|  |
| --- |
| **МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ** |
| **ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ «МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ТЕХНОЛОГИЙ И УПРАВЛЕНИЯ ИМЕНИ К.Г. РАЗУМОВСКОГО (ПЕРВЫЙ КАЗАЧИЙ УНИВЕРСИТЕТ)»**  **(ФГБОУ ВО «МГУТУ ИМ. К.Г. РАЗУМОВСКОГО (ПКУ)»)** |
|  |
| **УНИВЕРСИТЕТСКИЙ КОЛЛЕДЖ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ** |

|  |
| --- |
| **КУРСОВОЙ ПРОЕКТ** |
|  |
| по междисциплинарному курсу МДК.01.01. Разработка программных модулей систем |
|  |
| на тему: Разработка системы автоматизации учета проведенных занятий преподавателями колледжа |
|  |
| студента группы 090207-9о-20/2  специальности 09.02.07 Информационные системы и программирование |
| Асылбек уулу Бакыта |

|  |  |  |
| --- | --- | --- |
| Студент |  | Б. Асылбек уулу |
| Руководитель курсового проекта |  | А. И. Глускер |
| Председатель ПЦК специальности 09.02.07 Информационные системы и программирование |  | А. И. Глускер |

|  |  |  |
| --- | --- | --- |
| Дата защиты «\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_ 2023 г. | | |
| Оценка:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | | |
| Заведующий отделением № 3 |  | И. Г. Дзюба |

Москва

2023

**Содержание**

[ВВЕДЕНИЕ 4](#_Toc135658202)

[ОСНОВНАЯ ЧАСТЬ 5](#_Toc135658203)

[1 Описание предметной области 5](#_Toc135658204)

[2 Спецификации разрабатываемой системы 6](#_Toc135658205)

[3 Описание проектирования базы данных 9](#_Toc135658206)

[3.1 Описание логической модели 9](#_Toc135658207)

[3.2 Описание физической модели 11](#_Toc135658208)

[4 Стадии разработки приложения 17](#_Toc135658209)

[4.1 Авторизация и аутентификация 17](#_Toc135658210)

[4.2 Подключение базы данных 19](#_Toc135658211)

[4.3 Работа с базой данных через приложение 20](#_Toc135658212)

[4.4 Подсчет проведенных и не проведенных занятий 25](#_Toc135658213)

[4.5 Создание отчетов 26](#_Toc135658214)

[5 Тестирование приложения 28](#_Toc135658215)

[ЗАКЛЮЧЕНИЕ 35](#_Toc135658216)

[СПИСОК ЛИТЕРАТУРЫ 36](#_Toc135658217)

[ПРИЛОЖЕНИЕ А 37](#_Toc135658218)

[ПРИЛОЖЕНИЕ Б 39](#_Toc135658219)

[ПРИЛОЖЕНИЕ В 40](#_Toc135658220)

[ПРИЛОЖЕНИЕ Г 45](#_Toc135658221)

[ПРИЛОЖЕНИЕ Д 48](#_Toc135658222)

[ПРИЛОЖЕНИЕ Е 50](#_Toc135658223)

[ПРИЛОЖЕНИЕ Ж 56](#_Toc135658224)

[ПРИЛОЖЕНИЕ И 60](#_Toc135658225)

[ПРИЛОЖЕНИЕ К 65](#_Toc135658226)

[ПРИЛОЖЕНИЕ Л 69](#_Toc135658227)

[ПРИЛОЖЕНИЕ М 73](#_Toc135658228)

[ПРИЛОЖЕНИЕ Н 77](#_Toc135658229)

# ВВЕДЕНИЕ

Темой данной курсовой работы является «Разработка системы автоматизации учета проведенных занятий преподавателями колледжа».

Актуальность темы заключается в следующем:

1. Упрощение процесса учета проведенных занятий. Ручной учет занимает много времени и может быть неточным. Автоматизированная система учета значительно упрощает процесс и позволяет сократить время, затрачиваемое на подсчеты.
2. Повышение уровня эффективности работы преподавателей. Система учета позволит преподавателям точно контролировать число проведенных ими занятий и определить объем работы, которую они смогут выполнить в определенный период времени.
3. Соответствие требованиям современного образования. Система автоматизации учета является необходимым инструментом современного образования, так как отвечает требованиям информационной технологии и снижает риски ошибок и неточностей.

Цель проекта: упростить и ускорить процесс учета проведенных занятий, а также поднять эффективность работы колледжа в целом.

Задачи:

* описание предметной области;
* спецификации разрабатываемой системы;
* описание проектирования базы данных;
* стадии разработки приложения:
  1. авторизация и аутентификация;
  2. подключение базы данных к приложению;
  3. работа с базой данных через приложение;
  4. подсчет проведенных и не проведенных занятий;
  5. создание отчетов;
  + тестирование приложения.

# ОСНОВНАЯ ЧАСТЬ

## Описание предметной области

Каждый колледж предоставляет конкретные специальности и учебные планы в зависимости от выбранного направления обучения.

Специальности представляют собой определенный набор профессиональных знаний и навыков, которые студент получает на образовательной программе. В колледже могут предлагаться специальности в различных областях, таких как медицина, бизнес, информационные технологии, дизайн и так далее.

Учебные планы – это определенная последовательность учебных предметов и практических занятий, необходимых для получения квалификации по выбранной специальности. Каждый учебный план может включать как обязательные, так и выборочные дисциплины.

Предметы могут быть теоретическими и практическими, направленные на формирование определенных компетенций у студента. Предметы часто группируются по категориям, таким как математика, английский язык, компьютерные науки и так далее.

Обычно учебный процесс в колледже рассчитан на 2-4 года и включает несколько семестров. Семестры представляют собой периоды времени, в течение которых проходят занятия. В колледже может быть от 4 до 8 семестров, в зависимости от образовательной программы.

Учебный план включает в себя предмет, кол-во часов обучения, тип занятий (лекции и практические работы) и номер семестра, в котором он проходит. Каждый учебный план связан с определенной специальностью, а каждый предмет связан с определенным учебным планом. Таким образом, связь между специальностями, учебными планами, семестрами и предметами является важным элементом колледжей и других образовательных учреждений.

## Спецификации разрабатываемой системы

В программе будет реализован функционал для ввода и исправления информации в следующих сущностях:

* сущность «Специальности» будет содержать в себе:

1. код специальности;
2. наименование специальности;
3. форма обучения;
4. квалификация;

* сущность «Учебные планы» будет содержать в себе:

1. наименование учебных планов.
2. код предмета;
3. код специальности;
4. код группы;
5. номер семестра;
6. общее количество часов;
7. практические часы;
8. теоретические часы;

* сущность «Семестры» будет содержать в себе:

1. номер семестра;
2. семестр;
3. начало семестра;
4. конец семестра;

* сущность «Предметы» будет содержать в себе:

1. код предмета;
2. наименование предмета;

* сущность «Группы» будет содержать в себе следующие атрибуты:

1. номер группы;
2. наименование группы;
3. курс;
4. количество студентов;

* сущность «Учеты» будет содержать в себе следующие атрибуты:

1. дата;
2. код предмета;
3. основной преподаватель;
4. заменяющий преподаватель;
5. количество проведенных занятий;
6. код группы;

* сущность «Преподаватели» будет содержать в себе:

1. логин;
2. фамилия;
3. имя;
4. отчество;
5. пароль;
6. пол;
7. дата рождения;
8. телефон;
9. почта.

«Учеты» и «Преподаватели» это единственные сущности, в которых будет реализован функционал удаления записей.

Система будет учитывать дни проведения занятий, а также возможность замены преподавателей. Программа также будет подсчитывать часы проведенных и не проведенных занятий по предмету какой-либо группы.

С помощью данной программы на основе информации о проведенных занятиях будет возможно создание следующих отчетов:

1. Отчет о проведенных часах по дням преподавателем будет содержать в себе такую информацию, как дата, проведенные часы за весь день и логин преподавателя. Данный отчет будет реализовываться таким образом, что можно будет сделать его только сегодня.
2. Отчет о проведенных часах по предметам будет включать в себя общее количество проведенных часов, код предмета и логин преподавателя.

Система будет обеспечивать аутентификацию и авторизацию пользователей с разными уровнями доступа.

Преподавателям будет предоставлен доступ, позволяющий создавать отчеты, смотреть количество проведённых и не проведённых занятий, а также добавлять, изменять и удалять данные в сущности «Учеты».

Администратору будут доступны все возможности и функции приложения.

## Описание проектирования базы данных

### Описание логической модели

Логическая модель состоит из 7 таблиц:

* таблица «Специальности» будет содержать в себе:

1. код специальности. Первичный ключ;
2. наименование специальности;
3. форма обучения;
4. квалификация;

* таблица «Учебные планы» будет содержать в себе:

1. id учебных планов. Первичный ключ;
2. наименование учебных планов;
3. код предмета. Внешний ключ;
4. код специальности. Внешний ключ;
5. код группы. Внешний ключ;
6. номер семестра. Внешний ключ;
7. общее количество часов;
8. практические часы;
9. теоретические часы;

* таблица «Семестры» будет содержать в себе:

1. номер семестра. Первичный ключ;
2. семестр;
3. начало семестра;
4. конец семестра;

* таблица «Предметы» будет содержать в себе:

1. код предмета. Первичный ключ;
2. наименование предмета;

* таблица «Группы» будет содержать в себе следующие атрибуты:

1. номер группы. Первичный ключ;
2. наименование группы;
3. курс;
4. количество студентов;

* таблица «Учеты» будет содержать в себе следующие атрибуты:

1. id учета. Первичный ключ;
2. дата;
3. код предмета. Внешний ключ;
4. логин. Внешний ключ;
5. заменяющий логин. Внешний ключ;
6. количество проведенных занятий;
7. код группы. Внешний ключ;

* таблица «Преподаватели» будет содержать в себе:

1. логин. Первичный ключ;
2. фамилия;
3. имя;
4. отчество;
5. пароль;
6. пол;
7. дата рождения;
8. телефон;
9. почта.

Таблица «Группы» имеет связь один ко многим таблицей «Учеты», так как у одной группы могут быть пары с различными предметами и преподавателями.

Таблица «Преподаватели» имеет связи один ко многим с таблицей «Учеты», так как один преподаватель может вести несколько предметов и занятия с различными группами.

Таблица «Предметы» имеет связь один ко многим с таблицей «Учеты», так как один предмет могут вести несколько преподавателей, которые проводят их с различными группами.

Таблица «Специальности» имеет связь один ко многим с таблицей «Учебные планы», так как у одной специальности много учебных планов.

Таблица «Семестры» имеет связь один ко многим с таблицей «Учебные планы», потому что в одном семестре много учебных планов.

Таблица «Предметы» имеет связь многие к одному с таблицей «Учебные планы», так как в одном учебном плане может быть много предметов.

Таблица «Группы» имеет связь один к одному с таблицей «Учебные планы», потому что предоставленного времени оказалось недостаточно на реализацию более реалистичного варианта.

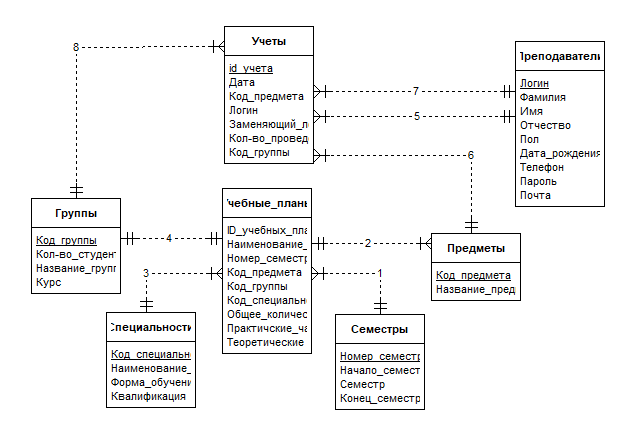


Рисунок - ER-диаграмма, которая отображает сущности и их связи

### Описание физической модели

Физическая модель базы данных состоит из 7 таблиц:

Таблица 1 – Специальности

|  |  |  |
| --- | --- | --- |
| Атрибуты | Типы данных | Ограничения |
| Код специальности | varchar | 10 |
| Наименование специальности | varchar | max |
| Форма обучения | varchar | 17 |
| Квалификация | varchar | max |

Все атрибуты таблицы имеют один тип данных, так как в данном учебном заведении учили использовать выбранный тип данных, если формат информации текстовый.

Код специальности ограничивается 10 символами, поскольку предполагалось, что код специальности с наибольшим количеством знаков будет иметь это ограничение.

Наименование специальности имеет ограничение max, так как нет смыла высчитывать каждый символ в специальностях и указывать максимально возможное их количество.

Форма обучения имеет ограничение в 17 символов, так как существует 3 формы обучения, и необходимая информация с наибольшим количеством символов составляет именно указанное ограничение.

Квалификация имеет ограничение max, потому что профессий много, и иногда в данном атрибуте могут быть указаны несколько профессий. А ведь они еще и с большим количеством символов могут быть. Поэтому и был выбран данный вариант.

Таблица 2 – Учебные планы

|  |  |  |
| --- | --- | --- |
| Атрибуты таблицы | Типы данных | Ограничения |
| ID учебных планов | int | - |
| Наименование учебных планов | varchar | 50 |
| Код предмета | varchar | 10 |
| Общее количество часов | int | - |
| Практические часы | int | - |
| Теоретические часы | int | - |
| Номер семестра | int | - |
| Код специальности | varchar | 10 |
| Код группы | varchar | 7 |

ID учебных планов, номер семестра, общее количество часов, практические часы и теоретические часы имеют числовой формат данных, поэтому указан выбранный тип данных. Ограничения не обозначены, так как очень сложно дойти до предела. К тому же не ожидается, что приложением будут пользоваться в дальнейшем.

Наименование учебных планов имеют указанный тип данных, так как в данном учебном заведении учили использовать выбранный тип данных, если формат информации текстовый.

Базой данных было предложено ограничение в пятьдесят символов для наименования учебных планов.

Остальные атрибуты являются внешними ключами, следовательно, их тип данных и ограничения будут соответствовать своим первичным ключам.

Таблица 3 – Семестры

|  |  |  |
| --- | --- | --- |
| Атрибуты | Типы данных | Ограничения |
| Номер семестра | int | - |
| Семестр | varchar | 10 |
| Начало семестра | varchar | 5 |
| Конец семестра | varchar | 5 |

Номер семестра является первичным ключом и имеет числовой формат, поэтому и был выбран указанный тип данных. Ограничения не обозначены, так как достаточно двух записей для работы с базой данных.

Семестр, начало семестра и конец семестра имеют данный тип данных, так как в данном учебном заведении учили использовать выбранный тип данных, если формат информации текстовый.

Семестр имеет ограничение в десятью символов, так как подразумевалось, что максимально возможное слово содержит данное количество символов.

Начало семестра и конец семестра имеют указанное ограничение, так как данные атрибуты содержат в себе день и месяц, разделенный точкой.

Таблица 4 – Предметы

|  |  |  |
| --- | --- | --- |
| Атрибуты | Типы данных | Ограничения |
| Код предмета | varchar | 10 |
| Наименование предмета | varchar | max |

Все атрибуты таблицы имеют один тип данных, так как в данном учебном заведении учили использовать выбранный тип данных, если формат информации текстовый.

Код предмета ограничивается десятью символами, поскольку предполагалось, что код предмета с наибольшим количеством знаков будет иметь это ограничение.

Наименование предмета имеет ограничение max, так как нет времени и смыла высчитывать каждый символ в предметах и указывать максимально возможное их количество.

Таблица 5 – Группы

|  |  |  |
| --- | --- | --- |
| Атрибуты | Типы данных | Ограничения |
| Код группы | varchar | 7 |
| Название группы | varchar | max |
| Курс | varchar | 1 |
| Количество студентов | int | - |

Код группы, название группы имеют данный тип данных, так как в данном учебном заведении учили использовать выбранный тип данных, если формат информации текстовый.

Курс имеет указанный тип данных, так как можно ограничить ввод необходимой информации.

Количество студентов имеет числовой формат данных, поэтому и выбран данный тип данных.

Код группы ограничивается семью символами, так как подразумевалось, что максимально возможный код группы состоит из данного количества.

Название группы имеет ограничение max, так как в данном атрибуте указывается наименование специальности.

Курс имеет ограничение в один символ, так как последним курсом в колледжах является четвертый. К тому же подразумевается, что в данном атрибуте следует указывать число.

Ограничение количества студентов не обозначены, так как в группе не могут учиться студенты более сорока.

Таблица 6 – Учеты

|  |  |  |
| --- | --- | --- |
| Атрибуты | Типы данных | Ограничения |
| ID учета | int | - |
| Дата | date | - |
| Код предмета | varchar | 10 |
| Логин | varchar | 50 |
| Заменяющий логин | varchar | 50 |
| Проведенные занятия | int | - |
| Код группы | varchar | 7 |

ID учета и проведенные занятия имеют числовой формат информации, следовательно, имеют соответствующий тип данных. Ограничения не обозначены, так как очень сложно дойти до предела.

Дата имеет соответствующий ей тип данных, потому что она более подходящая. Ограничения не указаны, так как их нельзя задать в базе данных.

Остальные атрибуты являются внешними ключами, поэтому их тип данных и ограничения будут соответствовать своим первичным ключам.

Таблица 7 – Преподаватели

|  |  |  |
| --- | --- | --- |
| Атрибуты | Типы данных | Ограничения |
| Логин | varchar | 50 |
| Фамилия | varchar | 50 |
| Имя | varchar | 50 |
| Отчество | varchar | 50 |
| Пароль | varchar | max |
| Пол | char | 1 |
| День рождения | date | - |
| Номер телефона | char | 15 |
| Почта | varchar | 50 |

Пол имеет данный тип данных, так как содержит в себе текстовый формат информации.

День рождения имеет указанный тип данных, так как содержит в себе формат информации дата. Ограничения не указаны, так как их нельзя задать в базе данных.

Номер телефона имеет выбранный тип данных, так как может содержать в себе специальные символы.

Остальные имеют указанный тип данных, так как в данном учебном заведении учили использовать выбранный тип данных, если формат информации текстовый.

Базой данных было предложено ограничение в пятьдесят символов для логина, фамилии, имени, отчества и почты.

Пароль имеет ограничение max, так как пароль будет хеширован, из-за чего количество символов в данном атрибуте будет меняться.

Пол имеет ограничение в один символ, потому что так легче заполнять таблицу. К тому же подразумевается, что в данном атрибуте следует указывать букву.

Телефон имеет ограничение в пятнадцать символов, так как подразумевалось, что номер будет вписан, используя пробелы в необходимых местах.

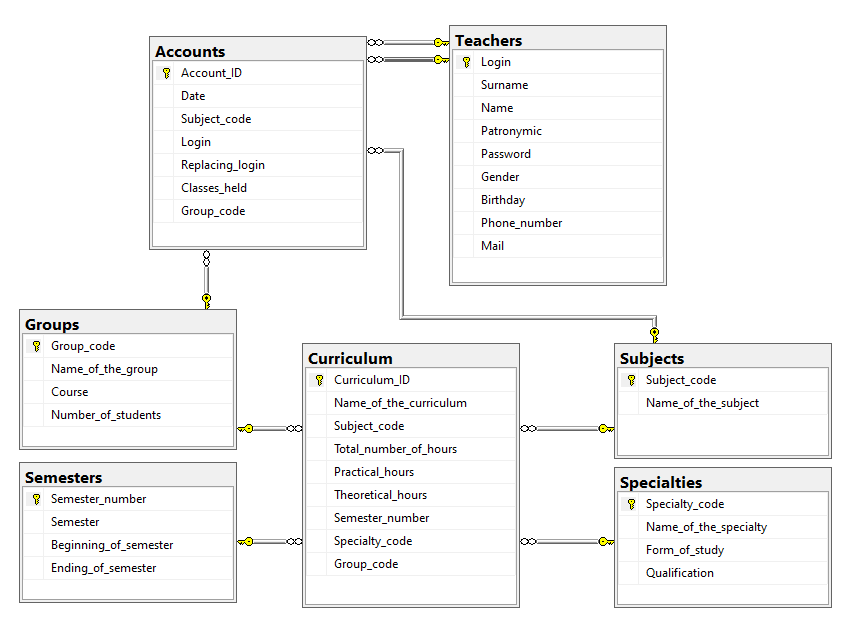


Рисунок 2 - Диаграмма физической модели

## Стадии разработки приложения

Приложение реализовано с использованием фреймворка WPF, так как уже работали с ним. Для работы с данной программой устройству необходимо иметь следующие характеристики:

* характеристики компьютера:

1. процессор Intel(R) Core(TM) i3-10110U CPU @ 2.10GHz 2.59 GHz;
2. оперативная память в 8,00 ГБ;
3. 64-разрядная операционная система, процессор x64;
4. для этого монитора недоступен ввод с помощью пера и сенсорный ввод;

* характеристики операционной системы:

1. windows 10 Pro;
2. версия 22H2;
3. сборка ОС 19045.2965;

взаимодействие Windows Feature Experience Pack 1000.19041.1000.0.

### Авторизация и аутентификация

Самым первым в приложении было сделано окно «Авторизация» (Рисунок 3, ПРИЛОЖЕНИЕ А), так как уже удавалось создавать ее. Данное окно разработано с помощью использования прошлой работы как шаблон. Отличие заключается в том, что в этом есть уровни доступа и использован класс «Password\_hashing» (ПРИЛОЖЕНИЕ Б) для того, чтобы введенный пароль хешировался с помощью SHA256. А для уровней доступа применялся класс «For\_admin» (ПРИЛОЖЕНИЕ Б).

Аутентификация для преподавателей реализована с использованием языка запросов, сравнивая заполненные поля с информацией из базы данных. Данные администратора (ПРИЛОЖЕНИЕ А) указаны в самом приложении для простоты.

Для сравнивания значений из полей с информацией из базы данных использован класс «DataBase» (ПРИЛОЖЕНИЕ Б). В случае правильного ввода данных отображается сообщение об успешном входе в систему. После чего переходим на другое окно.

Если авторизовался администратор, то его логин передается в класс, с помощью которого следующее окно после данного опознает пользователя и отображает скрытые для преподавателей кнопки.

Рисунки необходимы, чтобы заполнить пустое пространство. Кнопка имеет белый оттенок специально под цвет фона.

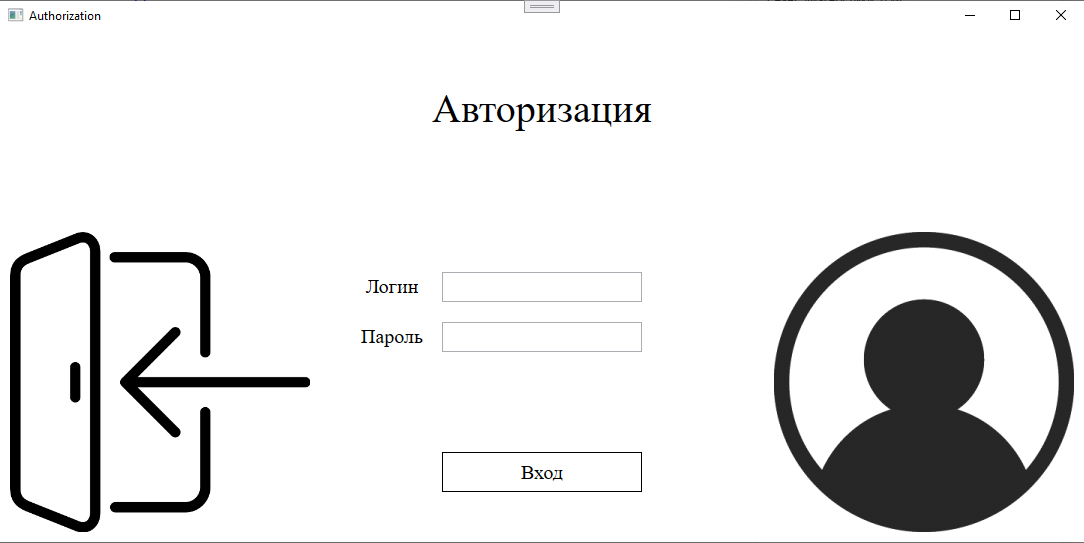


Рисунок - Окно «Авторизация»

### Подключение базы данных

Для подключения базы данных использована библиотека Entity Framework 6.x, так как оно уменьшает несогласованность между реляционным и объектно-ориентированным мирами, позволяя разработчикам создавать приложения, которые взаимодействуют с данными, хранящимися в реляционных базах данных и устраняя необходимость писать большой объем инфраструктурного кода для доступа к данным.

Этапы подключения базы данных:

1. Нажимаем на «Создать элемент» (Рисунок 4).

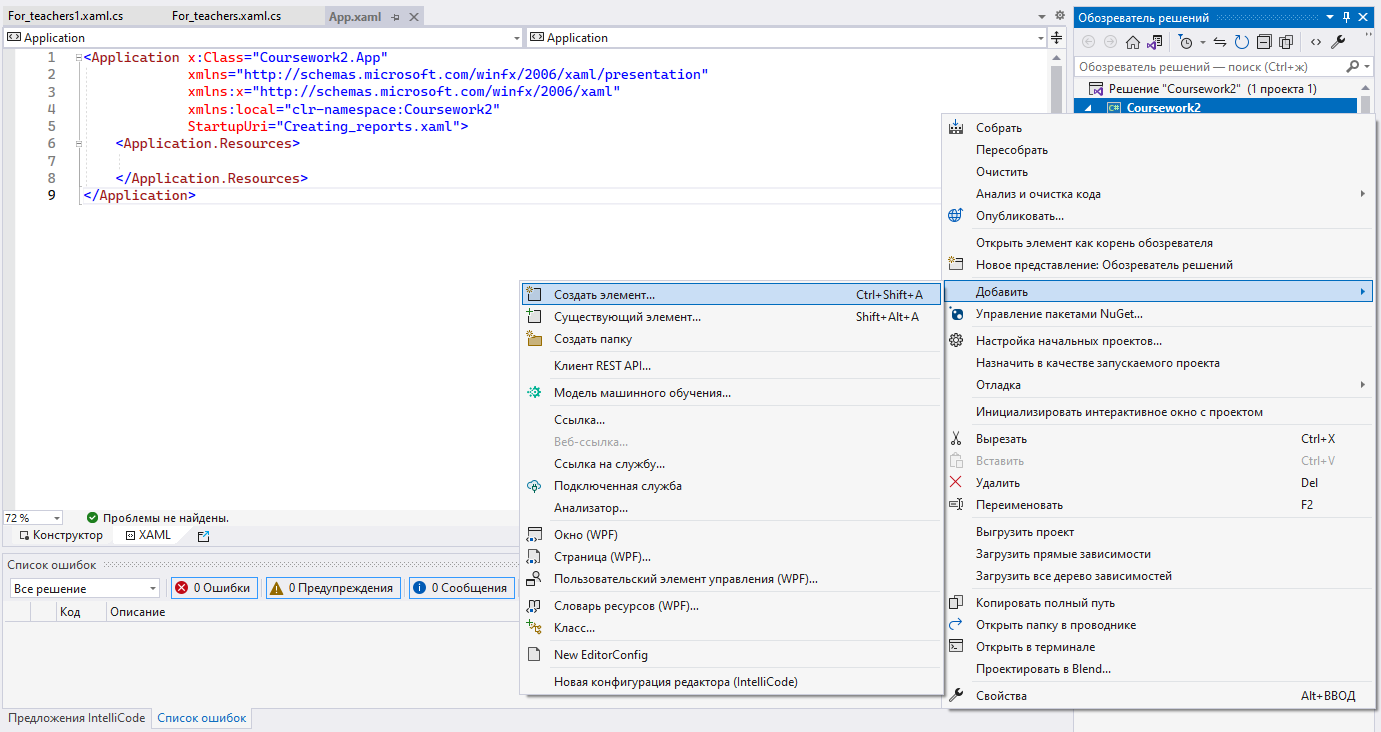


Рисунок - Создание элемента

1. Выбираем элемент «Модель ADO.NET EDM» и даем ему название.
2. Выбираем «Конструктор EF из базы данных»
3. Создаем подключение, вводим имя сервера и выбираем базу данных.
4. После того как выбрали базу данных, нажимаем «Далее».
5. Выбираем версию Entity Framework, которую будем использовать.
6. Выбираем все необходимое и нажимаем готово.

