GUI. Рассмотрим созданную главную форму:



## Рисунок 2.2 – Главное окно.

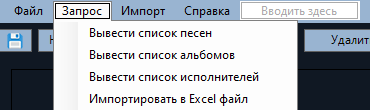
На скриншоте мы видим множество кнопок, созданных с помощью Metro Forms, элемент Tool Strip и Data Grid View. Все кнопки – элементы добавленной библиотеки. Основная и единственная причина, по которой были выбраны именно эти, а не встроенные кнопки – наличие бóльших возможностей по выбору фонового цвета. Сверху мы видим элемент Panel темно-синего цвета, с помощью которого можно перемещать окно по экрану компьютера. На элементе в верхнем правом углу расположены кнопки сворачивания и закрытия зеленого и красного цвета соответственно. Под панелью расположен элемент, содержащий выпадающие подпункты. Под ним расположены кнопки сохранения, настройки вывода, обновления базы данных и удаления выбранного поля. Справа над таблицей расположены textbox и кнопка для поиска в запросе. Всю остальную часть экрана занимает таблица с запросом пользователя.

Нажав на кнопку “Файл”, появляются кнопки для добавления файлов и папки в программу. По нажатию на них открывается диалоговое окно, и пользователь выбирает нужные ему композиции.



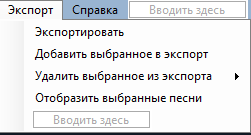
## Рисунок 2.3 – Меню “Файл”.

Следующая вкладка содержит запросы к базе данных. В программе дана возможность вывести список песен, альбомов и исполнителей. Также здесь расположена функция экспорта в Excel файл текущего запроса.



## Рисунок 2.4 – Меню “Запрос”.

Во вкладке “Экспорт” содержатся все необходимые функции для работы с копированием файлов в другую директорию. Выбранные песни можно добавить с помощью кнопки “Добавить выбранное в экспорт” и удалить кнопкой “Удалить выбранное из экспорта”. Последняя кнопка в списке отображает все выбранные композиции.



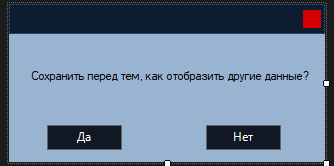
## Рисунок 2.5 – Меню “Экспорт”.

Вкладка “Справка” содержит кнопку “О программе”, с помощью которой можно открыть окно с информацией.



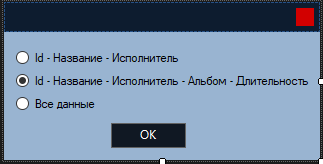
## Рисунок 2.6 – Меню “Справка”.

Одна из самых кропотливых задач – сделать интерфейс максимально удобным и понятным для пользователя. Программа не должна вводить пользователя в ступор, на каждом шаге использования должен быть простой для понимания GUI. Также пользователей стоит предохранять от возможных ошибок, которые они могут совершить по невнимательности. Например, пользователь захочет перейти к другому запросу. Скажем, к списку альбомов. Но перед этим он работал с запросом песен, редактировал данные, заполнял пустые поля. Не нажав кнопку сохранения, пользователь делает новый запрос. Поэтому в программе реализовано окно, спрашивающее пользователя, действительно ли он хочет это сделать:



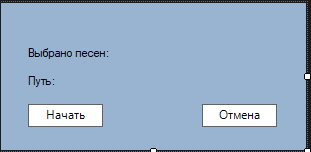
## Рисунок 2.7 – Предупреждающее окно.

По нажатии на кнопку “Настройка вывода”, появляется окно тремя возможными типами отображения. Однако это окно будет невозможно вызвать, если текущий вывод – это альбомы или исполнители.



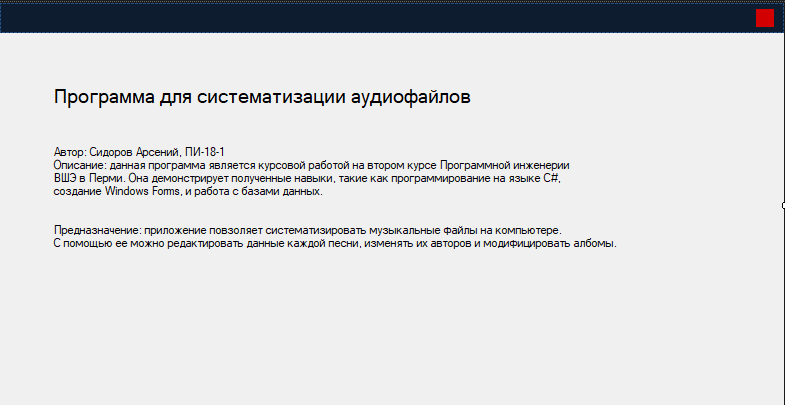
## Рисунок 2.8 – Выбор отображения таблиц.

