Министерство образования и молодежной политики

Свердловской области

ГАПОУ СО «Ирбитский политехникум»

Курсовая работа

по ПМ.02 Разработка и администрирование баз данных

МДК 02.02. Технология разработки и защиты баз данных

**Разработка программного комплекса "Фабрика новостей" по автоматизации бизнес-процессов редакции телеканала средствами баз данных**

Выполнил:

студент группы ПКС-406

Останин Иван Васильевич

Проверил:

преподаватель

Вишнякова Наталья Викторовна

Ирбит, 2023

Содержание

[Введение 4](#_Toc129872620)

[1. Постановка задачи программного комплекса «Фабрика новостей» 5](#_Toc129872621)

[2. Концептуальная модель базы данных 6](#_Toc129872622)

[3. Логическая модель базы данных 7](#_Toc129872623)

[4. Разработка физической модели базы данных 8](#_Toc129872624)

[5. Разработка интерфейса 10](#_Toc129872625)

[5.1. Форма соединения с сервером БД 10](#_Toc129872626)

[5.2. Главное окно 11](#_Toc129872627)

[5.3. Окна редактирования объектов 11](#_Toc129872628)

[5.4. Окна справки 13](#_Toc129872629)

[Заключение 14](#_Toc129872630)

[Список использованной литературы 15](#_Toc129872631)

[Приложение 1 Программный код DBProvider.cs (Статический провайдер методов работы с БД) 17](#_Toc129872632)

[Приложение 2 Программный код FileDialogs.cs (Набор файловых диалогов) 22](#_Toc129872633)

[Приложение 3 Программный код MaterialViewModel.cs (Модель представления материала) 29](#_Toc129872634)

[Приложение 4 Программный код EditPersonDialog.xaml (Диалог редактирования работника) 33](#_Toc129872635)

[Приложение 5 Программный код MainWindow.xaml (Главное окно) 35](#_Toc129872636)

[Приложение 6 Программный код EditProjectDialog.xaml (Диалог редактирования проекта) 38](#_Toc129872637)

[Приложение 7 Программный код EditFilmingDialog.xaml (Диалог редактирования съёмки) 40](#_Toc129872638)

[Приложение 8 Программный код EditMaterialDialog.xaml (Диалог редактирования материала) 42](#_Toc129872639)

[Приложение 9 Программный код ExplorerPage.xaml (Страница браузера объектов фабрики) 44](#_Toc129872640)

[Приложение 10 Программный код TagList.cs (Класс, представляющий список тегов объекта фабрики) 45](#_Toc129872641)

[Приложение 11 Программный код SyntaxHelpWindow.cs (Окно синтаксиса строки поиска) 48](#_Toc129872642)

[Приложение 12 Программный код EditTagDialog.xaml (Окно редактирования метки) 49](#_Toc129872643)

[Приложение 13 Программный код NewsFactoryObjectViewModel.cs (Базовый класс объекта фабрики новостей) 51](#_Toc129872644)

[Приложение 14 Программный код TagViewModel.cs (Модель представления метки) 53](#_Toc129872645)

[Приложение 15 Программный код ProjectViewModel.cs (Модель представления проекта) 56](#_Toc129872646)

[Приложение 16 Программный код ExplorerViewModel.cs (Модель представления браузера) 58](#_Toc129872647)

[Приложение 17 Программный код CrewListSerializer.cs (Сериалайзер списка работников объекта фабрики) 60](#_Toc129872648)

# Введение

Телеканалы имеют множество бизнес-задач, которые должны быть решены для успешной работы. Эти задачи включают в себя различные аспекты бизнеса, включая управление процессами, управление персоналом, анализ данных и т. д. Для решения этих задач необходимо разработать соответствующий программный комплекс. Основными целями разработки такого комплекса являются автоматизация бизнес-процессов, повышение эффективности и производительности бизнеса, упрощение управления бизнесом и улучшение предоставления услуг.

Создание программного комплекса для решения бизнес-задач телеканала актуально и необходимо для решения многих проблем. Так, он поможет повысить эффективность работы телеканала, а также позволит автоматизировать многие процессы такие как: систематизация и катологизация электронного хранения оперативных материалов работы телеканал, организация архивного хранения материалов и т.д. Кроме того, программный комплекс улучшит аналитические возможности телеканала, что поможет в принятии правильных решений и повышении эффективности бизнес-процессов.

**Цель курсовой работы**– проектирование, разработка, тестирование программного комплекса "Фабрика новостей" с помощью WPF и MongoDB.

Задачи курсовой работы:

* Анализ бизнес-процессов, сформировавшихся в редакции телеканала;
* Проектирование архитектуры программного комплекса «Фабрика новостей»;
* Разработка базы данных для хранения информации о процессах редакции телеканала в СУБД MongoDB;
* Разработка интерфейса пользователя для программного комплекса "Фабрика новостей" с использованием WPF и языка программирования C#;
* Тестирование и отладка программного комплекса;
* Оформление документации к работе.

Перейдем к рссмотрению постановки задачи для проектирования программного комплекса «Фабрика новостей».

# 1. Постановка задачи программного комплекса «Фабрика новостей»

Основные задачи, которые должна решать информационная система:

Хранить данные о съёмках телеканала:

* Название, описание
* Адрес
* Время съёмки
* Информация о работниках и выполняемых ими обязанностях на съёмке
* Метки

Хранить данные о материалах телеканала:

* Название, описание
* Путь к материалу в файловой системе
* Тип материала
* Метки

Хранить данные о проектах телеканала:

* Название, описание
* Использованные материалы
* Путь к проекту в файловой системе
* Срок сдачи проекта
* Тип проекта
* Информация о работниках и выполняемых ими обязанностях в проекте
* Метки

Хранить данные о работниках телеканала:

* Имя, Фамилия
* Должность

Реализовать возможность присвоить хранимым данным меток, указывающих на область, к которой они относится, для упрощения поиска. У метки может быть установлен родитель, и при поиске по метке-родителю в результате поиска должны быть представлены и объекты, содержащие дочерние теги. Каждому объекту можно присвоить неограниченное количество меток.

Реализовать продвинутую систему поиска материалов, съёмок и проектов по множеству настраиваемых условий.

# 2. Концептуальная модель базы данных

Концептуальная модель базы данных при использовании NoSQL СУБД MongoDB отличается от модели реляционных баз данных, так как MongoDB является документноориентированной СУБД, использующий использует формат документов BSON (Binary JSON), который позволяет хранить документы в виде иерархических структур. При проектировании базы данных необходимо учитывать эту структуру и определять иерархию данных.

MongoDB не требует определения схемы данных заранее, что позволяет гибче работать с данными и вносить изменения в структуру базы данных без необходимости пересоздания схемы. Также, это может приводить к неожиданным ошибкам в работе программного комплекса из-за отсутствия жесткой структуры данных.

|  |  |
| --- | --- |
| **Сущности** | **Атрибуты сущностей** |
| Съёмка | Название, описание  Адрес  Время съёмки  Информация о работниках и выполняемых ими обязанностях на съёмке  Присвоенные метки |
| Материал | Название, описание  Путь к материалу в файловой системе  Тип материала  Присвоенные метки  Информация о работниках ответственных за материал |
| Проект | Название, описание  Использованные материалы  Путь к проекту в файловой системе  Срок сдачи проекта  Тип проекта  Информация о работниках и выполняемых ими обязанностях в проекте  Присвоенные метки |
| Работник | Имя, Фамилия  Должность |
| Метка (Тег) | Название, описание  Тег-родитель  Цвет |

# 3. Логическая модель базы данных

В MongoDB для хранения данных используется документоориентированная модель, что позволяет хранить связанные данные внутри одного документа или в нескольких документах с использованием ссылок. В случае использования нескольких коллекций, для программной реализации соединения таблиц в MongoDB можно использовать ссылки на документы.

В данном случае, связанные данные хранятся в разных коллекциях, а в документе коллекции, который ссылается на другой документ, хранится только идентификатор (ObjectId) этого документа.

Таблица 1. Связи между таблицами базы данных

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Связь | Главный документ | Дочерний документ | Поле для связи главного документа | Поле для связи дочернего документа |
| Метки съёмки | Filmings | Tags | Элемент поля «Tags» | \_id |
| Метки материала | Materials | Tags | Элемент поля «Tags» | \_id |
| Метки проекта | Projects | Tags | Элемент поля «Tags» | \_id |
| Работники съёмки | Filmings | Persons | Элемент поля «MyCrew» | \_id |
| Работники материала | Materials | Persons | Элемент поля «MyCrew» | \_id |
| Работники проекта | Projects | Persons | Элемент поля «MyCrew» | \_id |
| Материалы проекта | Projects | Materials | Элемент поля «Materials» | \_id |
| Родитель тега | Tags | Tags | «ParentTagId» | \_id |

# 4. Разработка физической модели базы данных

В случае MongoDB, данные хранятся в виде BSON-документов (Binary JSON), каждый из которых содержит поля и значения, а также ссылки на другие документы.

Физическая модель базы данных программного комплекса "Фабрика новостей" по автоматизации бизнес-процессов редакции телеканала может быть описана следующим образом:

Таблица 2. Коллекции базы данных «NewsFactory»

|  |  |
| --- | --- |
| Коллекция БД | Индексы |
| Filmings | \_id\_ - уникальность поля «\_id» |
| Materials | \_id\_ - уникальность поля «\_id» |
| Persons | \_id\_ - уникальность поля «\_id» |
| Projects | \_id\_ - уникальность поля «\_id» |
| Tag | \_id\_ - уникальность поля «\_id»  Title\_ - уникальность поля «Title» |

Таблица 3. Структура документов коллекции «Filmings»

|  |  |  |
| --- | --- | --- |
| Имя поля | Тип данных | Описание |
| \_id | ObjectId | Идентификатор съёмки |
| Title | String | Название съёмки |
| Description | String | Описание съёмки |
| CreationTime | Date | Дата создания объекта |
| Tags | Array | Метки |
| Status | Int32 | Текущий статус съёмки |
| MyTime | Date | Время съёмки |
| Address | String | Адрес съёмки |

Таблица 4. Структура документов коллекции «Materials»

|  |  |  |
| --- | --- | --- |
| Имя поля | Тип данных | Описание |
| \_id | ObjectId | Идентификатор материала |
| Title | String | Название материала |
| Description | String | Описание материала |
| CreationTime | Date | Дата создания объекта |
| Tags | Array | Метки |
| MyCrew | Array | Работники и выполняемые ими обязанности для материала |
| Path | String | Путь к материалу в файловой системе |
| MyType | Int32 | Тип материала |

Таблица 5. Структура документов коллекции «Persons»

|  |  |  |
| --- | --- | --- |
| Имя поля | Тип данных | Описание |
| \_id | ObjectId | Идентификатор сотрудника |
| FirstName | String | Имя сотрудника |
| LastName | String | Фамилия сотрудника |
| CreationTime | Date | Дата создания объекта |
| Post | String | Должность |

Таблица 6. Структура документов коллекции «Projects»

|  |  |  |
| --- | --- | --- |
| Имя поля | Тип данных | Описание |
| \_id | ObjectId | Идентификатор проекта |
| Title | String | Название проекта |
| Description | String | Описание проекта |
| CreationTime | Date | Дата создания объекта |
| Tags | Array | Метки |
| MyCrew | Array | Работники и выполняемые ими обязанности в проекте |
| Path | String | Путь к проекту в файловой системе |
| MyType | Int32 | Тип проекта |
| DeadLine | Date | Срок сдачи проекта |
| Status | Int32 | Текущий статус работы над проектом |
| Materials | Array | Массив идентификаторов материалов, использованных в проекте |

Таблица 7. Структура документов коллекции «Tags»

|  |  |  |
| --- | --- | --- |
| Имя поля | Тип данных | Описание |
| \_id | ObjectId | Идентификатор метки |
| Title | String | Название метки |
| Description | String | Описание метки |
| CreationTime | Date | Дата создания объекта |
| ParentTagId | ObjectId | Метка-родитель |
| MyColor | Object | Цвет метки |

# 5. Разработка интерфейса

## 5.1. Форма соединения с сервером БД

Перед началом работы «Менеджер проектов» должен связаться с MongoDB сервером. В случае, если это не удаётся по адресу по умолчанию (127.0.0.1: 27017), пользователь должен ввести эти данные в представленной форме:

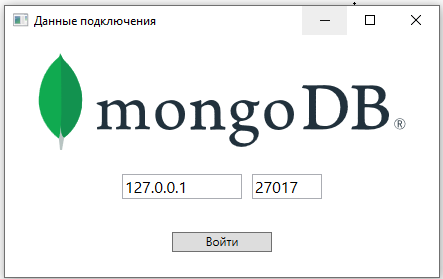


Рис. 1 Форма соединения с MongoDB сервером.

По нажатию кнопки «Войти» приложение попытается связаться с сервером. Если соединение будет установлено, пользователь попадает в главное окно, иначе, форма соединения с сервером будет показана снова.

Если соединение с сервером пропадёт после установления соединения, работа программы будет экстренно завершена, и пользователь получит сообщение об ошибке следующего вида:

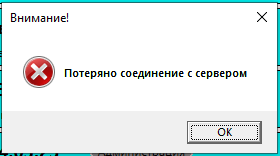


Рис. 2 Сообщение об ошибке при потере соединения.

## 5.2. Главное окно

Главное окно имеет следующий вид:

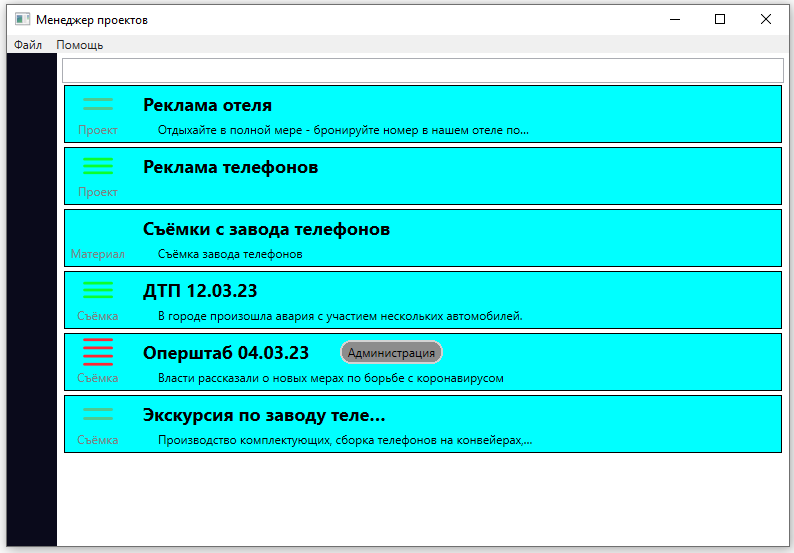


Рис. 2 Главное окно.