### Работа с базой данных через приложение

Для обращения к базе данных создан метод получение контекста (Рисунок 5), с помощью которого можно обращаться к ней.

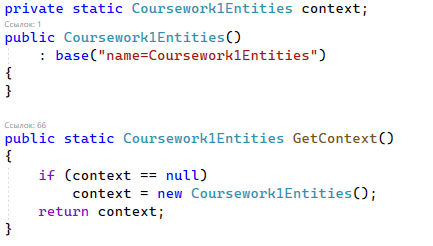


Рисунок - Метод для обращения к базе данных

После того как мы войдем в систему, окно «Авторизация» скроется, и будет отображено окно «Для перемещения 1» (Рисунок 6, ПРИЛОЖЕНИЕ А), с помощью которого можно переходить в такие окна, как «Учеты» (ПРИЛОЖЕНИЕ В), «Создание отчетов» (ПРИЛОЖЕНИЕ Г), «Для перемещения 2» (ПРИЛОЖЕНИЕ Д) и «Информация об учителях» (ПРИЛЖЕНИЕ Е).

Рисунки в данном окне необходимы, чтобы заполнить пустое пространство.

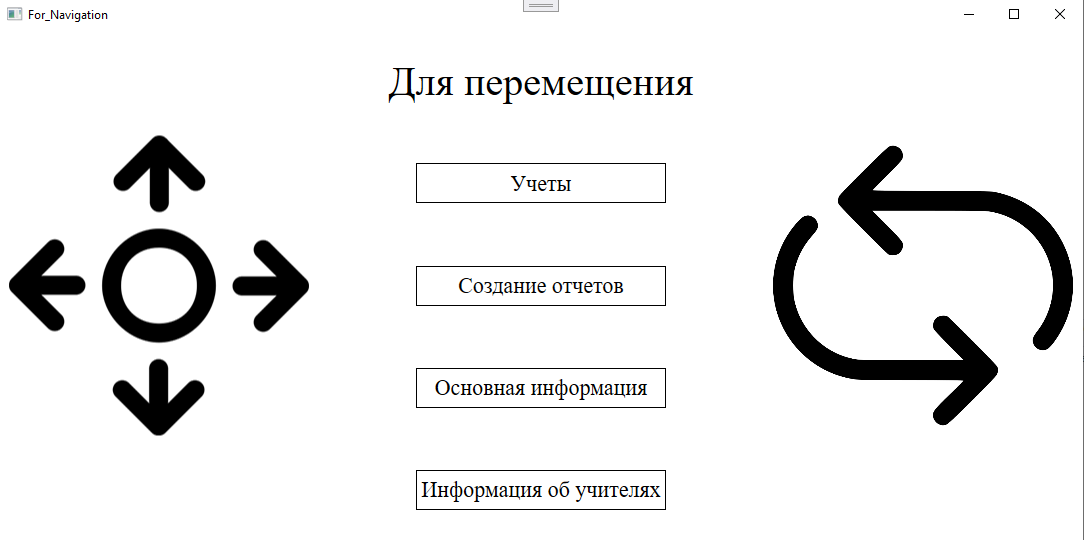


Рисунок - Окно «Для перемещения 1»

При нажатии кнопки «Основная информация» отображается окно «Для перемещения 2» (ПРИЛОЖЕНИЕ Д), с помощью которого можно переходить в такие окна, как «Специальности» (ПРИЛОЖЕНИЕ Ж), «Учебные планы» (ПРИЛОЖЕНИЕ И), «Семестры» (ПРИЛОЖЕНИЕ К), «Предметы» (ПРИЛОЖЕНИЕ Л), «Группы» (ПРИЛОЖЕНИЕ М).

В данном окне имеется семь кнопок. Двое из них реализуют функционал возврата и выхода, а остальные предназначены для работы с базой данных. Некоторые кнопки большие, чтобы указать их важность.

В данных иллюстрированных окнах реализовано только перемещение между окнами и ничего более.

Чтобы работать с базой данных созданы следующие окна:

* 1. Окно «Специальности» (ПРИЛОЖЕНИЕ Ж) имеет функционал возврата, выхода, ввода и исправления данных. Информация отображается посредством привязки данных к необходимым атрибутам таблицы 1.
  2. Окно «Учебные планы» (ПРИЛОЖЕНИЕ И) имеет функционал возврата, выхода, ввода и исправления данных. Информация отображается посредством привязки данных к необходимым атрибутам таблицы 2.
  3. Окно «Семестры» (ПРИЛОЖЕНИЕ К) имеет функционал возврата, выхода, ввода и исправления данных. Информация отображается посредством привязки данных к необходимым атрибутам таблицы 3.
  4. Окно «Предметы» (ПРИЛОЖЕНИЕ Л) имеет функционал возврата, выхода, ввода и исправления данных. Информация отображается посредством привязки данных к необходимым атрибутам таблицы 4.
  5. Окно «Группы» (ПРИЛОЖЕНИЕ М) имеет функционал возврата, выхода, ввода и исправления данных. Информация отображается посредством привязки данных к необходимым атрибутам таблицы 5.
  6. Окно «Преподаватели» (ПРИЛОЖЕНИЕ Е) имеет функционал возврата, выхода, ввода, исправления и удаления данных. Информация отображается посредством привязки данных к необходимым атрибутам таблицы 7. В данном окне были ошибки с хешированием пароля, а именно метод для хеширования использовалась неправильно. Оказалось, что результат метода не присваивался атрибуту таблицы 7.
  7. Окно «Учеты» (ПРИЛОЖЕНИЕ В) имеет функционал возврата, выхода, ввода, исправления и удаления данных. Информация отображается посредством привязки данных к необходимым атрибутам таблицы 6.

Кнопки «Назад», «Добавить» и «Изменить» выполняют перемещение в разные окна. А «Выход» отключает приложение.

Кнопка «Изменить» реализована так, чтобы можно было легче взять контекст данных строки, в которой он расположен, и передавать эти данные в окно для изменения информации.

Удаление реализовано таким образом, что выделенные записи передаются в кое-какую переменную, которая преобразует данные в коллекцию, и при нажатии соответствующей кнопки отображается диалоговое окно удаления (Рисунок 7). Если выбрать да, то выше упомянутая коллекция удаляется, после чего произведенные изменения сохраняются в базе данных и отображается сообщение об успешном удалении (Рисунок 8) записей.

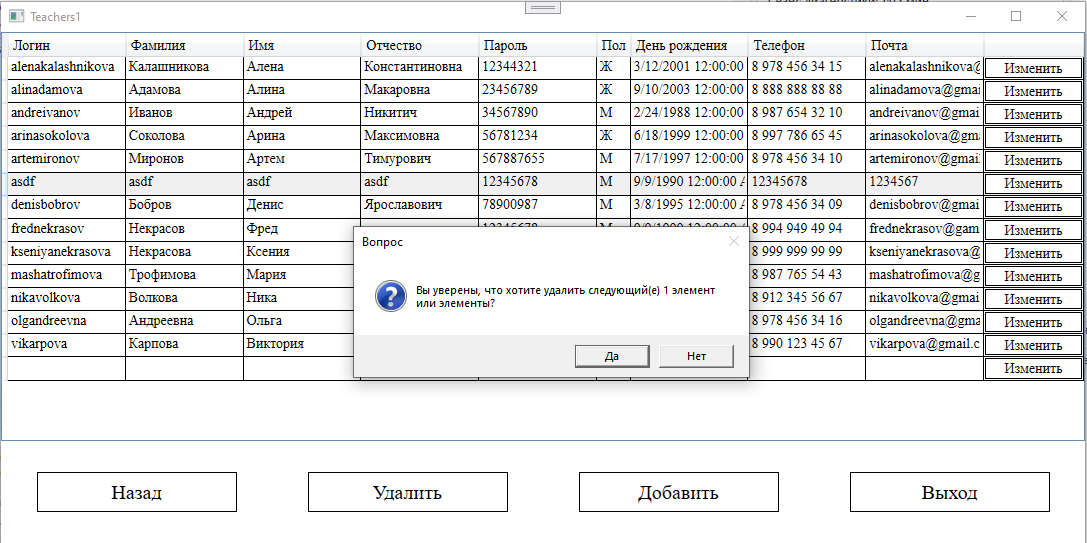


Рисунок - Диалоговое окно удаления

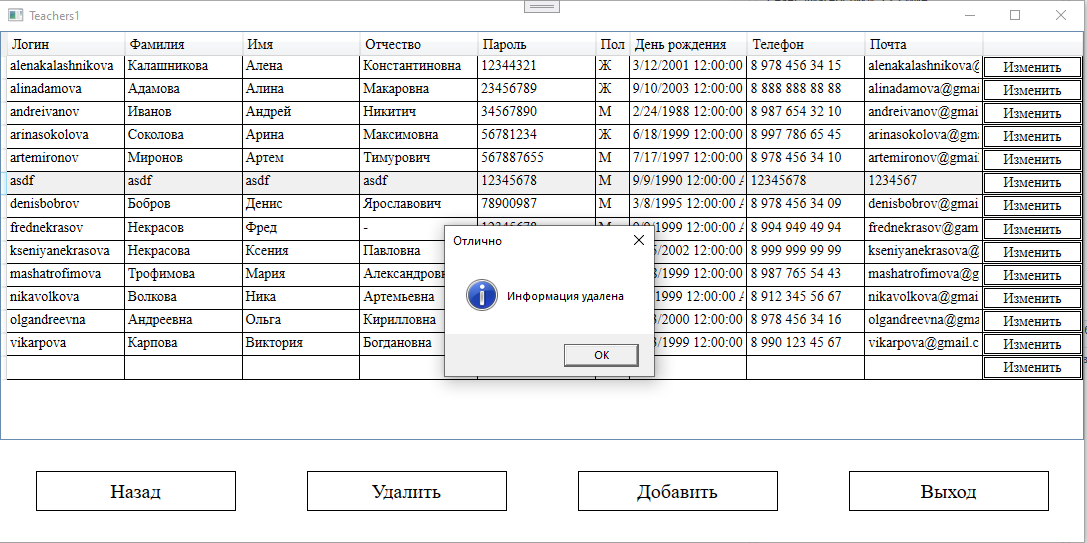


Рисунок - Сообщение об успешном удалении

Каждое окно ввода и изменения данных имеет контекст данных, определенной таблицы базы данных, и информацию, которая должна быть введена в соответствующее поле. Окна для редактирования (Рисунок 9) и добавления (Рисунок 10) данных реализованы одинаково, за исключением того, что в добавлении записей все поля пустые. А при исправлении будут уже заполненные строки, так как передается контекст данных.

Все строки и выпадающие списки необходимо заполнить, иначе выводится сообщение об ошибке с указанием ограничений.

Каждое поле привязана к определенному атрибуту таблицы базы данных, поэтому строки имеют почти такие же типы данных и ограничения. Но выпадающие списки выводят уже существующую в базе данных информацию, из-за чего ограничением данного элемента является выбор необходимого варианта.

Были ошибки в окнах изменений данных, использующие выпадающие списки, в которых текст не отображался, а для решения данной проблемы следовало привязать текст упомянутого элемента с соответствующим атрибутом таблицы.

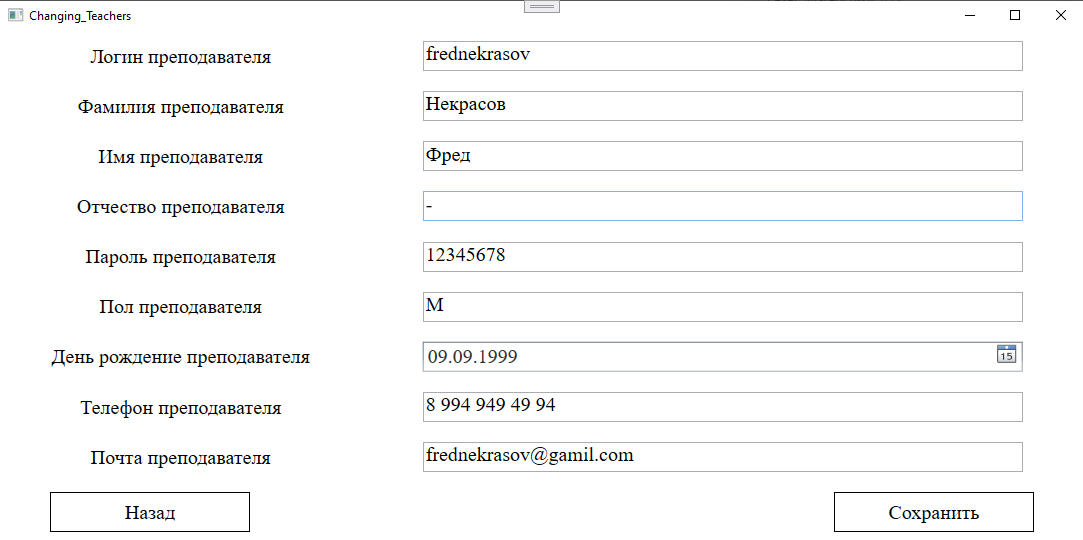


Рисунок - Окно для изменения записей

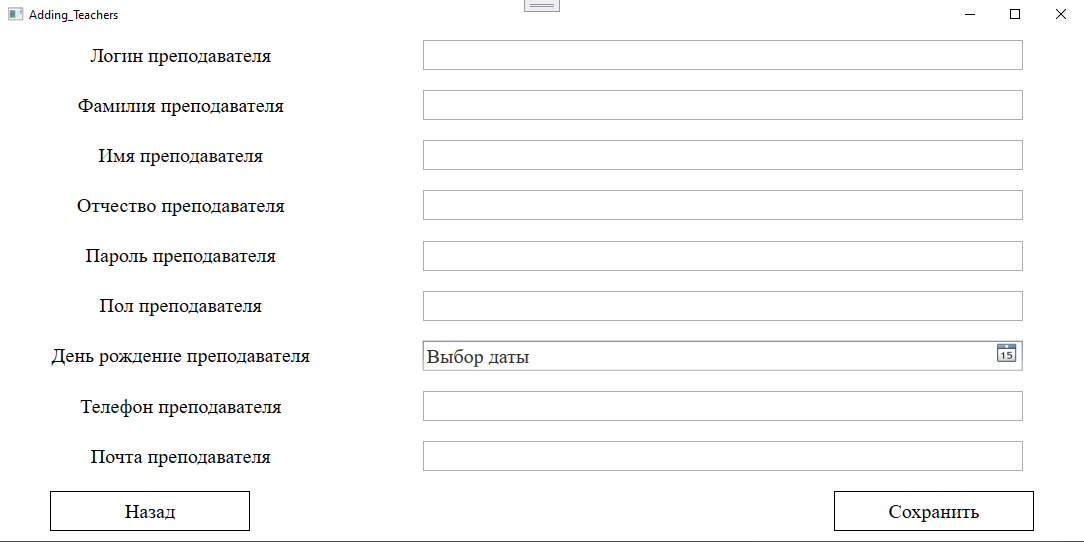


Рисунок - Окно для добавления записей

### Подсчет проведенных и не проведенных занятий

Для того чтобы посмотреть количество проведенных и не проведенных занятий, необходимо нажать кнопку «Создание отчетов» в окне «Для перемещения 1» (Рисунок 6, ПРИЛОЖЕНИЕ А) и следует выбрать предмет и группу, в которой преподаватель вел занятие.

Выше упомянутое необходимо для данной информации и еще на создание отчетов, поэтому остальной функционал окна скрыт. Но после того, как поля буду заполнены и нажата кнопка «Далее», иллюстрированные выпадающие списки, метка и кнопка станут скрытыми, а остальные будут видимы пользователю. Выпадающие списки привязаны к соответствующим атрибутам таблиц, поэтому отображают информацию без дублирования.

Проведенные и не проведенные занятия отображаются с помощью элемента, который может содержать текст. В вычислении данной информации были ошибки, а именно в условии использовалась пустая переменная, из-за чего необходимая информация не отображалась. Несколько дней было непонятно, в чем проблема. А решить ее оказалось просто. Нужно было использовать списки и присвоить им информацию из таблицы базы данных и с их помощью сравнить, существует ли выбранная информация в атрибуте. После этого все написанное сработало.

### Создание отчетов

Для того, чтобы создать отчет, нужно нажать кнопку «Создание отчетов» в окне «Для перемещения 1» (Рисунок 6, ПРИЛОЖЕНИЕ А). Для начала работы с отчетами необходимо заполнить данные, после чего будет доступна функция создания отчета. В приложении существуют две формы отчетов:

1. Отчет о проведенных часах по предметам (ПРИЛОЖЕНИЕ Н).
2. Отчет о проведенных часах по дням преподавателем (ПРИЛОЖЕНИЕ Н).

При нажатии кнопок выводится сообщение об просмотре и печати отчета. После чего перемещают пользователя в окно печати отчета. Для печати достаточно код предмета, благодаря чему можно выбрать всю необходимую информацию. При нажатии кнопки печати открывается окно «Печать» (Рисунок 20), в котором можно выбрать, каким образом печатать данные и так далее. Потом следует нажать печать, после чего кнопки возврата и выхода скрываются на некоторое время, и отобразится окно сохранения результата печати для наименования файла. Когда печать прошла успешно, кнопки возвращают свои изначальные свойства видимости.

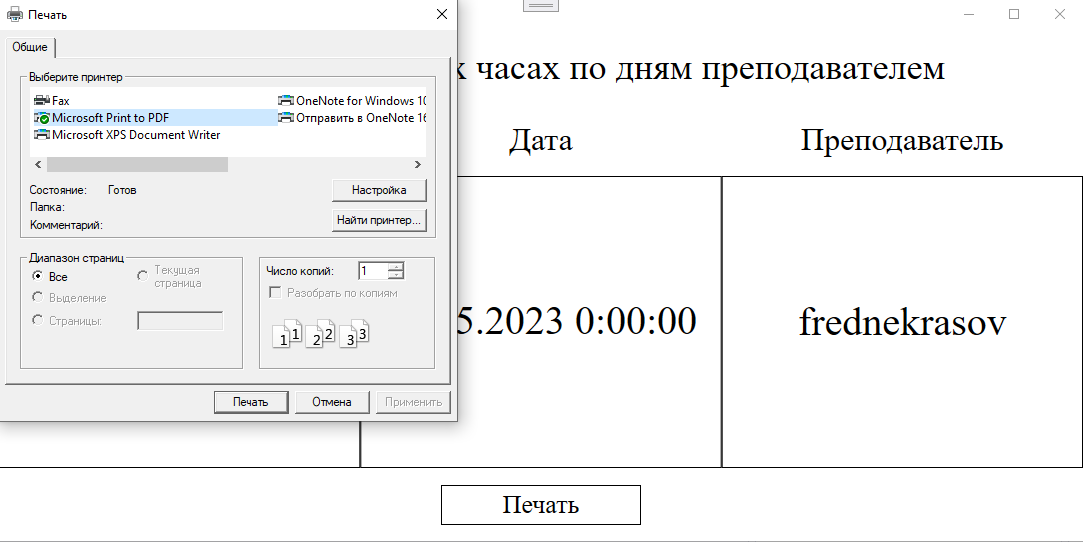


Рисунок - Окно «Печати»

## Тестирование приложения

Тестирование состоит из нескольких частей в виде таблиц, каждая таблица будет включать в себя проверку всего функционала одного окна.

Таблица 8 – Тестирование окна «Авторизация» (ПРИЛОЖЕНИЕ А)

|  |  |  |  |
| --- | --- | --- | --- |
| № | Проверяемая возможность | Ожидаемый результат | Фактический результат |
| 1 | Пустые поля | Сообщение об ошибке | Сообщение об ошибке |
| 2 | Пароль шифруется знаками звездочки | Поле пароля заполнено знаками звездочки | Поле пароля заполнено знаками звездочки |
| 3 | Правильный логин и неправильный пароль | Сообщение об ошибке | Сообщение об ошибке |
| 4 | Неправильный логин и правильный пароль | Сообщение об ошибке | Сообщение об ошибке |
| 5 | Правильный логин и пароль | Сообщение об успешном входе в систему | Сообщение об успешном входе в систему |
| 6 | Переход на окно «Для перемещения 1», после ввода правильных данных | Переходит на окно «Для перемещения 1» | Переходит на окно «Для перемещения 1» |
| 7 | Уровень доступа «Преподаватель» | Отображаются две кнопки в следующем окне | Отображаются две кнопки в следующем окне |
| 8 | Уровень доступа «Администратор» | Отображаются все кнопки в следующем окне | Отображаются все кнопки в следующем окне |

Таблица 9 – Тестирование окна «Для перемещения 1» (ПРИЛОЖЕНИЕ А)

|  |  |  |  |
| --- | --- | --- | --- |
| № | Проверяемый функционал | Ожидаемый результат | Фактический результат |
| 1 | Переход на окно «Учеты» | Переходит на окно «Учеты» | Переходит на окно «Учеты» |
| 2 | Переход на окно «Создание отчетов» | Переходит на окно «Создание отчетов» | Переходит на окно «Создание отчетов» |
| 3 | Переход на окно «Основная информация» | Переходит на окно «Основная информация» | Переходит на окно «Основная информация» |
| 4 | Переход на окно «Преподаватели» | Переходит на окно «Преподаватели» | Переходит на окно «Преподаватели» |

Таблица 10 – Тестирование окна «Основная информация» (ПРИЛОЖЕНИЕ Д)

|  |  |  |  |
| --- | --- | --- | --- |
| № | Проверяемая возможность | Ожидаемый результат | Фактический результат |
| 1 | Переход на окно «Специальности» | Переходит на окно «Специальности» | Переходит на окно «Специальности» |
| 2 | Переход на окно «Учебные планы» | Переходит на окно «Учебные планы» | Переходит на окно «Учебные планы» |
| 3 | Переход на окно «Семестры» | Переходит на окно «Семестры» | Переходит на окно «Семестры» |
| 4 | Переход на окно «Предметы» | Переходит на окно «Предметы» | Переходит на окно «Предметы» |
| 5 | Переход на окно «Группы» | Переходит на окно «Группы» | Переходит на окно «Группы» |
| 6 | Возвращение на предыдущее окно | Возвращается на предыдущее окно | Возвращается на предыдущее окно |
| 7 | Выход из приложения | Работа приложения завершается | Работа приложения завершается |

Таблица 11 – Тестирование окна «Специальности» (ПРИЛОЖЕНИЕ Ж)

|  |  |  |  |
| --- | --- | --- | --- |
| № | Проверяемая возможность | Ожидаемый результат | Фактический результат |
| 1 | Просмотр информации | Информация отображается | Информация отображается |
| 2 | Переход на окно добавления записи | Переходит на окно добавления записи | Переходит на окно добавления записи |
| 3 | Переход на окно изменения записи | Переходит на окно изменения записи | Переходит на окно изменения записи |
| 4 | Возвращение на предыдущее окно | Возвращается на предыдущее окно | Возвращается на предыдущее окно |
| 5 | Выход из приложения | Работа приложения завершается | Работа приложения завершается |

Таблица 12 – Тестирование окна «Учебные планы» (ПРИЛОЖЕНИЕ И)

|  |  |  |  |
| --- | --- | --- | --- |
| № | Проверяемая возможность | Ожидаемый результат | Фактический результат |
| 1 | Просмотр информации | Информация отображается | Информация отображается |
| 2 | Переход на окно добавления записи | Переходит на окно добавления записи | Переходит на окно добавления записи |
| 3 | Переход на окно изменения записи | Переходит на окно изменения записи | Переходит на окно изменения записи |
| 4 | Возвращение на предыдущее окно | Возвращается на предыдущее окно | Возвращается на предыдущее окно |
| 5 | Выход из приложения | Работа приложения завершается | Работа приложения завершается |

Таблица 13 – Тестирование окна «Семестры» (ПРИЛОЖЕНИЕ К)

|  |  |  |  |
| --- | --- | --- | --- |
| № | Проверяемая возможность | Ожидаемый результат | Фактический результат |
| 1 | Просмотр информации | Информация отображается | Информация отображается |
| 2 | Переход на окно добавления записи | Переходит на окно добавления записи | Переходит на окно добавления записи |
| 3 | Переход на окно изменения записи | Переходит на окно изменения записи | Переходит на окно изменения записи |
| 4 | Возвращение на предыдущее окно | Возвращается на предыдущее окно | Возвращается на предыдущее окно |
| 5 | Выход из приложения | Работа приложения завершается | Работа приложения завершается |

Таблица 14 – Тестирование окна «Предметы» (ПРИЛОЖЕНИЕ Л)

|  |  |  |  |
| --- | --- | --- | --- |
| № | Проверяемая возможность | Ожидаемый результат | Фактический результат |
| 1 | Просмотр информации | Информация отображается | Информация отображается |
| 2 | Переход на окно добавления записи | Переходит на окно добавления записи | Переходит на окно добавления записи |
| 3 | Переход на окно изменения записи | Переходит на окно изменения записи | Переходит на окно изменения записи |
| 4 | Возвращение на предыдущее окно | Возвращается на предыдущее окно | Возвращается на предыдущее окно |
| 5 | Выход из приложения | Работа приложения завершается | Работа приложения завершается |

Таблица 15 – Тестирование окна «Группы» (ПРИЛОЖЕНИЕ М)

|  |  |  |  |
| --- | --- | --- | --- |
| № | Проверяемая возможность | Ожидаемый результат | Фактический результат |
| 1 | Просмотр информации | Информация отображается | Информация отображается |
| 2 | Переход на окно добавления записи | Переходит на окно добавления записи | Переходит на окно добавления записи |
| 3 | Переход на окно изменения записи | Переходит на окно изменения записи | Переходит на окно изменения записи |
| 4 | Возвращение на предыдущее окно | Возвращается на предыдущее окно | Возвращается на предыдущее окно |
| 5 | Выход из приложения | Работа приложения завершается | Работа приложения завершается |