Когда пользователь выбрал нужные ему песни для экспорта, он нажимает кнопку “Экспортировать”. Если не было выбрано ни одной песни, появляется сообщение о невозможности данной операции. На форме отображается количество выбранных композиций и директория, которая выбрана в диалоговом окне. Снизу находятся кнопки для начала копирования и отмены действия.



## Рисунок 2.9 – Окно экспорта.

При нажатии на кнопку “О программе”, появляется данное окно:



## Рисунок 2.10 – Справка.

Итоги главы

В данной главе был описан дизайн будущей программы, тем самым получилась структура приложения. Также была разработана база данных для хранения информации о музыке.

Глава 3 Реализация программы

В предыдущей главе был спроектирован дизайн программы и создана диаграмма базы данных. В данной главе будет описываться реализация программы, ее классы и некоторые методы, требующие наибольшего внимания. Также будет разобрано подключение к Microsoft SQL Server Management Studio.

В процессе разработки программы необходимо использовать данные инструменты:

1. Разработка на языке программирования C#

Программу необходимо писать на языке программирования C#, так как это один из основных языков в среде .NET, и один из наилучших вариантов для программирования на Windows Forms.

1. Программирование на Windows Forms.

Программа должна быть разработана на интерфейсе программирования приложений Windows Forms, так как данный API позволяет быстро и удобно реализовать несложный интерфейс для взаимодействия с БД и прочими функциями программы.

Одной из альтернатив данному API является WPF. Исходя из излишней функциональности, выбор был в сторону простоты и скорости разработки.

1. Использование MsSQL [5]

MsSQL ­– использование расширенного языка SQL от Microsoft является наилучшим выбором в следствие хорошей совместимости с другими продуктами этой компании, такими как C# и WinForms.

1. Использование Microsoft Management Studio

Для работы с базами данных и написания запросов используется данная среда разработки. В ней есть весь необходимый функционал для работы, включая создание, редактирование и удаление таблиц, создание диаграмм и т. д.

1. ADO.NET

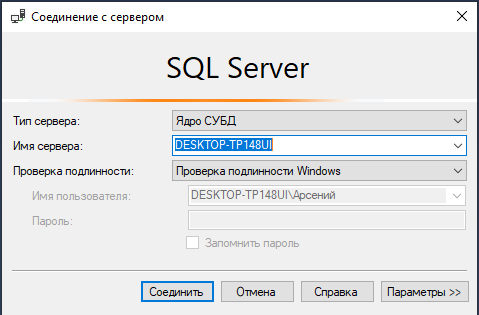
Использование технологии ADO.NET позволит работать с данными, хранящимися в базе данных в языке C#.

1. Metro Forms

Данное расширение добавляет новые элементы. Эти элементы не являются готовыми изначально, но имеют намного больше свойств по сравнению со встроенными элементами. Так что данная библиотека делает дизайн проекта более гибким.

3.1 Создание базы данных

Для начала необходимо установить все необходимые элементы для работы с базами данных. Первое что надо сделать, установить SQL Server последней версии. На момент написания работы последней версией являлся SQL Server 2019. Затем переходим к подключению в SSMS.



## Рисунок 3.1 – Окно входа в БД.

В данном окне выбрано подключение с проверкой подлинности Windows, однако существуют и другие способы. Выбранный же является самым простым, так как не требует ввода логина и пароля.

В непосредственно самой студии мы создаем базу данных. Она будет называться Music Library. Создать таблицы можно двумя способами: прямым, то есть создавать отдельно каждую таблицу или через диаграмму. Более удобным способом является создание через диаграмму, так как в ней можно установить связи простыми движениями мыши.

Разберем поподробнее правила обновления и удаления таблиц.

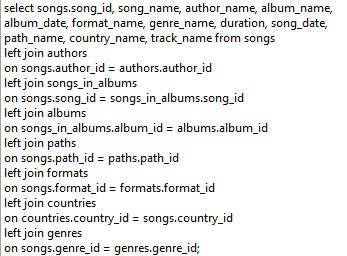
1. В таблицах countries и songs правило удаления и обновления – присвоить NULL, так как в случае изменения таблицы countries в songs должно оставаться пустое значение.
2. В таблицах genres и songs аналогичный тип связи, при изменении жанра в песне будет пустое значение.
3. В таблице authors связь также с присвоением NULL.
4. Если мы изменим таблицу paths и formats, вызовется исключение, так как изменение этих таблиц не предусмотрено в программе.
5. Songs\_in\_albums имеет каскадный тип связи с другими таблицами, так как в случае удаления данных из нее также должны удаляться данные, и так же при обновлении.

Далее созданную базу данных подключаем в Visual Studio для удобного взаимодействия прямо в этой программе.