Большую часть главного окна занимает интерфейс браузера объектов фабрики, позволяющий искать проекты, съёмки и материалы по множеству настраиваемых условий. Данное условия формулируются в запрос, который необходимо ввести в поисковую строку, находящуюся в верхней части браузера. Запрос формулируется на специальном языке. На данный момент поддерживается иерархический поиск по метке, поиск по пути материала или проекта, поиск по типу объекта, поиск по статусу объекта, поиск по дате создания объекта, поиск по сотруднику и исполняемыми им обязательствам. По двойному клику на результате поиска, откроется окно редактирования данного объекта.

В верхней части главного окна расположено меню, состоящие из двух разделов: «Файл», «Помощь». В меню «Файл» находятся команды для создания новых объектов фабрики: «Проект», «Материал», «Съёмка», «Тег», «Работник». В разделе меню «Помощь» помещены команды открытия окон справочной информацией: «Синтаксис поисковой строки» и «О программе».

В левой части главного окна расположено боковое меню, расширяющееся при наведении курсора. В данном меню находятся кнопки, с помощью которых можно перейти в браузер проектов с предустановленным запросом.

## 5.3. Окна редактирования объектов

Всего для редактирования доступно 5 объектов: «Проект», «Материал», «Съёмка», «Тег», «Работник». Вызвать окно редактирования объекта можно двойным кликом по нему из браузера или из контекстного меню представления данного объекта.

В верхней части окон редактирования можно изменить заголовок объекта. Описание объекта изменяется ниже. В нижней части окна находятся две кнопки: «Отмена» и «Сохранить». По нажатию кнопки отмены все изменения внесённые локально будут сброшены и состояние объекта будет сброшено до последнего сохранённого в БД. При нажатии кнопки сохранения все локальные изменения будут сохранены в БД.

Если объект содержит ссылки на другие объекты, например, проект ссылается на материалы, использованные в нём, и на работников, которые ответственны за данный проект, то редактирование подобных ссылок вынесено на отдельную вкладку, как показано на данном рисунке:

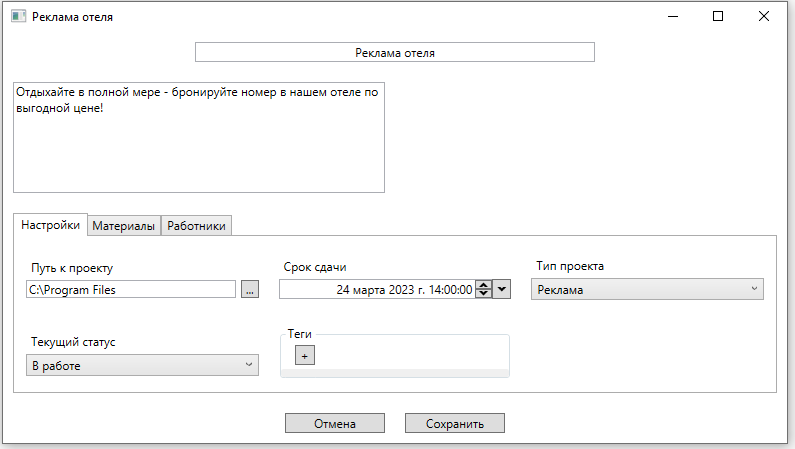


Рис. 3 Окно редактирования проекта.

Исключением являются метки (теги). Их редактирование вынесено в область основных настроек. И так как теги имеют свойства наследовать друг друга, то при попытке добавить к объекту родителя или потомка уже присвоенного тега, будет появляться сообщение об особенности данного действия, как показано на рисунках:

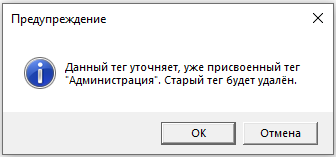


Рис. 4 Предупреждение при попытке добавить тег, являющийся потомком уже присвоенному.

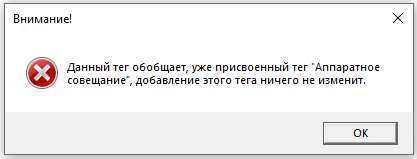


Рис. 5 Сообщение об ошибке при попытке добавить тег, являющийся родителем уже присвоенному.

## 5.4. Окна справки

Используя пункт меню «Помощь» главного окна, пользователь может открыть окна со справочной информацией: «Синтаксис поиска» и «О программе».

Окно «Синтаксис поиска» предоставляет информацию о правилах формирования запроса для поисковой строки браузера объектов фабрики. К данному окну пользователь может обратиться для уточнения правил или констант формирования запроса во время работы с программой. Данное окно имеет следующий вид:

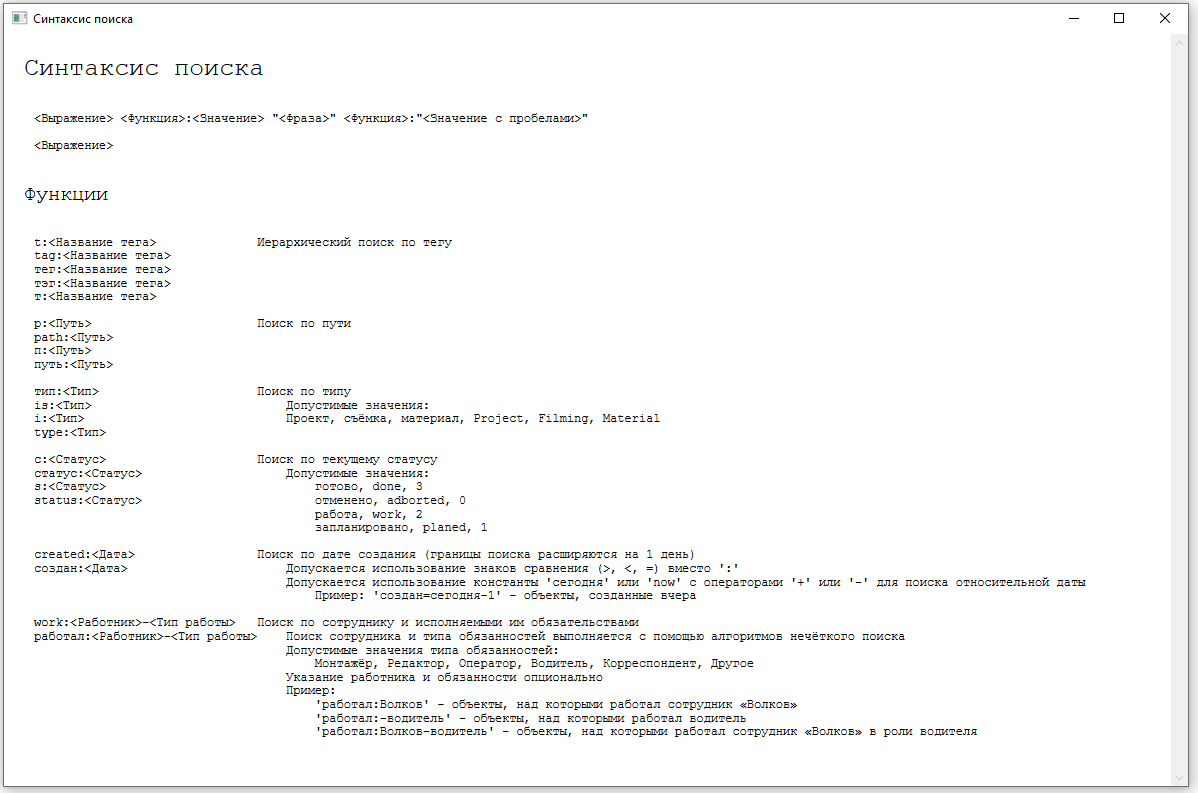


Рис. 6 Окно «Синтаксис поиска».

# Заключение

В результате выполнения курсовой работы был разработан программный комплекс «Фабрика новостей», основанный на технологиях WPF и MongoDB.

В процессе работы были изучены технологий WPF и MongoDB, была спроектирована архитектура базы данных, был разработан пользовательский интерфейс с использованием WPF и шаблона MVVM, написан код для связи пользовательского интерфейса с базой данных, проведено тестирование программного комплекса.

В результате выполнения курсовой работы было установлено, что использование WPF позволяет создать удобный и привлекательный пользовательский интерфейс с возможностью применения визуальных эффектов и анимации. MongoDB, в свою очередь, обеспечивает высокую скорость чтения и записи данных, что позволяет быстро и эффективно обрабатывать большое количество запросов.

В целом, программный комплекс "Фабрика новостей" является эффективным и удобным инструментом для автоматизации бизнес-процессов телеканала. Дальнейшее развитие проекта может быть связано с реализацией дополнительных возможностей програмного комплекса таких как: формирование отчётности, ведение мониторинга работы в реальном времнени, а также улучшением механизмов обработки ошибок и усовершенствованием пользовательского интерфейса.

В процессе работы над курсовым проектом получил практический опыт в области разработки базы данных и приложений с помощью технологий WPF и MongoDB, закрепил знания по основным принципам построения концептуальной, логической и физической модели данных, освоил современные инструментальные средства разработки схемы базы данных, методы описания схем баз данных в современных системах управления базами данных.

# Список использованной литературы

1. The Official C# .NET Driver for MongoDB [Электронный ресурс] // GitHub : Хостинг IT-проектов. URL : https://github.com/mongodb/mongo-csharp-driver (дата обращения: 22.02.2023)
2. MongoDB Documentation [Электронный ресурс] // MongoDB : Сайт разработчиков СУБД MongoDB. URL : https://www.mongodb.com/docs/ (дата обращения: 27.02.2023)
3. MongoDB .NET Driver API Documentation [Электронный ресурс] // MongoDB .NET Driver : Документация по драйверу .NET MongoDB. URL: https://mongodb.github.io/mongo-csharp-driver/2.18/apidocs/html/N\_MongoDB\_Driver.htm (дата обращения: 24.02.2023)
4. Official .NET driver for MongoDB [Электронный ресурс] // NuGet : Система управления пакетами для .NET. URL : https://www.nuget.org/packages/mongodb.driver (дата обращения: 21.02.2023)
5. MongoDB Driver Quick Tour [Электронный ресурс] // MongoDB .NET Driver: Документация по драйверу .NET MongoDB. URL: https://mongodb.github.io/mongo-csharp-driver/2.18/getting\_started/quick\_tour/ (дата обращения: 28.02.2023)
6. Serialization Tutorial [Электронный ресурс] // MongoDB .NET Driver : Документация по драйверу .NET MongoDB. URL: https://mongodb.github.io/mongo-csharp-driver/1.11/serialization/ (дата обращения: 27.01.2023)
7. C# Class MongoDB.Bson.IO.BsonDocumentWriter [Электронный ресурс] // Hot Examples : Агрегатор примеров использования исходного кода. URL : https://csharpdoc.hotexamples.com/class/MongoDB.Bson.IO/BsonDocumentWriter# (дата обращения: 24.01.2023)
8. Using C# MongoDB Driver, how to serialize a collection of object referements? [Электронный ресурс] // Stack Overflow : Система вопросов и ответов о программировании. URL : https://stackoverflow.com/questions/27537543/using-c-sharp-mongodb-driver-how-to-serialize-a-collection-of-object-referement (дата обращения: 21.01.2023)
9. Inserting .NET object into MongoDB [Электронный ресурс] // Stack Overflow : Система вопросов и ответов о программировании. URL : https://stackoverflow.com/questions/15224048/inserting-net-object-into-mongodb (дата обращения: 26.02.2023)
10. Full text search in mongodb in .net [Электронный ресурс] // Stack Overflow : Система вопросов и ответов о программировании. URL : https://stackoverflow.com/questions/41356544/full-text-search-in-mongodb-in-net (дата обращения: 28.01.2023)
11. In c# what does 'where T : class' mean? [Электронный ресурс] // Stack Overflow : Система вопросов и ответов о программировании. URL : https://stackoverflow.com/questions/3786774/in-c-sharp-what-does-where-t-class-mean (дата обращения: 22.01.2023)
12. MongoDb custom collection serializer [Электронный ресурс] // Stack Overflow : Система вопросов и ответов о программировании. URL : https://stackoverflow.com/questions/30421379/mongodb-custom-collection-serializer (дата обращения: 25.02.2023)
13. How to get File Created Date and Modified Date [Электронный ресурс] // Stack Overflow : Система вопросов и ответов о программировании. URL : https://stackoverflow.com/questions/23243336/how-to-get-file-created-date-and-modified-date (дата обращения: 23.01.2023)
14. BsonExtraElements with custom serializer [Электронный ресурс] // MongoDB : Сайт разработчиков СУБД MongoDB. URL : https://www.mongodb.com/community/forums/t/bsonextraelements-with-custom-serializer/16742 (дата обращения: 01.03.2023)
15. Словари ресурсов [Электронный ресурс] // METANIT : Сайт о программировании. URL : https://metanit.com/sharp/wpf/9.3.php (дата обращения: 20.02.2023)
16. Generic type constraint [Электронный ресурс] // Microsoft Learn : онлайн-платформа для обучения по технологиям Microsoft. URL : https://learn.microsoft.com/en-us/dotnet/csharp/language-reference/keywords/where-generic-type-constraint?source=recommendations (дата обращения: 26.01.2023)
17. Constraints on type parameters [Электронный ресурс] // Microsoft Learn : онлайн-платформа для обучения по технологиям Microsoft. URL : https://learn.microsoft.com/en-us/dotnet/csharp/programming-guide/generics/constraints-on-type-parameters?redirectedfrom=MSDN (дата обращения: 20.02.2023)