Таблица 16 – Тестирование окна «Преподаватели» (ПРИЛОЖЕНИЕ Е)

|  |  |  |  |
| --- | --- | --- | --- |
| № | Проверяемая возможность | Ожидаемый результат | Фактический результат |
| 1 | Просмотр информации | Информация отображается | Информация отображается |
| 2 | Переход на окно добавления записи | Переходит на окно добавления записи | Переходит на окно добавления записи |
| 3 | Переход на окно изменения записи | Переходит на окно изменения записи | Переходит на окно изменения записи |
| 4 | Возвращение на предыдущее окно | Возвращается на предыдущее окно | Возвращается на предыдущее окно |
| 5 | Удаление записей | Вывод окна удаления, оповещение об удалении | Вывод окна удаления, оповещение об удалении |
| 6 | Выход из приложения | Работа приложения завершается | Работа приложения завершается |

Таблица 17 – Тестирование окна «Учеты» (ПРИЛОЖЕНИЕ В)

|  |  |  |  |
| --- | --- | --- | --- |
| № | Проверяемая возможность | Ожидаемый результат | Фактический результат |
| 1 | Просмотр информации | Информация отображается | Информация отображается |
| 2 | Переход на окно добавления записи | Переходит на окно добавления записи | Переходит на окно добавления записи |
| 3 | Переход на окно изменения записи | Переходит на окно изменения записи | Переходит на окно изменения записи |
| 4 | Возвращение на предыдущее окно | Возвращается на предыдущее окно | Возвращается на предыдущее окно |
| 5 | Удаление записей | Вывод окна удаления, оповещение об удалении | Вывод окна удаления, оповещение об удалении |
| 6 | Выход из приложения | Работа приложения завершается | Работа приложения завершается |

Таблица 18 – Тестирование окон для изменения записи.

|  |  |  |  |
| --- | --- | --- | --- |
| № | Проверяемая возможность | Ожидаемы результат | Фактический результат |
| 1 | Возвращение на предыдущее окно | Возвращается на предыдущее окно | Возвращается на предыдущее окно |
| 2 | Ввод данных | Информация записывается | Информация записывается |
| 3 | Выбор данных, если это выпадающий список | Присутствует выбор данных | Присутствует выбор данных |
| 4 | Оставить поля пустыми или пустое поле | Сообщение с описанием ограничений | Сообщение с описанием ограничений |
| 5 | Вывод ограничений в случае неправильного заполнения | Сообщение с описанием ограничений | Сообщение с описанием ограничений |
| 6 | Изменение записей | Оповещение об успешном изменении записи | Оповещение об успешном изменении записи |

Таблица 19 – Тестирование окон для добавления записи.

|  |  |  |  |
| --- | --- | --- | --- |
| № | Проверяемая возможность | Ожидаемы результат | Фактический результат |
| 1 | Возвращение на предыдущее окно | Возвращается на предыдущее окно | Возвращается на предыдущее окно |
| 2 | Ввод данных | Информация записывается | Информация записывается |
| 3 | Выбор данных, если это выпадающий список | Присутствует выбор данных | Присутствует выбор данных |
| 4 | Добавить пустые поля или пустое поле | Сообщение с описанием ограничений | Сообщение с описанием ограничений |
| 5 | Вывод ограничений в случае неправильного заполнения | Сообщение с описанием ограничений | Сообщение с описанием ограничений |
| 6 | Добавление записей | Оповещение об успешном добавлении записи | Оповещение об успешном добавлении записи |

Таблица 20 – Тестирование окна «Создание отчетов» (ПРИЛОЖЕНИЕ Г)

|  |  |  |  |
| --- | --- | --- | --- |
| № | Проверяемая возможность | Ожидаемый результат | Фактический результат |
| 1 | Выбор кода группы | Присутствует выбор кода группы | Присутствует выбор кода группы |
| 2 | Выбор кода предмета | Присутствует выбор кода предмета | Присутствует выбор кода предмета |
| 3 | Отображение проведенных и не проведенных занятий | Отображается проведенные и не проведенные занятий | Отображается проведенные и не проведенные занятий |
| 4 | Переход на окно отчета по предметам | Переход на окно отчета по предметам | Переход на окно отчета по предметам |
| 5 | Переход на окно отчета по дням преподавателем | Переход на окно отчета по дням преподавателем | Переход на окно отчета по дням преподавателем |
| 6 | Возвращение на предыдущее окно | Возвращается на предыдущее окно | Возвращается на предыдущее окно |
| 7 | Выход из приложения | Работа приложения завершается | Работа приложения завершается |

Таблица 21 – Тестирование окна «Отчет о проведенных часах по предметам» (ПРИЛОЖЕНИЕ Н)

|  |  |  |  |
| --- | --- | --- | --- |
| № | Проверяемая возможность | Ожидаемый результат | Фактический результат |
| 1 | Возвращение на предыдущее окно | Возвращается на предыдущее окно | Возвращается на предыдущее окно |
| 2 | Вывод соответствующей информации | Необходимая информация отображается | Необходимая информация отображается |
| 3 | Печать отчета | Выводятся окна печати и сохранение результата печати | Выводятся окна печати и сохранение результата печати |
| 4 | Сохранение печати | Печать сохранена | Печать сохранена |
| 5 | Просмотр печати | Печать можно рассмотреть | Печать можно рассмотреть |
| 6 | Выход из приложения | Работа приложения завершается | Работа приложения завершается |

Таблица 22 – Тестирование окна «Отчет о проведенных часах по дням преподавателем» (ПРИЛОЖЕНИЕ Н)

|  |  |  |  |
| --- | --- | --- | --- |
| № | Проверяемая возможность | Ожидаемый результат | Фактический результат |
| 1 | Возвращение на предыдущее окно | Возвращается на предыдущее окно | Возвращается на предыдущее окно |
| 2 | Вывод соответствующей информации | Необходимая информация отображается | Необходимая информация отображается |
| 3 | Печать отчета | Выводятся окна печати и сохранение результата печати | Выводятся окна печати и сохранение результата печати |
| 4 | Сохранение печати | Печать сохранена | Печать сохранена |
| 5 | Просмотр печати | Печать можно рассмотреть | Печать можно рассмотреть |
| 6 | Выход из приложения | Работа приложения завершается | Работа приложения завершается |

# ЗАКЛЮЧЕНИЕ

В ходе выполнения курсовой работы было разработано приложение автоматизации учета проведенных занятий преподавателями колледжа. Целью работы было упростить и ускорить процесс учета проведенных занятий, а также поднять эффективность работы колледжа в целом.

В результате проведенной работы была создана база данных для хранения информации о специальностях, учебных планах, семестрах, предметах, учебных часах в разрезе типов (теория, практика), группах, а также учет по дням проведения занятий закрепленными преподавателями и замен преподавателей. Выполнен анализ и проектирование функционала системы, проведены требования к системе: аутентификация и авторизация пользователей, ввод/исправление необходимых информаций, учет по дням проведения занятий закрепленными преподавателями и замен преподавателей, подсчет часов проведенных и не проведённых занятий, создание отчетов.

В результате было установлено, что применение разработанной системы позволяет существенно сократить время и уменьшить количество ошибок в учете проведенных занятий, сделать процесс более удобным для преподавателей и улучшить эффективность работы колледжа в целом.

Рекомендации по оптимизации процесса могут быть использованы для построения дальнейшей модели, более адаптированной к специфике колледжа.

Таким образом, можно заключить, что выполненная работа позволила получить значимый результат и положительное воздействие на работу многих преподавателей. Цель работы с её задачами были достигнуты и приложены рекомендации для дальнейшего совершенствования приложения.

# СПИСОК ЛИТЕРАТУРЫ

1. Entity Framework 6. -URL: <https://learn.microsoft.com/ru-ru/ef/ef6/> (дата обращения 18.05.2023).
2. Подключение базы данных к приложению. Обращение к модели данных. Паттерн SingleTone. Реализация чтения, добавления, редактирования, удаления данных. Часть 1. -URL: <https://nationalteam.worldskills.ru/skills/rabota-s-bazoy-dannykh-v-prilozhenii-chtenie-dobavlenie-redaktirovanie-udalenie-dannykh-chast-1/> (дата обращения 16.05.2023).
3. Подключение базы данных к приложению. Обращение к модели данных. Паттерн SingleTone. Реализация чтения, добавления, редактирования, удаления данных. Часть 2. -URL: <https://nationalteam.worldskills.ru/skills/rabota-s-bazoy-dannykh-v-prilozhenii-chtenie-dobavlenie-redaktirovanie-udalenie-dannykh-chast-2/> (дата обращения 16.05.2023).

# ПРИЛОЖЕНИЕ А

Окно «Авторизация»

Title="Authorization" Height="550" Width="1100">

<Grid>

<Label Content="Авторизация" HorizontalAlignment="Center" VerticalAlignment="Top" Margin="0,50,0,0" FontSize="40" FontFamily="Times New Roman"/>

<Label Content="Логин" HorizontalAlignment="Center" Margin="0,0,300,0" VerticalAlignment="Center" FontSize="20" FontFamily="Times New Roman"/>

<Label Content="Пароль" HorizontalAlignment="Center" Margin="0,100,300,0" VerticalAlignment="Center" FontSize="20" FontFamily="Times New Roman"/>

<Image Visibility="Visible" HorizontalAlignment="Left" Height="300" Margin="10,0,0,10" VerticalAlignment="Bottom" Width="300" Source="/Images/login-pngrepo-com.png"/>

<Image Visibility="Visible" HorizontalAlignment="Right" Height="300" Margin="0,0,10,10" VerticalAlignment="Bottom" Width="300" Source="/Images/users's-icon2.png"/>

<Button Height="40" Width="200" Content="Вход" HorizontalAlignment="Center" Margin="0,0,0,50" VerticalAlignment="Bottom" FontSize="20" FontFamily="Times New Roman" BorderBrush="Black" Background="White" Click="Login\_Button\_Click"/>

<TextBox x:Name="TBuser\_name" HorizontalAlignment="Center" VerticalAlignment="Center" Height="30" Width="200" FontSize="22" FontFamily="Times New Roman"/>

<PasswordBox x:Name="PasswordBox1" HorizontalAlignment="Center" VerticalAlignment="Center" Height="30" Width="200" FontSize="22" FontFamily="Times New Roman" Margin="0,100,0,0"/>

</Grid>

public partial class MainWindow : Window

{

readonly private DataBase dataBase = new DataBase();

readonly private Password\_hashing password = new Password\_hashing();

public MainWindow()

{

InitializeComponent();

WindowStartupLocation = WindowStartupLocation.CenterScreen;

TBuser\_name.MaxLength = 20;

PasswordBox1.PasswordChar = '\*';

}

private void Login\_Button\_Click(object sender, RoutedEventArgs e)

{

var Username = TBuser\_name.Text;

var Password = password.Sha256(PasswordBox1.Password.ToString());

if (Username == "admin" && Password == "8c6976e5b5410415bde908bd4dee15dfb167a9c873fc4bb8a81f6f2ab448a918")

{

For\_admin.Text = "admin";

MessageBox.Show("Добрый день, вы успешно вошли в систему!", "Отлично!", MessageBoxButton.OK, MessageBoxImage.Information);

this.Hide();

For\_Navigation navigation = new For\_Navigation();

navigation.ShowDialog();

}

else

{

SqlDataAdapter adapter = new SqlDataAdapter();

DataTable table = new DataTable();

string querystring = $"select Login, Surname, Name, Patronymic, Password, Gender, Birthday, Phone\_number, Mail from Teachers where Login = '{Username}' and Password = '{Password}'";

SqlCommand command = new SqlCommand(querystring)

{

Connection = dataBase.GetConnection()

};

adapter.SelectCommand = command;

adapter.Fill(table);

if (table.Rows.Count > 0)

{

MessageBox.Show("Добрый день, вы успешно вошли в систему!", "Отлично!", MessageBoxButton.OK, MessageBoxImage.Information);

this.Hide();

For\_Navigation navigation = new For\_Navigation();

navigation.ShowDialog();

}

else

{

MessageBox.Show("Не оставляйте поля пустыми. Вводите свой правильный логин и пароль корректно! Попробуйте снова.", "Ошибка", MessageBoxButton.OK, MessageBoxImage.Error);

}

}

}

}

Окно «Для перемещения 1»

Title="For\_Navigation" Height="550" Width="1100">

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="\*"/>

<RowDefinition Height="\*"/>

<RowDefinition Height="\*"/>

<RowDefinition Height="\*"/>

<RowDefinition Height="\*"/>

</Grid.RowDefinitions>

<Button Content="Основная информация" Width="250" Height="40" FontSize="22" FontFamily="Times New Roman" BorderBrush="Black" Background="White" Click="To\_the\_necessary\_information\_Click" Grid.Row="3"/>

<Button Content="Создание отчетов" Width="250" Height="40" FontSize="22" FontFamily="Times New Roman" BorderBrush="Black" Background="White" Click="To\_create\_reports\_Click" Grid.Row="2"/>

<Button Content="Учеты" Width="250" Height="40" FontSize="22" FontFamily="Times New Roman" BorderBrush="Black" Background="White" Click="To\_accounts\_Click" Grid.Row="1"/>

<Button x:Name="To\_inf\_about\_teachers" Content="Информация об учителях" Width="250" Height="40" FontSize="22" FontFamily="Times New Roman" BorderBrush="Black" Background="White" Click="To\_inf\_about\_teachers\_Click" Grid.Row="4"/>

<Label Content="Для перемещения" HorizontalAlignment="Center" VerticalAlignment="Center" FontSize="40" FontFamily="Times New Roman" Grid.Row="0"/>

<Image Visibility="Visible" HorizontalAlignment="Left" Height="300" Margin="10,0,0,0" VerticalAlignment="Center" Width="300" Source="/Images/moving-icon.png" Grid.RowSpan="5"/>

<Image Visibility="Visible" HorizontalAlignment="Right" Height="300" Margin="0,0,10,0" VerticalAlignment="Center" Width="300" Source="/Images/moving-icon1.png" Grid.RowSpan="5"/>

</Grid>

public partial class For\_Navigation : Window

{

public For\_Navigation()

{

InitializeComponent();

WindowStartupLocation = WindowStartupLocation.CenterScreen;

if (For\_admin.Text == "admin")

{

Basic\_inf.Visibility = Visibility.Visible;

To\_inf\_about\_teachers.Visibility = Visibility.Visible;

}

}

private void To\_accounts\_Click(object sender, RoutedEventArgs e)

{

this.Hide();

Accounts1 accounts1 = new Accounts1();

accounts1.ShowDialog();

}

private void To\_create\_reports\_Click(object sender, RoutedEventArgs e)

{

this.Hide();

Creating\_reports creating\_Reports = new Creating\_reports();

creating\_Reports.ShowDialog();

}

private void To\_the\_necessary\_information\_Click(object sender, RoutedEventArgs e)

{

this.Hide();

To\_the\_necessary\_inf inf = new To\_the\_necessary\_inf();

inf.ShowDialog();

}

private void To\_inf\_about\_teachers\_Click(object sender, RoutedEventArgs e)

{

this.Hide();

Teachers1 teachers1 = new Teachers1();

teachers1.ShowDialog();

}

}

# ПРИЛОЖЕНИЕ Б

Класс Password\_hashing

internal class Password\_hashing

{

public string Sha256(string randomString)

{

var crypt = new System.Security.Cryptography.SHA256Managed();

var hash = new StringBuilder();

byte[] crypto = crypt.ComputeHash(Encoding.UTF8.GetBytes(randomString));

foreach (byte theByte in crypto)

{

hash.Append(theByte.ToString("x2"));

}

return hash.ToString();

}

}

Класс DataBase

internal class DataBase

{

readonly SqlConnection sqlConnection = new SqlConnection(@"Data Source=DESKTOP-TP5U6CA\SQLEXPRESS;Initial Catalog=Курсовая\_работа2;Integrated Security=True");

public void OpenConnection()

{

if (sqlConnection.State == System.Data.ConnectionState.Closed)

{

sqlConnection.Open();

}

}

public void CloseConnection()

{

if (sqlConnection.State == System.Data.ConnectionState.Open)

{

sqlConnection.Close();

}

}

public SqlConnection GetConnection()

{

return sqlConnection;

}

}

Класс For\_admin

static class For\_admin

{

public static string Text;

}

# ПРИЛОЖЕНИЕ В

Окно «Учеты»

Title="Accounts1" Height="550" Width="1100" IsVisibleChanged="Window\_IsVisibleChanged">

<Grid>

<Grid.ColumnDefinitions>

<ColumnDefinition Width="\*"/>

<ColumnDefinition Width="\*"/>

<ColumnDefinition Width="\*"/>

<ColumnDefinition Width="\*"/>

</Grid.ColumnDefinitions>

<Grid.RowDefinitions>

<RowDefinition Height="4\*"/>

<RowDefinition Height="\*"/>

</Grid.RowDefinitions>

<Button Content="Назад" Width="200" Height="40" FontSize="20" FontFamily="Times New Roman" BorderBrush="Black" Background="White" Click="Go\_back\_Click" Grid.Row="1" Grid.Column="0"/>

<Button Content="Удалить" Width="200" Height="40" FontSize="20" FontFamily="Times New Roman" BorderBrush="Black" Background="White" Grid.Row="1" Click="Delete\_Click" Grid.Column="1"/>

<Button Content="Добавить" Width="200" Height="40" FontSize="20" FontFamily="Times New Roman" BorderBrush="Black" Background="White" Grid.Row="1" Click="Add\_Click" Grid.Column="2"/>

<Button Content="Выход" Width="200" Height="40" FontSize="20" FontFamily="Times New Roman" BorderBrush="Black" Background="White" Click="Exit\_Click" Grid.Row="1" Grid.Column="3"/>

<DataGrid x:Name="Grid1" AutoGenerateColumns="False" Background="White" FontSize="14" FontFamily="Times New Roman" Grid.Row="0" Grid.ColumnSpan="4">

<DataGrid.Columns>

<DataGridTextColumn Header="Дата" Binding="{Binding Date}" Width="\*"/>

<DataGridTextColumn Header="Код предмета" Binding="{Binding Subject\_code}" Width="Auto"/>

<DataGridTextColumn Header="Преподаватель" Binding="{Binding Login}" Width="\*"/>

<DataGridTextColumn Header="Преподаватель на замену" Binding="{Binding Replacing\_login}" Width="\*"/>

<DataGridTextColumn Header="Проведенные пары" Binding="{Binding Classes\_held}" Width="Auto"/>

<DataGridTextColumn Header="Код группы" Binding="{Binding Group\_code}" Width="Auto"/>

<DataGridTemplateColumn Width="100">

<DataGridTemplateColumn.CellTemplate>

<DataTemplate>

<Button Content="Изменить" FontFamily="Times New Roman" BorderBrush="Black" Background="White" Click="Edit\_Click"/>

</DataTemplate>

</DataGridTemplateColumn.CellTemplate>

</DataGridTemplateColumn>

</DataGrid.Columns>

</DataGrid>

</Grid>

public partial class Accounts1 : Window

{

public Accounts1()

{

InitializeComponent();

WindowStartupLocation = WindowStartupLocation.CenterScreen;

}

private void Edit\_Click(object sender, RoutedEventArgs e)

{

Changing\_Accounts accounts = new Changing\_Accounts((sender as Button).DataContext as Accounts);

accounts.ShowDialog();

}

private void Add\_Click(object sender, RoutedEventArgs e)

{

Adding\_Accounts accounts = new Adding\_Accounts();

accounts.ShowDialog();

}

private void Delete\_Click(object sender, RoutedEventArgs e)

{

var Accounts\_For\_Removing = Grid1.SelectedItems.Cast<Accounts>().ToList();

if (MessageBox.Show($"Вы уверены, что хотите удалить следующий(е) {Accounts\_For\_Removing.Count()} элемент или элементы?", "Вопрос", MessageBoxButton.YesNo, MessageBoxImage.Question) == MessageBoxResult.Yes)

{

try

{

Coursework1Entities.GetContext().Accounts.RemoveRange(Accounts\_For\_Removing);

Coursework1Entities.GetContext().SaveChanges();

MessageBox.Show("Информация удалена", "Отлично", MessageBoxButton.OK, MessageBoxImage.Information);

}

catch (Exception ex)

{

MessageBox.Show(ex.Message.ToString());

}

}

}

private void Go\_back\_Click(object sender, RoutedEventArgs e)

{

this.Hide();

For\_Navigation for\_Navigation = new For\_Navigation();

for\_Navigation.ShowDialog();

}

private void Exit\_Click(object sender, RoutedEventArgs e)

{

Application.Current.Shutdown();

}

private void Window\_IsVisibleChanged(object sender, DependencyPropertyChangedEventArgs e)

{

if (Visibility == Visibility.Visible)

{

Coursework1Entities.GetContext().ChangeTracker.Entries().ToList().ForEach(p => p.Reload());

Grid1.ItemsSource = Coursework1Entities.GetContext().Accounts.ToList();

}

}

}

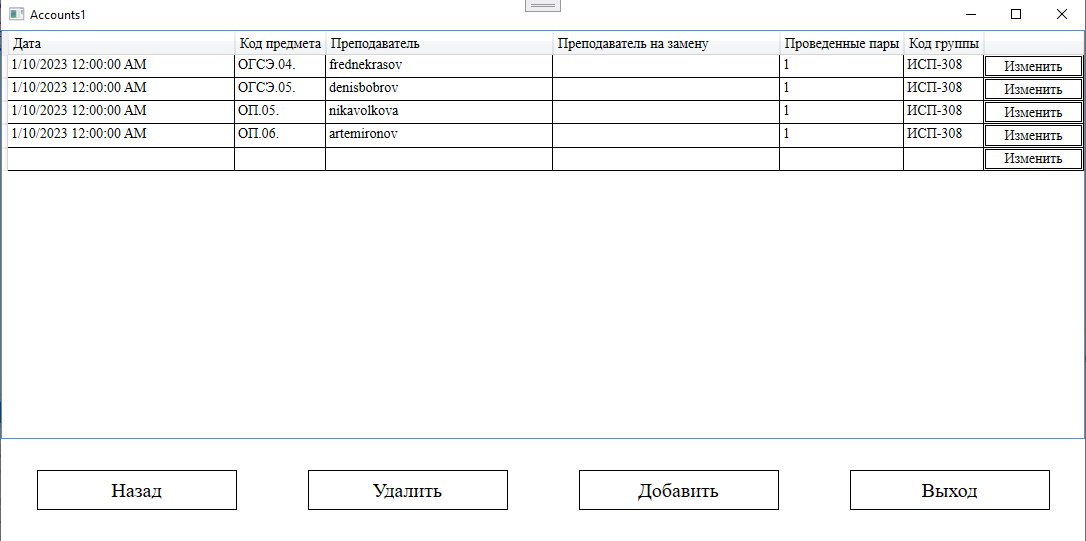


Рисунок - Окно «Учеты»

Окно добавление учетов

Title="Adding\_Accounts" Height="550" Width="1100">

<Grid>

<Grid.ColumnDefinitions>

<ColumnDefinition Width="\*"/>

<ColumnDefinition Width="2\*"/>

</Grid.ColumnDefinitions>

<Grid.RowDefinitions>

<RowDefinition Height="\*"/>

<RowDefinition Height="\*"/>

<RowDefinition Height="\*"/>

<RowDefinition Height="\*"/>

<RowDefinition Height="\*"/>

<RowDefinition Height="\*"/>

<RowDefinition Height="\*"/>

</Grid.RowDefinitions>

<Label Content="Дата" FontSize="20" FontFamily="Times New Roman" HorizontalAlignment="Center" VerticalAlignment="Center" Grid.Row="0" Grid.Column="0"/>

<Label Content="Код предмета" FontSize="20" FontFamily="Times New Roman" HorizontalAlignment="Center" VerticalAlignment="Center" Grid.Row="1" Grid.Column="0"/>

<Label Content="Преподаватель" FontSize="20" FontFamily="Times New Roman" HorizontalAlignment="Center" VerticalAlignment="Center" Grid.Row="2" Grid.Column="0"/>

<Label Content="Преподаватель на замену" FontSize="20" FontFamily="Times New Roman" HorizontalAlignment="Center" VerticalAlignment="Center" Grid.Row="3" Grid.Column="0"/>

<Label Content="Проведенные пары" FontSize="20" FontFamily="Times New Roman" HorizontalAlignment="Center" VerticalAlignment="Center" Grid.Row="4" Grid.Column="0"/>

<Label Content="Код группы" FontSize="20" FontFamily="Times New Roman" HorizontalAlignment="Center" VerticalAlignment="Center" Grid.Row="5" Grid.Column="0"/>

<DatePickerTextBox Text="{Binding Date}" Width="600" Height="30" HorizontalAlignment="Center" VerticalAlignment="Center" Grid.Column="1" FontSize="20" FontFamily="Times New Roman"/>

<ComboBox x:Name="CBsubject\_code" SelectedItem="{Binding Subject\_code}" DisplayMemberPath="Subject\_code" Width="600" Height="30" HorizontalAlignment="Center" VerticalAlignment="Center" FontSize="20" FontFamily="Times New Roman" Grid.Row="1" Grid.Column="1"/>

<ComboBox x:Name="CBteachers" SelectedItem="{Binding Login}" DisplayMemberPath="Login" Width="600" Height="30" HorizontalAlignment="Center" VerticalAlignment="Center" FontSize="20" FontFamily="Times New Roman" Grid.Row="2" Grid.Column="1"/>

<TextBox Text="{Binding Replacing\_login}" Width="600" Height="30" HorizontalAlignment="Center" VerticalAlignment="Center" FontSize="20" FontFamily="Times New Roman" Grid.Row="3" Grid.Column="1"/>

<TextBox Text="{Binding Classes\_held}" Width="600" Height="30" HorizontalAlignment="Center" VerticalAlignment="Center" Grid.Row="4" Grid.Column="1" FontSize="20" FontFamily="Times New Roman"/>

<ComboBox x:Name="CBgroup\_code" SelectedItem="{Binding Group\_code}" DisplayMemberPath="Group\_code" Width="600" Height="30" HorizontalAlignment="Center" VerticalAlignment="Center" FontSize="20" FontFamily="Times New Roman" Grid.Row="5" Grid.Column="1"/>

<Button Content="Сохранить" Height="40" Width="200" HorizontalAlignment="Right" VerticalAlignment="Center" Grid.Row="6" FontSize="20" FontFamily="Times New Roman" BorderBrush="Black" Background="White" Margin="0,0,50,0" Click="Save\_Click" Grid.Column="1"/>

<Button Content="Назад" Height="40" Width="200" HorizontalAlignment="Left" VerticalAlignment="Center" Grid.Row="6" FontSize="20" FontFamily="Times New Roman" BorderBrush="Black" Background="White" Margin="50,0,0,0" Click="Go\_back\_Click" Grid.Column="0"/>

</Grid>

public partial class Adding\_Accounts : Window

{

readonly private Accounts accounts = new Accounts();

public Adding\_Accounts()

{

InitializeComponent();

WindowStartupLocation = WindowStartupLocation.CenterScreen;

DataContext = accounts;

CBsubject\_code.ItemsSource = Coursework1Entities.GetContext().Subjects.ToList();

CBteachers.ItemsSource = Coursework1Entities.GetContext().Teachers.ToList();

CBgroup\_code.ItemsSource = Coursework1Entities.GetContext().Groups.ToList();

}

private void Go\_back\_Click(object sender, RoutedEventArgs e)

{

this.Hide();

}

private void Save\_Click(object sender, RoutedEventArgs e)