3.2 Описание запросов

В программе невозможно взаимодействовать с базой данных без запросов, поэтому было создано несколько получения, изменения и удаления данных из таблиц.

Первые три использованные запроса – получение полного списка песен, альбомов и исполнителей. Запрос на получение песен выглядит следующим образом:



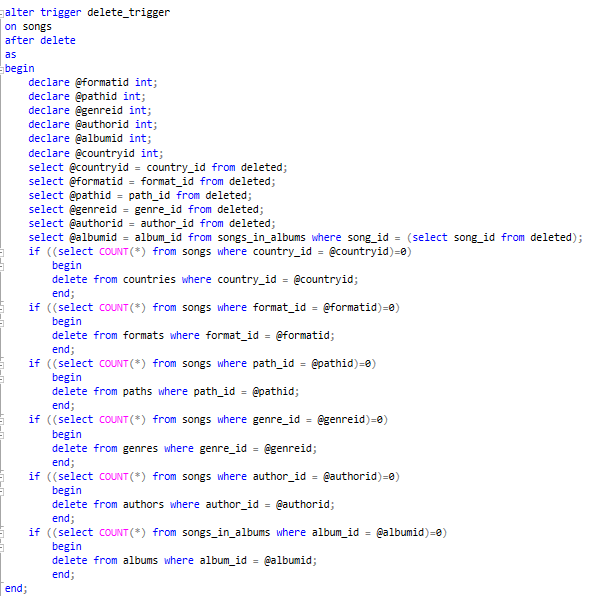
## Рисунок 3.2 – Запрос на вывод песен.

Так как для отображения всех данных берутся строки из всех таблиц, сформирован такой запрос, в котором есть несколько left join.

В программе реализовано редактирование данных прямо в таблице. То есть при изменении ячейки и сохранении таблицы вызывается соответствующий запрос. Например, если мы изменим страну, в которой была написана песня, будет вызван запрос на изменение только этого поля. Таким образом, мы не изменяем все поля записи, а только то, которое подверглось изменению в таблице.

Следующий тип запроса – добавление элементов. У песни есть альбом, автор и директория. Мы должны проверить, есть ли уже такие значения в таблице. Если их нет, то создаем их, иначе присваиваем существующие. Главная проблема, с которой предстояло столкнуться было как раз-таки добавление песен. Язык MsSql не предусматривает триггер before, а значит мы не можем уместить все в одном запросе, приходилось добавлять все шаг за шагом: сначала исполнителя, затем альбомы, форматы и так далее, пока наконец не будет добавлена сама песня.

Удаление элементов происходит намного проще, так как был добавлен триггер на удаление песни. Выглядит он следующим образом:



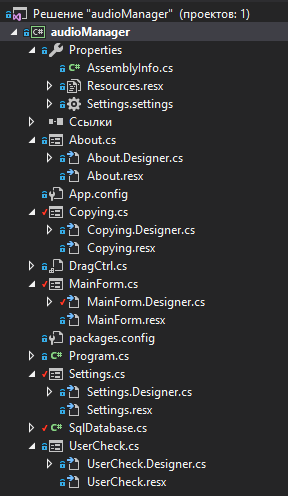
## Рисунок 3.3 – Триггер.

Перед удалением мы делаем проверку в каждой таблице, действительно ли это последняя запись, которая имеет данный жанр, альбом и так далее. Если больше таких записей в базе данных нет, мы удаляем поле.

Все запросы хранятся в ресурсах программы, так что в любом классе мы можем получить их как строку.

3.3 Описание программы

Разрабатываемая программа разбита на несколько классов, отвечающих за какую-то область действий. Так как программа представляет из себя, по сути, интерфейс для работы с базой данных, весь код был написан в двух классах: MainForm и SqlDatabase.

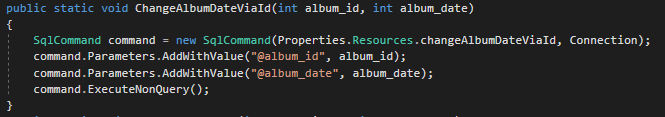


## Рисунок 3.4 – Обозреватель решений.

Рассмотрим каждый класс программы.

SqlDatabase.cs

Для начала возьмем класс SqlDatabase. В нем содержится все необходимое для работы с базой данных. По сути, данный класс является надстройкой над встроенным классом SqlConnection. Чтобы не работать с ним в форме и не ошибиться с лишним подключением или отключением весь функционал, ответственный за это содержится в этом классе. Он содержит множество методов, включая примитивные Connect() и Close(), которые просто открывают и закрывают подключение. Большинство из этих методов – это запросы к бд. Все они имеют более-менее похожую структуру. Для начала мы создаем экземпляр класса SqlCommand, в который в качестве аргументов конструктора заносится необходимый нам запрос и соединение, которое мы используем. В случае, если в запрос нужно вставить какое-то значение, мы используем метод AddWithValue(). Затем выполняем этот запрос и получаем ответ в случае, если это необходимо через ExecuteReader().