## Приложение 1 Программный код DBProvider.cs (Статический провайдер методов работы с БД)

using System;

using System.Collections.Generic;

using System.IO;

using System.Linq;

using System.Threading;

using System.Windows;

using System.Windows.Media;

using System.Xml.Serialization;

using MongoDB.Bson;

using MongoDB.Bson.Serialization;

using MongoDB.Driver;

using MongoDB.Driver.Core.Clusters;

using NfModels.Services.Serializers;

using NfModels.ViewModels.Base;

using NfModels.ViewModels.NewsFactoryObjects;

using NfModels.Views.Dialogs;

namespace NfModels.Services;

/// <summary>

/// Статический провайдер методов работы с БД

/// </summary>

public static class DBProvider

{

private const string RecentConnectionPath = "recent.d";

private static readonly IMongoCollection<ProjectViewModel> ProjectsCollection;

private static readonly IMongoCollection<FilmingViewModel> FilmingsCollection;

private static readonly IMongoCollection<MaterialViewModel> MaterialsCollection;

private static readonly IMongoCollection<PersonViewModel> PersonsCollection;

private static readonly IMongoCollection<TagViewModel> TagsCollection;

static DBProvider()

{

BsonSerializer.RegisterSerializer(typeof(Color), new ColorSerializer());

BsonSerializer.RegisterSerializer(typeof(TagList), new TagListSerializer());

BsonSerializer.RegisterSerializer(typeof(CrewList), new CrewListSerializer());

BsonSerializer.RegisterSerializer(typeof(MaterialList), new MaterialSerializer());

var client = Connect();

var db = client.GetDatabase("NewsFactory");

ProjectsCollection = db.GetCollection<ProjectViewModel>("Projects");

FilmingsCollection = db.GetCollection<FilmingViewModel>("Filmings");

MaterialsCollection = db.GetCollection<MaterialViewModel>("Materials");

PersonsCollection = db.GetCollection<PersonViewModel>("Persons");

TagsCollection = db.GetCollection<TagViewModel>("Tags");

}

/// <summary>

/// Наладка соединения с БД

/// </summary>

/// <returns>Клиент MongoDB</returns>

private static MongoClient Connect()

{

var firstTry = true;

var settings = LoadConnectionSettings();

while (true)

{

MongoClient client;

if (!firstTry) {

settings = DBEnterDialog.PickSettings();

if (settings is null) {Application.Current.Shutdown();

return null!;

}

}

client = new MongoClient(settings);

if (CheckConnection(client))

{

SaveConnectionSettings((settings.Server.Host, settings.Server.Port));

return client;

}

firstTry = false;

}

}

/// <summary>

/// Сохранение настроек соединения в файл

/// </summary>

/// <param name="settings">Настройки соединения</param>

private static void SaveConnectionSettings((string, int) settings)

{

var xml = new XmlSerializer(typeof((string, int)));

using var fs = new FileStream(RecentConnectionPath, FileMode.Create, FileAccess.Write);

xml.Serialize(fs, settings);

fs.Close();

}

/// <summary>

/// Загрузка настроек соединения из файла

/// </summary>

/// <returns>Настройки соединения</returns>

private static MongoClientSettings LoadConnectionSettings()

{

if (File.Exists(RecentConnectionPath))

{

var xml = new XmlSerializer(typeof((string, int)));

using var fs = new FileStream(RecentConnectionPath, FileMode.Open);

var newSettings = ((string, int))xml.Deserialize(fs);

fs.Close();

return new MongoClientSettings {

Server = new MongoServerAddress(newSettings.Item1, newSettings.Item2),

ConnectTimeout = TimeSpan.FromSeconds(5),

};

}

var settings = new MongoClientSettings {

Server = new MongoServerAddress("localhost", 27017),

ConnectTimeout = TimeSpan.FromSeconds(5),

};

return settings;

}

/// <summary>

/// Проверка статуса соединения с БД

/// </summary>

/// <param name="client">Клиент соединения</param>

/// <returns>Есть ли соединение</returns>

private static bool CheckConnection(IMongoClient client)

{

var count = 0;

while (client.Cluster.Description.State == ClusterState.Disconnected)

{

Thread.Sleep(100);

if (count++ >= 50) return false;

}

return true;

}

/// <summary>

/// Сохранение объекта фабрики в БД

/// </summary>

/// <param name="instance">Экземпляр объекта фабрики</param>

public static void Save(NewsFactoryObjectViewModel instance)

{

switch (instance)

{

case ProjectViewModel pr:

Save(pr);

break;

case FilmingViewModel pr:

Save(pr);

break;

case MaterialViewModel pr:

Save(pr);

break;

case TagViewModel pr:

Save(pr);

break;

case PersonViewModel pr:

Save(pr);

break;

}

}

/// <summary>

/// Загрузка сохранённых данных об объекте в БД

/// </summary>

/// <param name="viewModel">Экземпляр объекта фабрики</param>

/// <returns>Последние сохранённые данные объекта фабрики</returns>

/// <exception cref="ArgumentOutOfRangeException">Неизвестный тип объекта</exception>

public static NewsFactoryObjectViewModel? LoadSavedData(NewsFactoryObjectViewModel viewModel)

{

return viewModel switch

{

ProjectViewModel pr => ProjectsCollection.FindSync(GetVmFindFilter(pr)).FirstOrDefault(),

FilmingViewModel pr => FilmingsCollection.FindSync(GetVmFindFilter(pr)).FirstOrDefault(),

MaterialViewModel pr => MaterialsCollection.FindSync(GetVmFindFilter(pr)).FirstOrDefault(),

TagViewModel pr => TagsCollection.FindSync(GetVmFindFilter(pr)).FirstOrDefault(),

PersonViewModel pr => PersonsCollection.FindSync(GetVmFindFilter(pr)).FirstOrDefault(),

\_ => throw new ArgumentOutOfRangeException()

};

}

/// <summary>

/// Сохранение модели представления проекта в БД

/// </summary>

/// <param name="instance">Экземпляр проекта</param>

private static void Save(ProjectViewModel instance)

{

SaveObject(ProjectsCollection, instance);

}

/// <summary>

/// Сохранение модели представления съёмки в БД

/// </summary>

/// <param name="instance">Экземпляр съёмки</param>

private static void Save(FilmingViewModel instance)

{

SaveObject(FilmingsCollection, instance);

}

/// <summary>

/// Сохранение модели представления материала в БД

/// </summary>

/// <param name="instance">Экземпляр материала</param>

private static void Save(MaterialViewModel instance)

{

SaveObject(MaterialsCollection, instance);

}

/// <summary>

/// Сохранение модели представления тега в БД

/// </summary>

/// <param name="instance">Экземпляр тега</param>

private static void Save(TagViewModel instance)

{

SaveObject(TagsCollection, instance);

}

/// <summary>

/// Сохранение модели представления работника в БД

/// </summary>

/// <param name="instance">Экземпляр работника</param>

private static void Save(PersonViewModel instance)

{

SaveObject(PersonsCollection, instance);

}

/// <summary>

/// Универсальное сохранение модели представления объекта фабрики в БД

/// </summary>

/// <param name="collection">Ссылка на коллекцию для сохранения</param>

/// <param name="vm">Экземпляр модели представления</param>

/// <typeparam name="T">Тип объекта фабрики</typeparam>

private static void SaveObject<T>(IMongoCollection<T> collection, T vm) where T : NewsFactoryObjectViewModel

{

try

{

var r = collection.FindOneAndReplace(GetVmFindFilter(vm), vm);

if (r is not null) return;

collection.InsertOne(vm);

if (typeof(T) == typeof(TagViewModel)) TagManager.Refresh();

if (typeof(T) == typeof(PersonViewModel)) PersonManager.Refresh();

}

catch (TimeoutException)

{

LostConnection();

}

}

/// <summary>

/// Вывод сообщения и прекращение работы приложения при разрыве соединения с БД

/// </summary>

private static void LostConnection()

{

MessageBox.Show("Потеряно соединение с сервером", "Внимание!", MessageBoxButton.OK,

MessageBoxImage.Error);

Application.Current.Shutdown();

}

/// <summary>

/// Получение фильтра для нахождения данных о модели представления в БД

/// </summary>

/// <param name="vm">Искомая модель представления</param>

/// <typeparam name="T">Тип модели представления</typeparam>

/// <returns>Фильтр для нахождения данных о модели представления в БД</returns>

private static FilterDefinition<T> GetVmFindFilter<T>(T vm) where T : NewsFactoryObjectViewModel

{

return Builders<T>.Filter.Eq(r => r.Id, vm.Id);

}

/// <summary>

/// Поиск объектов фабрики по запросу поисковой строки

/// </summary>

/// <param name="search">Запросу поисковой строки</param>

/// <returns>Объекты фабрики, удовлетворяющие запросу</returns>

public static IEnumerable<NewsFactoryObjectViewModel> LoadObjects(Search search)

{

try

{

var projects = search.Filter.ProjectFilter is null

? Enumerable.Empty<NewsFactoryObjectViewModel>()

: ProjectsCollection.FindSync(search.Filter.ProjectFilter).ToEnumerable();

var materials = search.Filter.MaterialFilter is null

? Enumerable.Empty<NewsFactoryObjectViewModel>()

: MaterialsCollection.FindSync(search.Filter.MaterialFilter).ToEnumerable()

.Cast<CrewedObjectViewModel>();

var filmings = search.Filter.FilmingFilter is null

? Enumerable.Empty<NewsFactoryObjectViewModel>()

: FilmingsCollection.FindSync(search.Filter.FilmingFilter).ToEnumerable();

return projects.Concat(materials).Concat(filmings);

}

catch (MongoCommandException e)

{

return Enumerable.Empty<NewsFactoryObjectViewModel>();

}

catch (TimeoutException e)

{

LostConnection();

return Enumerable.Empty<NewsFactoryObjectViewModel>();

}

}

/// <summary>

/// Загрузка всех тегов

/// </summary>

/// <returns>Все теги</returns>

public static IQueryable<TagViewModel> LoadTags()

{

return TagsCollection.AsQueryable();

}

/// <summary>

/// Загрузка всех работников

/// </summary>

/// <returns>Все работники</returns>

public static IQueryable<PersonViewModel> LoadPersons()

{

return PersonsCollection.AsQueryable();

}

/// <summary>

/// Загрузка всех материалов по набору их идентификаторов

/// </summary>

/// <param name="ids">Набору искомых идентификаторов </param>

/// <returns>Набор материалов</returns>

public static IEnumerable<MaterialViewModel> LoadMaterials(IEnumerable<ObjectId> ids)

{

return MaterialsCollection.AsQueryable().Where(m => ids.Contains(m.Id));

}

}

## Приложение 2 Программный код FileDialogs.cs (Набор файловых диалогов)

using System;

using System.Collections.Generic;

using System.IO;

using System.Linq;

using System.Windows.Forms;

using System.Xml.Serialization;

using NfModels.ViewModels.NewsFactoryObjects;

using NfModels.ViewModels.ScriptParts.Base;

using OpenFileDialog = Microsoft.Win32.OpenFileDialog;

using SaveFileDialog = Microsoft.Win32.SaveFileDialog;

namespace NfModels.Services;

public static class FileDialogs

{

/// <summary>

/// Путь по умолчанию для сохранения сюжетов

/// </summary>

public static string DefaultScriptPath =

Path.Join(Environment.GetFolderPath(Environment.SpecialFolder.MyDocuments), "Сюжеты");

/// <summary>

/// Событие, вызываемое при изменении файлов

/// </summary>

public static event EventHandler<FileManagementEventArgs> FileManagement = null!;

/// <summary>

/// Сохранение сюжета

/// </summary>

/// <param name="script">Сюжет</param>

/// <param name="saveAs">Сохранить по новому пути</param>

/// <returns>Удалось ли сохранить</returns>

public static bool Save(ScriptViewModel script, bool saveAs = false)

{

string path;

if (string.IsNullOrWhiteSpace(script.Path) || saveAs)

{

string? newPath;

if (script.IsDefaultName && script.ScriptParts.Count > 0 && script.ScriptParts[0] is TextPart textPart &&

textPart.Content.Length > 10)

newPath = GetNewScriptPath(script.Path,

textPart.Content[..Math.Min(textPart.Content.Length, 50)]);

else newPath = GetNewScriptPath(script.Path, script.Title);

if (newPath is null) return false;

path = newPath;

script.Path = path;

if (script.IsDefaultName)

{

script.Title = Path.GetFileNameWithoutExtension(path);

script.IsDefaultName = false;

}

}

else

{

path = script.Path;

}

try

{

var xml = new XmlSerializer(typeof(ScriptViewModel));

using var fs = new FileStream(path, FileMode.Create, FileAccess.Write);

xml.Serialize(fs, script);

}

catch

{

OnFileManagement(new FileManagementEventArgs

{

ActionType = FileManagementType.Save,

IsSuccesses = false

});

return false;

}

script.IsNotSaved = false;

OnFileManagement(new FileManagementEventArgs

{

Script = script,

ActionType = FileManagementType.Save,

IsSuccesses = true

});

return true;

}

/// <summary>

/// Загрузка сюжета по его пути

/// </summary>

/// <param name="path">Путь к сюжету</param>

/// <returns>Сюжет</returns>

public static ScriptViewModel? Load(string path)

{

try

{

var xml = new XmlSerializer(typeof(ScriptViewModel));

using var fs = new FileStream(path, FileMode.Open);

var newScript = (ScriptViewModel)xml.Deserialize(fs);

newScript.Path = path;

newScript.OnLoaded();

OnFileManagement(new FileManagementEventArgs

{

Script = newScript,

ActionType = FileManagementType.Load,

IsSuccesses = true

});

return newScript;

}

catch (Exception e)

{

return null;