{

StringBuilder errors = new StringBuilder();

if (accounts.Date == null)

errors.AppendLine("Пожалуйста, введите дату. Подсказка ММ.ДД.ГГ");

if ((CBsubject\_code == null) || (accounts.Subject\_code == null))

errors.AppendLine("Пожалуйста, выберите код предмета");

if ((CBteachers == null) || (accounts.Login == null))

errors.AppendLine("Пожалуйста, выберите логин преподавателя");

if ((accounts.Replacing\_login.Length > 50) || string.IsNullOrEmpty(accounts.Replacing\_login))

errors.AppendLine("Пожалуйста, напишите логин заменяющего преподавателя или напишите NULL, если вас никто не заменял. Максимальная длина 50 символов");

if (accounts.Classes\_held <= 0)

errors.AppendLine("Пожалуйста, введите количество проведенных занятий");

if ((CBgroup\_code == null) || (accounts.Group\_code == null))

errors.AppendLine("Пожалуйста, выберите код группы");

if (errors.Length > 0)

{

MessageBox.Show(errors.ToString(), "Ошибка", MessageBoxButton.OK, MessageBoxImage.Error);

return;

}

if (accounts.Account\_ID == 0)

{

Coursework1Entities.GetContext().Accounts.Add(accounts);

}

try

{

Coursework1Entities.GetContext().SaveChanges();

MessageBox.Show("Информация добавлена", "Отлично", MessageBoxButton.OK, MessageBoxImage.Information);

}

catch (Exception ex)

{

MessageBox.Show(ex.Message.ToString());

}

this.Hide();

}

}

Окно изменения учетов

Title="Changing\_Accounts" Height="550" Width="1100">

<Grid>

<Grid.ColumnDefinitions>

<ColumnDefinition Width="\*"/>

<ColumnDefinition Width="2\*"/>

</Grid.ColumnDefinitions>

<Grid.RowDefinitions>

<RowDefinition Height="\*"/>

<RowDefinition Height="\*"/>

<RowDefinition Height="\*"/>

<RowDefinition Height="\*"/>

<RowDefinition Height="\*"/>

<RowDefinition Height="\*"/>

<RowDefinition Height="\*"/>

</Grid.RowDefinitions>

<Label Content="Дата" FontSize="20" FontFamily="Times New Roman" HorizontalAlignment="Center" VerticalAlignment="Center" Grid.Row="0" Grid.Column="0"/>

<Label Content="Код предмета" FontSize="20" FontFamily="Times New Roman" HorizontalAlignment="Center" VerticalAlignment="Center" Grid.Row="1" Grid.Column="0"/>

<Label Content="Преподаватель" FontSize="20" FontFamily="Times New Roman" HorizontalAlignment="Center" VerticalAlignment="Center" Grid.Row="2" Grid.Column="0"/>

<Label Content="Преподаватель на замену" FontSize="20" FontFamily="Times New Roman" HorizontalAlignment="Center" VerticalAlignment="Center" Grid.Row="3" Grid.Column="0"/>

<Label Content="Проведенные пары" FontSize="20" FontFamily="Times New Roman" HorizontalAlignment="Center" VerticalAlignment="Center" Grid.Row="4" Grid.Column="0"/>

<Label Content="Код группы" FontSize="20" FontFamily="Times New Roman" HorizontalAlignment="Center" VerticalAlignment="Center" Grid.Row="5" Grid.Column="0"/>

<DatePickerTextBox Text="{Binding Date}" Width="600" Height="30" HorizontalAlignment="Center" VerticalAlignment="Center" Grid.Column="1" FontSize="20" FontFamily="Times New Roman"/>

<ComboBox x:Name="CBsubject\_code" Text="{Binding Subject\_code}" SelectedItem="{Binding Subject\_code}" DisplayMemberPath="Subject\_code" Width="600" Height="30" HorizontalAlignment="Center" VerticalAlignment="Center" FontSize="20" FontFamily="Times New Roman" Grid.Row="1" Grid.Column="1"/>

<ComboBox x:Name="CBteachers" Text="{Binding Login}" SelectedItem="{Binding Login}" DisplayMemberPath="Login" Width="600" Height="30" HorizontalAlignment="Center" VerticalAlignment="Center" FontSize="20" FontFamily="Times New Roman" Grid.Row="2" Grid.Column="1"/>

<TextBox Text="{Binding Replacing\_login}" Width="600" Height="30" HorizontalAlignment="Center" VerticalAlignment="Center" FontSize="20" FontFamily="Times New Roman" Grid.Row="3" Grid.Column="1"/>

<TextBox Text="{Binding Classes\_held}" Width="600" Height="30" HorizontalAlignment="Center" VerticalAlignment="Center" Grid.Row="4" Grid.Column="1" FontSize="20" FontFamily="Times New Roman"/>

<ComboBox x:Name="CBgroup\_code" Text="{Binding Group\_code}" SelectedItem="{Binding Group\_code}" DisplayMemberPath="Group\_code" Width="600" Height="30" HorizontalAlignment="Center" VerticalAlignment="Center" FontSize="20" FontFamily="Times New Roman" Grid.Row="5" Grid.Column="1"/>

<Button Content="Сохранить" Height="40" Width="200" HorizontalAlignment="Right" VerticalAlignment="Center" Grid.Row="6" FontSize="20" FontFamily="Times New Roman" BorderBrush="Black" Background="White" Margin="0,0,50,0" Click="Save\_Click" Grid.Column="1"/>

<Button Content="Назад" Height="40" Width="200" HorizontalAlignment="Left" VerticalAlignment="Center" Grid.Row="6" FontSize="20" FontFamily="Times New Roman" BorderBrush="Black" Background="White" Margin="50,0,0,0" Click="Go\_back\_Click" Grid.Column="0"/>

</Grid>

public partial class Changing\_Accounts : Window

{

readonly private Accounts accounts = new Accounts();

public Changing\_Accounts(Accounts selected\_row)

{

InitializeComponent();

WindowStartupLocation = WindowStartupLocation.CenterScreen;

if (selected\_row != null)

{

accounts = selected\_row;

}

DataContext = accounts;

CBsubject\_code.ItemsSource = Coursework1Entities.GetContext().Subjects.ToList();

CBteachers.ItemsSource = Coursework1Entities.GetContext().Teachers.ToList();

CBgroup\_code.ItemsSource = Coursework1Entities.GetContext().Groups.ToList();

}

private void Go\_back\_Click(object sender, RoutedEventArgs e)

{

this.Hide();

}

private void Save\_Click(object sender, RoutedEventArgs e)

{

StringBuilder errors = new StringBuilder();

if (accounts.Date == null)

errors.AppendLine("Пожалуйста, введите дату. Подсказка ММ.ДД.ГГ");

if ((CBsubject\_code == null) || (accounts.Subject\_code == null))

errors.AppendLine("Пожалуйста, выберите код предмета");

if ((CBteachers == null) || (accounts.Login == null))

errors.AppendLine("Пожалуйста, выберите логин преподавателя");

if (accounts.Replacing\_login.Length > 50)

errors.AppendLine("Пожалуйста, напишите логин заменяющего преподавателя или оставьте его пустым, если вас никто не заменял. Максимальная длина 50 символов");

if (accounts.Classes\_held <= 0)

errors.AppendLine("Пожалуйста, введите количество проведенных занятий");

if ((CBgroup\_code == null) || (accounts.Group\_code == null))

errors.AppendLine("Пожалуйста, выберите код группы");

if (errors.Length > 0)

{

MessageBox.Show(errors.ToString(), "Ошибка", MessageBoxButton.OK, MessageBoxImage.Error);

return;

}

if (accounts.Account\_ID == 0)

{

Coursework1Entities.GetContext().Accounts.Add(accounts);

}

try

{

Coursework1Entities.GetContext().SaveChanges();

MessageBox.Show("Информация изменена", "Отлично", MessageBoxButton.OK, MessageBoxImage.Information);

}

catch (Exception ex)

{

MessageBox.Show(ex.Message.ToString());

}

this.Hide();

}

}

# ПРИЛОЖЕНИЕ Г

Окно «Создание отчетов»

Title="Creating\_reports" Height="550" Width="1100">

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="1\*"/>

<RowDefinition Height="2\*"/>

<RowDefinition Height="1\*"/>

</Grid.RowDefinitions>

<Label x:Name="Lmain" Content="Создание отчетов и подсчёт проведенных и не проведенных часов" HorizontalAlignment="Center" VerticalAlignment="Center" FontSize="32" FontFamily="Times New Roman" Grid.Row="0"/>

<Label x:Name="Lfor\_necessary\_inf" Content="Выберите код предмета и группу, в которой вы вели предмет, чтобы в отчет вносить необходимые данные" HorizontalAlignment="Center" VerticalAlignment="Center" FontSize="20" FontFamily="Times New Roman" Grid.Row="0"/>

<Button Content="Назад" HorizontalAlignment="Left" Margin="20,0,0,20" VerticalAlignment="Bottom" Width="200" Height="40" FontSize="20" FontFamily="Times New Roman" BorderBrush="Black" Background="White" Click="Go\_back\_Click" Grid.Row="2"/>

<Button x:Name="Report\_like\_accounts" Content="Создать отчет о проведенных часах по дням преподавателем" HorizontalAlignment="Center" Margin="0,0,0,80" VerticalAlignment="Bottom" Width="550" Height="40" FontSize="20" FontFamily="Times New Roman" BorderBrush="Black" Background="White" Click="Create\_report1\_Click" Grid.Row="2"/>

<Button x:Name="Report\_about\_subjects" Content="Создать отчет о проведенных часах по предметам" HorizontalAlignment="Center" Margin="0,0,0,20" VerticalAlignment="Bottom" Width="450" Height="40" FontSize="20" FontFamily="Times New Roman" BorderBrush="Black" Background="White" Click="Create\_report2\_Click" Grid.Row="2"/>

<Button Content="Выход" HorizontalAlignment="Right" Margin="0,0,20,20" VerticalAlignment="Bottom" Width="200" Height="40" FontSize="20" FontFamily="Times New Roman" BorderBrush="Black" Background="White" Click="Exit\_Click" Grid.Row="2"/>

<Label x:Name="Lconducted\_hours" Content="Проведенные занятия" HorizontalAlignment="Left" VerticalAlignment="Top" FontSize="20" FontFamily="Times New Roman" Grid.Row="1" Margin="50,20,0,0"/>

<Label x:Name="Lnot\_conducted\_hours" Content="Не проведенные занятия" HorizontalAlignment="Right" VerticalAlignment="Top" FontSize="20" FontFamily="Times New Roman" Grid.Row="1" Margin="0,20,50,0"/>

<TextBlock x:Name="TBconducted\_hours" HorizontalAlignment="Left" VerticalAlignment="Center" Width="Auto" Height="Auto" FontSize="60" FontFamily="Times New Roman" Grid.Row="1" Margin="120,0,0,0"/>

<TextBlock x:Name="TBnot\_conducted\_hours" HorizontalAlignment="Right" VerticalAlignment="Center" Width="Auto" Height="Auto" FontSize="60" FontFamily="Times New Roman" Grid.Row="1" Margin="0,0,120,0"/>

<Label x:Name="Lsubject\_code" Content="Код предмета" HorizontalAlignment="Center" Margin="0,0,350,0" VerticalAlignment="Center" FontSize="20" FontFamily="Times New Roman" Grid.Row="1"/>

<Label x:Name="Lgroup\_code" Content="Код группы" HorizontalAlignment="Center" Margin="0,0,350,80" VerticalAlignment="Center" FontSize="20" FontFamily="Times New Roman" Grid.Row="1"/>

<Button x:Name="Bsubject" Content="Далее" Height="40" Width="550" HorizontalAlignment="Center" VerticalAlignment="Bottom" FontSize="20" FontFamily="Times New Roman" BorderBrush="Black" Background="White" Click="Login\_Button\_Click" Grid.Row="1"/>

<ComboBox x:Name="CBsubject\_code" SelectedItem="{Binding Subject\_code}" DisplayMemberPath="Subject\_code" HorizontalAlignment="Center" VerticalAlignment="Center" Height="30" Width="200" FontSize="22" FontFamily="Times New Roman" Grid.Row="1"/>

<ComboBox x:Name="CBgroup\_code" SelectedItem="{Binding Group\_code}" DisplayMemberPath="Group\_code" HorizontalAlignment="Center" Margin="0,0,0,80" VerticalAlignment="Center" Height="30" Width="200" FontSize="22" FontFamily="Times New Roman" Grid.Row="1"/>

</Grid>

public partial class Creating\_reports : Window

{

readonly private List<Curriculum> curriculum;

readonly private List<Accounts> accounts;

public Creating\_reports()

{

InitializeComponent();

WindowStartupLocation = WindowStartupLocation.CenterScreen;

Lmain.Visibility = Visibility.Hidden;

Lconducted\_hours.Visibility = Visibility.Hidden;

Lnot\_conducted\_hours.Visibility = Visibility.Hidden;

TBconducted\_hours.Visibility = Visibility.Hidden;

TBnot\_conducted\_hours.Visibility = Visibility.Hidden;

Report\_like\_accounts.Visibility = Visibility.Hidden;

Report\_about\_subjects.Visibility = Visibility.Hidden;

CBsubject\_code.ItemsSource = Coursework1Entities.GetContext().Subjects.ToList();

CBgroup\_code.ItemsSource = Coursework1Entities.GetContext().Groups.ToList();

curriculum = Coursework1Entities.GetContext().Curriculum.ToList();

accounts = Coursework1Entities.GetContext().Accounts.ToList();

}

private void Go\_back\_Click(object sender, RoutedEventArgs e)

{

this.Hide();

For\_Navigation for\_Navigation = new For\_Navigation();

for\_Navigation.ShowDialog();

}

private void Create\_report1\_Click(object sender, RoutedEventArgs e)

{

if (MessageBox.Show("Отчет создан! Посмотреть и распечатать?", "Информация", MessageBoxButton.YesNo, MessageBoxImage.Question) == MessageBoxResult.Yes)

{

this.Hide();

var subject = (Subjects)CBsubject\_code.SelectedItem;

Report\_like\_account report = new Report\_like\_account(subject.Subject\_code);

report.ShowDialog();

}

}

private void Create\_report2\_Click(object sender, RoutedEventArgs e)

{

if (MessageBox.Show("Отчет создан! Посмотреть и распечатать?", "Информация", MessageBoxButton.YesNo, MessageBoxImage.Question) == MessageBoxResult.Yes)

{

this.Hide();

var subject = (Subjects)CBsubject\_code.SelectedItem;

Report\_for\_subjects report = new Report\_for\_subjects(subject.Subject\_code);

report.ShowDialog();

}

}

private void Exit\_Click(object sender, RoutedEventArgs e)

{

Application.Current.Shutdown();

}

private void Login\_Button\_Click(object sender, RoutedEventArgs e)

{

if (CBgroup\_code.SelectedItem != null && CBsubject\_code.SelectedItem != null)

{

var subject = (Subjects)CBsubject\_code.SelectedItem;

var group = (Groups)CBgroup\_code.SelectedItem;

var ob = accounts.Where(a => a.Group\_code == group.Group\_code && a.Subject\_code == subject.Subject\_code);

var obj = curriculum.Where(c => c.Group\_code == group.Group\_code && c.Subject\_code == subject.Subject\_code);

int Total\_number\_of\_hours = 0;

foreach (var item in obj)

{

Total\_number\_of\_hours += (int)item.Total\_number\_of\_hours;

}

int Classes\_held = 0;

foreach (var item in ob)

{

Classes\_held += (int)item.Classes\_held;

}

Lmain.Visibility = Visibility.Visible;

Lconducted\_hours.Visibility = Visibility.Visible;

Lnot\_conducted\_hours.Visibility = Visibility.Visible;

TBconducted\_hours.Visibility = Visibility.Visible;

TBnot\_conducted\_hours.Visibility = Visibility.Visible;

Report\_like\_accounts.Visibility = Visibility.Visible;

Report\_about\_subjects.Visibility = Visibility.Visible;

Lfor\_necessary\_inf.Visibility = Visibility.Hidden;

Lsubject\_code.Visibility = Visibility.Hidden;

Lgroup\_code.Visibility = Visibility.Hidden;

CBgroup\_code.Visibility = Visibility.Hidden;

CBsubject\_code.Visibility = Visibility.Hidden;

Bsubject.Visibility = Visibility.Hidden;

int not\_conductedhours = Math.Abs((Total\_number\_of\_hours / 2) - Classes\_held);

TBnot\_conducted\_hours.Text = not\_conductedhours.ToString();

int conductedhours = Math.Abs((Total\_number\_of\_hours / 2) - not\_conductedhours);

TBconducted\_hours.Text = conductedhours.ToString();

}

else MessageBox.Show("Не оставляйте поля пустыми. Вводите существующие коды предмета и группы, которых вы ведете. Попробуйте снова.", "Ошибка", MessageBoxButton.OK, MessageBoxImage.Error);

}

}

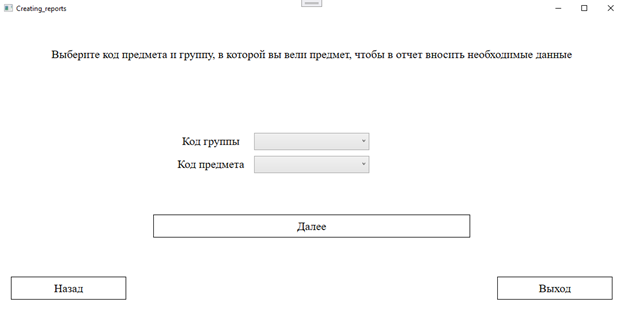


Рисунок .1 - Окно «Создание отчетов» первая форма

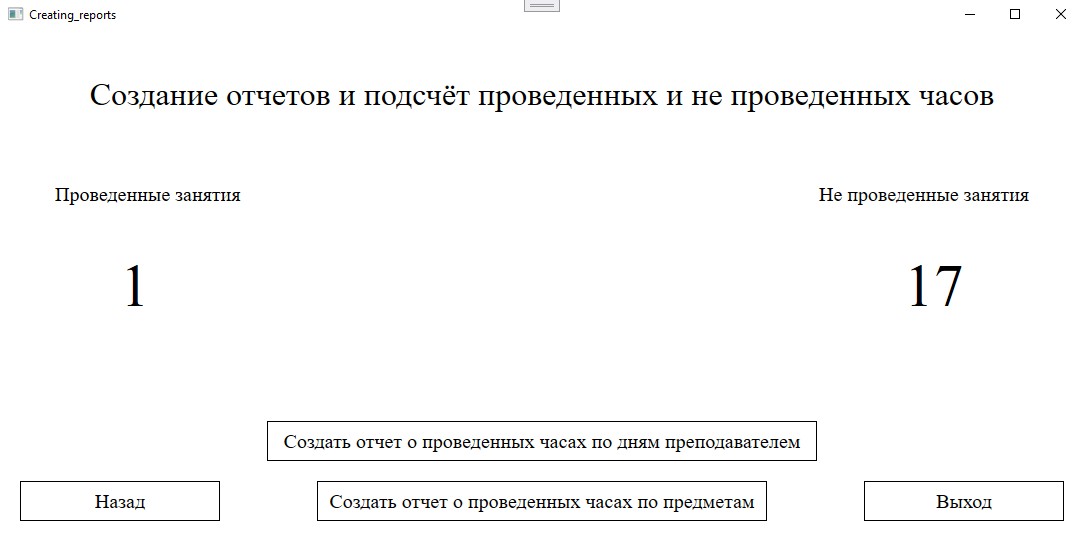


Рисунок 13.2 - Окно «Создание отчетов» вторая форма

# ПРИЛОЖЕНИЕ Д

Окно «Для перемещения 2»

Title="To\_the\_necessary\_inf" Height="550" Width="1100">

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="\*"/>

<RowDefinition Height="\*"/>

<RowDefinition Height="\*"/>

</Grid.RowDefinitions>

<Grid.ColumnDefinitions>

<ColumnDefinition Width="\*"/>

<ColumnDefinition Width="\*"/>

</Grid.ColumnDefinitions>

<Button Content="Назад" HorizontalAlignment="Left" Margin="20,0,0,20" VerticalAlignment="Bottom" Width="200" Height="40" FontSize="22" FontFamily="Times New Roman" BorderBrush="Black" Background="White" Click="Go\_back\_Click" Grid.Column="0" Grid.Row="2"/>

<Button Content="Выход" HorizontalAlignment="Right" Margin="0,0,20,20" VerticalAlignment="Bottom" Width="200" Height="40" FontSize="22" FontFamily="Times New Roman" BorderBrush="Black" Background="White" Click="Exit\_Click" Grid.Column="1" Grid.Row="2"/>

<Button Content="Специальности" Width="300" Height="50" FontSize="32" FontFamily="Times New Roman" BorderBrush="Black" Background="White" Grid.Column="0" Grid.Row="0" Click="To\_specialties\_Click"/>

<Button Content="Учебные планы" Width="300" Height="50" FontSize="32" FontFamily="Times New Roman" BorderBrush="Black" Background="White" Grid.Column="1" Grid.Row="0" Click="To\_curriculum\_Click"/>

<Button Content="Семестры" Width="300" Height="50" FontSize="32" FontFamily="Times New Roman" BorderBrush="Black" Background="White" Grid.Column="0" Grid.Row="1" Click="To\_semesters\_Click"/>

<Button Content="Предметы" Width="300" Height="50" FontSize="32" FontFamily="Times New Roman" BorderBrush="Black" Background="White" Grid.Column="1" Grid.Row="1" Click="To\_subjects\_Click"/>

<Button Content="Группы" Width="300" Height="50" FontSize="32" FontFamily="Times New Roman" BorderBrush="Black" Background="White" Grid.ColumnSpan="2" Grid.Row="2" Click="To\_groups\_Click"/>

</Grid>

public partial class To\_the\_necessary\_inf : Window

{

public To\_the\_necessary\_inf()

{

InitializeComponent();

WindowStartupLocation = WindowStartupLocation.CenterScreen;

}

private void To\_specialties\_Click(object sender, RoutedEventArgs e)

{

this.Hide();

Specialties1 specialties1 = new Specialties1();

specialties1.ShowDialog();

}

private void To\_curriculum\_Click(object sender, RoutedEventArgs e)

{

this.Hide();

Curriculums1 curriculums1 = new Curriculums1();

curriculums1.ShowDialog();

}

private void To\_semesters\_Click(object sender, RoutedEventArgs e)

{

this.Hide();

Semesters1 semesters1 = new Semesters1();

semesters1.ShowDialog();

}

private void To\_subjects\_Click(object sender, RoutedEventArgs e)

{

this.Hide();

Subjects1 subjects1 = new Subjects1();

subjects1.ShowDialog();

}

private void To\_groups\_Click(object sender, RoutedEventArgs e)

{

this.Hide();

Groups1 groups1 = new Groups1();

groups1.ShowDialog();

}

private void Go\_back\_Click(object sender, RoutedEventArgs e)

{

this.Hide();

For\_Navigation for\_Navigation = new For\_Navigation();

for\_Navigation.ShowDialog();

}

private void Exit\_Click(object sender, RoutedEventArgs e)

{

Application.Current.Shutdown();

}

}



Рисунок - Окно «Для перемещения 2»

# ПРИЛОЖЕНИЕ Е

Окно «Преподаватели»

Title="Teachers1" Height="550" Width="1100" IsVisibleChanged="Window\_IsVisibleChanged">

<Grid>

<Grid.ColumnDefinitions>

<ColumnDefinition Width="\*"/>

<ColumnDefinition Width="\*"/>

<ColumnDefinition Width="\*"/>

<ColumnDefinition Width="\*"/>

</Grid.ColumnDefinitions>

<Grid.RowDefinitions>

<RowDefinition Height="4\*"/>

<RowDefinition Height="\*"/>

</Grid.RowDefinitions>

<Button Content="Назад" Width="200" Height="40" FontSize="20" FontFamily="Times New Roman" BorderBrush="Black" Background="White" Click="Go\_back\_Click" Grid.Row="1" Grid.Column="0"/>

<Button Content="Выход" Width="200" Height="40" FontSize="20" FontFamily="Times New Roman" BorderBrush="Black" Background="White" Click="Exit\_Click" Grid.Row="1" Grid.Column="3"/>

<Button Content="Удалить" Width="200" Height="40" FontSize="20" FontFamily="Times New Roman" BorderBrush="Black" Background="White" Grid.Row="1" Click="Delete\_Click" Grid.Column="1"/>

<Button Content="Добавить" Width="200" Height="40" FontSize="20" FontFamily="Times New Roman" BorderBrush="Black" Background="White" Grid.Row="1" Click="Add\_Click" Grid.Column="2"/>

<DataGrid x:Name="Grid1" AutoGenerateColumns="False" Background="White" FontSize="14" FontFamily="Times New Roman" Grid.Row="0" Grid.ColumnSpan="4">

<DataGrid.Columns>

<DataGridTextColumn Header="Логин" Binding="{Binding Login}" Width="\*"></DataGridTextColumn>

<DataGridTextColumn Header="Фамилия" Binding="{Binding Surname}" Width="\*"></DataGridTextColumn>

<DataGridTextColumn Header="Имя" Binding="{Binding Name}" Width="\*"></DataGridTextColumn>

<DataGridTextColumn Header="Отчество" Binding="{Binding Patronymic}" Width="\*"></DataGridTextColumn>

<DataGridTextColumn Header="Пароль" Binding="{Binding Password}" Width="\*"></DataGridTextColumn>

<DataGridTextColumn Header="Пол" Binding="{Binding Gender}" Width="Auto"></DataGridTextColumn>

<DataGridTextColumn Header="День рождения" Binding="{Binding Birthday}" Width="\*"></DataGridTextColumn>

<DataGridTextColumn Header="Телефон" Binding="{Binding Phone\_number}" Width="\*"></DataGridTextColumn>

<DataGridTextColumn Header="Почта" Binding="{Binding Mail}" Width="\*"></DataGridTextColumn>

<DataGridTemplateColumn Width="100">

<DataGridTemplateColumn.CellTemplate>

<DataTemplate>

<Button Content="Изменить" FontFamily="Times New Roman" BorderBrush="Black" Background="White" Click="Edit\_Click"/>

</DataTemplate>

</DataGridTemplateColumn.CellTemplate>

</DataGridTemplateColumn>

</DataGrid.Columns>

</DataGrid>

</Grid>

public partial class Teachers1 : Window

{

public Teachers1()

{

InitializeComponent();

WindowStartupLocation = WindowStartupLocation.CenterScreen;

}

private void Edit\_Click(object sender, RoutedEventArgs e)

{

Changing\_Teachers teachers = new Changing\_Teachers((sender as Button).DataContext as Teachers);

teachers.ShowDialog();

}

private void Add\_Click(object sender, RoutedEventArgs e)

{

Adding\_Teachers teachers = new Adding\_Teachers();

teachers.ShowDialog();

}

private void Delete\_Click(object sender, RoutedEventArgs e)

{

var Teachers\_For\_Removing = Grid1.SelectedItems.Cast<Teachers>().ToList();

if (MessageBox.Show($"Вы уверены, что хотите удалить следующий(е) {Teachers\_For\_Removing.Count()} элемент или элементы?", "Вопрос", MessageBoxButton.YesNo, MessageBoxImage.Question) == MessageBoxResult.Yes)

{

try

{

Coursework1Entities.GetContext().Teachers.RemoveRange(Teachers\_For\_Removing);

Coursework1Entities.GetContext().SaveChanges();