## Рисунок 3.5 – Метод изменения даты альбома.

MainForm

MainForm является главной формой программы, в нем описаны все элементы и их действия. Выносить функции каждой кнопки в отдельный класс бессмысленно, так как методы класса MainForm будут тогда содержать просто ссылки на методы другого класса. Поэтому было принято решение функционал описывать прямо в этой форме. Разрабатываемая программа может получить несколько запросов, и эти запросы могут отличаться типом. К примеру, таблица альбомов и таблица песен имеют разную структуру. Поэтому функционал программы должен отличаться, точнее, адаптироваться к текущему запросу. В MainForm имеются два перечисления: тип текущего запроса и форма отображения. Первый меняется в зависимости от того, что мы запросили от базы данных. Затем все кнопки и прочие элементы формы обращаются к этому перечислению чтобы изменить свое свойство Enabled. Например, кнопка с изменением отображения запроса не будет активна, если вызван список авторов. Второе перечисление как раз-таки меняется по нажатии кнопки изменения отображения. В программе предусмотрено три вида таблицы: подробный, обычный и краткий. Это полезно, если пользователь захочет сам выбрать, нужно ли ему видеть подробно все столбцы или же предпочтет лаконичное отображение. Запрос, однако, мы имеем один и тот же, то есть получаем все данные, и данная настройка просто скрывает некоторые колонки таблицы DataGridView.

Работа с сохранением ячеек

В программе предоставлена возможность редактировать ячейки прямо в таблице DataGridView и описание этой возможности тоже требует внимания. Когда пользователь нажимает на ячейку два раза, вызывается метод CellBeginEdit(), в котором мы запоминаем текущее значение. Когда пользователь заканчивает редактирование, программа вызывает CellEndEdit(), в котором идет проверка на корректность ввода. Если ввод некорректен, например, если в названии альбома присутствуют недопустимые знаки или год написания альбома в неверном формате, то тогда ячейке мы присваиваем значение, которое запомнили в CellBeginEdit().



## Рисунок 3.6 – Редактирование ячейки.

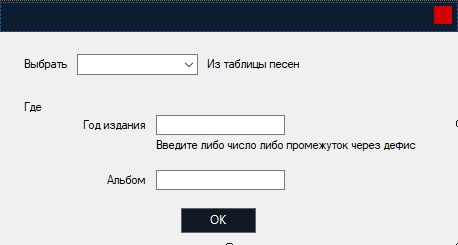
Если все условия форматирования выполнены, то запоминаем данную ячейку. Затем с помощью оператора switch мы смотрим, какое название у столбца и вызываем соответствующий метод из класса SqlDatabase.

Вывод информации в Excel файл

В программе также предусмотрен вывод таблицы в отдельный файл Excel. Данная функция разработана с помощью библиотеки ClosedXml. Для начала таблицу DataGridView необходимо преобразовать в DataTable, это происходит простым обходом в цикле и созданию столбцов и ячеек. Вызывается диалоговое окно, мы получаем директорию. Затем мы у класса XLWorkbook вызываем метод SaveAs(), на вход которой подается DataTable и директория.

Конструктор отчетов

При таком сценарии, когда пользователю нужно получить список композиций по каким-то критериям, следует воспользоваться конструктором отчетов. В выпадающем списке необходимо выбрать те столбцы, которые будут отображаться. В поле “Год издания” можно ввести год или диапазон годов, в котором была записана песня. В последнее поле можно ввести желаемый альбом. Оставив поля пустыми, запрос не будет иметь никаких ограничений.



## Рисунок 3.7 – Конструктор отчетов.

3.4 Тестирование

По окончании разработки было проведено тестирование [4], в котором описывается успешность выполнения каждой функции программы. Было выявлено 9 ошибок, все из которых успешно исправлены. Результат тестирования находится в приложении В.

3.5 Код программы

Весь программный код приложения и запросы, использованные в нём, хранятся на Github. Ссылка: <https://github.com/Marshall200900/audioManager>.

Итоги главы

В главе описаны конкретные способы реализации поставленных задач. Было разработано приложение на Windows Forms с использованием C# и подключена база данных. Разработанная программа успешно протестирована.

Заключение

По окончании разработки программы все поставленные цели были достигнуты. В приложении реализованы все функции, которые необходимо было добавить.

Результаты выполнения работы:

1. Сформулированы требования к приложению.
2. Спроектирован макет приложения.
3. Описан дизайн.
4. Описана созданная база данных.
5. Разработана программа и база данных.
6. Проведено тестирование.

В итоге получилось приложение, которое может использоваться повседневно как помощник в организации аудиофайлов. Также реализован функционал, требуемый по критериям курсовой работы 2 курса.