}

}

/// <summary>

/// Вызов диалога получения нового пути сюжета

/// </summary>

/// <param name="previousPath">Предыдущий путь сюжета</param>

/// <param name="title">Заголовок окна</param>

/// <param name="filter">Фильтр</param>

/// <returns>Новый путь сохранения</returns>

public static string? GetNewScriptPath(string? previousPath, string title,

string filter = "Сюжеты|\*.nscr|All files|\*.\*")

{

var dialog = new SaveFileDialog

{

Filter = filter,

RestoreDirectory = true,

OverwritePrompt = true,

Title = title

};

if (!string.IsNullOrWhiteSpace(previousPath))

{

dialog.FileName = Path.GetFileName(previousPath);

dialog.InitialDirectory = Path.GetDirectoryName(previousPath);

}

else

{

dialog.InitialDirectory = DefaultScriptPath;

dialog.FileName = title;

}

var isSuccesses = dialog.ShowDialog();

switch (isSuccesses)

{

case null:

case false:

return null;

default:

return dialog.FileName;

}

}

/// <summary>

/// Вызов диалога открытия материалов из файлов

/// </summary>

/// <param name="previousPath">Предыдущий путь</param>

/// <param name="filter">Фильтр</param>

/// <param name="multiselect">Возможность множественного выбора</param>

/// <returns>Набор материалов</returns>

public static IEnumerable<Material>? OpenMaterial(string? previousPath = null, string filter = "Видео|\*.\*",

bool multiselect = true)

{

var dialog = new OpenFileDialog

{

Filter = filter,

RestoreDirectory = true,

Title = "Открыть материал",

Multiselect = multiselect

};

if (!string.IsNullOrWhiteSpace(previousPath))

{

dialog.FileName = Path.GetFileName(previousPath);

dialog.InitialDirectory = Path.GetDirectoryName(previousPath);

}

// TODO: Настройки пути по умолчанию поиска видео

// dialog.InitialDirectory = SettingsManager.CurrentSettings.DefaultScriptPath;

var isSuccesses = dialog.ShowDialog();

switch (isSuccesses)

{

case null:

case false:

return null;

default:

return dialog.FileNames.Select(m => new Material { Path = m });

}

}

/// <summary>

/// Вызов диалога открытия файлов

/// </summary>

/// <param name="previousPath">Предыдущий путь</param>

/// <param name="filter">Фильтр</param>

/// <param name="multiselect">Возможность множественного выбора</param>

/// <returns>Набор путей</returns>

public static IEnumerable<string>? OpenFiles(string? previousPath = null, string filter = "Файлы |\*.\*",

bool multiselect = true)

{

var dialog = new OpenFileDialog

{

Filter = filter,

RestoreDirectory = true,

Title = "Открыть материал",

Multiselect = multiselect

};

if (!string.IsNullOrWhiteSpace(previousPath))

{

dialog.FileName = Path.GetFileName(previousPath);

dialog.InitialDirectory = Path.GetDirectoryName(previousPath);

}

// TODO: Настройки пути по умолчанию поиска видео

// dialog.InitialDirectory = SettingsManager.CurrentSettings.DefaultScriptPath;

var isSuccesses = dialog.ShowDialog();

switch (isSuccesses)

{

case null:

case false:

return null;

default:

return dialog.FileNames;

}

}

/// <summary>

/// Вызов диалога получения пути для сохранения

/// </summary>

/// <param name="previousPath">Предыдущий путь</param>

/// <param name="filter">Фильтр</param>

/// <param name="title">Заголовок окна</param>

/// <returns>Путь сохранения</returns>

public static string GetNewFilePath(string? previousPath = null, string filter = "Файлы |\*.\*", string title = "Экспорт")

{

var dialog = new SaveFileDialog

{

Filter = filter,

RestoreDirectory = true,

Title = title,

};

if (!string.IsNullOrWhiteSpace(previousPath))

{

dialog.FileName = Path.GetFileName(previousPath);

dialog.InitialDirectory = Path.GetDirectoryName(previousPath);

}

var isSuccesses = dialog.ShowDialog();

switch (isSuccesses)

{

case null:

case false:

return "";

default:

return dialog.FileName;

}

}

/// <summary>

/// Вызов диалога открытия папки

/// </summary>

/// <param name="initialDirectory">Начальная директория</param>

/// <param name="title">Заголов окна</param>

/// <returns>Путь к директории</returns>

public static string? GetDirectory(string? initialDirectory, string? title)

{

using var dialog = new FolderBrowserDialog();

dialog.Description = title;

dialog.SelectedPath = initialDirectory;

var result = dialog.ShowDialog();

switch (result)

{

case DialogResult.Yes:

case DialogResult.OK:

return dialog.SelectedPath;

case DialogResult.None:

case DialogResult.Cancel:

case DialogResult.Abort:

case DialogResult.Retry:

case DialogResult.Ignore:

case DialogResult.No:

default:

return null;

}

}

/// <summary>

/// Вызов диалога выбора файла пути сюжета

/// </summary>

/// <returns>Путь к файлу сюжета</returns>

public static string? GetExistsScriptFileName()

{

var dialog = new OpenFileDialog

{

CheckFileExists = true,

Filter = "Сюжет|\*.nscr",

InitialDirectory = DefaultScriptPath

};

var result = dialog.ShowDialog();

switch (result)

{

case null:

case false:

return null;

}

return dialog.FileName;

}

/// <summary>

/// Вызов события при изменении файлов

/// </summary>

/// <param name="e">Аргументы события</param>

private static void OnFileManagement(FileManagementEventArgs e)

{

FileManagement?.Invoke(null, e);

}

/// <summary>

/// Запись данных во временны файл

/// </summary>

/// <param name="data">Данные для записи</param>

public static void WriteTempFile(string data)

{

File.WriteAllText(Path.Join(Path.GetTempPath(), "ScriptEditorTempFile.dat"), data);

}

/// <summary>

/// Чтение данных из временного файла

/// </summary>

/// <returns>Данные временного файла</returns>

public static string? ReadTempFile()

{

var path = Path.Join(Path.GetTempPath(), "ScriptEditorTempFile.dat");

if (!File.Exists(path)) return null;

var data = File.ReadAllText(path);

File.Delete(path);

return data;

}

}

/// <summary>

/// Аргумент события изменения файлов

/// </summary>

public class FileManagementEventArgs : EventArgs

{

public ScriptViewModel? Script { get; set; }

public FileManagementType ActionType { get; set; }

public bool IsSuccesses { get; set; }

}

/// <summary>

/// Тип действия

/// </summary>

public enum FileManagementType

{

Save,

Load

}

## Приложение 3 Программный код MaterialViewModel.cs (Модель представления материала)

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.IO;

using System.Linq;

using System.Windows.Input;

using NfModels.Infrastructure.Commands;

using NfModels.Services;

using NfModels.ViewModels.Base;

namespace NfModels.ViewModels.NewsFactoryObjects;

/// <summary>

/// Модель представления материала

/// </summary>

public class MaterialViewModel : CrewedObjectViewModel

{

/// <summary>

/// Массив известных расширений видеофайлов

/// </summary>

public static readonly string[] VideoFileExtensions = new[]

{

".webm", ".mkv", ".flv", ".flv", ".vob", ".ogv", ".ogg", ".drc", ".gif", ".gifv", ".mng", ".avi",

".MTS", ".M2TS", ".TS", ".mov", ".qt", ".wmv", ".yuv", ".rm", ".rmvb", ".viv", ".asf", ".amv",

".mp4", ".m4p", ".m4v", ".mpg", ".mp2", ".mpeg", ".mpe", ".mpv", ".mpg", ".mpeg", ".m2v", ".m4v",

".svi", ".3gp", ".3g2", ".mxf", ".roq", ".nsv", ".flv", ".f4v", ".f4p", ".f4a", ".f4b"

}.Select(t => t.ToLower()).ToArray();

/// <summary>

/// Массив известных расширений аудиофайлов

/// </summary>

public static readonly string[] AudioFileExtensions = new[]

{

".3gp", ".aa", ".aac", ".aax", ".act", ".aiff", ".alac", ".amr", ".ape", ".au", ".awb", ".dss", ".dvf", ".flac",

".gsm", ".iklax", ".ivs", ".m4a", ".m4b", ".m4p", ".mmf", ".mp3", ".mpc", ".msv", ".nmf", ".ogg", ".oga",

".mogg", ".opus", ".ra", ".rm", ".raw", ".rf64", ".sln", ".tta", ".voc", ".vox", ".wav", ".wma", ".wv", ".webm",

".8svx", ".cda"

}.Select(t => t.ToLower()).ToArray();

/// <summary>

/// Массив известных расширений изображений

/// </summary>

public static readonly string[] PictureFileExtensions = new[]

{

".006", ".007", ".3DMF", ".3DX", ".8PBS", ".ABM", ".ABR", ".ADI", ".AEX", ".AI", ".AIS", ".ALBM", ".AMU",

".ARD", ".ART", ".ARW", ".ASAT", ".B16", ".BIL", ".BLEND", ".BLKRT", ".BLZ", ".BMC", ".BMC", ".BMP", ".BOB",

".BR4", ".BR5", ".C4", ".CADRG", ".CATPART", ".CCX", ".CDR", ".CDT", ".CDX", ".CGM", ".CHT", ".CM2", ".CMX",

".CMZ", ".COMICDOC", ".CPL", ".CPS", ".CPT", ".CR2", ".CSF", ".CV5", ".CVG", ".CVI", ".CVI", ".CVX", ".DAE",

".DCIM", ".DCM", ".DCR", ".DCS", ".DDS", ".DESIGN", ".DIB", ".DJV", ".DJVU", ".DNG", ".DRG", ".DRW", ".DRWDOT",

".DT2", ".DVL", ".DWB", ".DWF", ".DXB", ".EASM", ".EC3", ".EDP", ".EDRW", ".EDW", ".EMF", ".EPRT", ".EPS",

".EPSF", ".EPSI", ".EXR", ".FAC", ".FACE", ".FBM", ".FBX", ".FC2", ".FCZ", ".FD2", ".FH11", ".FHD", ".FIT",

".FLIC", ".FLM", ".FM", ".FPF", ".FS", ".FW.PNG", ".FXG", ".GIF", ".GRAFFLE", ".GTX", ".HD2", ".HDZ", ".HPD",

".HPI", ".HR2", ".HTZ4", ".ICL", ".ICS", ".IDW", ".IEF", ".IGES", ".IGR", ".ILBM", ".ILM", ".IMA", ".IME",

".IMI", ".IMS", ".INDD", ".INDT", ".IPJ", ".IRF", ".ITC2", ".ITHMB", ".J2K", ".JIFF", ".JNG", ".JPEG", ".JPF",

".JPG", ".JPG2", ".JPS", ".JPW", ".JT", ".JWL", ".JXR", ".KDC", ".KODAK", ".KPG", ".LDA", ".LDM", ".LET",

".LT2", ".LTZ", ".LVA", ".LVF", ".LXF", ".MAC", ".MACP", ".MCS", ".MCZ", ".MDI", ".MGS", ".MGX", ".MIC", ".MIP",

".MNG", ".MPF", ".MPO", ".MTZ", ".MUR", ".MUR", ".NAV", ".NCR", ".NEU", ".NFF", ".NJB", ".NTC", ".NTH", ".ODI",

".ODIF", ".OLA", ".OPD", ".ORA", ".OTA", ".OTB", ".OTC", ".OTG", ".OTI", ".OVW", ".P21", ".P2Z", ".PAT", ".PC6",

".PC7", ".PCD", ".PCT", ".PCX", ".PDF", ".PDN", ".PEF", ".PI2", ".PIC", ".PIC", ".PICNC", ".PICTCLIPPING",

".PL0", ".PL2", ".PLN", ".PMB", ".PNG", ".POL", ".PP2", ".PPSX", ".PRW", ".PS", ".PS", ".PSB", ".PSD", ".PSF",

".PSG", ".PSP", ".PSPIMAGE", ".PSQ", ".PVL", ".PWD", ".PWS", ".PX", ".PXR", ".PZ2", ".PZ3", ".QTIF", ".QTZ",

".QXD", ".RIC", ".RLC", ".RLE", ".RW2", ".SDK", ".SDR", ".SEC", ".SFW", ".SIG", ".SKP", ".SLDASM", ".SLDDRW",

".SLDPRT", ".SNX", ".SRF", ".SST", ".SUN", ".SVG", ".SVGZ", ".TARGA", ".TCW", ".TCX", ".TEX", ".TGA", ".TIF",

".TIFF", ".TJP", ".TN", ".TPF", ".TPX", ".TRIF", ".TRX", ".U3D", ".UPX", ".URT", ".UTX", ".V00", ".V3D", ".VFS",

".VGA", ".VHD", ".VIS", ".VRL", ".VTX", ".WB1", ".WBC", ".WBD", ".WBZ", ".WEBP", ".WGS", ".WI", ".WMF", ".WNK",

".XDW", ".XIP", ".XSI", ".X\_B", ".X\_T", ".ZDL", ".ZIF", ".ZNO", ".ZPRF", ".ZT"

}.Select(t => t.ToLower()).ToArray();

/// <summary>

/// Массив известных расширений текстовых файлов

/// </summary>

public static readonly string[] TextFileExtensions = new[]

{

".txt", ".doc", ".docx"

}.Select(t => t.ToLower()).ToArray();

/// <summary>

/// Словарь связывающий тип материала и его расширения

/// </summary>

public static readonly Dictionary<MaterialType, string[]> TypesExtensions = new()

{

{ MaterialType.Video, VideoFileExtensions },

{ MaterialType.Audio, AudioFileExtensions },

{ MaterialType.Picture, PictureFileExtensions },

{ MaterialType.Text, TextFileExtensions }

};

public MaterialViewModel()

{

ChangePathCommand = new LambdaCommand(OnChangePathCommandExecute, CanChangePathCommandExecute);

}

public override string TypeName => "Материал";

protected override void Reload(NewsFactoryObjectViewModel other)

{

base.Reload(other);

var viewModel = (MaterialViewModel)other;

Path = viewModel.Path;

Duration = viewModel.Duration;

MyType = viewModel.MyType;