MessageBox.Show("Информация удалена", "Отлично", MessageBoxButton.OK, MessageBoxImage.Information);

}

catch (Exception ex)

{

MessageBox.Show(ex.Message.ToString());

}

}

}

private void Go\_back\_Click(object sender, RoutedEventArgs e)

{

this.Hide();

For\_Navigation for\_Navigation = new For\_Navigation();

for\_Navigation.ShowDialog();

}

private void Exit\_Click(object sender, RoutedEventArgs e)

{

Application.Current.Shutdown();

}

private void Window\_IsVisibleChanged(object sender, DependencyPropertyChangedEventArgs e)

{

if (Visibility == Visibility.Visible)

{

Coursework1Entities.GetContext().ChangeTracker.Entries().ToList().ForEach(p => p.Reload());

Grid1.ItemsSource = Coursework1Entities.GetContext().Teachers.ToList();

}

}

}

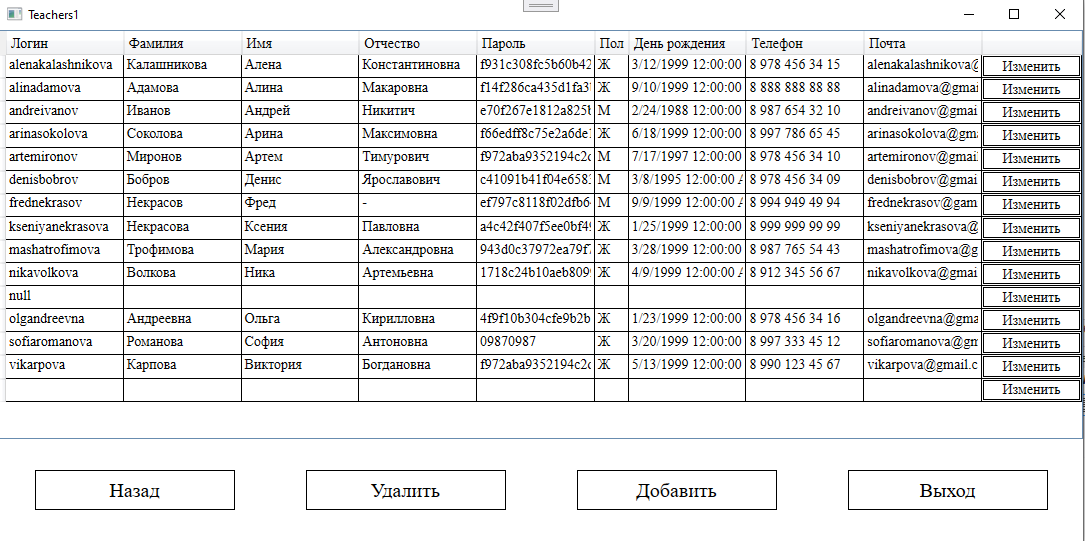


Рисунок - Окно «Преподаватели»

Окно «Добавление преподавателей»

Title="Adding\_Teachers" Height="550" Width="1100">

<Grid>

<Grid.ColumnDefinitions>

<ColumnDefinition Width="\*"/>

<ColumnDefinition Width="2\*"/>

</Grid.ColumnDefinitions>

<Grid.RowDefinitions>

<RowDefinition Height="\*"/>

<RowDefinition Height="\*"/>

<RowDefinition Height="\*"/>

<RowDefinition Height="\*"/>

<RowDefinition Height="\*"/>

<RowDefinition Height="\*"/>

<RowDefinition Height="\*"/>

<RowDefinition Height="\*"/>

<RowDefinition Height="\*"/>

<RowDefinition Height="60"/>

</Grid.RowDefinitions>

<Label Content="Логин преподавателя" FontSize="20" FontFamily="Times New Roman" HorizontalAlignment="Center" VerticalAlignment="Center" Grid.Column="0" Grid.Row="0"/>

<Label Content="Фамилия преподавателя" FontSize="20" FontFamily="Times New Roman" HorizontalAlignment="Center" VerticalAlignment="Center" Grid.Column="0" Grid.Row="1"/>

<Label Content="Имя преподавателя" FontSize="20" FontFamily="Times New Roman" HorizontalAlignment="Center" VerticalAlignment="Center" Grid.Column="0" Grid.Row="2"/>

<Label Content="Отчество преподавателя" FontSize="20" FontFamily="Times New Roman" HorizontalAlignment="Center" VerticalAlignment="Center" Grid.Column="0" Grid.Row="3"/>

<Label Content="Пароль преподавателя" FontSize="20" FontFamily="Times New Roman" HorizontalAlignment="Center" VerticalAlignment="Center" Grid.Column="0" Grid.Row="4"/>

<Label Content="Пол преподавателя" FontSize="20" FontFamily="Times New Roman" HorizontalAlignment="Center" VerticalAlignment="Center" Grid.Column="0" Grid.Row="5"/>

<Label Content="День рождение преподавателя" FontSize="20" FontFamily="Times New Roman" HorizontalAlignment="Center" VerticalAlignment="Center" Grid.Column="0" Grid.Row="6"/>

<Label Content="Телефон преподавателя" FontSize="20" FontFamily="Times New Roman" HorizontalAlignment="Center" VerticalAlignment="Center" Grid.Column="0" Grid.Row="7"/>

<Label Content="Почта преподавателя" FontSize="20" FontFamily="Times New Roman" HorizontalAlignment="Center" VerticalAlignment="Center" Grid.Column="0" Grid.Row="8"/>

<TextBox Text="{Binding Login}" Width="600" Height="30" HorizontalAlignment="Center" VerticalAlignment="Center" FontSize="20" FontFamily="Times New Roman" Grid.Row="0" Grid.Column="1"/>

<TextBox Text="{Binding Surname}" Width="600" Height="30" HorizontalAlignment="Center" VerticalAlignment="Center" FontSize="20" FontFamily="Times New Roman" Grid.Row="1" Grid.Column="1"/>

<TextBox Text="{Binding Name}" Width="600" Height="30" HorizontalAlignment="Center" VerticalAlignment="Center" FontSize="20" FontFamily="Times New Roman" Grid.Row="2" Grid.Column="1"/>

<TextBox Text="{Binding Patronymic}" Width="600" Height="30" HorizontalAlignment="Center" VerticalAlignment="Center" FontSize="20" FontFamily="Times New Roman" Grid.Row="3" Grid.Column="1"/>

<TextBox x:Name="TBpassword" Text="{Binding Password}" Width="600" Height="30" HorizontalAlignment="Center" VerticalAlignment="Center" FontSize="20" FontFamily="Times New Roman" Grid.Row="4" Grid.Column="1"/>

<TextBox Text="{Binding Gender}" Width="600" Height="30" HorizontalAlignment="Center" VerticalAlignment="Center" FontSize="20" FontFamily="Times New Roman" Grid.Row="5" Grid.Column="1"/>

<DatePickerTextBox Text="{Binding Birthday}" Width="600" Height="30" HorizontalAlignment="Center" VerticalAlignment="Center" FontSize="20" FontFamily="Times New Roman" Grid.Column="1" Grid.Row="6"/>

<TextBox Text="{Binding Phone\_number}" Width="600" Height="30" HorizontalAlignment="Center" VerticalAlignment="Center" FontSize="20" FontFamily="Times New Roman" Grid.Row="7" Grid.Column="1"/>

<TextBox Text="{Binding Mail}" Width="600" Height="30" HorizontalAlignment="Center" VerticalAlignment="Center" FontSize="20" FontFamily="Times New Roman" Grid.Row="8" Grid.Column="1"/>

<Button Content="Сохранить" HorizontalAlignment="Right" VerticalAlignment="Center" Width="200" Height="40" Click="Save\_Click" FontSize="20" FontFamily="Times New Roman" Grid.Row="9" Grid.Column="1" Margin="0,0,50,0" BorderBrush="Black" Background="White"/>

<Button Content="Назад" HorizontalAlignment="Left" VerticalAlignment="Center" Width="200" Height="40" Click="Go\_back\_Click" FontSize="20" FontFamily="Times New Roman" Grid.Row="9" Grid.Column="0" Margin="50,0,0,0" BorderBrush="Black" Background="White"/>

</Grid>

public partial class Adding\_Teachers : Window

{

readonly private Teachers teachers = new Teachers();

readonly private Password\_hashing password = new Password\_hashing();

public Adding\_Teachers()

{

InitializeComponent();

WindowStartupLocation = WindowStartupLocation.CenterScreen;

DataContext = teachers;

}

private void Go\_back\_Click(object sender, RoutedEventArgs e)

{

this.Hide();

}

private void Save\_Click(object sender, RoutedEventArgs e)

{

StringBuilder errors = new StringBuilder();

if (string.IsNullOrWhiteSpace(teachers.Login) || (teachers.Login.Length > 50))

errors.AppendLine("Пожалуйста, напишите логин преподавателя. Максимальная длина 50 символов");

if (string.IsNullOrWhiteSpace(teachers.Surname) || (teachers.Surname.Length > 50))

errors.AppendLine("Пожалуйста, напишите фамилию преподавателя. Максимальная длина 50 символов");

if (string.IsNullOrWhiteSpace(teachers.Name) || (teachers.Name.Length > 50))

errors.AppendLine("Пожалуйста, напишите имя преподавателя. Максимальная длина 50 символов");

if (string.IsNullOrWhiteSpace(teachers.Patronymic) || (teachers.Patronymic.Length > 50))

errors.AppendLine("Пожалуйста, напишите отчество преподавателя. Максимальная длина 50 символов");

if (string.IsNullOrWhiteSpace(teachers.Password) || (teachers.Password.Length < 4) || (teachers.Password.Length > 8))

errors.AppendLine("Пожалуйста, введите пароль преподавателя. Минимальная длина 4, а максимальная 8");

else

{

TBpassword.Text = password.Sha256(teachers.Password);

teachers.Password = TBpassword.Text;

}

if (string.IsNullOrWhiteSpace(teachers.Gender) || (teachers.Gender.Length != 1))

errors.AppendLine("Пожалуйста, напишите пол преподавателя. Максимальная длина 1 символ");

DateTime minDate = new DateTime(1950, 1, 1);

DateTime maxDate = new DateTime(2000, 1, 1);

if ((teachers.Birthday == null) || (teachers.Birthday < minDate) || (teachers.Birthday > maxDate))

errors.AppendLine("Пожалуйста, напишите день рождения преподавателя. Минимально вводимая дата 01.01.1950, а максимальная 01.01.2000");

if (string.IsNullOrWhiteSpace(teachers.Phone\_number) || (teachers.Phone\_number.Length > 15))

errors.AppendLine("Пожалуйста, напишите телефон преподавателя. Максимальная длина 15 символов");

if (string.IsNullOrWhiteSpace(teachers.Mail) || (teachers.Mail.Length > 50))

errors.AppendLine("Пожалуйста, напишите почту преподавателя. Максимальная длина 50 символов");

if (errors.Length > 0)

{

MessageBox.Show(errors.ToString(), "Ошибка", MessageBoxButton.OK, MessageBoxImage.Error);

return;

}

if (teachers.Login.Length > 0)

{

Coursework1Entities.GetContext().Teachers.Add(teachers);

}

try

{

password.Sha256(teachers.Password);

Coursework1Entities.GetContext().SaveChanges();

MessageBox.Show("Информация добавлена", "Отлично", MessageBoxButton.OK, MessageBoxImage.Information);

}

catch (Exception ex)

{

MessageBox.Show(ex.Message.ToString());

}

this.Hide();

}

}

Окно «Изменение преподавателей»

Title="Changing\_Teachers" Height="550" Width="1100">

<Grid>

<Grid.ColumnDefinitions>

<ColumnDefinition Width="\*"/>

<ColumnDefinition Width="2\*"/>

</Grid.ColumnDefinitions>

<Grid.RowDefinitions>

<RowDefinition Height="\*"/>

<RowDefinition Height="\*"/>

<RowDefinition Height="\*"/>

<RowDefinition Height="\*"/>

<RowDefinition Height="\*"/>

<RowDefinition Height="\*"/>

<RowDefinition Height="\*"/>

<RowDefinition Height="\*"/>

<RowDefinition Height="\*"/>

<RowDefinition Height="60"/>

</Grid.RowDefinitions>

<Label Content="Логин преподавателя" FontSize="20" FontFamily="Times New Roman" HorizontalAlignment="Center" VerticalAlignment="Center" Grid.Column="0" Grid.Row="0"/>

<Label Content="Фамилия преподавателя" FontSize="20" FontFamily="Times New Roman" HorizontalAlignment="Center" VerticalAlignment="Center" Grid.Column="0" Grid.Row="1"/>

<Label Content="Имя преподавателя" FontSize="20" FontFamily="Times New Roman" HorizontalAlignment="Center" VerticalAlignment="Center" Grid.Column="0" Grid.Row="2"/>

<Label Content="Отчество преподавателя" FontSize="20" FontFamily="Times New Roman" HorizontalAlignment="Center" VerticalAlignment="Center" Grid.Column="0" Grid.Row="3"/>

<Label Content="Пароль преподавателя" FontSize="20" FontFamily="Times New Roman" HorizontalAlignment="Center" VerticalAlignment="Center" Grid.Column="0" Grid.Row="4"/>

<Label Content="Пол преподавателя" FontSize="20" FontFamily="Times New Roman" HorizontalAlignment="Center" VerticalAlignment="Center" Grid.Column="0" Grid.Row="5"/>

<Label Content="День рождение преподавателя" FontSize="20" FontFamily="Times New Roman" HorizontalAlignment="Center" VerticalAlignment="Center" Grid.Column="0" Grid.Row="6"/>

<Label Content="Телефон преподавателя" FontSize="20" FontFamily="Times New Roman" HorizontalAlignment="Center" VerticalAlignment="Center" Grid.Column="0" Grid.Row="7"/>

<Label Content="Почта преподавателя" FontSize="20" FontFamily="Times New Roman" HorizontalAlignment="Center" VerticalAlignment="Center" Grid.Column="0" Grid.Row="8"/>

<TextBox Text="{Binding Login}" Width="600" Height="30" HorizontalAlignment="Center" VerticalAlignment="Center" FontSize="20" FontFamily="Times New Roman" Grid.Row="0" Grid.Column="1"/>

<TextBox Text="{Binding Surname}" Width="600" Height="30" HorizontalAlignment="Center" VerticalAlignment="Center" FontSize="20" FontFamily="Times New Roman" Grid.Row="1" Grid.Column="1"/>

<TextBox Text="{Binding Name}" Width="600" Height="30" HorizontalAlignment="Center" VerticalAlignment="Center" FontSize="20" FontFamily="Times New Roman" Grid.Row="2" Grid.Column="1"/>

<TextBox Text="{Binding Patronymic}" Width="600" Height="30" HorizontalAlignment="Center" VerticalAlignment="Center" FontSize="20" FontFamily="Times New Roman" Grid.Row="3" Grid.Column="1"/>

<TextBox x:Name="TBpassword" Text="{Binding Password}" Width="600" Height="30" HorizontalAlignment="Center" VerticalAlignment="Center" FontSize="20" FontFamily="Times New Roman" Grid.Row="4" Grid.Column="1"/>

<TextBox Text="{Binding Gender}" Width="600" Height="30" HorizontalAlignment="Center" VerticalAlignment="Center" FontSize="20" FontFamily="Times New Roman" Grid.Row="5" Grid.Column="1"/>

<DatePickerTextBox Text="{Binding Birthday}" Width="600" Height="30" HorizontalAlignment="Center" VerticalAlignment="Center" FontSize="20" FontFamily="Times New Roman" Grid.Column="1" Grid.Row="6"/>

<TextBox Text="{Binding Phone\_number}" Width="600" Height="30" HorizontalAlignment="Center" VerticalAlignment="Center" FontSize="20" FontFamily="Times New Roman" Grid.Row="7" Grid.Column="1"/>

<TextBox Text="{Binding Mail}" Width="600" Height="30" HorizontalAlignment="Center" VerticalAlignment="Center" FontSize="20" FontFamily="Times New Roman" Grid.Row="8" Grid.Column="1"/>

<Button Content="Сохранить" HorizontalAlignment="Right" VerticalAlignment="Center" Width="200" Height="40" Click="Save\_Click" FontSize="20" FontFamily="Times New Roman" Grid.Row="9" Grid.Column="1" Margin="0,0,50,0" BorderBrush="Black" Background="White"/>

<Button Content="Назад" HorizontalAlignment="Left" VerticalAlignment="Center" Width="200" Height="40" Click="Go\_back\_Click" FontSize="20" FontFamily="Times New Roman" Grid.Row="9" Grid.Column="0" Margin="50,0,0,0" BorderBrush="Black" Background="White"/>

</Grid>

public partial class Changing\_Teachers : Window

{

readonly private Teachers teachers = new Teachers();

readonly private Password\_hashing password = new Password\_hashing();

public Changing\_Teachers(Teachers selected\_row)

{

InitializeComponent();

WindowStartupLocation = WindowStartupLocation.CenterScreen;

if (selected\_row != null)

{

teachers = selected\_row;

}

DataContext = teachers;

}

private void Go\_back\_Click(object sender, RoutedEventArgs e)

{

this.Hide();

}

private void Save\_Click(object sender, RoutedEventArgs e)

{

StringBuilder errors = new StringBuilder();

if (string.IsNullOrWhiteSpace(teachers.Login) || (teachers.Login.Length > 50))

errors.AppendLine("Пожалуйста, напишите логин преподавателя. Максимальная длина 50 символов");

if (string.IsNullOrWhiteSpace(teachers.Surname) || (teachers.Surname.Length > 50))

errors.AppendLine("Пожалуйста, напишите фамилию преподавателя. Максимальная длина 50 символов");

if (string.IsNullOrWhiteSpace(teachers.Name) || (teachers.Name.Length > 50))

errors.AppendLine("Пожалуйста, напишите имя преподавателя. Максимальная длина 50 символов");

if (string.IsNullOrWhiteSpace(teachers.Patronymic) || (teachers.Patronymic.Length > 50))

errors.AppendLine("Пожалуйста, напишите отчество преподавателя. Максимальная длина 50 символов");

if (string.IsNullOrWhiteSpace(teachers.Password) || (teachers.Password.Length < 4) || (teachers.Password.Length > 8))

errors.AppendLine("Пожалуйста, введите пароль преподавателя. Минимальная длина 4, а максимальная 8");

else

{

TBpassword.Text = password.Sha256(teachers.Password);

teachers.Password = TBpassword.Text;

}

if (string.IsNullOrWhiteSpace(teachers.Gender) || (teachers.Gender.Length != 1))

errors.AppendLine("Пожалуйста, напишите пол преподавателя. Максимальная длина 1 символ");

DateTime minDate = new DateTime(1950, 1, 1);

DateTime maxDate = new DateTime(2000, 1, 1);

if ((teachers.Birthday == null) || (teachers.Birthday < minDate) || (teachers.Birthday > maxDate))

errors.AppendLine("Пожалуйста, напишите день рождения преподавателя. Минимально вводимая дата 01.01.1950, а максимальная 01.01.2000");

if (string.IsNullOrWhiteSpace(teachers.Phone\_number) || (teachers.Phone\_number.Length > 15))

errors.AppendLine("Пожалуйста, напишите телефон преподавателя. Максимальная длина 15 символов");

if (string.IsNullOrWhiteSpace(teachers.Mail) || (teachers.Mail.Length > 50))

errors.AppendLine("Пожалуйста, напишите почту преподавателя. Максимальная длина 50 символов");

if (errors.Length > 0)

{

MessageBox.Show(errors.ToString(), "Ошибка", MessageBoxButton.OK, MessageBoxImage.Error);

return;

}

if (teachers.Login == null)

{

Coursework1Entities.GetContext().Teachers.Add(teachers);

}

try

{

Coursework1Entities.GetContext().SaveChanges();

MessageBox.Show("Информация изменена", "Отлично", MessageBoxButton.OK, MessageBoxImage.Information);

}

catch (Exception ex)

{

MessageBox.Show(ex.Message.ToString());

}

this.Hide();

}

}

# ПРИЛОЖЕНИЕ Ж

Окно «Специальности»

Title="Specialties1" Height="550" Width="1100" IsVisibleChanged="Window\_IsVisibleChanged">

<Grid>

<Grid.ColumnDefinitions>

<ColumnDefinition Width="\*"/>

<ColumnDefinition Width="\*"/>

<ColumnDefinition Width="\*"/>

</Grid.ColumnDefinitions>

<Grid.RowDefinitions>

<RowDefinition Height="4\*"/>

<RowDefinition Height="\*"/>

</Grid.RowDefinitions>

<Button Content="Добавить" Width="200" Height="40" FontSize="20" FontFamily="Times New Roman" BorderBrush="Black" Background="White" Click="Add\_Click" Grid.Row="1" Grid.Column="1"/>

<Button Content="Назад" Width="200" Height="40" FontSize="20" FontFamily="Times New Roman" BorderBrush="Black" Background="White" Click="Go\_back\_Click" Grid.Row="1" Grid.Column="0"/>

<Button Content="Выход" Width="200" Height="40" FontSize="20" FontFamily="Times New Roman" BorderBrush="Black" Background="White" Click="Exit\_Click" Grid.Row="1" Grid.Column="2"/>

<DataGrid x:Name="Grid1" AutoGenerateColumns="False" Background="White" FontFamily="Times New Roman" FontSize="14" Grid.Row="0" Grid.ColumnSpan="3">

<DataGrid.Columns>

<DataGridTextColumn Header="Код специальности" Binding="{Binding Specialty\_code}" Width="Auto"></DataGridTextColumn>

<DataGridTextColumn Header="Наименование специальности" Binding="{Binding Name\_of\_the\_specialty}" Width="\*"></DataGridTextColumn>

<DataGridTextColumn Header="Форма обучения" Binding="{Binding Form\_of\_study}" Width="Auto"></DataGridTextColumn>

<DataGridTextColumn Header="Квалификация" Binding="{Binding Qualification}" Width="\*"></DataGridTextColumn>

<DataGridTemplateColumn Width="100">

<DataGridTemplateColumn.CellTemplate>

<DataTemplate>

<Button Content="Изменить" FontFamily="Times New Roman" BorderBrush="Black" Background="White" Click="Edit\_Click"/>

</DataTemplate>

</DataGridTemplateColumn.CellTemplate>

</DataGridTemplateColumn>

</DataGrid.Columns>

</DataGrid>

</Grid>

public partial class Specialties1 : Window

{

public Specialties1()

{

InitializeComponent();

WindowStartupLocation = WindowStartupLocation.CenterScreen;

}

private void Edit\_Click(object sender, RoutedEventArgs e)

{

Changing\_Specialties specialties = new Changing\_Specialties((sender as Button).DataContext as Specialties);

specialties.ShowDialog();

}

private void Add\_Click(object sender, RoutedEventArgs e)

{

Adding\_Specialties specialties = new Adding\_Specialties();

specialties.ShowDialog();

}

private void Go\_back\_Click(object sender, RoutedEventArgs e)

{

this.Hide();

To\_the\_necessary\_inf inf = new To\_the\_necessary\_inf();

inf.ShowDialog();

}

private void Exit\_Click(object sender, RoutedEventArgs e)

{

Application.Current.Shutdown();

}

private void Window\_IsVisibleChanged(object sender, DependencyPropertyChangedEventArgs e)

{

if (Visibility == Visibility.Visible)

{

Coursework1Entities.GetContext().ChangeTracker.Entries().ToList().ForEach(p => p.Reload());

Grid1.ItemsSource = Coursework1Entities.GetContext().Specialties.ToList();

}

}

}

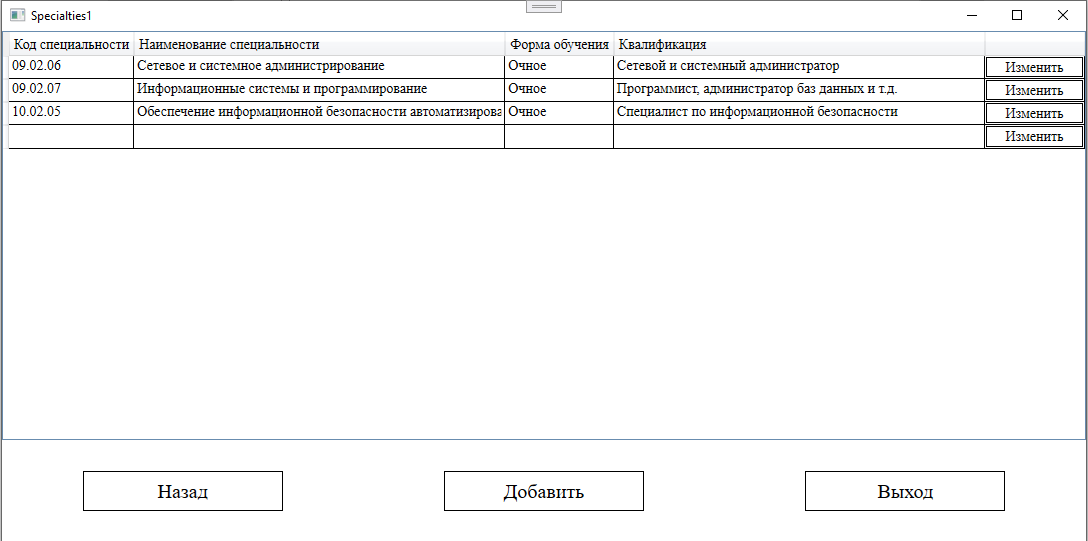


Рисунок - Окно «Специальности»

Окно «Добавление специальностей»

Title="Adding\_Specialties" Height="550" Width="1100">

<Grid>

<Grid.ColumnDefinitions>

<ColumnDefinition Width="\*"/>

<ColumnDefinition Width="2\*"/>

</Grid.ColumnDefinitions>

<Grid.RowDefinitions>

<RowDefinition Height="\*"/>

<RowDefinition Height="\*"/>

<RowDefinition Height="\*"/>

<RowDefinition Height="\*"/>

<RowDefinition Height="\*"/>

</Grid.RowDefinitions>

<Label Content="Код специальности" FontSize="20" FontFamily="Times New Roman" HorizontalAlignment="Center" VerticalAlignment="Center" Grid.Column="0" Grid.Row="0"/>

<Label Content="Название специальности" FontSize="20" FontFamily="Times New Roman" HorizontalAlignment="Center" VerticalAlignment="Center" Grid.Column="0" Grid.Row="1"/>

<Label Content="Форма обучения" FontSize="20" FontFamily="Times New Roman" HorizontalAlignment="Center" VerticalAlignment="Center" Grid.Column="0" Grid.Row="2"/>

<Label Content="Квалификация" FontSize="20" FontFamily="Times New Roman" HorizontalAlignment="Center" VerticalAlignment="Center" Grid.Column="0" Grid.Row="3"/>

<TextBox Text="{Binding Specialty\_code}" Height="30" Width="600" FontSize="20" FontFamily="Times New Roman" HorizontalAlignment="Center" VerticalAlignment="Center" Grid.Column="1" Grid.Row="0"/>

<TextBox Text="{Binding Name\_of\_the\_specialty}" Height="30" Width="600" FontSize="20" FontFamily="Times New Roman" HorizontalAlignment="Center" VerticalAlignment="Center" Grid.Column="1" Grid.Row="1"/>