В дальнейшем планируется улучшить программу, добавив в нее полноценный проигрыватель. Таким образом, оно будет использоваться как многофункциональный и единственный на компьютере проигрыватель аудиофайлов.

Список литературы

1. ITunes // apple.com URL: <https://www.apple.com/ru/itunes/> (дата обращения:24.01.2020).
2. AIMP // aimp.ru URL: <http://ru.aimp.ru/> (дата обращения:24.01.2020).
3. VLC Media Player // softportal.com URL: <https://www.softportal.com/software-1427-vlc-media-player-videolan.html> (дата обращения:24.01.2020).
4. Плаксин. М. А. Тестирование и отладка программ – для профессионалов будущих и настоящих / М. А. Плаксин. –М.: БИНОМ. Лаборатория знаний, 2007. – 167 с.: ил.
5. SQL Server // Microsoft.com // URL: <https://www.microsoft.com/ru-ru/sql-server/sql-server-2019>
6. <https://docs.microsoft.com/ru-ru/> (дата обращения:20.03.2020).

ПРИЛОЖЕНИЕ А. Техническое задание

|  |  |
| --- | --- |
| **ПРОГРАММЫ ДЛЯ СИСТЕМАТИЗАЦИИ ХРАНИМЫХ АУДИОФАЙЛОВ** | |
| **Техническое задание** | |
| ***Лист утверждения*** | |
|  | Руководитель разработки: |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Кушев В.О. |
| “\_\_\_\_\_”\_\_\_\_\_\_\_\_\_\_\_\_\_2020 |
|  |
| Исполнитель: |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Сидоров А.А. |
| “\_\_\_\_\_”\_\_\_\_\_\_\_\_\_\_\_\_\_2020 |
| Пермь, 2020 | |

1. Введение

**Наименование программы**

Наименование – Разработка программы для систематизации хранимых аудиофайлов

**Краткая характеристика области применения программы**

Данная программа предназначена для менеджмента музыкальных файлов, хранящихся на персональном компьютере.

**Краткая характеристика объекта, в котором используют программу.**

Программа предназначена для любого пользователя компьютера, который хранит на жестком диске аудиофайлы.

1. **Основание для разработки**

**Документ (документы), на основании которых ведется разработка**

Юридическим основанием для проведения разработки является приказ об утверждении тем и руководителей курсовых работ студентов образовательной программы «Программная инженерия» факультета экономики, менеджмента и бизнес-информатики от 10.12.2019 года.

**Организация, утвердившая этот документ и дата его утверждения**

Приказ был утвержден 10.12.2019 НИУ-ВШЭ Перми на образовательной программе «Программная инженерия» факультета экономики, менеджмента и бизнес-информатики.

**Наименование и (или) условное обозначение темы разработки**

Наименование темы разработки - «Разработка программы для систематизации хранимых аудиофайлов».

Условное обозначение темы разработки – «Программа для систематизации аудиофайлов».

1. **Назначение разработки**

**Функциональное назначение программы**

Функциональным назначением программы является создание базы данных аудиофайлов с последующим возможным их копированием в другую директорию.

**Эксплуатационное назначение программы**

Программа должна эксплуатироваться на домашнем или рабочем компьютере.

1. **Требования к программе или программному изделию**

**Требования к функциональным характеристикам программы**

1. Занесение в базу данных информации о песнях.
2. Обновление базы данных в случае редактирования информации.
3. Экспортирование аудиофайлов в другую директорию
4. Получение информации о песнях в базе данных.
   * 1. **Требования к составу выполняемых функций**
5. Создание новой записи о песне.
6. Редактирование записи о песне.
7. Удаление записи о песне.
8. Просмотр информации о песне.
9. Выгрузка записей о песне.
   * 1. **Требования к организации входных и выходных данных**

Входные данные – аудиофайлы, размещенные на компьютере. Выходные данные – информация об этих файлах.

Данные должны размещаться на локальных или съемных носителях, отформатированных согласно требованиям операционной системы.

* + 1. **Требования к временным характеристикам**

Временные рамки формируются пропорционально размеру и количеству добавляемых песен в базу данных.

**Требования к надежности**

* + 1. **Требования к обеспечению надежного (устойчивого) функционирования программы**

Надежное (устойчивое) функционирование программы должно быть обеспечено выполнением заказчиком совокупности организационно-технических мероприятий, перечень которых приведен ниже:

а) организацией бесперебойного питания технических средств;

Требование необходимо, поскольку отключение питания может привести к отключению устройства или же к отключению доступа к интернету для устройства.