}

/// <summary>

/// Обновления типа материала исходя из его расширения

/// </summary>

private void UpdateType()

{

foreach (var typeExtension in TypesExtensions.Where(typeExtension =>

typeExtension.Value.Contains(System.IO.Path.GetExtension(\_path).ToLower())))

{

MyType = typeExtension.Key;

return;

}

MyType = MaterialType.Other;

}

#region Path

private string \_path = string.Empty;

/// <summary>

/// Путь к материалу

/// </summary>

public string Path

{

get => \_path;

set

{

if (File.Exists(value)) CreationTime = new FileInfo(value).CreationTime;

if (!string.IsNullOrWhiteSpace(value) &&

(string.IsNullOrWhiteSpace(Title) || Title == System.IO.Path.GetFileName(\_path)) &&

System.IO.Path.GetFileName(value) is { } dn)

Title = dn;

Set(ref \_path, value);

UpdateType();

}

}

#endregion

#region Duration

/// <summary>

/// Длительность материала

/// </summary>

public TimeSpan? Duration

{

get => \_duration;

set => Set(ref \_duration, value);

}

private TimeSpan? \_duration;

#endregion

#region MyType

/// <summary>

/// Тип материала

/// </summary>

public MaterialType MyType

{

get => \_myType;

set => Set(ref \_myType, value);

}

private MaterialType \_myType;

#endregion

#region ChangePathCommand

/// <summary>

/// Команда изменения пути материала

/// </summary>

public ICommand ChangePathCommand { get; }

private bool CanChangePathCommandExecute(object p)

{

return true;

}

private void OnChangePathCommandExecute(object p)

{

if (FileDialogs.OpenFiles(Path, multiselect: false) is { } pa) Path = pa.First();

}

#endregion

}

/// <summary>

/// Перечисление возможных типов материала

/// </summary>

public enum MaterialType

{

[Description("Видео")] Video,

[Description("Аудио")] Audio,

[Description("Изображение")] Picture,

[Description("Текст")] Text,

[Description("Другое")] Other

}

## Приложение 4 Программный код EditPersonDialog.xaml (Диалог редактирования работника)

<editDialogs:EditDialogWindow x:Class="NfModels.Views.Dialogs.EditDialogs.EditPersonDialog"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:xctk="http://schemas.xceed.com/wpf/xaml/toolkit"

xmlns:newsFactoryObjects="clr-namespace:NfModels.ViewModels.NewsFactoryObjects"

xmlns:nfModels="clr-namespace:NfModels"

xmlns:dialogs="clr-namespace:NfModels.Views.Dialogs"

xmlns:editDialogs="clr-namespace:NfModels.Views.Dialogs.EditDialogs"

xmlns:converters="clr-namespace:NfModels.Infrastructure.Converters"

mc:Ignorable="d"

Width="700" Height="550" MinWidth="700" MinHeight="550">

<Window.Title>

<MultiBinding Converter="{converters:IsSavedTitleConverter}">

<Binding Path="FullName" />

<Binding Path="IsNotSaved" />

</MultiBinding>

</Window.Title>

<d:Window.DataContext>

<newsFactoryObjects:PersonViewModel />

</d:Window.DataContext>

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="30" MinHeight="30" MaxHeight="30" />

<RowDefinition Height="30" MinHeight="30" MaxHeight="30" />

<RowDefinition Height="30" MinHeight="30" MaxHeight="30" />

<RowDefinition Height="30" MinHeight="30" MaxHeight="30" />

<RowDefinition Height="30" MinHeight="30" MaxHeight="30" />

<RowDefinition Height="30" MinHeight="30" MaxHeight="30" />

<RowDefinition Height="\*" MinHeight="100" />

<RowDefinition Height="40" MinHeight="40" MaxHeight="40" />

</Grid.RowDefinitions>

<Grid.Resources>

<DataTemplate DataType="{x:Type nfModels:ReadTimeModel}" x:Key="ReadTimeModelPresentation">

<Border BorderBrush="DimGray" BorderThickness="1" Padding="5" CornerRadius="2" Height="230">

<Grid VerticalAlignment="Center">

<Grid.ColumnDefinitions>

<ColumnDefinition MinWidth="50" />

<ColumnDefinition Width="120" />

</Grid.ColumnDefinitions>

<Grid.RowDefinitions>

<RowDefinition Height="15" />

<RowDefinition Height="25" />

<RowDefinition Height="25" />

<RowDefinition Height="25" />

<RowDefinition Height="25" />

<RowDefinition Height="25" />

<RowDefinition Height="25" />

<RowDefinition Height="30" />

</Grid.RowDefinitions>

<TextBlock Grid.Row="0" Grid.Column="0" Grid.ColumnSpan="2" Text="{Binding Mode}"/>

<xctk:DoubleUpDown Grid.Row="1" Grid.Column="0" Value="{Binding SecondPerSymbol}" Minimum="-1"

Maximum="2" Margin="2" Increment="0.1" />

<xctk:DoubleUpDown Grid.Row="2" Grid.Column="0" Value="{Binding SecondPerWord}" Minimum="-1"

Maximum="2" Margin="2" Increment="0.1" />

<xctk:DoubleUpDown Grid.Row="3" Grid.Column="0" Value="{Binding SecondPerComa}" Minimum="-1"

Maximum="2" Margin="2" Increment="0.1" />

<xctk:DoubleUpDown Grid.Row="4" Grid.Column="0" Value="{Binding SecondPerDot}" Minimum="-1"

Maximum="2" Margin="2" Increment="0.1" />

<xctk:DoubleUpDown Grid.Row="5" Grid.Column="0" Value="{Binding SecondPerDigit}" Minimum="-1"

Maximum="2" Margin="2" Increment="0.1" />

<xctk:DoubleUpDown Grid.Row="6" Grid.Column="0" Value="{Binding SecondPerHyphen}" Minimum="-1"

Maximum="2" Margin="2" Increment="0.1" />

<TextBlock Grid.Row="1" Grid.Column="1" Text="Время на 1 символ" Margin="10,0,0,0" />

<TextBlock Grid.Row="2" Grid.Column="1" Text="Время на 1 слово" Margin="10,0,0,0" />

<TextBlock Grid.Row="3" Grid.Column="1" Text="Время на запятую" Margin="10,0,0,0" />

<TextBlock Grid.Row="4" Grid.Column="1" Text="Время на точку" Margin="10,0,0,0" />

<TextBlock Grid.Row="5" Grid.Column="1" Text="Время на цифру" Margin="10,0,0,0" />

<TextBlock Grid.Row="6" Grid.Column="1" Text="Время на тире" Margin="10,0,0,0" />

<StackPanel Grid.Column="0" Grid.Row="7" Grid.ColumnSpan="2" Orientation="Horizontal" HorizontalAlignment="Center">

<Button Content="Экспорт" Margin="10,5,10,0" Padding="10,0" Command="{Binding ExportCommand}"/>

<Button Content="Импорт" Margin="10,5,10,0" Padding="10,0" Command="{Binding ImportCommand}"/>

</StackPanel>

</Grid>

</Border>

</DataTemplate>

</Grid.Resources>

<TextBlock Grid.Row="0" Margin="5,0" Text="Имя" VerticalAlignment="Bottom" />

<TextBox Grid.Row="1" Margin="5" Text="{Binding FirstName}" Width="400" HorizontalAlignment="Left" />

<TextBlock Grid.Row="2" Margin="5,0" Text="Фамилия" VerticalAlignment="Bottom" />

<TextBox Grid.Row="3" Margin="5" Text="{Binding LastName}" Width="400" HorizontalAlignment="Left" />

<TextBlock Grid.Row="4" Margin="5,0" Text="Должность" VerticalAlignment="Bottom" />

<TextBox Grid.Row="5" Margin="5" Text="{Binding Post}" Width="400" HorizontalAlignment="Left" />

<Grid Grid.Row="6">

<Grid.ColumnDefinitions>

<ColumnDefinition Width="\*" />

<ColumnDefinition Width="\*" />

<ColumnDefinition Width="\*" />

</Grid.ColumnDefinitions>

<ContentPresenter Grid.Column="0" Content="{Binding VoiceReadTimeModel}"

ContentTemplate="{StaticResource ReadTimeModelPresentation}" Margin="10" />

<ContentPresenter Grid.Column="1" Content="{Binding DictorReadTimeModel}"

ContentTemplate="{StaticResource ReadTimeModelPresentation}" Margin="10" />

<ContentPresenter Grid.Column="2" Content="{Binding DigestReadTimeModel}"

ContentTemplate="{StaticResource ReadTimeModelPresentation}" Margin="10" />

</Grid>

<StackPanel Grid.Row="8" Orientation="Horizontal"

HorizontalAlignment="Center">

<Button Content="Отмена" Margin="10" Width="100" Command="{Binding CancelCommand}"

CommandParameter="{Binding RelativeSource={RelativeSource Mode=FindAncestor,AncestorType=Window}}" />

<Button Content="Сохранить" Margin="10" Width="100" Command="{Binding Save}"

CommandParameter="{Binding RelativeSource={RelativeSource Mode=FindAncestor,AncestorType=Window}}" />

</StackPanel>

</Grid>

</editDialogs:EditDialogWindow>

## Приложение 5 Программный код MainWindow.xaml (Главное окно)

<Window x:Class="ProjectManager.View.Windows.MainWindow"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008" mc:Ignorable="d"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:windows="clr-namespace:ProjectManager.ViewModels.Windows"

xmlns:localCommands="clr-namespace:ProjectManager.Infrastructure.Commands"

Title="Менеджер проектов" Height="550" Width="800">

<Window.DataContext>

<windows:MainWindowViewModel />

</Window.DataContext>

<DockPanel>

<Menu DockPanel.Dock="Top">

<MenuItem Header="Файл">

<!-- <MenuItem Height="Добавить..."> -->

<!-- <MenuItem Header="Добавить существующий объект" /> -->

<MenuItem Header="Добавить новый объект">

<MenuItem Header="Проект">

<MenuItem.Command>

<localCommands:CreateNewProjectCommand />

</MenuItem.Command>

</MenuItem>

<MenuItem Header="Материал">

<MenuItem.Command>

<localCommands:CreateNewMaterialCommand />

</MenuItem.Command>

</MenuItem>

<MenuItem Header="Съёмка">

<MenuItem.Command>

<localCommands:CreateNewFilmingCommand />

</MenuItem.Command>

</MenuItem>

<MenuItem Header="Тег">

<MenuItem.Command>

<localCommands:CreateNewTagCommand />

</MenuItem.Command>

</MenuItem>

<MenuItem Header="Работник">

<MenuItem.Command>

<localCommands:CreateNewPersonCommand />

</MenuItem.Command>

</MenuItem>

</MenuItem>

<!-- </MenuItem> -->

</MenuItem>

<MenuItem Header="Помощь">

<MenuItem Header="Синтаксис поисковой строки">

<MenuItem.Command>

<localCommands:OpenSyntaxHelpCommand />

</MenuItem.Command>

</MenuItem>

<Separator />

<MenuItem Header="О программе">

<MenuItem.Command>

<localCommands:OpenAboutCommand />

</MenuItem.Command>

</MenuItem>

</MenuItem>

</Menu>

<Grid>

<Grid.Resources>

<Style x:Key="DockPanelTrigger" TargetType="{x:Type DockPanel}">

<Style.Triggers>

<Trigger

Property="IsMouseOver"

Value="True">

<Trigger.EnterActions>

<BeginStoryboard>

<Storyboard>

<ThicknessAnimation Storyboard.TargetProperty="Margin"

To="0,0,0,0" Duration="0:0:0.2" />

</Storyboard>

</BeginStoryboard>

</Trigger.EnterActions>

<Trigger.ExitActions>

<BeginStoryboard>

<Storyboard>

<ThicknessAnimation Storyboard.TargetProperty="Margin"

To="-200,0,0,0" Duration="0:0:0.2" />

</Storyboard>

</BeginStoryboard>

</Trigger.ExitActions>

</Trigger>

</Style.Triggers>

</Style>

</Grid.Resources>

<Grid Background="Blue" Margin="50,0,0,0">

<Frame Content="{Binding CurrentPage}" NavigationUIVisibility="Hidden" />

</Grid>

<Canvas Background="#55000000" Opacity="0" Name="DropDown" IsHitTestVisible="False" />

<ContentControl MouseEnter="UIElement\_OnMouseEnter" MouseLeave="UIElement\_OnMouseLeave">

<DockPanel Background="{StaticResource DarkBG}" Width="250" HorizontalAlignment="Left"

Margin="-200,0,0,0"

Style="{StaticResource DockPanelTrigger}" x:Name="SidePanel">

<StackPanel Margin="0,10">

<StackPanel.Resources>

<Style TargetType="Button">

<Setter Property="Foreground" Value="White" />

<Setter Property="Background" Value="Transparent" />

<Setter Property="BorderBrush" Value="Transparent" />

<Setter Property="Margin" Value="10,5,20,5" />

<Setter Property="VerticalAlignment" Value="Center" />

<Setter Property="HorizontalAlignment" Value="Left" />

<Setter Property="FontFamily" Value="Gotham pro" />

<Setter Property="FontWeight" Value="Black" />

<Setter Property="FontSize" Value="18" />

</Style>

<Style TargetType="Image">

<Setter Property="DockPanel.Dock" Value="Right" />

<Setter Property="Height" Value="30" />

<Setter Property="Margin" Value="10" />

</Style>

</StackPanel.Resources>

<Button Content="News Factory" FontSize="25" HorizontalAlignment="Stretch" Margin="0,0,25,0" Command="{Binding GoToMainPageCommand}" />