<TextBox Text="{Binding Form\_of\_study}" Height="30" Width="600" FontSize="20" FontFamily="Times New Roman" HorizontalAlignment="Center" VerticalAlignment="Center" Grid.Column="1" Grid.Row="2"/>

<TextBox Text="{Binding Qualification}" Height="30" Width="600" FontSize="20" FontFamily="Times New Roman" HorizontalAlignment="Center" VerticalAlignment="Center" Grid.Column="1" Grid.Row="3"/>

<Button Content="Сохранить" HorizontalAlignment="Right" VerticalAlignment="Center" Height="40" Width="200" Click="Save\_Click" Grid.Column="1" Grid.Row="4" FontSize="20" FontFamily="Times New Roman" BorderBrush="Black" Background="White" Margin="0,0,50,0"/>

<Button Content="Назад" HorizontalAlignment="Left" VerticalAlignment="Center" Height="40" Width="200" Click="Go\_back\_Click" Grid.Column="0" Grid.Row="4" FontSize="20" FontFamily="Times New Roman" BorderBrush="Black" Background="White" Margin="50,0,0,0"/>

</Grid>

public partial class Adding\_Specialties : Window

{

readonly private Specialties specialties = new Specialties();

public Adding\_Specialties()

{

InitializeComponent();

WindowStartupLocation = WindowStartupLocation.CenterScreen;

DataContext = specialties;

}

private void Go\_back\_Click(object sender, RoutedEventArgs e)

{

this.Hide();

}

private void Save\_Click(object sender, RoutedEventArgs e)

{

StringBuilder errors = new StringBuilder();

if (string.IsNullOrEmpty(specialties.Specialty\_code) || (specialties.Specialty\_code.Length > 10))

errors.AppendLine("Пожалуйста, напишите код специальности. Максимальная длина 10 символов");

if (string.IsNullOrWhiteSpace(specialties.Name\_of\_the\_specialty))

errors.AppendLine("Пожалуйста, напишите наименование специальности");

if (string.IsNullOrWhiteSpace(specialties.Form\_of\_study) || (specialties.Form\_of\_study.Length > 17))

errors.AppendLine("Пожалуйста, напишите форму обучения. Максимальная длина 17 символов");

if (string.IsNullOrWhiteSpace(specialties.Qualification))

errors.AppendLine("Пожалуйста, напишите квалификацию");

if (errors.Length > 0)

{

MessageBox.Show(errors.ToString(), "Ошибка", MessageBoxButton.OK, MessageBoxImage.Error);

return;

}

if (specialties.Specialty\_code.Length > 0)

{

Coursework1Entities.GetContext().Specialties.Add(specialties);

}

try

{

Coursework1Entities.GetContext().SaveChanges();

MessageBox.Show("Информация добавлена", "Отлично", MessageBoxButton.OK, MessageBoxImage.Information);

}

catch (Exception ex)

{

MessageBox.Show(ex.Message.ToString());

}

this.Hide();

}

}

Окно «Изменение специальностей»

Title="Changing\_Specialties" Height="550" Width="1100">

<Grid>

<Grid.ColumnDefinitions>

<ColumnDefinition Width="\*"/>

<ColumnDefinition Width="2\*"/>

</Grid.ColumnDefinitions>

<Grid.RowDefinitions>

<RowDefinition Height="\*"/>

<RowDefinition Height="\*"/>

<RowDefinition Height="\*"/>

<RowDefinition Height="\*"/>

<RowDefinition Height="\*"/>

</Grid.RowDefinitions>

<Label Content="Код специальности" FontSize="20" FontFamily="Times New Roman" HorizontalAlignment="Center" VerticalAlignment="Center" Grid.Column="0" Grid.Row="0"/>

<Label Content="Название специальности" FontSize="20" FontFamily="Times New Roman" HorizontalAlignment="Center" VerticalAlignment="Center" Grid.Column="0" Grid.Row="1"/>

<Label Content="Форма обучения" FontSize="20" FontFamily="Times New Roman" HorizontalAlignment="Center" VerticalAlignment="Center" Grid.Column="0" Grid.Row="2"/>

<Label Content="Квалификация" FontSize="20" FontFamily="Times New Roman" HorizontalAlignment="Center" VerticalAlignment="Center" Grid.Column="0" Grid.Row="3"/>

<TextBox Text="{Binding Specialty\_code}" Height="30" Width="600" FontSize="20" FontFamily="Times New Roman" HorizontalAlignment="Center" VerticalAlignment="Center" Grid.Column="1" Grid.Row="0"/>

<TextBox Text="{Binding Name\_of\_the\_specialty}" Height="30" Width="600" FontSize="20" FontFamily="Times New Roman" HorizontalAlignment="Center" VerticalAlignment="Center" Grid.Column="1" Grid.Row="1"/>

<TextBox Text="{Binding Form\_of\_study}" Height="30" Width="600" FontSize="20" FontFamily="Times New Roman" HorizontalAlignment="Center" VerticalAlignment="Center" Grid.Column="1" Grid.Row="2"/>

<TextBox Text="{Binding Qualification}" Height="30" Width="600" FontSize="20" FontFamily="Times New Roman" HorizontalAlignment="Center" VerticalAlignment="Center" Grid.Column="1" Grid.Row="3"/>

<Button Content="Сохранить" HorizontalAlignment="Right" VerticalAlignment="Center" Height="40" Width="200" Click="Save\_Click" Grid.Column="1" Grid.Row="4" FontSize="20" FontFamily="Times New Roman" BorderBrush="Black" Background="White" Margin="0,0,50,0"/>

<Button Content="Назад" HorizontalAlignment="Left" VerticalAlignment="Center" Height="40" Width="200" Click="Go\_back\_Click" Grid.Column="0" Grid.Row="4" FontSize="20" FontFamily="Times New Roman" BorderBrush="Black" Background="White" Margin="50,0,0,0"/>

</Grid>

public partial class Changing\_Specialties : Window

{

readonly private Specialties specialties = new Specialties();

public Changing\_Specialties(Specialties selected\_row)

{

InitializeComponent();

WindowStartupLocation = WindowStartupLocation.CenterScreen;

if (selected\_row != null)

{

specialties = selected\_row;

}

DataContext = specialties;

}

private void Go\_back\_Click(object sender, RoutedEventArgs e)

{

this.Hide();

}

private void Save\_Click(object sender, RoutedEventArgs e)

{

StringBuilder errors = new StringBuilder();

if (string.IsNullOrEmpty(specialties.Specialty\_code) || (specialties.Specialty\_code.Length > 10))

errors.AppendLine("Пожалуйста, напишите код специальности. Максимальная длина 10 символов");

if (string.IsNullOrWhiteSpace(specialties.Name\_of\_the\_specialty))

errors.AppendLine("Пожалуйста, напишите наименование специальности");

if (string.IsNullOrWhiteSpace(specialties.Form\_of\_study) || (specialties.Form\_of\_study.Length > 17))

errors.AppendLine("Пожалуйста, напишите форму обучения. Максимальная длина 17 символов");

if (string.IsNullOrWhiteSpace(specialties.Qualification))

errors.AppendLine("Пожалуйста, напишите квалификацию");

if (errors.Length > 0)

{

MessageBox.Show(errors.ToString(), "Ошибка", MessageBoxButton.OK, MessageBoxImage.Error);

return;

}

if (specialties.Specialty\_code == null)

{

Coursework1Entities.GetContext().Specialties.Add(specialties);

}

try

{

Coursework1Entities.GetContext().SaveChanges();

MessageBox.Show("Информация изменена", "Отлично", MessageBoxButton.OK, MessageBoxImage.Information);

}

catch (Exception ex)

{

MessageBox.Show(ex.Message.ToString());

}

this.Hide();

}

}

# ПРИЛОЖЕНИЕ И

Окно «Учебные планы»

Title="Curriculum1" Height="550" Width="1100" IsVisibleChanged="Window\_IsVisibleChanged">

<Grid>

<Grid.ColumnDefinitions>

<ColumnDefinition Width="\*"/>

<ColumnDefinition Width="\*"/>

<ColumnDefinition Width="\*"/>

</Grid.ColumnDefinitions>

<Grid.RowDefinitions>

<RowDefinition Height="4\*"/>

<RowDefinition Height="\*"/>

</Grid.RowDefinitions>

<Button Content="Назад" Width="200" Height="40" FontSize="20" FontFamily="Times New Roman" BorderBrush="Black" Background="White" Click="Go\_back\_Click" Grid.Row="1" Grid.Column="0"/>

<Button Content="Выход" Width="200" Height="40" FontSize="20" FontFamily="Times New Roman" BorderBrush="Black" Background="White" Click="Exit\_Click" Grid.Row="1" Grid.Column="2"/>

<Button Content="Добавить" Width="200" Height="40" FontSize="20" FontFamily="Times New Roman" BorderBrush="Black" Background="White" Click="Add\_Click" Grid.Row="1" Grid.Column="1"/>

<DataGrid x:Name="Grid1" AutoGenerateColumns="False" Background="White" FontSize="14" FontFamily="Times New Roman" Grid.Row="0" Grid.ColumnSpan="3">

<DataGrid.Columns>

<DataGridTextColumn Header="Наименование учебного плана" Binding="{Binding Name\_of\_the\_curriculum}" Width="Auto"/>

<DataGridTextColumn Header="Код предмета" Binding="{Binding Subject\_code}" Width="Auto"/>

<DataGridTextColumn Header="Общее кол-во часов" Binding="{Binding Total\_number\_of\_hours}" Width="Auto"/>

<DataGridTextColumn Header="Практические часы" Binding="{Binding Practical\_hours}" Width="Auto"/>

<DataGridTextColumn Header="Теоретические часы" Binding="{Binding Theoretical\_hours}" Width="Auto"/>

<DataGridTextColumn Header="Номер семестра" Binding="{Binding Semester\_number}" Width="Auto"/>

<DataGridTextColumn Header="Код специальности" Binding="{Binding Specialty\_code}" Width="Auto"/>

<DataGridTextColumn Header="Код группы" Binding="{Binding Group\_code}" Width="Auto"/>

<DataGridTemplateColumn Width="100">

<DataGridTemplateColumn.CellTemplate>

<DataTemplate>

<Button Content="Изменить" FontFamily="Times New Roman" BorderBrush="Black" Background="White" Click="Edit\_Click"/>

</DataTemplate>

</DataGridTemplateColumn.CellTemplate>

</DataGridTemplateColumn>

</DataGrid.Columns>

</DataGrid>

</Grid>

public partial class Curriculums1 : Window

{

public Curriculums1()

{

InitializeComponent();

WindowStartupLocation = WindowStartupLocation.CenterScreen;

}

private void Edit\_Click(object sender, RoutedEventArgs e)

{

Changing\_Curriculums curriculums = new Changing\_Curriculums((sender as Button).DataContext as Curriculum);

curriculums.ShowDialog();

}

private void Add\_Click(object sender, RoutedEventArgs e)

{

Adding\_Curriculums curriculums = new Adding\_Curriculums();

curriculums.ShowDialog();

}

private void Go\_back\_Click(object sender, RoutedEventArgs e)

{

this.Hide();

To\_the\_necessary\_inf inf = new To\_the\_necessary\_inf();

inf.ShowDialog();

}

private void Exit\_Click(object sender, RoutedEventArgs e)

{

Application.Current.Shutdown();

}

private void Window\_IsVisibleChanged(object sender, DependencyPropertyChangedEventArgs e)

{

Coursework1Entities.GetContext().ChangeTracker.Entries().ToList().ForEach(p => p.Reload());

Grid1.ItemsSource = Coursework1Entities.GetContext().Curriculum.ToList();

}

}

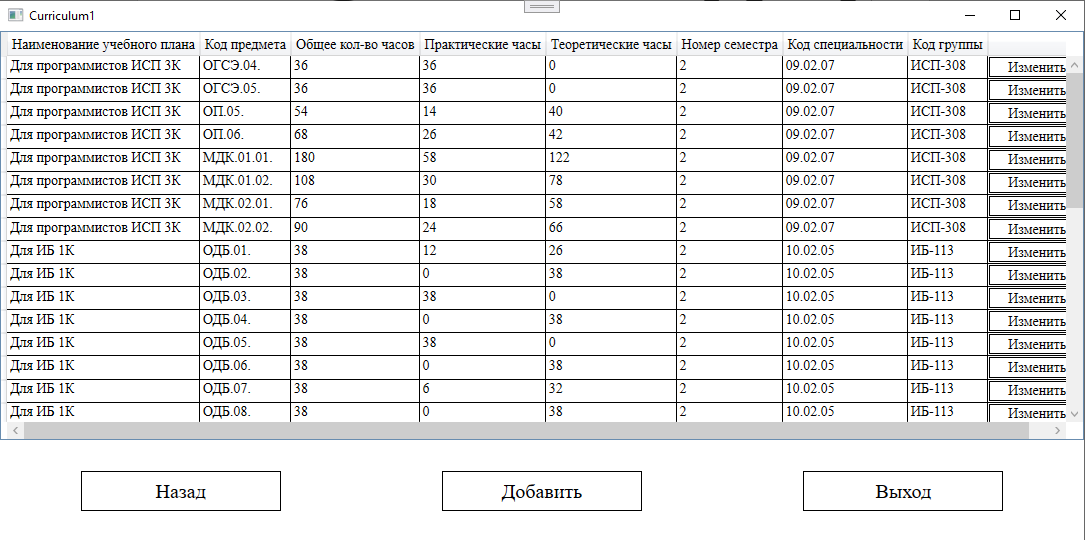


Рисунок - Окно «Учебные планы»

Окно «Добавление Учебных планов»

Title="Adding\_Curriculums" Height="550" Width="1100">

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="\*"/>

<RowDefinition Height="\*"/>

<RowDefinition Height="\*"/>

<RowDefinition Height="\*"/>

<RowDefinition Height="\*"/>

<RowDefinition Height="\*"/>

<RowDefinition Height="\*"/>

<RowDefinition Height="\*"/>

<RowDefinition Height="\*"/>

</Grid.RowDefinitions>

<Grid.ColumnDefinitions>

<ColumnDefinition Width="\*"/>

<ColumnDefinition Width="2\*"/>

</Grid.ColumnDefinitions>

<Label Content="Наименование учебного плана" FontSize="20" FontFamily="Times New Roman" HorizontalAlignment="Center" VerticalAlignment="Center" Grid.Column="0" Grid.Row="0"/>

<Label Content="Код предмета" FontSize="20" FontFamily="Times New Roman" HorizontalAlignment="Center" VerticalAlignment="Center" Grid.Column="0" Grid.Row="1"/>

<Label Content="Общее кол-во часов" FontSize="20" FontFamily="Times New Roman" HorizontalAlignment="Center" VerticalAlignment="Center" Grid.Column="0" Grid.Row="2"/>

<Label Content="Практические часы" FontSize="20" FontFamily="Times New Roman" HorizontalAlignment="Center" VerticalAlignment="Center" Grid.Column="0" Grid.Row="3"/>

<Label Content="Теоретические часы" FontSize="20" FontFamily="Times New Roman" HorizontalAlignment="Center" VerticalAlignment="Center" Grid.Column="0" Grid.Row="4"/>

<Label Content="Номер семестра" FontSize="20" FontFamily="Times New Roman" HorizontalAlignment="Center" VerticalAlignment="Center" Grid.Column="0" Grid.Row="5"/>

<Label Content="Код специальности" FontSize="20" FontFamily="Times New Roman" HorizontalAlignment="Center" VerticalAlignment="Center" Grid.Column="0" Grid.Row="6"/>

<Label Content="Код группы" FontSize="20" FontFamily="Times New Roman" HorizontalAlignment="Center" VerticalAlignment="Center" Grid.Column="0" Grid.Row="7"/>

<TextBox Text="{Binding Name\_of\_the\_curriculum}" Height="30" Width="600" FontSize="20" FontFamily="Times New Roman" HorizontalAlignment="Center" VerticalAlignment="Center" Grid.Column="1" Grid.Row="0"/>

<ComboBox x:Name="CBsubject\_code" SelectedItem="{Binding Subject\_code}" DisplayMemberPath="Subject\_code" Height="30" Width="600" FontSize="20" FontFamily="Times New Roman" HorizontalAlignment="Center" VerticalAlignment="Center" Grid.Column="1" Grid.Row="1"/>

<TextBox Text="{Binding Total\_number\_of\_hours}" Height="30" Width="600" FontSize="20" FontFamily="Times New Roman" Grid.Column="1" Grid.Row="2"/>

<TextBox Text="{Binding Practical\_hours}" Height="30" Width="600" FontSize="20" FontFamily="Times New Roman" Grid.Column="1" Grid.Row="3"/>

<TextBox Text="{Binding Theoretical\_hours}" Height="30" Width="600" FontSize="20" FontFamily="Times New Roman" Grid.Column="1" Grid.Row="4"/>

<ComboBox x:Name="CBsemester\_number" SelectedItem="{Binding Semester\_number}" DisplayMemberPath="Semester\_number" Height="30" Width="600" FontSize="20" FontFamily="Times New Roman" HorizontalAlignment="Center" VerticalAlignment="Center" Grid.Column="1" Grid.Row="5"/>

<ComboBox x:Name="CBspecialty\_code" SelectedItem="{Binding Specialty\_code}" DisplayMemberPath="Specialty\_code" Height="30" Width="600" FontSize="20" FontFamily="Times New Roman" HorizontalAlignment="Center" VerticalAlignment="Center" Grid.Column="1" Grid.Row="6"/>

<ComboBox x:Name="CBgroup\_code" SelectedItem="{Binding Group\_code}" DisplayMemberPath="Group\_code" Height="30" Width="600" FontSize="20" FontFamily="Times New Roman" HorizontalAlignment="Center" VerticalAlignment="Center" Grid.Column="1" Grid.Row="7"/>

<Button Content="Сохранить" HorizontalAlignment="Right" VerticalAlignment="Center" Height="40" Width="200" Click="Save\_Click" Grid.Column="1" Grid.Row="8" FontSize="20" FontFamily="Times New Roman" BorderBrush="Black" Background="White" Margin="0,0,50,0"/>

<Button Content="Назад" HorizontalAlignment="Left" VerticalAlignment="Center" Height="40" Width="200" Click="Go\_back\_Click" Grid.Column="0" Grid.Row="8" FontSize="20" FontFamily="Times New Roman" BorderBrush="Black" Background="White" Margin="50,0,0,0"/>

</Grid>

public partial class Adding\_Curriculums : Window

{

readonly private Curriculum curriculum = new Curriculum();

public Adding\_Curriculums()

{

InitializeComponent();

WindowStartupLocation = WindowStartupLocation.CenterScreen;

DataContext = curriculum;

CBsubject\_code.ItemsSource = Coursework1Entities.GetContext().Subjects.ToList();

CBsemester\_number.ItemsSource = Coursework1Entities.GetContext().Semesters.ToList();

CBspecialty\_code.ItemsSource = Coursework1Entities.GetContext().Specialties.ToList();

CBgroup\_code.ItemsSource = Coursework1Entities.GetContext().Groups.ToList();

}

private void Go\_back\_Click(object sender, RoutedEventArgs e)

{

this.Hide();

}

private void Save\_Click(object sender, RoutedEventArgs e)

{

StringBuilder errors = new StringBuilder();

if (string.IsNullOrWhiteSpace(curriculum.Name\_of\_the\_curriculum) || (curriculum.Name\_of\_the\_curriculum.Length > 50))

errors.AppendLine("Пожалуйста, напишите наименование учебного плана. Максимальная длина 50 символов");

if ((CBsubject\_code == null) || (curriculum.Subject\_code == null))

errors.AppendLine("Пожалуйста, выберите код предмета");

if ((curriculum.Total\_number\_of\_hours == null) || (curriculum.Total\_number\_of\_hours <= 0))

errors.AppendLine("Пожалуйста, введите общее кол-во часов");

if ((curriculum.Practical\_hours == null) || (curriculum.Practical\_hours < 0) || (curriculum.Practical\_hours > curriculum.Total\_number\_of\_hours))

errors.AppendLine("Пожалуйста, введите кол-во практических часов корректно");

if ((curriculum.Theoretical\_hours == null) || (curriculum.Theoretical\_hours < 0) || (curriculum.Practical\_hours > curriculum.Total\_number\_of\_hours))

errors.AppendLine("Пожалуйста, введите кол-во теоретических часов корректно");

if ((CBsemester\_number == null) || (curriculum.Semester\_number == null))

errors.AppendLine("Пожалуйста, выберите номер семестра");

if ((CBspecialty\_code == null) || (curriculum.Specialty\_code == null))

errors.AppendLine("Пожалуйста, выберите код специальности");

if ((CBgroup\_code == null) || (curriculum.Group\_code == null))

errors.AppendLine("Пожалуйста, выберите код группы");

if (errors.Length > 0)

{

MessageBox.Show(errors.ToString(), "Ошибка", MessageBoxButton.OK, MessageBoxImage.Error);

return;

}

if (curriculum.Curriculum\_ID == 0)

{

Coursework1Entities.GetContext().Curriculum.Add(curriculum);

}

try

{

Coursework1Entities.GetContext().SaveChanges();

MessageBox.Show("Информация добавлена", "Отлично", MessageBoxButton.OK, MessageBoxImage.Information);

}

catch (Exception ex)

{

MessageBox.Show(ex.Message.ToString());

}

this.Hide();

}

}

Окно «Изменение Учебных планов»

Title="Changing\_Curriculums" Height="550" Width="1100">

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="\*"/>

<RowDefinition Height="\*"/>

<RowDefinition Height="\*"/>

<RowDefinition Height="\*"/>

<RowDefinition Height="\*"/>

<RowDefinition Height="\*"/>

<RowDefinition Height="\*"/>

<RowDefinition Height="\*"/>

<RowDefinition Height="\*"/>

</Grid.RowDefinitions>

<Grid.ColumnDefinitions>

<ColumnDefinition Width="\*"/>

<ColumnDefinition Width="2\*"/>

</Grid.ColumnDefinitions>

<Label Content="Наименование учебного плана" FontSize="20" FontFamily="Times New Roman" HorizontalAlignment="Center" VerticalAlignment="Center" Grid.Column="0" Grid.Row="0"/>

<Label Content="Код предмета" FontSize="20" FontFamily="Times New Roman" HorizontalAlignment="Center" VerticalAlignment="Center" Grid.Column="0" Grid.Row="1"/>

<Label Content="Общее кол-во часов" FontSize="20" FontFamily="Times New Roman" HorizontalAlignment="Center" VerticalAlignment="Center" Grid.Column="0" Grid.Row="2"/>

<Label Content="Практические часы" FontSize="20" FontFamily="Times New Roman" HorizontalAlignment="Center" VerticalAlignment="Center" Grid.Column="0" Grid.Row="3"/>

<Label Content="Теоретические часы" FontSize="20" FontFamily="Times New Roman" HorizontalAlignment="Center" VerticalAlignment="Center" Grid.Column="0" Grid.Row="4"/>

<Label Content="Номер семестра" FontSize="20" FontFamily="Times New Roman" HorizontalAlignment="Center" VerticalAlignment="Center" Grid.Column="0" Grid.Row="5"/>

<Label Content="Код специальности" FontSize="20" FontFamily="Times New Roman" HorizontalAlignment="Center" VerticalAlignment="Center" Grid.Column="0" Grid.Row="6"/>

<Label Content="Код группы" FontSize="20" FontFamily="Times New Roman" HorizontalAlignment="Center" VerticalAlignment="Center" Grid.Column="0" Grid.Row="7"/>

<TextBox Text="{Binding Name\_of\_the\_curriculum}" Height="30" Width="600" FontSize="20" FontFamily="Times New Roman" HorizontalAlignment="Center" VerticalAlignment="Center" Grid.Column="1" Grid.Row="0"/>

<ComboBox x:Name="CBsubject\_code" Text="{Binding Subject\_code}" SelectedItem="{Binding Subject\_code}" DisplayMemberPath="Subject\_code" Height="30" Width="600" FontSize="20" FontFamily="Times New Roman" HorizontalAlignment="Center" VerticalAlignment="Center" Grid.Column="1" Grid.Row="1"/>

<TextBox Text="{Binding Total\_number\_of\_hours}" Height="30" Width="600" FontSize="20" FontFamily="Times New Roman" Grid.Column="1" Grid.Row="2"/>

<TextBox Text="{Binding Practical\_hours}" Height="30" Width="600" FontSize="20" FontFamily="Times New Roman" Grid.Column="1" Grid.Row="3"/>

<TextBox Text="{Binding Theoretical\_hours}" Height="30" Width="600" FontSize="20" FontFamily="Times New Roman" Grid.Column="1" Grid.Row="4"/>

<ComboBox x:Name="CBsemester\_number" Text="{Binding Semester\_number}" SelectedItem="{Binding Semester\_number}" DisplayMemberPath="Semester\_number" Height="30" Width="600" FontSize="20" FontFamily="Times New Roman" HorizontalAlignment="Center" VerticalAlignment="Center" Grid.Column="1" Grid.Row="5"/>

<ComboBox x:Name="CBspecialty\_code" Text="{Binding Specialty\_code}" SelectedItem="{Binding Specialty\_code}" DisplayMemberPath="Specialty\_code" Height="30" Width="600" FontSize="20" FontFamily="Times New Roman" HorizontalAlignment="Center" VerticalAlignment="Center" Grid.Column="1" Grid.Row="6"/>

<ComboBox x:Name="CBgroup\_code" Text="{Binding Group\_code}" SelectedItem="{Binding Group\_code}" DisplayMemberPath="Group\_code" Height="30" Width="600" FontSize="20" FontFamily="Times New Roman" HorizontalAlignment="Center" VerticalAlignment="Center" Grid.Column="1" Grid.Row="7"/>

<Button Content="Сохранить" HorizontalAlignment="Right" VerticalAlignment="Center" Height="40" Width="200" Click="Save\_Click" Grid.Column="1" Grid.Row="8" FontSize="20" FontFamily="Times New Roman" BorderBrush="Black" Background="White" Margin="0,0,50,0"/>

<Button Content="Назад" HorizontalAlignment="Left" VerticalAlignment="Center" Height="40" Width="200" Click="Go\_back\_Click" Grid.Column="0" Grid.Row="8" FontSize="20" FontFamily="Times New Roman" BorderBrush="Black" Background="White" Margin="50,0,0,0"/>

</Grid>

public partial class Changing\_Curriculums : Window

{

readonly private Curriculum curriculum = new Curriculum();

public Changing\_Curriculums(Curriculum selected\_row)

{

InitializeComponent();

WindowStartupLocation = WindowStartupLocation.CenterScreen;

if (selected\_row != null)

{

curriculum = selected\_row;

}

DataContext = curriculum;

CBsubject\_code.ItemsSource = Coursework1Entities.GetContext().Subjects.ToList();

CBsemester\_number.ItemsSource = Coursework1Entities.GetContext().Semesters.ToList();

CBspecialty\_code.ItemsSource = Coursework1Entities.GetContext().Specialties.ToList();

CBgroup\_code.ItemsSource = Coursework1Entities.GetContext().Groups.ToList();

}

private void Go\_back\_Click(object sender, RoutedEventArgs e)

{

this.Hide();

}

private void Save\_Click(object sender, RoutedEventArgs e)