б) регулярным выполнением рекомендаций Министерства труда и социального развития РФ, изложенных в Постановлении от 23 июля 1998 г. «Об утверждении межотраслевых типовых норм времени на работы по сервисному обслуживанию ПЭВМ и оргтехники и сопровождению программных средств»;

Несоблюдение может привести к отключению и некорректной работы ОС и технического средства, а, следовательно, и программы.

в) регулярным выполнением требований ГОСТ 51188-98. Защита информации. Испытания программных средств на наличие компьютерных вирусов;

Несоблюдение может привести к отключению и некорректной работы ОС и технического средства, а, следовательно, и программы. Необходимо проверить, что программа крутится на 4-5 уровне защиты от ядра и что не может на него забраться при желании.

г) необходимым уровнем квалификации сотрудников профильных подразделений.

Несоблюдение может привести к отключению и некорректной работы ОС и технического средства, а, следовательно, и программы.

д) регулярной проверкой целостности информационной базы и восстановления ее в случае необходимости, администрирования базы.

* + 1. **Время восстановления после отказа**

Время восстановления после отказа, вызванного сбоем электропитания технических средств (иными внешними факторами), не фатальным сбоем (не крахом) операционной системы, не должно превышать времени, необходимого на перезагрузку операционной системы и запуск программы, при условии соблюдения условий эксплуатации технических и программных средств.

Время восстановления после отказа, вызванного неисправностью технических средств, фатальным сбоем (крахом) операционной системы, не должно превышать времени, требуемого на устранение неисправностей технических средств и переустановки программных средств.

* + 1. **Отказы из-за некорректных действий оператора**

Отказы программы возможны вследствие некорректных действий оператора (пользователя) при взаимодействии с информационной системой (например, ввод неправильных, некорректных данных), взаимодействие с базой данных, не поддерживаемой в данной информационной системе. Во избежание возникновения отказов программы по указанной выше причине следует обеспечить работу конечного пользователя без предоставления ему административных привилегий.

Последние данные, с которыми работал пользователь до сбоя (краха) системы, должны быть сохранены. Допустимой потерей данных при отказе программы являются некорректные данные, введенные пользователем, то, что привело к сбою (краху) системы.

* + 1. **Допустимый объем данных, утрачиваемый во время отказа**

Допустимый объем данных, утрачиваемых в случае отказа, не должен превышать объема информации об изменениях в записях, изменяемых в момент отказа.

**Условия эксплуатации программы**

* + 1. **Условия эксплуатации, при которых должны обеспечиваться заданные характеристики**

Система должна использоваться на ПК в Операционных системах Windows версии 7 и более поздних версиях ОС Windows. Также на ПК должен быть установлен .NET framework 4.5.

* + 1. **Климатические условия эксплуатации**

Климатические условия эксплуатации, при которых должны обеспечиваться заданные характеристики, должны удовлетворять требованиям, предъявляемым к техническим средствам в части условий их эксплуатации.

* + 1. **Виды обслуживания программы**

Программа не требует проведения каких-либо видов обслуживания.

См. пункт №4.2.1 «Требования к обеспечению надежного (устойчивого) функционирования программы".

* + 1. **Необходимое количество и квалификация персонала**

Минимальное количество персонала, требуемого для работы программы, должно составлять не менее одной штатной единицы: конечный пользователь программы.

**Требования к составу и параметрам технических средств**

В состав технических средств должен входить:

* + 1. IBM-совместимый персональный компьютер (ПЭВМ), включающий в себя:
    2. процессор Pentium - 4 с тактовой частотой, 1.2 ГГц, не менее;
    3. оперативную память объемом, 512 Mб, не менее;
    4. жесткий диск объемом 40 Гб, и выше;
    5. устройства ввода (мышь и клавиатура).

**Требования к информационной и программной совместимости**

Программа должна быть совместима с ПК сотрудников университета.

* + 1. **Требования к информационным структурам и методам решения**

Источником входных данных должен быть ручной ввод в систему. Источником выходных данных должна быть экранная форма.

* + 1. **Требования к исходным кодам и языкам программирования**

Информационная система должна быть реализована на языке программирования C# с использованием ADO.NET

* + 1. **Требования к программным средствам, используемым программой**

Системные программные средства, используемые ИС, должны быть представлены лицензионной локализованной версией операционной системы Windows.

* + 1. **Требования к защите информации и программ**

Защита Системы должна обеспечиваться антивирусной программой, а также комплексом программно-технических средств и поддерживающих их организационных мер.

Программно-технические средства защиты не должны существенно ухудшать основные функциональные характеристики Системы.

* + 1. **Требования к маркировке и упаковке**

Программа поставляется на дистрибутивном (внешнем оптическом) носителе (компакт-диске), который должен иметь маркировку с обозначением типа (наименования), номера версии, порядкового номера, даты изготовления.