<DockPanel>

<Image DockPanel.Dock="Left" Source="{StaticResource LayersDrawingImage}" />

<Button Content="Проекты" Command="{Binding GoToProjectPageCommand}" />

</DockPanel>

<DockPanel>

<Image DockPanel.Dock="Left" Source="{StaticResource FilmDrawingImage}" />

<Button Content="Материалы" Command="{Binding GoToMaterialsPageCommand}" />

</DockPanel>

<DockPanel>

<Image DockPanel.Dock="Left" Source="{StaticResource VideocamDrawingImage}" />

<Button Content="Съёмки" Command="{Binding GoToFilmingPageCommand}" />

</DockPanel>

</StackPanel>

</DockPanel>

</ContentControl>

</Grid>

</DockPanel>

</Window>

## Приложение 6 Программный код EditProjectDialog.xaml (Диалог редактирования проекта)

<editDialogs:EditDialogWindow x:Class="NfModels.Views.Dialogs.EditDialogs.EditProjectDialog"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:newsFactoryObjects="clr-namespace:NfModels.ViewModels.NewsFactoryObjects"

xmlns:newsFactoryConverters="clr-namespace:NfModels.Infrastructure.Converters"

xmlns:nfEl="clr-namespace:NfModels.Views.Elements"

xmlns:xctk="http://schemas.xceed.com/wpf/xaml/toolkit"

xmlns:dialogs="clr-namespace:NfModels.Views.Dialogs"

xmlns:editDialogs="clr-namespace:NfModels.Views.Dialogs.EditDialogs"

mc:Ignorable="d"

Height="450" Width="800" MinHeight="400" MinWidth="650">

<Window.Title>

<MultiBinding Converter="{newsFactoryConverters:IsSavedTitleConverter}">

<Binding Path="Title" />

<Binding Path="IsNotSaved" />

</MultiBinding>

</Window.Title>

<d:Window.DataContext>

<newsFactoryObjects:ProjectViewModel />

</d:Window.DataContext>

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="40" />

<RowDefinition Height="\*" />

<RowDefinition Height="\*" MinHeight="200" />

<RowDefinition Height="40" />

</Grid.RowDefinitions>

<Grid.ColumnDefinitions>

<ColumnDefinition Width="\*" />

<ColumnDefinition Width="\*" />

</Grid.ColumnDefinitions>

<TextBox Grid.Row="0" Grid.Column="0" Grid.ColumnSpan="2"

HorizontalAlignment="Center" Width="400"

VerticalContentAlignment="Center" HorizontalContentAlignment="Center"

Text="{Binding Title, UpdateSourceTrigger=PropertyChanged}" Margin="10" />

<TextBox Grid.Row="1" Grid.Column="0" Margin="10"

TextWrapping="Wrap"

AcceptsReturn="True"

VerticalScrollBarVisibility="Auto"

Text="{Binding Description, UpdateSourceTrigger=PropertyChanged}" />

<!-- <Image Grid.Row="1" Grid.Column="1" Margin="10" /> -->

<TabControl Grid.Row="2" Grid.Column="0" Grid.ColumnSpan="2" Margin="10">

<TabItem Header="Настройки">

<UniformGrid Rows="2" Columns="3">

<StackPanel VerticalAlignment="Center" Margin="10">

<TextBlock Text="Путь к проекту" Margin="5" />

<DockPanel>

<Button DockPanel.Dock="Right" Content="..." Padding="4,0" Margin="5,0,0,0"

Command="{Binding ChangePathCommand}" />

<TextBox Text="{Binding Path, UpdateSourceTrigger=PropertyChanged}" />

</DockPanel>

</StackPanel>

<StackPanel VerticalAlignment="Center" Margin="10">

<TextBlock Text="Срок сдачи" Margin="5" />

<xctk:DateTimePicker Value="{Binding DeadLine}" />

</StackPanel>

<StackPanel VerticalAlignment="Center" Margin="10">

<TextBlock Text="Тип проекта" Margin="5" />

<ComboBox

ItemsSource="{Binding Path=MyType, Converter={newsFactoryConverters:EnumToCollectionConverter}, Mode=OneTime}"

SelectedValuePath="Value"

DisplayMemberPath="Description"

SelectedValue="{Binding Path=MyType, Mode=TwoWay, UpdateSourceTrigger=PropertyChanged}" />

</StackPanel>

<StackPanel VerticalAlignment="Center" Margin="10">

<TextBlock Text="Текущий статус" Margin="5" />

<ComboBox

ItemsSource="{Binding Path=Status, Converter={newsFactoryConverters:EnumToCollectionConverter}, Mode=OneTime}"

SelectedValuePath="Value"

DisplayMemberPath="Description"

SelectedValue="{Binding Path=Status, Mode=TwoWay, UpdateSourceTrigger=PropertyChanged}" />

</StackPanel>

<GroupBox VerticalAlignment="Center" Margin="10" Header="Теги" Padding="-4,3,-4,-4" Height="55">

<nfEl:TagListEdit DataContext="{Binding Tags}" Margin="0" />

</GroupBox>

</UniformGrid>

</TabItem>

<TabItem Header="Материалы">

<nfEl:MaterialListEdit DataContext="{Binding Materials}" />

</TabItem>

<TabItem Header="Работники">

<nfEl:CrewListEdit DataContext="{Binding MyCrew}" />

</TabItem>

</TabControl>

<StackPanel Grid.Row="3" Grid.Column="0" Grid.ColumnSpan="2" Orientation="Horizontal"

HorizontalAlignment="Center">

<Button Content="Отмена" Margin="10" Width="100" Command="{Binding CancelCommand}"

CommandParameter="{Binding RelativeSource={RelativeSource Mode=FindAncestor,AncestorType=Window}}" />

<Button Content="Сохранить" Margin="10" Width="100" Command="{Binding Save}"

CommandParameter="{Binding RelativeSource={RelativeSource Mode=FindAncestor,AncestorType=Window}}" />

</StackPanel>

</Grid>

</editDialogs:EditDialogWindow>

## Приложение 7 Программный код EditFilmingDialog.xaml (Диалог редактирования съёмки)

<editDialogs:EditDialogWindow x:Class="NfModels.Views.Dialogs.EditDialogs.EditFilmingDialog"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:newsFactoryObjects="clr-namespace:NfModels.ViewModels.NewsFactoryObjects"

xmlns:newsFactoryConverters="clr-namespace:NfModels.Infrastructure.Converters"

xmlns:xctk="http://schemas.xceed.com/wpf/xaml/toolkit"

xmlns:elements="clr-namespace:NfModels.Views.Elements"

xmlns:dialogs="clr-namespace:NfModels.Views.Dialogs"

xmlns:editDialogs="clr-namespace:NfModels.Views.Dialogs.EditDialogs"

mc:Ignorable="d"

Height="500" Width="400" MinHeight="450" MinWidth="650">

<d:Window.DataContext>

<newsFactoryObjects:FilmingViewModel />

</d:Window.DataContext>

<Window.Title>

<MultiBinding Converter="{newsFactoryConverters:IsSavedTitleConverter}">

<Binding Path="Title" />

<Binding Path="IsNotSaved" />

</MultiBinding>

</Window.Title>

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="40" />

<RowDefinition Height="\*" MinHeight="200" />

<RowDefinition Height="40" />

</Grid.RowDefinitions>

<Grid.ColumnDefinitions>

<ColumnDefinition Width="\*" />

<ColumnDefinition Width="\*" />

</Grid.ColumnDefinitions>

<TextBox Grid.Row="0" Grid.Column="0" Grid.ColumnSpan="2"

HorizontalAlignment="Center" Width="400"

VerticalContentAlignment="Center" HorizontalContentAlignment="Center"

Text="{Binding Title, UpdateSourceTrigger=PropertyChanged}" Margin="10" />

<TextBox Grid.Row="1" Grid.Column="0" Margin="10"

TextWrapping="Wrap"

AcceptsReturn="True"

VerticalScrollBarVisibility="Auto"

Text="{Binding Description, UpdateSourceTrigger=PropertyChanged}" />

<!-- <Image Grid.Row="1" Grid.Column="1" Margin="10" /> -->

<TabControl Grid.Row="1" Grid.Column="1">

<TabItem Header="Настройки">

<UniformGrid Grid.Row="1" Grid.Column="1" Columns="1" Margin="10">

<StackPanel VerticalAlignment="Center" Margin="10">

<TextBlock Text="Время" Margin="5" />

<xctk:DateTimePicker Value="{Binding MyTime}" />

</StackPanel>

<StackPanel VerticalAlignment="Top" Margin="10">

<TextBlock Text="Адрес" Margin="5" />

<TextBox Text="{Binding Address, UpdateSourceTrigger=PropertyChanged}" />

</StackPanel>

<StackPanel VerticalAlignment="Top" Margin="10">

<TextBlock Text="Статус" Margin="5" />

<ComboBox

ItemsSource="{Binding Path=Status, Converter={newsFactoryConverters:EnumToCollectionConverter}, Mode=OneTime}"

SelectedValuePath="Value"

DisplayMemberPath="Description"

SelectedValue="{Binding Path=Status, Mode=TwoWay, UpdateSourceTrigger=PropertyChanged}" />

</StackPanel>

<StackPanel VerticalAlignment="Center" Margin="0">

<TextBlock Text="Теги" Margin="15,5" />

<elements:TagListEdit DataContext="{Binding Tags}" />

</StackPanel>

</UniformGrid>

</TabItem>

<TabItem Header="Работники">

<elements:CrewListEdit DataContext="{Binding MyCrew}"/>

</TabItem>

</TabControl>

<StackPanel Grid.Row="2" Grid.Column="0" Grid.ColumnSpan="2" Orientation="Horizontal"

HorizontalAlignment="Center">

<Button Content="Отмена" Margin="10" Width="100" Command="{Binding CancelCommand}"

CommandParameter="{Binding RelativeSource={RelativeSource Mode=FindAncestor,AncestorType=Window}}" />

<Button Content="Сохранить" Margin="10" Width="100" Command="{Binding Save}"

CommandParameter="{Binding RelativeSource={RelativeSource Mode=FindAncestor,AncestorType=Window}}" />

</StackPanel>

</Grid>

</editDialogs:EditDialogWindow>

## Приложение 8 Программный код EditMaterialDialog.xaml (Диалог редактирования материала)

<editDialogs:EditDialogWindow x:Class="NfModels.Views.Dialogs.EditDialogs.EditMaterialDialog"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:newsFactoryObjects="clr-namespace:NfModels.ViewModels.NewsFactoryObjects"

xmlns:newsFactoryConverters="clr-namespace:NfModels.Infrastructure.Converters"

xmlns:elements="clr-namespace:NfModels.Views.Elements"

xmlns:dialogs="clr-namespace:NfModels.Views.Dialogs"

xmlns:editDialogs="clr-namespace:NfModels.Views.Dialogs.EditDialogs"

mc:Ignorable="d"

Height="450" Width="800" MinHeight="400" MinWidth="650">

<d:Window.DataContext>

<newsFactoryObjects:MaterialViewModel />

</d:Window.DataContext>

<Window.Title>

<MultiBinding Converter="{newsFactoryConverters:IsSavedTitleConverter}">

<Binding Path="Title" />

<Binding Path="IsNotSaved" />

</MultiBinding>

</Window.Title>

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="40" />

<RowDefinition Height="\*" MinHeight="100" />

<RowDefinition Height="40" />

</Grid.RowDefinitions>

<Grid.ColumnDefinitions>

<ColumnDefinition Width="\*" />

<ColumnDefinition Width="\*" />

</Grid.ColumnDefinitions>

<TextBox Grid.Row="0" Grid.Column="0" Grid.ColumnSpan="2"

HorizontalAlignment="Center" Width="400"

VerticalContentAlignment="Center" HorizontalContentAlignment="Center"

Text="{Binding Title, UpdateSourceTrigger=PropertyChanged}" Margin="10" />

<TextBox Grid.Row="1" Grid.Column="0" Margin="10"

TextWrapping="Wrap"

AcceptsReturn="True"

VerticalScrollBarVisibility="Auto"

Text="{Binding Description, UpdateSourceTrigger=PropertyChanged}" />

<!-- <Image Grid.Row="1" Grid.Column="1" Margin="10" /> -->

<TabControl Grid.Row="1" Grid.Column="1" Margin="10">

<TabItem Header="Настройки">

<UniformGrid Columns="1" Margin="5">

<StackPanel VerticalAlignment="Top" Margin="10">

<TextBlock Text="Путь к материалу" Margin="5" />

<DockPanel>

<Button DockPanel.Dock="Right" Content="..." Padding="4,0" Margin="5,0,0,0"

Command="{Binding ChangePathCommand}" />

<TextBox Text="{Binding Path, UpdateSourceTrigger=PropertyChanged}" />

</DockPanel>

</StackPanel>

<StackPanel VerticalAlignment="Top" Margin="10">

<TextBlock Text="Тип материала" Margin="5" />

<ComboBox

ItemsSource="{Binding Path=MyType, Converter={newsFactoryConverters:EnumToCollectionConverter}, Mode=OneTime}"

SelectedValuePath="Value"

DisplayMemberPath="Description"

SelectedValue="{Binding Path=MyType, Mode=TwoWay, UpdateSourceTrigger=PropertyChanged}" />

</StackPanel>

<StackPanel VerticalAlignment="Center" Margin="0">

<TextBlock Text="Теги" Margin="15,5" />

<elements:TagListEdit DataContext="{Binding Tags}" />

</StackPanel>

</UniformGrid>

</TabItem>

<TabItem Header="Работники">

<elements:CrewListEdit DataContext="{Binding MyCrew}"/>

</TabItem>

</TabControl>

<StackPanel Grid.Row="2" Grid.Column="0" Grid.ColumnSpan="2" Orientation="Horizontal"

HorizontalAlignment="Center">

<Button Content="Отмена" Margin="10" Width="100" Command="{Binding CancelCommand}"

CommandParameter="{Binding RelativeSource={RelativeSource Mode=FindAncestor,AncestorType=Window}}" />

<Button Content="Сохранить" Margin="10" Width="100" Command="{Binding Save}"

CommandParameter="{Binding RelativeSource={RelativeSource Mode=FindAncestor,AncestorType=Window}}" />

</StackPanel>

</Grid>

</editDialogs:EditDialogWindow>

## Приложение 9 Программный код ExplorerPage.xaml (Страница браузера объектов фабрики)

<Page x:Class="NfModels.Views.Pages.ExplorerPage"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:newsFactoryObjectsBase="clr-namespace:NfModels.ViewModels.Base"

xmlns:windows="clr-namespace:NfModels.ViewModels.Windows"

mc:Ignorable="d"