{

StringBuilder errors = new StringBuilder();

if (string.IsNullOrWhiteSpace(curriculum.Name\_of\_the\_curriculum) || (curriculum.Name\_of\_the\_curriculum.Length > 50))

errors.AppendLine("Пожалуйста, напишите наименование учебного плана. Максимальная длина 50 символов");

if ((CBsubject\_code == null) || (curriculum.Subject\_code == null))

errors.AppendLine("Пожалуйста, выберите код предмета");

if ((curriculum.Total\_number\_of\_hours == null) || (curriculum.Total\_number\_of\_hours <= 0))

errors.AppendLine("Пожалуйста, введите общее кол-во часов");

if ((curriculum.Practical\_hours == null) || (curriculum.Practical\_hours < 0) || (curriculum.Practical\_hours > curriculum.Total\_number\_of\_hours))

errors.AppendLine("Пожалуйста, введите кол-во практических часов корректно");

if ((curriculum.Theoretical\_hours == null) || (curriculum.Theoretical\_hours < 0) || (curriculum.Practical\_hours > curriculum.Total\_number\_of\_hours))

errors.AppendLine("Пожалуйста, введите кол-во теоретических часов корректно");

if ((CBsemester\_number == null) || (curriculum.Semester\_number == null))

errors.AppendLine("Пожалуйста, выберите номер семестра");

if ((CBspecialty\_code == null) || (curriculum.Specialty\_code == null))

errors.AppendLine("Пожалуйста, выберите код специальности");

if ((CBgroup\_code == null) || (curriculum.Group\_code == null))

errors.AppendLine("Пожалуйста, выберите код группы");

if (errors.Length > 0)

{

MessageBox.Show(errors.ToString(), "Ошибка", MessageBoxButton.OK, MessageBoxImage.Error);

return;

}

if (curriculum.Curriculum\_ID == 0)

{

Coursework1Entities.GetContext().Curriculum.Add(curriculum);

}

try

{

Coursework1Entities.GetContext().SaveChanges();

MessageBox.Show("Информация изменена", "Отлично", MessageBoxButton.OK, MessageBoxImage.Information);

}

catch (Exception ex)

{

MessageBox.Show(ex.Message.ToString());

}

this.Hide();

}

}

# ПРИЛОЖЕНИЕ К

Окно «Семестры»

Title="Semesters1" Height="550" Width="1100" IsVisibleChanged="Window\_IsVisibleChanged">

<Grid>

<Grid.ColumnDefinitions>

<ColumnDefinition Width="\*"/>

<ColumnDefinition Width="\*"/>

<ColumnDefinition Width="\*"/>

</Grid.ColumnDefinitions>

<Grid.RowDefinitions>

<RowDefinition Height="4\*"/>

<RowDefinition Height="\*"/>

</Grid.RowDefinitions>

<Button Content="Добавить" Width="200" Height="40" FontSize="20" FontFamily="Times New Roman" BorderBrush="Black" Background="White" Click="Add\_Click" Grid.Row="1" Grid.Column="1"/>

<Button Content="Назад" Width="200" Height="40" FontSize="20" FontFamily="Times New Roman" BorderBrush="Black" Background="White" Click="Go\_back\_Click" Grid.Row="1" Grid.Column="0"/>

<Button Content="Выход" Width="200" Height="40" FontSize="20" FontFamily="Times New Roman" BorderBrush="Black" Background="White" Click="Exit\_Click" Grid.Row="1" Grid.Column="2"/>

<DataGrid x:Name="Grid1" AutoGenerateColumns="False" Background="White" FontSize="14" FontFamily="Times New Roman" Grid.Row="0" Grid.ColumnSpan="3">

<DataGrid.Columns>

<DataGridTextColumn Header="Семестр" Binding="{Binding Semester}" Width="\*"/>

<DataGridTextColumn Header="Начало семестра" Binding="{Binding Beginning\_of\_semester}" Width="\*"/>

<DataGridTextColumn Header="Конец семестра" Binding="{Binding Ending\_of\_semester}" Width="\*"/>

<DataGridTemplateColumn Width="100">

<DataGridTemplateColumn.CellTemplate>

<DataTemplate>

<Button Content="Изменить" FontFamily="Times New Roman" BorderBrush="Black" Background="White" Click="Edit\_Click"/>

</DataTemplate>

</DataGridTemplateColumn.CellTemplate>

</DataGridTemplateColumn>

</DataGrid.Columns>

</DataGrid>

</Grid>

public partial class Semesters1 : Window

{

public Semesters1()

{

InitializeComponent();

WindowStartupLocation = WindowStartupLocation.CenterScreen;

}

private void Edit\_Click(object sender, RoutedEventArgs e)

{

Changing\_Semesters semesters = new Changing\_Semesters((sender as Button).DataContext as Semesters);

semesters.ShowDialog();

}

private void Add\_Click(object sender, RoutedEventArgs e)

{

Adding\_Semesters semesters = new Adding\_Semesters();

semesters.ShowDialog();

}

private void Go\_back\_Click(object sender, RoutedEventArgs e)

{

this.Hide();

To\_the\_necessary\_inf inf = new To\_the\_necessary\_inf();

inf.ShowDialog();

}

private void Exit\_Click(object sender, RoutedEventArgs e)

{

Application.Current.Shutdown();

}

private void Window\_IsVisibleChanged(object sender, DependencyPropertyChangedEventArgs e)

{

if (Visibility == Visibility.Visible)

{

Coursework1Entities.GetContext().ChangeTracker.Entries().ToList().ForEach(p => p.Reload());

Grid1.ItemsSource = Coursework1Entities.GetContext().Semesters.ToList();

}

}

}

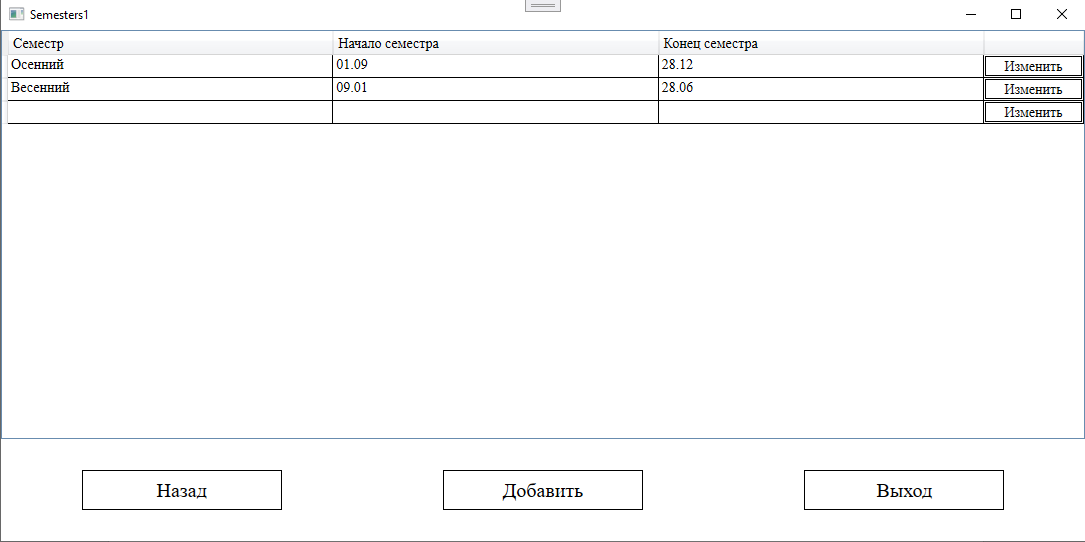


Рисунок - Окно «Семестры»

Окно «Добавление Семестров»

Title="Adding\_Semesters" Height="550" Width="1100">

<Grid>

<Grid.ColumnDefinitions>

<ColumnDefinition Width="\*"/>

<ColumnDefinition Width="2\*"/>

</Grid.ColumnDefinitions>

<Grid.RowDefinitions>

<RowDefinition Height="\*"/>

<RowDefinition Height="\*"/>

<RowDefinition Height="\*"/>

<RowDefinition Height="\*"/>

</Grid.RowDefinitions>

<Label Content="Семестр" FontSize="20" FontFamily="Times New Roman" HorizontalAlignment="Center" VerticalAlignment="Center" Grid.Column="0" Grid.Row="0"/>

<Label Content="Начало семестра" FontSize="20" FontFamily="Times New Roman" HorizontalAlignment="Center" VerticalAlignment="Center" Grid.Column="0" Grid.Row="1"/>

<Label Content="Конец семестра" FontSize="20" FontFamily="Times New Roman" HorizontalAlignment="Center" VerticalAlignment="Center" Grid.Column="0" Grid.Row="2"/>

<TextBox Text="{Binding Semester}" Height="30" Width="600" FontSize="20" FontFamily="Times New Roman" Grid.Column="1" Grid.Row="0"/>

<TextBox Text="{Binding Beginning\_of\_semester}" Height="30" Width="600" FontSize="20" FontFamily="Times New Roman" Grid.Column="1" Grid.Row="1"/>

<TextBox Text="{Binding Ending\_of\_semester}" Height="30" Width="600" FontSize="20" FontFamily="Times New Roman" Grid.Column="1" Grid.Row="2"/>

<Button HorizontalAlignment="Right" VerticalAlignment="Center" Height="40" Width="200" Content="Сохранить" Click="Save\_Click" Grid.Column="1" Grid.Row="3" FontSize="20" FontFamily="Times New Roman" BorderBrush="Black" Background="White" Margin="0,0,50,0"/>

<Button HorizontalAlignment="Left" VerticalAlignment="Center" Height="40" Width="200" Content="Назад" Click="Go\_back\_Click" Grid.Column="0" Grid.Row="3" FontSize="20" FontFamily="Times New Roman" BorderBrush="Black" Background="White" Margin="50,0,0,0"/>

</Grid>

public partial class Adding\_Semesters : Window

{

readonly private Semesters semesters = new Semesters();

public Adding\_Semesters()

{

InitializeComponent();

WindowStartupLocation = WindowStartupLocation.CenterScreen;

DataContext = semesters;

}

private void Go\_back\_Click(object sender, RoutedEventArgs e)

{

this.Hide();

}

private void Save\_Click(object sender, RoutedEventArgs e)

{

StringBuilder errors = new StringBuilder();

if (string.IsNullOrWhiteSpace(semesters.Semester))

errors.AppendLine("Пожалуйста, напишите семестр");

if (string.IsNullOrEmpty(semesters.Beginning\_of\_semester) || (semesters.Beginning\_of\_semester.Length > 5))

errors.AppendLine("Пожалуйста, напишите начало семестра. Максимальная длина 5 символов");

if (string.IsNullOrEmpty(semesters.Ending\_of\_semester) || (semesters.Ending\_of\_semester.Length > 5))

errors.AppendLine("Пожалуйста, напишите конец семестра. Максимальная длина 5 символов");

if (errors.Length > 0)

{

MessageBox.Show(errors.ToString(), "Ошибка", MessageBoxButton.OK, MessageBoxImage.Error);

return;

}

if (semesters.Semester\_number == 0)

{

Coursework1Entities.GetContext().Semesters.Add(semesters);

}

try

{

Coursework1Entities.GetContext().SaveChanges();

MessageBox.Show("Информация добавлена", "Отлично", MessageBoxButton.OK, MessageBoxImage.Information);

}

catch (Exception ex)

{

MessageBox.Show(ex.Message.ToString());

}

this.Hide();

}

}

Окно «Изменение Семестров»

Title="Changing\_Semesters" Height="550" Width="1100">

<Grid>

<Grid.ColumnDefinitions>

<ColumnDefinition Width="\*"/>

<ColumnDefinition Width="2\*"/>

</Grid.ColumnDefinitions>

<Grid.RowDefinitions>

<RowDefinition Height="\*"/>

<RowDefinition Height="\*"/>

<RowDefinition Height="\*"/>

<RowDefinition Height="\*"/>

</Grid.RowDefinitions>

<Label Content="Семестр" FontSize="20" FontFamily="Times New Roman" HorizontalAlignment="Center" VerticalAlignment="Center" Grid.Column="0" Grid.Row="0"/>

<Label Content="Начало семестра" FontSize="20" FontFamily="Times New Roman" HorizontalAlignment="Center" VerticalAlignment="Center" Grid.Column="0" Grid.Row="1"/>

<Label Content="Конец семестра" FontSize="20" FontFamily="Times New Roman" HorizontalAlignment="Center" VerticalAlignment="Center" Grid.Column="0" Grid.Row="2"/>

<TextBox Text="{Binding Semester}" Height="30" Width="600" FontSize="20" FontFamily="Times New Roman" Grid.Column="1" Grid.Row="0"/>

<TextBox Text="{Binding Beginning\_of\_semester}" Height="30" Width="600" FontSize="20" FontFamily="Times New Roman" Grid.Column="1" Grid.Row="1"/>

<TextBox Text="{Binding Ending\_of\_semester}" Height="30" Width="600" FontSize="20" FontFamily="Times New Roman" Grid.Column="1" Grid.Row="2"/>

<Button HorizontalAlignment="Right" VerticalAlignment="Center" Height="40" Width="200" Content="Сохранить" Click="Save\_Click" Grid.Column="1" Grid.Row="3" FontSize="20" FontFamily="Times New Roman" BorderBrush="Black" Background="White" Margin="0,0,50,0"/>

<Button HorizontalAlignment="Left" VerticalAlignment="Center" Height="40" Width="200" Content="Назад" Click="Go\_back\_Click" Grid.Column="0" Grid.Row="3" FontSize="20" FontFamily="Times New Roman" BorderBrush="Black" Background="White" Margin="50,0,0,0"/>

</Grid>

public partial class Changing\_Semesters : Window

{

readonly private Semesters semesters = new Semesters();

public Changing\_Semesters(Semesters selected\_row)

{

InitializeComponent();

WindowStartupLocation = WindowStartupLocation.CenterScreen;

if (selected\_row != null)

{

semesters = selected\_row;

}

DataContext = semesters;

}

private void Go\_back\_Click(object sender, RoutedEventArgs e)

{

this.Hide();

}

private void Save\_Click(object sender, RoutedEventArgs e)

{

StringBuilder errors = new StringBuilder();

if (string.IsNullOrWhiteSpace(semesters.Semester))

errors.AppendLine("Пожалуйста, напишите семестр");

if (string.IsNullOrEmpty(semesters.Beginning\_of\_semester) || (semesters.Beginning\_of\_semester.Length > 5))

errors.AppendLine("Пожалуйста, напишите начало семестра. Максимальная длина 5 символов");

if (string.IsNullOrEmpty(semesters.Ending\_of\_semester) || (semesters.Ending\_of\_semester.Length > 5))

errors.AppendLine("Пожалуйста, напишите конец семестра. Максимальная длина 5 символов");

if (errors.Length > 0)

{

MessageBox.Show(errors.ToString(), "Ошибка", MessageBoxButton.OK, MessageBoxImage.Error);

return;

}

if (semesters.Semester\_number == 0)

{

Coursework1Entities.GetContext().Semesters.Add(semesters);

}

try

{

Coursework1Entities.GetContext().SaveChanges();

MessageBox.Show("Информация изменена", "Отлично", MessageBoxButton.OK, MessageBoxImage.Information);

}

catch (Exception ex)

{

MessageBox.Show(ex.Message.ToString());

}

this.Hide();

}

}

# ПРИЛОЖЕНИЕ Л

Окно «Предметы»

Title="Subjects1" Height="550" Width="1100" IsVisibleChanged="Window\_IsVisibleChanged">

<Grid>

<Grid.ColumnDefinitions>

<ColumnDefinition Width="\*"/>

<ColumnDefinition Width="\*"/>

<ColumnDefinition Width="\*"/>

</Grid.ColumnDefinitions>

<Grid.RowDefinitions>

<RowDefinition Height="4\*"/>

<RowDefinition Height="\*"/>

</Grid.RowDefinitions>

<Button Content="Добавить" Width="200" Height="40" FontSize="20" FontFamily="Times New Roman" BorderBrush="Black" Background="White" Click="Add\_Click" Grid.Row="1" Grid.Column="1"/>

<Button Content="Назад" Width="200" Height="40" FontSize="20" FontFamily="Times New Roman" BorderBrush="Black" Background="White" Click="Go\_back\_Click" Grid.Row="1" Grid.Column="0"/>

<Button Content="Выход" Width="200" Height="40" FontSize="20" FontFamily="Times New Roman" BorderBrush="Black" Background="White" Click="Exit\_Click" Grid.Row="1" Grid.Column="2"/>

<DataGrid x:Name="Grid1" AutoGenerateColumns="False" Background="White" FontFamily="Times New Roman" FontSize="14" Grid.Row="0" Grid.ColumnSpan="3">

<DataGrid.Columns>

<DataGridTextColumn Header="Код предмета" Binding="{Binding Subject\_code}" Width="Auto"/>

<DataGridTextColumn Header="Название предмета" Binding="{Binding Name\_of\_the\_subject}" Width="\*"/>

<DataGridTemplateColumn Width="100">

<DataGridTemplateColumn.CellTemplate>

<DataTemplate>

<Button Content="Изменить" FontFamily="Times New Roman" BorderBrush="Black" Background="White" Click="Edit\_Click"/>

</DataTemplate>

</DataGridTemplateColumn.CellTemplate>

</DataGridTemplateColumn>

</DataGrid.Columns>

</DataGrid>

</Grid>

public partial class Subjects1 : Window

{

public Subjects1()

{

InitializeComponent();

WindowStartupLocation = WindowStartupLocation.CenterScreen;

}

private void Edit\_Click(object sender, RoutedEventArgs e)

{

Changing\_Subjects subjects = new Changing\_Subjects((sender as Button).DataContext as Subjects);

subjects.ShowDialog();

}

private void Add\_Click(object sender, RoutedEventArgs e)

{

Adding\_Subjects subjects = new Adding\_Subjects();

subjects.ShowDialog();

}

private void Go\_back\_Click(object sender, RoutedEventArgs e)

{

this.Hide();

To\_the\_necessary\_inf inf = new To\_the\_necessary\_inf();

inf.ShowDialog();

}

private void Exit\_Click(object sender, RoutedEventArgs e)

{

Application.Current.Shutdown();

}

private void Window\_IsVisibleChanged(object sender, DependencyPropertyChangedEventArgs e)

{

if (Visibility == Visibility.Visible)

{

Coursework1Entities.GetContext().ChangeTracker.Entries().ToList().ForEach(p => p.Reload());

Grid1.ItemsSource = Coursework1Entities.GetContext().Subjects.ToList();

}

}

}

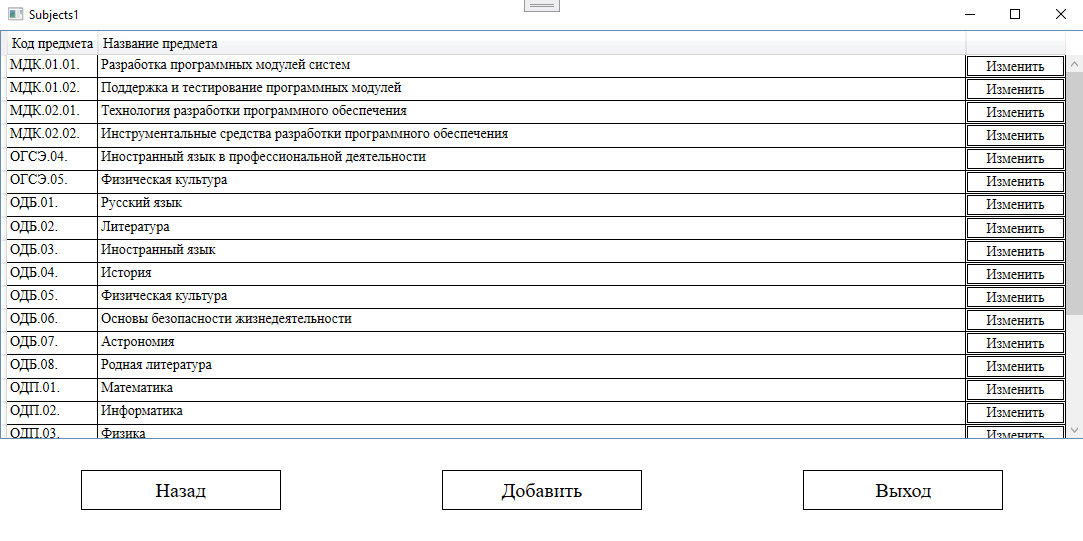


Рисунок - Окно «Предметы»

Окно «Добавление предметов»

Title="Adding\_Subjects" Height="550" Width="1100">

<Grid>

<Grid.ColumnDefinitions>

<ColumnDefinition Width="\*"/>

<ColumnDefinition Width="2\*"/>

</Grid.ColumnDefinitions>

<Grid.RowDefinitions>

<RowDefinition Height="\*"/>

<RowDefinition Height="\*"/>

<RowDefinition Height="\*"/>

</Grid.RowDefinitions>

<Label Content="Код предмета" FontSize="20" FontFamily="Times New Roman" HorizontalAlignment="Center" VerticalAlignment="Center" Grid.Column="0" Grid.Row="0"/>

<Label Content="Название предмета" FontSize="20" FontFamily="Times New Roman" HorizontalAlignment="Center" VerticalAlignment="Center" Grid.Column="0" Grid.Row="1"/>

<TextBox Text="{Binding Subject\_code}" Height="30" Width="600" FontSize="20" FontFamily="Times New Roman" Grid.Column="1" Grid.Row="0"/>

<TextBox Text="{Binding Name\_of\_the\_subject}" Height="30" Width="600" FontSize="20" FontFamily="Times New Roman" Grid.Column="1" Grid.Row="1"/>

<Button HorizontalAlignment="Right" VerticalAlignment="Center" Height="40" Width="200" Content="Сохранить" Click="Save\_Click" Grid.Column="1" Grid.Row="3" FontSize="20" FontFamily="Times New Roman" BorderBrush="Black" Background="White" Margin="0,0,50,0"/>

<Button HorizontalAlignment="Left" VerticalAlignment="Center" Height="40" Width="200" Content="Назад" Click="Go\_back\_Click" Grid.Column="0" Grid.Row="3" FontSize="20" FontFamily="Times New Roman" BorderBrush="Black" Background="White" Margin="50,0,0,0"/>

</Grid>

public partial class Adding\_Subjects : Window

{

readonly private Subjects subjects = new Subjects();

public Adding\_Subjects()

{

InitializeComponent();

WindowStartupLocation = WindowStartupLocation.CenterScreen;

DataContext = subjects;

}

private void Go\_back\_Click(object sender, RoutedEventArgs e)

{

this.Hide();

}

private void Save\_Click(object sender, RoutedEventArgs e)

{

StringBuilder errors = new StringBuilder();

if (string.IsNullOrEmpty(subjects.Subject\_code) || (subjects.Subject\_code.Length > 10))

errors.AppendLine("Пожалуйста, напишите код предмета. Максимальная длина 10 символов");

if (string.IsNullOrWhiteSpace(subjects.Name\_of\_the\_subject) || (subjects.Name\_of\_the\_subject.Length > 50))

errors.AppendLine("Пожалуйста, напишите название предмета. Максимальная длина 50 символов");

if (errors.Length > 0)

{

MessageBox.Show(errors.ToString(), "Ошибка", MessageBoxButton.OK, MessageBoxImage.Error);

return;

}

if (subjects.Subject\_code.Length > 0)

{

Coursework1Entities.GetContext().Subjects.Add(subjects);

}

try

{

Coursework1Entities.GetContext().SaveChanges();

MessageBox.Show("Информация добавлена", "Отлично", MessageBoxButton.OK, MessageBoxImage.Information);

}

catch (Exception ex)

{

MessageBox.Show(ex.Message.ToString());

}

this.Hide();

}

}

Окно «Изменение предметов»

Title="Changing\_Subjects" Height="550" Width="1100">

<Grid>

<Grid.ColumnDefinitions>

<ColumnDefinition Width="\*"/>

<ColumnDefinition Width="2\*"/>

</Grid.ColumnDefinitions>

<Grid.RowDefinitions>

<RowDefinition Height="\*"/>

<RowDefinition Height="\*"/>

<RowDefinition Height="\*"/>

</Grid.RowDefinitions>

<Label Content="Код предмета" FontSize="20" FontFamily="Times New Roman" HorizontalAlignment="Center" VerticalAlignment="Center" Grid.Column="0" Grid.Row="0"/>

<Label Content="Название предмета" FontSize="20" FontFamily="Times New Roman" HorizontalAlignment="Center" VerticalAlignment="Center" Grid.Column="0" Grid.Row="1"/>

<TextBox Text="{Binding Subject\_code}" Height="30" Width="600" FontSize="20" FontFamily="Times New Roman" Grid.Column="1" Grid.Row="0"/>

<TextBox Text="{Binding Name\_of\_the\_subject}" Height="30" Width="600" FontSize="20" FontFamily="Times New Roman" Grid.Column="1" Grid.Row="1"/>

<Button HorizontalAlignment="Right" VerticalAlignment="Center" Height="40" Width="200" Content="Сохранить" Click="Save\_Click" Grid.Column="1" Grid.Row="3" FontSize="20" FontFamily="Times New Roman" BorderBrush="Black" Background="White" Margin="0,0,50,0"/>

<Button HorizontalAlignment="Left" VerticalAlignment="Center" Height="40" Width="200" Content="Назад" Click="Go\_back\_Click" Grid.Column="0" Grid.Row="3" FontSize="20" FontFamily="Times New Roman" BorderBrush="Black" Background="White" Margin="50,0,0,0"/>

</Grid>

public partial class Changing\_Subjects : Window

{

readonly private Subjects subjects = new Subjects();

public Changing\_Subjects(Subjects selected\_row)

{

InitializeComponent();

WindowStartupLocation = WindowStartupLocation.CenterScreen;

if (selected\_row != null)

{

subjects = selected\_row;

}

DataContext = subjects;

}

private void Go\_back\_Click(object sender, RoutedEventArgs e)

{

this.Hide();

}

private void Save\_Click(object sender, RoutedEventArgs e)

{

StringBuilder errors = new StringBuilder();

if (string.IsNullOrEmpty(subjects.Subject\_code) || (subjects.Subject\_code.Length > 10))

errors.AppendLine("Пожалуйста, напишите код предмета. Максимальная длина 10 символов");

if (string.IsNullOrWhiteSpace(subjects.Name\_of\_the\_subject) || (subjects.Name\_of\_the\_subject.Length > 50))

errors.AppendLine("Пожалуйста, напишите название предмета. Максимальная длина 50 символов");

if (errors.Length > 0)

{

MessageBox.Show(errors.ToString(), "Ошибка", MessageBoxButton.OK, MessageBoxImage.Error);

return;

}

if (subjects.Subject\_code == null)

{

Coursework1Entities.GetContext().Subjects.Add(subjects);

}

try

{

Coursework1Entities.GetContext().SaveChanges();

MessageBox.Show("Информация изменена", "Отлично", MessageBoxButton.OK, MessageBoxImage.Information);

}

catch (Exception ex)

{

MessageBox.Show(ex.Message.ToString());

}

this.Hide();

}

}

# ПРИЛОЖЕНИЕ М

Окно «Группы»

Title="Groups1" Height="550" Width="1100" IsVisibleChanged="Window\_IsVisibleChanged">