* + 1. **Требования к транспортированию и хранению**

Заказчик не вправе нарушать условий транспортирования и хранения. Исполнитель сможет отказать заказчику в возврате программного изделия, утверждая, что ненадлежащий внешний вид программного изделия является следствием несоблюдения условий транспортирования и хранения.

* + - 1. **Требования к условиям транспортирования, месту, а также условиям хранения**

Допускается транспортирование программного изделия в транспортной таре всеми видами транспорта. При транспортировании и хранении программного изделия должна быть предусмотрена защита от попадания пыли и атмосферных осадков. Не допускается кантование программного изделия.

При хранении и использовании дисков принять меры к предотвращению ударов, перемещению и вибрации дисков внутри футляров.

При хранении и использовании дисков принять меры к предотвращению проникновения влаги, агрессивных жидкостей (растворителей, клея и т.п.), вредных газов, пыли, солнечных лучей и образованию конденсата внутри футляров.

**Специальные требования**

Программа должна обеспечивать взаимодействие с пользователем (оператором) посредством графического пользовательского интерфейса, разработанного согласно рекомендациям компании-производителя операционной системы. Также к пользовательскому интерфейсу предъявляются следующие требования:

Защита от некорректного ввода данных.

Вывод различных сообщений об ошибках при их появлении.

1. **Требования к программной документации**

**Предварительный состав программной документации**

Состав программной документации должен включать в себя:

1. техническое задание;
2. программу и методики испытаний;
3. исходный код;
4. руководство для программиста;
5. руководство для оператора (пользователя);
6. протокол испытаний;
7. свидетельство о приемке;
8. формуляр.

**Специальные требования к программной документации**

Программная документация должна быть разработана в соответствии с ГОСТ 19.201-78.

1. **Технико-экономические показатели**

**Ориентировочная экономическая эффективность**

Информационная система позволит сократить время сотрудников студии дизайна, что позволит им повысить качество обработки и выполнения заказов.

**Экономические преимущества разработки**

Информационная система позволит сократить время сотрудников на поиск информации, а также позволит сократить время на выполнение заказа.

**Предполагаемая годовая потребность в программе**

Предполагаемое число использования программы в год – 50 сеансов работы на одном рабочем месте.

1. **Стадии и этапы разработки**

**Стадии разработки**

Разработка должна быть проведена в несколько стадий:

1. Анализ предметной области и разработка технического задания;
2. Проектирование информационной системы.
3. Реализация информационной системы.

**Этапы разработки**

На стадии анализа предметной области и разработки технического задания должна быть исследована предметная область информационной системы, выполнен анализ существующих решений, выявлены требования к разрабатываемой системе и разработано техническое задание.

На стадии проектирования информационной системы должна быть спроектирована архитектура разрабатываемой информационной системы и спроектирована база данных.

На стадии реализации информационной системы должна быть разработана информационная система для работы с информацией о мероприятиях для абитуриентов.

1. **Порядок контроля и приемки**

**Виды испытаний**

Конкретные требования к видам испытаний не предъявляются.

**Общие требования к приемке работы**

Прием работы проходит на защите курсовых работ 2 курса.

ПРИЛОЖЕНИЕ Б. Руководство пользователя

**УТВЕРЖДЕНО**

А.В.00001-01 РПЛ 01-лу

**EVENT BOOK**

**Руководство пользователя**

**А.В.00001-01 ТЗ 01**

Инв. № подл.

Подпись и дата

Взам. инв. №

Инв. № дубл.

Подпись и дата

**А.В.00001-01 РПЛ 01-лу**

|  |  |  |
| --- | --- | --- |
|  |  | Руководитель разработки |
|  |  |  |
|  |  | \_\_\_\_\_\_\_\_\_Кушев В. О. |
|  |  | “ ” 2020 |
|  |  |  |
|  |  | Исполнитель |
|  |  |  |
|  |  | \_\_\_\_\_\_\_\_Сидоров А.А. |
|  |  | “ ” 2020 |
|  |  |  |

**Введение**

Руководство пользователя представлено для программы, систематизирующей аудиофайлы на персональном компьютере.

* 1. **Область применения**

Программа применима для использования на домашнем компьютере, предназначенного для одного пользователя.

* 1. **Краткое описание возможностей**

Программа применяется для редактирования информации о музыкальных композициях, хранящихся на жестком диске и копирования плейлистов.

* 1. **Уровень подготовки пользователя**

Пользователь должен уметь работать с программами ОС Windows, разбираться в форматах аудиофайлов.

* 1. **Перечень эксплуатационной документации, с которой необходимо ознакомиться пользователю**

Пользователи, которые будут использовать систему, в обязательном порядке должны быть ознакомлены с настоящим Руководством.

**Назначение и условия применения**

* 1. **Функциональное назначение программы**

Назначением программы является создание базы данных с информацией о композициях, хранящихся на компьютере, добавление и удаление полей, редактирование информации о песнях. Также программа позволяет получить список песен, альбомов и авторов по определенным критериям. Также одной из функций является копирование выбранных треков в другую директорию.