Title="Браузер" Background="White" d:DesignHeight="450" d:DesignWidth="800">

<Page.DataContext>

<windows:ExplorerViewModel />

</Page.DataContext>

<Grid Margin="5">

<Grid.RowDefinitions>

<RowDefinition Height="25" />

<RowDefinition />

</Grid.RowDefinitions>

<TextBox Grid.Row="0" VerticalContentAlignment="Center"

Text="{Binding Filter, UpdateSourceTrigger=PropertyChanged}" />

<ScrollViewer Grid.Row="1" VerticalScrollBarVisibility="Auto" HorizontalScrollBarVisibility="Auto">

<ItemsControl ItemsSource="{Binding Objects}">

<!-- <ItemsControl.ItemsPanel> -->

<!-- <ItemsPanelTemplate> -->

<!-- -->

<!-- </ItemsPanelTemplate> -->

<!-- </ItemsControl.ItemsPanel> -->

<ItemsControl.Resources>

<DataTemplate DataType="{x:Type newsFactoryObjectsBase:TaggedObjectViewModel}">

<ContentControl PreviewMouseDoubleClick="OnItemDoubleClick">

<Border BorderThickness="1" BorderBrush="Black" CornerRadius="0" Padding="3,3"

Background="Aqua"

Margin="2">

<!-- <Border.ToolTip> -->

<!-- <ToolTip Content="{Binding Id}" Placement="Bottom" /> -->

<!-- </Border.ToolTip> -->

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="30" />

<RowDefinition Height="20" />

</Grid.RowDefinitions>

<Grid.ColumnDefinitions>

<ColumnDefinition Width="60" />

<ColumnDefinition Width="15" />

<ColumnDefinition Width="\*" />

<ColumnDefinition Width="30" />

</Grid.ColumnDefinitions>

<ContentPresenter Grid.Row="0" Grid.Column="0" Content="{Binding Status}"

VerticalAlignment="Center" HorizontalAlignment="Center"

Width="30" Height="30" />

<StackPanel Grid.Row="0" VerticalAlignment="Center" Grid.Column="2"

Orientation="Horizontal">

<TextBlock VerticalAlignment="Center" Text="{Binding Title}" FontWeight="Bold"

FontSize="18" Margin="0,0,30,0" MaxWidth="250" TextTrimming="CharacterEllipsis"/>

<ContentPresenter Content="{Binding Tags}" />

</StackPanel>

<TextBlock Grid.Row="1" VerticalAlignment="Center" Grid.Column="2"

Text="{Binding Description}" TextTrimming="WordEllipsis" MaxWidth="400"

HorizontalAlignment="Left" Padding="15,0,0,0" />

<TextBlock Grid.Row="1" VerticalAlignment="Center" Grid.Column="0"

Text="{Binding TypeName}" HorizontalAlignment="Center" Foreground="Gray" />

</Grid>

</Border>

</ContentControl>

</DataTemplate>

</ItemsControl.Resources>

</ItemsControl>

</ScrollViewer>

</Grid>

</Page>

## Приложение 10 Программный код TagList.cs (Класс, представляющий список тегов объекта фабрики)

using System.Collections.Generic;

using System.Collections.ObjectModel;

using System.Linq;

using System.Windows;

using System.Windows.Input;

using NfModels.Infrastructure.Commands;

using NfModels.ViewModels.NewsFactoryObjects;

using NfModels.Views.Dialogs;

using NfModels.Views.Dialogs.EditDialogs;

using NfModels.Views.Dialogs.PickDialogs;

namespace NfModels.ViewModels.Base;

/// <summary>

/// Класс, представляющий список тегов объекта фабрики

/// </summary>

public class TagList : ObservableCollection<TagViewModel>

{

public TagList()

{

AddTagCommand = new LambdaCommand(OnAddTagCommandExecute, CanAddTagCommandExecute);

RemoveTagCommand = new LambdaCommand(OnRemoveTagCommandExecute, CanRemoveTagCommandExecute);

EditTagCommand = new LambdaCommand(OnEditTagCommandExecute, CanEditTagCommandExecute);

}

/// <summary>

/// Получение списка всех тегов и тегов-родителей для поиска

/// </summary>

/// <returns>Список всех тегов и тегов-родителей для поиска</returns>

public IEnumerable<TagViewModel> GetAllTagsToSearch()

{

var prev = 0;

var tags = new List<TagViewModel>(Items);

while (prev != tags.Count)

{

prev = tags.Count;

foreach (var parentTag in tags.Select(tag => tag.ParentTag)

.Where(parentTag => parentTag is not null && !tags.Contains(parentTag)))

tags.Add(parentTag!);

}

return tags;

}

#region AddTagCommand

/// <summary>

/// Команда добавления тега

/// </summary>

public ICommand AddTagCommand { get; }

private bool CanAddTagCommandExecute(object p)

{

return true;

}

private void OnAddTagCommandExecute(object? p)

{

p ??= PickTagDialog.PickTag();

if (p is TagViewModel tag) AddTag(tag);

}

/// <summary>

/// Метод добавления тега

/// </summary>

/// <param name="tag">Тег</param>

private void AddTag(TagViewModel tag)

{

if (!Contains(tag))

{

if ((from oldTag in Items.ToList() where oldTag.Parents().Contains(tag) select MessageBox.Show(

$"Данный тег обобщает, уже присвоенный тег \"{oldTag.Title}\", добавление этого тега ничего не изменит.",

"Внимание!", MessageBoxButton.OK, MessageBoxImage.Error)).Any())

{

return;

}

foreach (var newTagParent in tag.Parents())

{

if (!Contains(newTagParent)) continue;

var messageBoxResult = MessageBox.Show(

$"Данный тег уточняет, уже присвоенный тег \"{newTagParent.Title}\". Старый тег будет удалён.",

"Предупреждение", MessageBoxButton.OKCancel, MessageBoxImage.Information);

if (messageBoxResult == MessageBoxResult.Cancel) return;

Remove(newTagParent);

}

Add(tag);

}

else

{

MessageBox.Show("Данный тег уже присвоен данном объекту", "Ошибка", MessageBoxButton.OK,

MessageBoxImage.Warning);

}

}

#endregion

#region RemoveTagCommand

/// <summary>

/// Команда исключения тега

/// </summary>

public ICommand RemoveTagCommand { get; }

private bool CanRemoveTagCommandExecute(object p)

{

return p is TagViewModel tag && Items.Contains(tag);

}

private void OnRemoveTagCommandExecute(object p)

{

Remove((TagViewModel)p);

}

#endregion

#region EditTagCommand

/// <summary>

/// Команда изменения тега

/// </summary>

public ICommand EditTagCommand { get; }

private bool CanEditTagCommandExecute(object p)

{

return true;

}

private void OnEditTagCommandExecute(object p)

{

new EditTagDialog((TagViewModel)p).Show();

}

#endregion

}

## Приложение 11 Программный код SyntaxHelpWindow.cs (Окно синтаксиса строки поиска)

using System.Windows;

namespace ProjectManager.View.Windows;

/// <summary>

/// Окно синтаксиса строки поиска

/// </summary>

public partial class SyntaxHelpWindow : Window

{

public SyntaxHelpWindow()

{

InitializeComponent();

Basics.Text =

"<Выражение> <Функция>:<Значение> \"<Фраза>\" <Функция>:\"<Значение с пробелами>\"\n" +

"\n" +

"<Выражение>";

Functions.Text =

"t:<Название тега> Иерархический поиск по тегу\n" +

"tag:<Название тега>\n" +

"тег:<Название тега>\n" +

"тэг:<Название тега>\n" +

"т:<Название тега>\n" +

"\n" +

"p:<Путь> Поиск по пути\n" +

"path:<Путь>\n" +

"п:<Путь>\n" +

"путь:<Путь>\n" +

"\n" +

"тип:<Тип> Поиск по типу\n" +

"is:<Тип> Допустимые значения:\n" +

"i:<Тип> Проект, съёмка, материал, Project, Filming, Material\n" +

"type:<Тип>\n" +

"\n" +

"с:<Статус> Поиск по текущему статусу\n" +

"статус:<Статус> Допустимые значения:\n" +

"s:<Статус> готово, done, 3\n" +

"status:<Статус> отменено, adborted, 0\n" +

" работа, work, 2\n" +

" запланировано, planed, 1\n" +

"\n" +

"created:<Дата> Поиск по дате создания (границы поиска расширяются на 1 день)\n" +

"создан:<Дата> Допускается использование знаков сравнения (>, <, =) вместо ':' \n" +

" Допускается использование константы 'сегодня' или 'now' с операторами '+' или '-' для поиска относительной даты \n" +

" Пример: 'создан=сегодня-1' - объекты, созданные вчера\n" +

"\n" +

"work:<Работник>-<Тип работы> Поиск по сотруднику и исполняемыми им обязательствами\n" +

"работал:<Работник>-<Тип работы> Поиск сотрудника и типа обязанностей выполняется с помощью алгоритмов нечёткого поиска\n" +

" Допустимые значения типа обязанностей:\n" +

" Монтажёр, Редактор, Оператор, Водитель, Корреспондент, Другое\n" +

" Указание работника и обязанности опционально\n" +

" Пример: \n" +

" 'работал:Волков' - объекты, над которыми работал сотрудник «Волков»\n" +

" 'работал:-водитель' - объекты, над которыми работал водитель\n" +

" 'работал:Волков-водитель' - объекты, над которыми работал сотрудник «Волков» в роли водителя\n" +

"";

}

}

## Приложение 12 Программный код EditTagDialog.xaml (Окно редактирования метки)

<editDialogs:EditDialogWindow x:Class="NfModels.Views.Dialogs.EditDialogs.EditTagDialog"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:xctk="http://schemas.xceed.com/wpf/xaml/toolkit"

xmlns:newsFactoryObjects="clr-namespace:NfModels.ViewModels.NewsFactoryObjects;assembly=NfModels"

xmlns:elements="clr-namespace:NfModels.Views.Elements"

xmlns:dialogs="clr-namespace:NfModels.Views.Dialogs"

xmlns:editDialogs="clr-namespace:NfModels.Views.Dialogs.EditDialogs"

mc:Ignorable="d" d:DataContext="{d:DesignInstance newsFactoryObjects:TagViewModel}"

Title="{Binding Title}" Height="300" Width="400" MinHeight="250" MinWidth="350">

<Grid Margin="5">

<Grid.RowDefinitions>

<RowDefinition Height="50" />

<RowDefinition Height="35" />

<RowDefinition Height="2\*" MinHeight="30" />

<RowDefinition Height="\*" MinHeight="30" />

<RowDefinition Height="\*" MinHeight="40" />

<RowDefinition Height="30" />

</Grid.RowDefinitions>

<Grid.ColumnDefinitions>

<ColumnDefinition Width="\*" MaxWidth="180" MinWidth="140" />

<ColumnDefinition Width="\*" />

</Grid.ColumnDefinitions>

<elements:Tag Grid.Column="0" Grid.Row="0" Grid.ColumnSpan="2" DataContext="{Binding}"

HorizontalAlignment="Center" VerticalAlignment="Center" />

<TextBlock Grid.Column="0" Grid.Row="1" VerticalAlignment="Top" Text="Название" Margin="0" />

<TextBox Grid.Column="1" Grid.Row="1" VerticalAlignment="Stretch"

Text="{Binding Title, UpdateSourceTrigger=PropertyChanged}" Margin="5"

VerticalContentAlignment="Center" />

<TextBlock Grid.Column="0" Grid.Row="2" VerticalAlignment="Top" Text="Описание" Margin="0" />

<TextBox Grid.Column="1" Grid.Row="2" VerticalAlignment="Stretch"

Text="{Binding Description, UpdateSourceTrigger=PropertyChanged}" Margin="5" />

<TextBlock Grid.Column="0" Grid.Row="3" VerticalAlignment="Top" Text="Родительский тег" Margin="0" />

<DockPanel Grid.Column="1" Grid.Row="3" LastChildFill="False" Margin="5,0">

<elements:Tag DataContext="{Binding ParentTag}" DockPanel.Dock="Left" Height="25" />

<Button Content="..." DockPanel.Dock="Right" Width="25" Height="20" Padding="0"

Command="{Binding ChangeParentCommand}" />

</DockPanel>

<xctk:ColorPicker Grid.Column="1" Grid.Row="4" VerticalAlignment="Stretch" Margin="5"

SelectedColor="{Binding MyColor}" UsingAlphaChannel="False"

AdvancedTabHeader="RGB" RecentColorsHeader="Недавние"

StandardColorsHeader="Стандартные цвета"

StandardTabHeader="Стандартные" AvailableColorsHeader="Библиотека" />

<TextBlock Grid.Column="0" Grid.Row="4" VerticalAlignment="Top" Text="Цвет" Margin="0" />

<StackPanel Grid.Row="5" Grid.Column="0" Grid.ColumnSpan="2" Orientation="Horizontal"

HorizontalAlignment="Center">

<Button Content="Отмена" Margin="10,4" Width="100" Command="{Binding CancelCommand}"

CommandParameter="{Binding RelativeSource={RelativeSource Mode=FindAncestor,AncestorType=Window}}" />

<Button Content="Сохранить" Margin="10,4" Width="100" Command="{Binding Save}"

CommandParameter="{Binding RelativeSource={RelativeSource Mode=FindAncestor,AncestorType=Window}}" />

</StackPanel>

</Grid>

</editDialogs:EditDialogWindow>

## Приложение 13 Программный код NewsFactoryObjectViewModel.cs (Базовый класс объекта фабрики новостей)

using System;

using System.ComponentModel;

using System.Windows.Input;

using MongoDB.Bson;

using MongoDB.Bson.Serialization.Attributes;

using NfModels.Infrastructure.Commands;

using NfModels.Services;

using NfModels.Views.Dialogs;

using NfModels.Views.Dialogs.EditDialogs;

namespace NfModels.ViewModels.Base;

/// <summary>

/// Базовый класс объекта фабрики

/// </summary>

public abstract class NewsFactoryObjectViewModel : ViewModel

{

protected NewsFactoryObjectViewModel()