<Grid>

<Grid.ColumnDefinitions>

<ColumnDefinition Width="\*"/>

<ColumnDefinition Width="\*"/>

<ColumnDefinition Width="\*"/>

</Grid.ColumnDefinitions>

<Grid.RowDefinitions>

<RowDefinition Height="4\*"/>

<RowDefinition Height="\*"/>

</Grid.RowDefinitions>

<Button Content="Назад" Width="200" Height="40" FontSize="20" FontFamily="Times New Roman" BorderBrush="Black" Background="White" Click="Go\_back\_Click" Grid.Row="1" Grid.Column="0"/>

<Button Content="Выход" Width="200" Height="40" FontSize="20" FontFamily="Times New Roman" BorderBrush="Black" Background="White" Click="Exit\_Click" Grid.Row="1" Grid.Column="2"/>

<Button Content="Добавить" Width="200" Height="40" FontSize="20" FontFamily="Times New Roman" BorderBrush="Black" Background="White" Click="Add\_Click" Grid.Row="1" Grid.Column="1"/>

<DataGrid x:Name="Grid1" AutoGenerateColumns="False" Background="White" FontSize="14" FontFamily="Times New Roman" Grid.Row="0" Grid.ColumnSpan="3">

<DataGrid.Columns>

<DataGridTextColumn Header="Код группы" Binding="{Binding Group\_code}" Width="\*"/>

<DataGridTextColumn Header="Название группы" Binding="{Binding Name\_of\_the\_group}" Width="\*"/>

<DataGridTextColumn Header="Курс" Binding="{Binding Course}" Width="Auto"/>

<DataGridTextColumn Header="Количества студентов" Binding="{Binding Number\_of\_students}" Width="Auto"/>

<DataGridTemplateColumn Width="100">

<DataGridTemplateColumn.CellTemplate>

<DataTemplate>

<Button Content="Изменить" FontFamily="Times New Roman" BorderBrush="Black" Background="White" Click="Edit\_Click"/>

</DataTemplate>

</DataGridTemplateColumn.CellTemplate>

</DataGridTemplateColumn>

</DataGrid.Columns>

</DataGrid>

</Grid>

public partial class Groups1 : Window

{

public Groups1()

{

InitializeComponent();

WindowStartupLocation = WindowStartupLocation.CenterScreen;

}

private void Edit\_Click(object sender, RoutedEventArgs e)

{

Changing\_Groups groups = new Changing\_Groups((sender as Button).DataContext as Groups);

groups.ShowDialog();

}

private void Add\_Click(object sender, RoutedEventArgs e)

{

Adding\_Groups groups = new Adding\_Groups();

groups.ShowDialog();

}

private void Go\_back\_Click(object sender, RoutedEventArgs e)

{

this.Hide();

To\_the\_necessary\_inf inf = new To\_the\_necessary\_inf();

inf.ShowDialog();

}

private void Exit\_Click(object sender, RoutedEventArgs e)

{

Application.Current.Shutdown();

}

private void Window\_IsVisibleChanged(object sender, DependencyPropertyChangedEventArgs e)

{

if (Visibility == Visibility.Visible)

{

Coursework1Entities.GetContext().ChangeTracker.Entries().ToList().ForEach(p => p.Reload());

Grid1.ItemsSource = Coursework1Entities.GetContext().Groups.ToList();

}

}

}

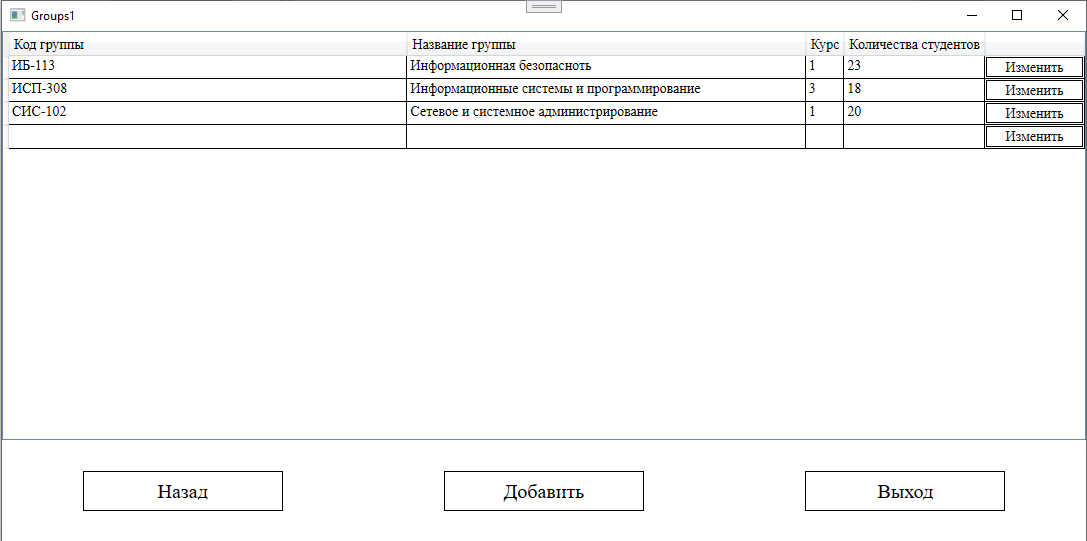


Рисунок - Окно «Группы»

Окно «Добавление групп»

Title="Adding\_Groups" Height="550" Width="1100">

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="\*"/>

<RowDefinition Height="\*"/>

<RowDefinition Height="\*"/>

<RowDefinition Height="\*"/>

<RowDefinition Height="\*"/>

</Grid.RowDefinitions>

<Grid.ColumnDefinitions>

<ColumnDefinition Width="\*"/>

<ColumnDefinition Width="2\*"/>

</Grid.ColumnDefinitions>

<Label Content="Код группы" FontSize="20" FontFamily="Times New Roman" HorizontalAlignment="Center" VerticalAlignment="Center" Grid.Column="0" Grid.Row="0"/>

<Label Content="Название группы" FontSize="20" FontFamily="Times New Roman" HorizontalAlignment="Center" VerticalAlignment="Center" Grid.Column="0" Grid.Row="1"/>

<Label Content="Курс" FontSize="20" FontFamily="Times New Roman" HorizontalAlignment="Center" VerticalAlignment="Center" Grid.Column="0" Grid.Row="2"/>

<Label Content="Количество студентов" FontSize="20" FontFamily="Times New Roman" HorizontalAlignment="Center" VerticalAlignment="Center" Grid.Column="0" Grid.Row="3"/>

<TextBox Text="{Binding Group\_code}" Width="600" Height="30" HorizontalAlignment="Center" VerticalAlignment="Center" FontSize="20" FontFamily="Times New Roman" Grid.Row="0" Grid.Column="1"/>

<TextBox Text="{Binding Name\_of\_the\_group}" Width="600" Height="30" HorizontalAlignment="Center" VerticalAlignment="Center" FontSize="20" FontFamily="Times New Roman" Grid.Row="1" Grid.Column="1"/>

<TextBox Text="{Binding Course}" Width="600" Height="30" HorizontalAlignment="Center" VerticalAlignment="Center" FontSize="20" FontFamily="Times New Roman" Grid.Row="2" Grid.Column="1"/>

<TextBox Text="{Binding Number\_of\_students}" Width="600" Height="30" HorizontalAlignment="Center" VerticalAlignment="Center" FontSize="20" FontFamily="Times New Roman" Grid.Row="3" Grid.Column="1"/>

<Button Content="Сохранить" HorizontalAlignment="Right" VerticalAlignment="Center" Width="200" Height="40" Click="Save\_Click" FontSize="20" FontFamily="Times New Roman" Grid.Row="6" Grid.Column="1" Margin="0,0,50,0" BorderBrush="Black" Background="White"/>

<Button Content="Назад" HorizontalAlignment="Left" VerticalAlignment="Center" Width="200" Height="40" Click="Go\_back\_Click" FontSize="20" FontFamily="Times New Roman" Grid.Row="6" Grid.Column="0" Margin="50,0,0,0" BorderBrush="Black" Background="White"/>

</Grid>

public partial class Adding\_Groups : Window

{

readonly private Groups groups = new Groups();

public Adding\_Groups()

{

InitializeComponent();

WindowStartupLocation = WindowStartupLocation.CenterScreen;

DataContext = groups;

}

private void Go\_back\_Click(object sender, RoutedEventArgs e)

{

this.Hide();

}

private void Save\_Click(object sender, RoutedEventArgs e)

{

StringBuilder errors = new StringBuilder();

if (string.IsNullOrEmpty(groups.Group\_code) || (groups.Group\_code.Length > 7))

errors.AppendLine("Пожалуйста, напишите код группы. Максимальна длина 7 символов");

if (string.IsNullOrEmpty(groups.Name\_of\_the\_group) || (groups.Name\_of\_the\_group.Length > 50))

errors.AppendLine("Пожалуйста, напишите название группы. Максимальная 50 символов");

if (string.IsNullOrWhiteSpace(groups.Course) || (groups.Course.Length != 1))

errors.AppendLine("Пожалуйста, напишите курс. Максимальная длина 1 символ");

if ((groups.Number\_of\_students == null) || (groups.Number\_of\_students <= 0))

errors.AppendLine("Пожалуйст, введите кол-во студентов");

if (errors.Length > 0)

{

MessageBox.Show(errors.ToString(), "Ошибка", MessageBoxButton.OK, MessageBoxImage.Error);

return;

}

if (groups.Group\_code.Length > 0)

{

Coursework1Entities.GetContext().Groups.Add(groups);

}

try

{

Coursework1Entities.GetContext().SaveChanges();

MessageBox.Show("Информация добавлена", "Отлично", MessageBoxButton.OK, MessageBoxImage.Information);

}

catch (Exception ex)

{

MessageBox.Show(ex.Message.ToString());

}

this.Hide();

}

}

Окно «Изменение групп»

Title="Changing\_Groups" Height="550" Width="1100">

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="\*"/>

<RowDefinition Height="\*"/>

<RowDefinition Height="\*"/>

<RowDefinition Height="\*"/>

<RowDefinition Height="\*"/>

</Grid.RowDefinitions>

<Grid.ColumnDefinitions>

<ColumnDefinition Width="\*"/>

<ColumnDefinition Width="2\*"/>

</Grid.ColumnDefinitions>

<Label Content="Код группы" FontSize="20" FontFamily="Times New Roman" HorizontalAlignment="Center" VerticalAlignment="Center" Grid.Column="0" Grid.Row="0"/>

<Label Content="Название группы" FontSize="20" FontFamily="Times New Roman" HorizontalAlignment="Center" VerticalAlignment="Center" Grid.Column="0" Grid.Row="1"/>

<Label Content="Курс" FontSize="20" FontFamily="Times New Roman" HorizontalAlignment="Center" VerticalAlignment="Center" Grid.Column="0" Grid.Row="2"/>

<Label Content="Количество студентов" FontSize="20" FontFamily="Times New Roman" HorizontalAlignment="Center" VerticalAlignment="Center" Grid.Column="0" Grid.Row="3"/>

<TextBox Text="{Binding Group\_code}" Width="600" Height="30" HorizontalAlignment="Center" VerticalAlignment="Center" FontSize="20" FontFamily="Times New Roman" Grid.Row="0" Grid.Column="1"/>

<TextBox Text="{Binding Name\_of\_the\_group}" Width="600" Height="30" HorizontalAlignment="Center" VerticalAlignment="Center" FontSize="20" FontFamily="Times New Roman" Grid.Row="1" Grid.Column="1"/>

<TextBox Text="{Binding Course}" Width="600" Height="30" HorizontalAlignment="Center" VerticalAlignment="Center" FontSize="20" FontFamily="Times New Roman" Grid.Row="2" Grid.Column="1"/>

<TextBox Text="{Binding Number\_of\_students}" Width="600" Height="30" HorizontalAlignment="Center" VerticalAlignment="Center" FontSize="20" FontFamily="Times New Roman" Grid.Row="3" Grid.Column="1"/>

<Button Content="Сохранить" HorizontalAlignment="Right" VerticalAlignment="Center" Width="200" Height="40" Click="Save\_Click" FontSize="20" FontFamily="Times New Roman" Grid.Row="6" Grid.Column="1" Margin="0,0,50,0" BorderBrush="Black" Background="White"/>

<Button Content="Назад" HorizontalAlignment="Left" VerticalAlignment="Center" Width="200" Height="40" Click="Go\_back\_Click" FontSize="20" FontFamily="Times New Roman" Grid.Row="6" Grid.Column="0" Margin="50,0,0,0" BorderBrush="Black" Background="White"/>

</Grid>

public partial class Changing\_Groups : Window

{

readonly private Groups groups = new Groups();

public Changing\_Groups(Groups selected\_row)

{

InitializeComponent();

WindowStartupLocation = WindowStartupLocation.CenterScreen;

if (selected\_row != null)

{

groups = selected\_row;

}

DataContext = groups;

}

private void Go\_back\_Click(object sender, RoutedEventArgs e)

{

this.Hide();

}

private void Save\_Click(object sender, RoutedEventArgs e)

{

StringBuilder errors = new StringBuilder();

if (string.IsNullOrEmpty(groups.Group\_code) || (groups.Group\_code.Length > 7))

errors.AppendLine("Пожалуйста, напишите код группы. Максимальна длина 7 символов");

if (string.IsNullOrEmpty(groups.Name\_of\_the\_group) || (groups.Name\_of\_the\_group.Length > 50))

errors.AppendLine("Пожалуйста, напишите название группы. Максимальная 50 символов");

if (string.IsNullOrWhiteSpace(groups.Course) || (groups.Course.Length != 1))

errors.AppendLine("Пожалуйста, напишите курс. Максимальная длина 1 символ");

if ((groups.Number\_of\_students == null) || (groups.Number\_of\_students <= 0))

errors.AppendLine("Пожалуйст, введите кол-во студентов");

if (errors.Length > 0)

{

MessageBox.Show(errors.ToString(), "Ошибка", MessageBoxButton.OK, MessageBoxImage.Error);

return;

}

if (groups.Group\_code == null)

{

Coursework1Entities.GetContext().Groups.Add(groups);

}

try

{

Coursework1Entities.GetContext().SaveChanges();

MessageBox.Show("Информация изменена", "Отлично", MessageBoxButton.OK, MessageBoxImage.Information);

}

catch (Exception ex)

{

MessageBox.Show(ex.Message.ToString());

}

this.Hide();

}

}

# ПРИЛОЖЕНИЕ Н

Окно «Отчет о проведенных часах по предметам»

Title="Report\_for\_subjects" Height="550" Width="1100">

<Grid x:Name="RAS">

<Grid.ColumnDefinitions>

<ColumnDefinition Width="\*"/>

<ColumnDefinition Width="\*"/>

<ColumnDefinition Width="\*"/>

</Grid.ColumnDefinitions>

<Grid.RowDefinitions>

<RowDefinition Height="1\*"/>

<RowDefinition Height="1\*"/>

<RowDefinition Height="4\*"/>

<RowDefinition Height="1\*"/>

</Grid.RowDefinitions>

<Label Content="Отчет о проведенных часах по предметам" HorizontalAlignment="Center" VerticalAlignment="Center" FontSize="36" FontFamily="Times New Roman" Grid.ColumnSpan="3" Grid.Row="0"/>

<Label Content="Проведенные часы" HorizontalAlignment="Center" VerticalAlignment="Center" FontSize="32" FontFamily="Times New Roman" Grid.Column="0" Grid.Row="1"/>

<Label Content="Название предмета" HorizontalAlignment="Center" VerticalAlignment="Center" FontSize="32" FontFamily="Times New Roman" Grid.Column="1" Grid.Row="1"/>

<Label Content="Преподаватель" HorizontalAlignment="Center" VerticalAlignment="Center" FontSize="32" FontFamily="Times New Roman" Grid.Column="2" Grid.Row="1"/>

<Border Grid.Column="0" Grid.Row="2" BorderBrush="Black" Background="Transparent" BorderThickness="1">

<TextBlock x:Name="TBconducted\_hours" HorizontalAlignment="Center" VerticalAlignment="Center" FontSize="40" FontFamily="Times New Roman"/>

</Border>

<Border Grid.Column="1" Grid.Row="2" BorderBrush="Black" Background="Transparent" BorderThickness="1">

<TextBlock x:Name="TBsubject\_s\_name" HorizontalAlignment="Center" VerticalAlignment="Center" FontSize="40" FontFamily="Times New Roman"/>

</Border>

<Border Grid.Column="2" Grid.Row="2" BorderBrush="Black" Background="Transparent" BorderThickness="1">

<TextBlock x:Name="TBteacher" HorizontalAlignment="Center" VerticalAlignment="Center" FontSize="40" FontFamily="Times New Roman"/>

</Border>

<Button x:Name="Go\_back" Content="Назад" HorizontalAlignment="Left" Margin="20,0,0,20" VerticalAlignment="Bottom" Width="200" Height="40" FontSize="26" FontFamily="Times New Roman" BorderBrush="Black" Background="White" Click="Go\_back\_Click" Grid.Column="0" Grid.Row="3"/>

<Button x:Name="Exit" Content="Выход" HorizontalAlignment="Right" Margin="0,0,20,20" VerticalAlignment="Bottom" Width="200" Height="40" FontSize="26" FontFamily="Times New Roman" BorderBrush="Black" Background="White" Click="Exit\_Click" Grid.Column="2" Grid.Row="3"/>

<Button Content="Печать" HorizontalAlignment="Center" VerticalAlignment="Center" Width="200" Height="40" FontSize="26" FontFamily="Times New Roman" BorderBrush="Black" Background="White" Click="Print\_Click" Grid.Column="1" Grid.Row="3"/>

</Grid>

public partial class Report\_for\_subjects : Window

{

readonly private List<Accounts> accounts;

public Report\_for\_subjects(string subjects)

{

InitializeComponent();

WindowStartupLocation = WindowStartupLocation.CenterScreen;

TBsubject\_s\_name.Text = subjects;

accounts = Coursework1Entities.GetContext().Accounts.ToList();

var obj = accounts.Where(a => a.Subject\_code == subjects);

int conducted\_hours = 0;

foreach (var item in obj)

{

conducted\_hours += (int)item.Classes\_held \* 2;

TBteacher.Text = item.Login;

}

TBconducted\_hours.Text = conducted\_hours.ToString();

}

private void Go\_back\_Click(object sender, RoutedEventArgs e)

{

this.Hide();

Creating\_reports creating\_Reports = new Creating\_reports();

creating\_Reports.ShowDialog();

}

private void Exit\_Click(object sender, RoutedEventArgs e)

{

Application.Current.Shutdown();

}

private void Print\_Click(object sender, RoutedEventArgs e)

{

PrintDialog print = new PrintDialog();

Go\_back.Visibility = Visibility.Hidden;

Exit.Visibility = Visibility.Hidden;

if (print.ShowDialog() == true)

{

print.PrintVisual(RAS, "Печать");

}

Go\_back.Visibility = Visibility.Visible;

Exit.Visibility = Visibility.Visible;

}

}

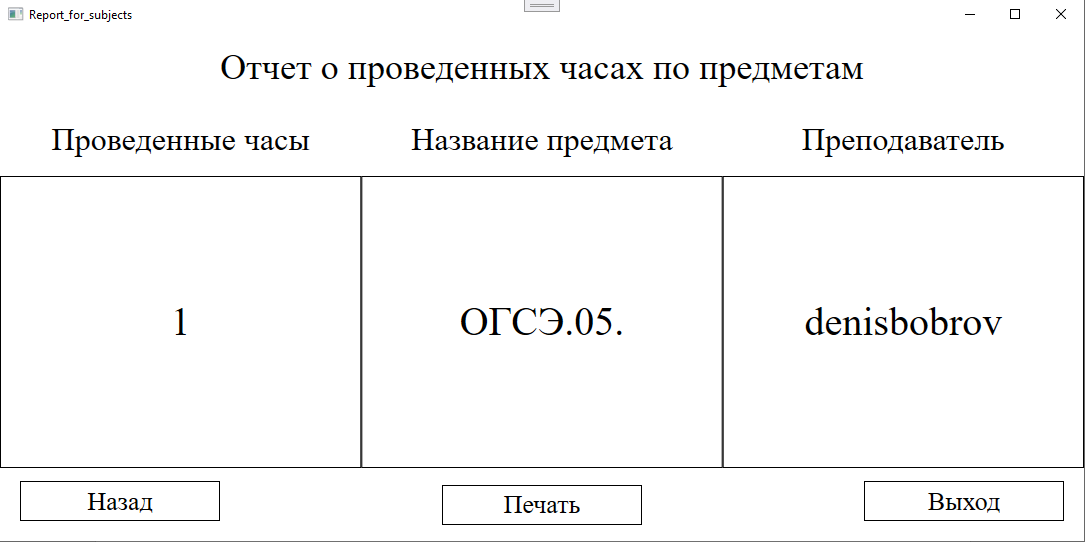


Рисунок - Окно «Отчет о проведенных часах по предметам»

Окно «Отчет о проведенных часах по дням преподавателем»

Title="Report\_like\_accounts" Height="550" Width="1100">

<Grid x:Name="RLA">

<Grid.ColumnDefinitions>

<ColumnDefinition Width="\*"/>

<ColumnDefinition Width="\*"/>

<ColumnDefinition Width="\*"/>

</Grid.ColumnDefinitions>

<Grid.RowDefinitions>

<RowDefinition Height="1\*"/>

<RowDefinition Height="1\*"/>

<RowDefinition Height="4\*"/>

<RowDefinition Height="1\*"/>

</Grid.RowDefinitions>

<Label Content="Отчет о проведенных часах по дням преподавателем" HorizontalAlignment="Center" VerticalAlignment="Center" FontSize="36" FontFamily="Times New Roman" Grid.ColumnSpan="3" Grid.Row="0"/>

<Label Content="Проведенные часы" HorizontalAlignment="Center" VerticalAlignment="Center" FontSize="32" FontFamily="Times New Roman" Grid.Column="0" Grid.Row="1"/>

<Label Content="Дата" HorizontalAlignment="Center" VerticalAlignment="Center" FontSize="32" FontFamily="Times New Roman" Grid.Column="1" Grid.Row="1"/>

<Label Content="Преподаватель" HorizontalAlignment="Center" VerticalAlignment="Center" FontSize="32" FontFamily="Times New Roman" Grid.Column="2" Grid.Row="1"/>

<Border Grid.Column="0" Grid.Row="2" BorderBrush="Black" Background="Transparent" BorderThickness="1">

<TextBlock x:Name="TBconducted\_hours" HorizontalAlignment="Center" VerticalAlignment="Center" FontSize="40" FontFamily="Times New Roman"/>

</Border>

<Border Grid.Column="1" Grid.Row="2" BorderBrush="Black" Background="Transparent" BorderThickness="1">

<DatePickerTextBox x:Name="TBDate" HorizontalAlignment="Center" VerticalAlignment="Center" FontSize="40" FontFamily="Times New Roman"/>

</Border>

<Border Grid.Column="2" Grid.Row="2" BorderBrush="Black" Background="Transparent" BorderThickness="1">

<TextBlock x:Name="TBteacher" HorizontalAlignment="Center" VerticalAlignment="Center" FontSize="40" FontFamily="Times New Roman"/>

</Border>

<Button x:Name="Go\_back" Content="Назад" HorizontalAlignment="Left" Margin="20,0,0,20" VerticalAlignment="Bottom" Width="200" Height="40" FontSize="26" FontFamily="Times New Roman" BorderBrush="Black" Background="White" Click="Go\_back\_Click" Grid.Column="0" Grid.Row="3"/>

<Button x:Name="Exit" Content="Выход" HorizontalAlignment="Right" Margin="0,0,20,20" VerticalAlignment="Bottom" Width="200" Height="40" FontSize="26" FontFamily="Times New Roman" BorderBrush="Black" Background="White" Click="Exit\_Click" Grid.Column="2" Grid.Row="3"/>

<Button Content="Печать" HorizontalAlignment="Center" VerticalAlignment="Center" Width="200" Height="40" FontSize="26" FontFamily="Times New Roman" BorderBrush="Black" Background="White" Click="Print\_Click" Grid.Column="1" Grid.Row="3"/>

</Grid>

public partial class Report\_like\_account : Window

{

readonly private List<Accounts> accounts;

public Report\_like\_account(string subjects)

{

InitializeComponent();

WindowStartupLocation = WindowStartupLocation.CenterScreen;

accounts = Coursework1Entities.GetContext().Accounts.ToList();

var obj = accounts.Where(a => a.Subject\_code == subjects && a.Date == DateTime.Today);

int hours = 0;

foreach (var item in obj)

{

TBDate.Text = item.Date.ToString();

TBteacher.Text = item.Login;

hours = (int)item.Classes\_held \* 2;

}

TBconducted\_hours.Text = hours.ToString();

}

private void Go\_back\_Click(object sender, RoutedEventArgs e)

{

this.Hide();

Creating\_reports creating\_Reports = new Creating\_reports();

creating\_Reports.ShowDialog();

}

private void Exit\_Click(object sender, RoutedEventArgs e)

{

Application.Current.Shutdown();

}

private void Print\_Click(object sender, RoutedEventArgs e)

{

PrintDialog print = new PrintDialog();

Go\_back.Visibility = Visibility.Hidden;

Exit.Visibility = Visibility.Hidden;

if (print.ShowDialog() == true)

{

print.PrintVisual(RLA, "Печать");

}

Go\_back.Visibility = Visibility.Visible;

Exit.Visibility = Visibility.Visible;

}

}

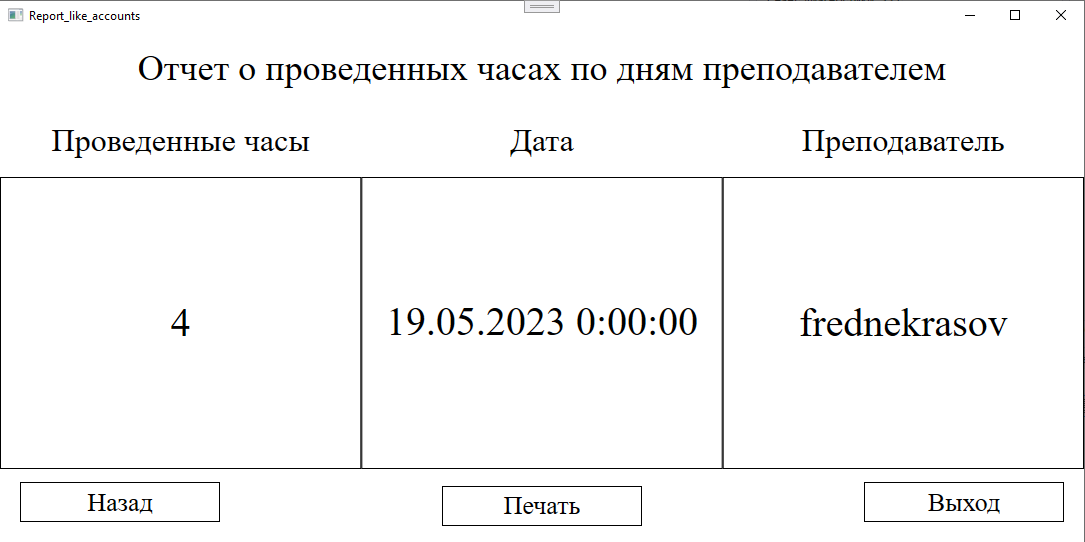


Рисунок - Окно «Отчет о проведенных часах по дням преподавателем»