* 1. **Условия применения**

Система должна использовать на персональных компьютерах на операционных системах Windows версии 7 и более поздних версиях ОС Windows. Необходима версия .Net Framework 4.5.

**Подготовка к работе**

* 1. **Состав и содержание дистрибутивного носителя данных**

1. IBM PC совместимый с процессором Pentium-4 и выше.

2. ОЗУ 512 Мб и выше.

4. Устройства ввода (мышь и клавиатура).

5. Операционная система: Windows 7 и выше.

* 1. **Порядок загрузки данных и программ**

Если база данных для хранения информации об аудиофайлах хранится на компьютере, необходимо в первую очередь запустить ее, затем программу.

* 1. **Порядок проверки работоспособности**

Для проверки работоспособности необходимо запустить программу, добавить аудиофайлы и сохранить. Рабочая программа должна выдать успешный результат.

**Описание операций**

* 1. **Выполняемые функции и задачи**

Программа обеспечивает возможность выполнения перечисленных ниже функций:

1. Вывод аудиофайлов по критериям;

2. Копирование выбранных файлов в другую директорию;

3. Удаление записей;

4. Добавление записей;

5. Редактирование записей;

6. Поиск добавленных записей.

Программа обеспечивает выполнение следующих задач:

* + - 1. Создание плейлиста для копирования в другую директорию;
      2. Менеджмент хранимых аудиофайлов.

**Аварийные ситуации**

В случае возникновения аварийных ситуаций перезапустите программу. Если ошибки появляются регулярно, напишите по адресу: arseniy.sidorov2013@gmail.com.

ПРИЛОЖЕНИЕ В. Тестирование

|  |  |  |  |
| --- | --- | --- | --- |
| Номер теста | Событие | Ожидаемый результат | Реальный результат |
| 1 | Добавление файла | Успешность операции | Успешность операции |
| 2 | Добавление нескольких файлов | Успешность операции | Успешность операции |
| 3 | Добавления папки | Успешность операции | Успешность операции |
| 4 | Вывести список песен | Успешность операции | Успешность операции |
| 5 | Вывести список альбомов | Успешность операции | Успешность операции |
| 6 | Вывести список исполнителей | Успешность операции | Успешность операции |
| 7 | Обновить таблицу | Успешность операции | Успешность операции |
| 8 | Вывести песни в кратком виде | Успешность операции | Успешность операции |
| 9 | Вывести песни в нормальном виде | Успешность операции | Успешность операции |
| 10 | Вывести песни в подробном виде | Успешность операции | Успешность операции |
| 11 | Настроить вывод запроса альбома | Неактивная кнопка | Неактивная кнопка |
| 12 | Настроить вывод запроса исполнителя | Неактивная кнопка | Неактивная кнопка |
| 13 | Найти по названию песни | Успешность операции | Успешность операции |
| 14 | Найти по названию исполнителя | Успешность операции | Успешность операции |
| 15 | найти по названию альбома | Успешность операции | Успешность операции |
| 16 | Удалить поле | Успешность операции | Успешность операции |
| 17 | Удалить несколько полей | Успешность операции | Успешность операции |
| 18 | Изменить поле альбома с корректным вводом | Успешность операции | Успешность операции |
| 19 | Изменить поле исполнителя с корректным вводом | Успешность операции | Успешность операции |
| 20 | Изменить поле названия песни | Успешность операции | Успешность операции |
| 21 | Изменить идентификатор | Успешность операции | Успешность операции |
| 22 | Изменить длительность песни | Успешность операции | Успешность операции |
| 23 | Изменить дату альбома | Успешность операции | Успешность операции |
| 24 | Изменить дату песни | Успешность операции | Успешность операции |
| 25 | Вывести справку | Успешность операции | Успешность операции |
| 26 | Вывести таблицу в Excel | Успешность операции | Успешность операции |
| 27 | Добавить песню в экспорт | Успешность операции | Успешность операции |
| 28 | Добавить существующую песню в экспорт | Ничего не происходит | Ничего не происходит |
| 29 | Удалить песню из экспорта | Успешность операции | Успешность операции |
| 30 | Удалить несуществующую песню из экспорта | Ничего не происходит | Ничего не происходит |
| 31 | Экспортировать | Успешность операции | Успешность операции |
| 32 | Экспортировать в ту же директорию | Ошибка | Ошибка |
| 33 | Сворачивание программы | Успешность операции | Успешность операции |
| 34 | Закрытие программы | Успешность операции | Успешность операции |
| 35 | Перетаскивание окна программы | Успешность операции | Успешность операции |
| 36 | Изменение размера окна программы | Успешность операции | Успешность операции |