{

Save = new LambdaCommand(OnSaveExecute, CanSaveExecute);

CancelCommand = new LambdaCommand(OnCancelCommandExecute, CanCancelCommandExecute);

}

/// <summary>

/// Идентификатор объекта фабрики

/// </summary>

[BsonElement("\_id")] public ObjectId Id { get; set; }

/// <summary>

/// Словесное название типа объекта фабрики

/// </summary>

public virtual string TypeName => GetType().ToString();

protected override void OnPropertyChanged(string PropertyName = null)

{

base.OnPropertyChanged(PropertyName);

if (PropertyName == nameof(IsNotSaved)) return;

IsNotSaved = true;

}

/// <summary>

/// Метод перезаписи всех полей данного объекта по примеру другого

/// </summary>

/// <param name="other">Другой объект фабрики</param>

/// <exception cref="ArgumentException">Тип данного объекта не совпадает с типом аргумента</exception>

protected virtual void Reload(NewsFactoryObjectViewModel other)

{

if (other.GetType() != GetType()) throw new ArgumentException();

IsNotSaved = false;

}

#region CreationTime

/// <summary>

/// Дата и время создания объекта

/// </summary>

public DateTime CreationTime

{

get => \_creationTime;

set => Set(ref \_creationTime, value);

}

private DateTime \_creationTime = DateTime.Now;

#endregion

#region Save

/// <summary>

/// Команда сохранения объекта

/// </summary>

public ICommand Save { get; }

private bool CanSaveExecute(object p)

{

return true;

}

private void OnSaveExecute(object p)

{

DBProvider.Save(this);

if (p is EditDialogWindow window) window.SilentClose();

}

#endregion

#region CancelCommand

/// <summary>

/// Команда отмены несохранённых изменений

/// </summary>

public ICommand CancelCommand { get; }

private bool CanCancelCommandExecute(object? p)

{

return true;

}

private void OnCancelCommandExecute(object? p)

{

var savedData = DBProvider.LoadSavedData(this);

if (savedData is not null)Reload(savedData);

if (p is EditDialogWindow window) window.SilentClose();

}

#endregion

#region IsSaved

/// <summary>

/// Сохранён ли данный объект

/// </summary>

[BsonIgnore]

public bool IsNotSaved

{

get => \_isNotSaved;

set => Set(ref \_isNotSaved, value);

}

private bool \_isNotSaved;

#endregion

protected bool Equals(NewsFactoryObjectViewModel other)

{

return Id.Equals(other.Id);

}

public override bool Equals(object? obj)

{

if (ReferenceEquals(null, obj)) return false;

if (ReferenceEquals(this, obj)) return true;

if (obj.GetType() != this.GetType()) return false;

return Equals((NewsFactoryObjectViewModel)obj);

}

public override int GetHashCode()

{

return Id.GetHashCode();

}

}

public enum ActionStatus

{

[Description("Запланировано")] Planed,

[Description("В работе")] InWork,

[Description("Готово")] Done,

[Description("Отменено")] Aborted

}

## Приложение 14 Программный код TagViewModel.cs (Модель представления метки)

using System.Collections.Generic;

using System.Windows.Input;

using System.Windows.Media;

using MongoDB.Bson;

using MongoDB.Bson.Serialization.Attributes;

using NfModels.Infrastructure.Commands;

using NfModels.Services;

using NfModels.ViewModels.Base;

using NfModels.Views.Dialogs;

using NfModels.Views.Dialogs.PickDialogs;

namespace NfModels.ViewModels.NewsFactoryObjects;

/// <summary>

/// Модель представления тега

/// </summary>

public class TagViewModel : TitledObjectViewModel

{

private Color \_myColor;

public TagViewModel()

{

ChangeParentCommand = new LambdaCommand(OnChangeParentCommandExecute, CanChangeParentCommandExecute);

}

/// <summary>

/// Основной цвет тега

/// </summary>

public Color MyColor

{

get => \_myColor;

set

{

Set(ref \_myColor, value);

OnPropertyChanged(nameof(TextColor));

OnPropertyChanged(nameof(BgColor));

OnPropertyChanged(nameof(BorderColor));

}

}

/// <summary>

/// Вторичный цвет тега

/// </summary>

public Color MySecondColor

{

get

{

var c = Color.Multiply(MyColor, 0.4f);

c.A = byte.MaxValue;

return c;

}

}

/// <summary>

/// Цвет текста тега

/// </summary>

public Color TextColor => MySecondColor.R + MySecondColor.G + MySecondColor.B > 350 ? Colors.Black : Colors.White;

/// <summary>

/// Цвет фона тега

/// </summary>

public Color BgColor => MySecondColor;

/// <summary>

/// Цвет границы тега

/// </summary>

public Color BorderColor => MyColor;

#region ParentTag

/// <summary>

/// Родительский тег

/// </summary>

[BsonIgnore]

public TagViewModel? ParentTag

{

get => TagManager.LoadTag(ParentTagId);

set

{

if (value is not null) ParentTagId = value.Id;

}

}

#endregion

public override string TypeName => "Тег";

protected override void Reload(NewsFactoryObjectViewModel other)

{

base.Reload(other);

var viewModel = (TagViewModel)other;

MyColor = viewModel.MyColor;

ParentTagId = viewModel.ParentTagId;

}

/// <summary>

/// Получение коллекции тегов родителей

/// </summary>

/// <param name="addSelf">Добавление в коллекцию себя</param>

/// <returns>Коллекция тегов родителей</returns>

public IEnumerable<TagViewModel> Parents(bool addSelf = false)

{

var tag = addSelf ? this : ParentTag;

while (tag is not null)

{

yield return tag;

tag = tag.ParentTag;

if (tag == this) yield break;

}

}

#region ParentTagId

/// <summary>

/// Идентификатор родительского тега

/// </summary>

public ObjectId? ParentTagId

{

get => \_parentTagId;

set

{

Set(ref \_parentTagId, value);

OnPropertyChanged(nameof(ParentTag));

}

}

private ObjectId? \_parentTagId;

#endregion

#region ChangeParentCommand

/// <summary>

/// Команда изменения родительского тега

/// </summary>

public ICommand ChangeParentCommand { get; }

private bool CanChangeParentCommandExecute(object p)

{

return true;

}

private void OnChangeParentCommandExecute(object p)

{

if (PickTagDialog.PickTag() is not { } tag) return;

ParentTag = tag;

}

#endregion

}

## Приложение 15 Программный код ProjectViewModel.cs (Модель представления проекта)

using System;

using System.ComponentModel;

using System.IO;

using System.Windows.Input;

using NfModels.Infrastructure.Commands;

using NfModels.Services;

using NfModels.ViewModels.Base;

namespace NfModels.ViewModels.NewsFactoryObjects;

/// <summary>

/// Модель представления проекта

/// </summary>

public class ProjectViewModel : CrewedObjectViewModel

{

public ProjectViewModel()

{

ChangePathCommand = new LambdaCommand(OnChangePathCommandExecute, CanChangePathCommandExecute);

}

public override string TypeName => "Проект";

protected override void Reload(NewsFactoryObjectViewModel other)

{

base.Reload(other);

var viewModel = (ProjectViewModel)other;

Status = viewModel.\_status;

Path = viewModel.\_path;

DeadLine = viewModel.\_deadLine;

MyType = viewModel.\_MyType;

Materials = viewModel.\_materials;

}

#region Path

private string \_path = string.Empty;

/// <summary>Путь к проекту</summary>

public string Path

{

get => \_path;

set

{

if (Directory.Exists(value)) CreationTime = new DirectoryInfo(value).CreationTime;

if (!string.IsNullOrWhiteSpace(value) &&

(string.IsNullOrWhiteSpace(Title) || Title == System.IO.Path.GetFileName(\_path)) &&

System.IO.Path.GetFileName(value) is { } dn)

Title = dn;

Set(ref \_path, value);

}

}

#endregion

#region MyType

/// <summary>

/// Тип проекта

/// </summary>

public ProjectType MyType

{

get => \_MyType;

set => Set(ref \_MyType, value);

}

private ProjectType \_MyType;

#endregion

#region DeadLine

/// <summary>

/// Срок сдачи проекта

/// </summary>

public DateTime DeadLine

{

get => \_deadLine;

set => Set(ref \_deadLine, value);

}

private DateTime \_deadLine = DateTime.Today + TimeSpan.FromHours(19);

#endregion

#region Status

/// <summary>

/// Статус проекта

/// </summary>

public ActionStatus Status

{

get => \_status;

set => Set(ref \_status, value);

}

private ActionStatus \_status = ActionStatus.Planed;

#endregion

#region ChangePathCommand

/// <summary>

/// Команда изменения пути проекта

/// </summary>

public ICommand ChangePathCommand { get; }

private bool CanChangePathCommandExecute(object p)

{

return true;

}

private void OnChangePathCommandExecute(object p)

{

if (FileDialogs.GetDirectory(Path, "Путь к проекту") is { } pa) Path = pa;

}

#endregion

#region Materials

/// <summary>

/// Список материалов проекта

/// </summary>

public MaterialList Materials

{

get => \_materials;

set

{

Set(ref \_materials, value);

\_materials.CollectionChanged += (\_, \_) => { IsNotSaved = true; };

}

}

private MaterialList \_materials = new();

#endregion

}

/// <summary>

/// Тип проекта

/// </summary>

public enum ProjectType

{

[Description("Эфир")] Efir,

[Description("Сюжет")] Report,

[Description("Реклама")] Advertising,

[Description("Фильм")] Film,

[Description("Другое")] Other

}

## Приложение 16 Программный код ExplorerViewModel.cs (Модель представления браузера)

using System;

using System.Collections.Generic;

using NfModels.Services;

using NfModels.ViewModels.Base;

namespace NfModels.ViewModels.Windows;

/// <summary>

/// Модель представления браузера

/// </summary>

public class ExplorerViewModel : ViewModel

{

private string \_filter;

/// <summary>

/// Текущий запрос

/// </summary>

public string Filter

{

get => \_filter;

set

{

Set(ref \_filter, value);

OnPropertyChanged(nameof(Objects));

}

}

#region HiddenFilter

/// <summary>

/// Скрытый запрос

/// </summary>

public string HiddenFilter

{

get => \_hiddenFilter;

set

{

Set(ref \_hiddenFilter, value);

OnPropertyChanged(nameof(Objects));

}

}

private string \_hiddenFilter = "";

#endregion

/// <summary>

/// Полный фильтр

/// </summary>

public string FullFilter => \_hiddenFilter + " " + \_filter;

/// <summary>

/// Объекты для отображения

/// </summary>

public IEnumerable<NewsFactoryObjectViewModel> Objects => DBProvider.LoadObjects(FilterParser.Parse(FullFilter));

/// <summary>

/// Загрузка предустановленной запроса

/// </summary>

/// <param name="preset">Предустановленный запрос</param>

/// <param name="hidden">Загрузить в скрытом режиме</param>

/// <exception cref="ArgumentOutOfRangeException">Неизвестная предустановка</exception>

public void LoadPreset(ExplorerPreset preset, bool hidden = false)

{

if (!hidden)

Filter = preset switch

{

ExplorerPreset.Projects => "Тип:Проект",

ExplorerPreset.Materials => "Тип:Материал",

ExplorerPreset.Filmings => "Тип:Съёмка",

\_ => throw new ArgumentOutOfRangeException(nameof(preset), preset, null)

};

else

HiddenFilter = preset switch

{

ExplorerPreset.Projects => "Тип:Проект",

ExplorerPreset.Materials => "Тип:Материал",

ExplorerPreset.Filmings => "Тип:Съёмка",

\_ => throw new ArgumentOutOfRangeException(nameof(preset), preset, null)

};

}

}

/// <summary>

/// Предустановленные запросы

/// </summary>

public enum ExplorerPreset

{

Projects,

Materials,

Filmings

}

## Приложение 17 Программный код CrewListSerializer.cs (Сериалайзер списка работников объекта фабрики)

using System;

using MongoDB.Bson.IO;

using MongoDB.Bson.Serialization;

using NfModels.ViewModels.Base;

namespace NfModels.Services.Serializers;

/// <summary>

/// Сериалайзер списка работников объекта фабрики

/// </summary>

public class CrewListSerializer : IBsonSerializer<CrewList>, IBsonArraySerializer

{

public object Deserialize(BsonDeserializationContext context, BsonDeserializationArgs args)

{

var result = new CrewList();

var reader = context.Reader;

reader.ReadStartArray();

while (true)

{

reader.ReadBsonType();

if (reader.State == BsonReaderState.EndOfArray) break;

reader.ReadStartDocument();

var crew = new Crew();

var person = PersonManager.LoadPerson(reader.ReadObjectId("Person"));

if (person is null)

{

reader.ReadInt32("WorkType");

reader.ReadEndDocument();

continue;

}

crew.Person = person;

crew.WorkType = (Work)reader.ReadInt32("WorkType");

reader.ReadEndDocument();

result.Add(crew);

}

reader.ReadEndArray();

return result;

}

public void Serialize(BsonSerializationContext context, BsonSerializationArgs args, CrewList value)

{

Serialize(context, args, (object)value);

}

CrewList IBsonSerializer<CrewList>.Deserialize(BsonDeserializationContext context, BsonDeserializationArgs args)

{

return (CrewList)Deserialize(context, args);

}

public void Serialize(BsonSerializationContext context, BsonSerializationArgs args, object value)

{

var l = (CrewList)value;

var writer = context.Writer;

writer.WriteStartArray();

foreach (var crew in l)

{

writer.WriteStartDocument();

writer.WriteObjectId("Person", crew.Person.Id);

writer.WriteInt32("WorkType", (int)crew.WorkType);

writer.WriteEndDocument();

}

writer.WriteEndArray();

}

public Type ValueType { get; } = typeof(CrewList);

public bool TryGetItemSerializationInfo(out BsonSerializationInfo serializationInfo)

{

serializationInfo = new BsonSerializationInfo("MyCrew", new CrewListSerializer(), ValueType);

return true;

}

}