**CОДЕРЖАНИЕ**

[Введение 6](#_Toc104138326)

[Глава 1. Анализ предметной области и определение требований к программному продукту 7](#_Toc104138327)

[1.1. Анализ предметной области 7](#_Toc104138328)

[1.2. Анализ технического задания на программный продукт 8](#_Toc104138329)

[1.3. Выбор инструментальных программных средств 10](#_Toc104138330)

[1.4. Определение системных требований к программному продукту 12](#_Toc104138331)

[Глава 2. Разработка программного продукта 14](#_Toc104138332)

[2.1. Проектирование структуры программного продукта 14](#_Toc104138333)

[2.2. Создание программного продукта 19](#_Toc104138334)

[2.3. Тестирование программного продукта 38](#_Toc104138335)

[Глава 3. Экономическая часть 39](#_Toc104138336)

[Глава 4. Охрана труда и безопасность жизнедеятельности 45](#_Toc104138337)

[Заключение 53](#_Toc104138338)

[Список использованных источников 54](#_Toc104138339)

Приложение А

Приложение Б

Приложение В

Приложение Г

# ВВЕДЕНИЕ

Одна из основной сложности организации работы учебного процесса является автоматизация распределения учебной нагрузки и формирование учебных поручений педагогам.

Учебная нагрузка – основа педагогического рабочего времени, устанавливаемая руководством учебного заведения, с учетом профессиональной компетентности педагога, количества часов по учебному плану и учебных программ, и множества других особенностей учебного заведения.

Оперативный расчет учебной нагрузки педагога позволяет повысить скорость управления учебным процессом. Таким образом, автоматизация оформления расписания колоссально снижает объем затрат рабочего времени на разработку организации учебного процесса и повышает качество управления учебного процесса.

Актуальность данной темы заключается в том, что разработка программного продукта облегчит процесс распределения учебной нагрузки, ускорит разработку и проконтролирует правильность заполнения, минимизируя экономические затраты.

Целью выпускной квалификационной работы является разработка программного продукта для составления расписания.

Для достижения поставленной цели, необходимо решить ряд следующих задач:

1. Проанализировать предметную область, техническое задание и возможные способы реализации поставленной цели;
2. Выбрать инструментальные программные средства для разработки программного продукта и определить системные требования к программному продукту;
3. Спроектировать структуру программного продукта, создать программный продукт и протестировать его;
4. Рассчитать затраты и цену на реализацию программного продукта;
5. Разработать программную документацию.

# ГЛАВА 1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ И ОПРЕДЕЛЕНИЕ ТРЕБОВАНИЙ К ПРОГРАММНОМУ ПРОДУКТУ

## Анализ предметной области

Для анализа предметной области необходимо использовать методологию моделирования IDEF0, которая облегчает и стандартизирует описание процесса работы какого-либо объекта.

IDEF0 – методология функционального моделирования и графическая нотация, предназначенная для формализации и описания бизнес-процессов. Отличительной особенностью IDEF0 является её акцент на соподчинённость объектов. В IDEF0 рассматриваются логические отношения между работами, а не их временная последовательность.[[1]](#footnote-1)

Один из основных компонентов учебного процесса является расписание занятий, который регламентирует трудовую деятельности и влияет на творческую отдачу педагогов. По своей сути, он является фактором оптимизации ограниченных ресурсов в виде педагогического состава. Технологию же разработки расписания следует воспринимать не только как трудоемкий процесс, объект механизации и автоматизации с использованием ЭВМ, но и также как предложение оптимального управления. Ниже, на рисунке 1, представлена контекстная диаграмма процесса составления расписания.

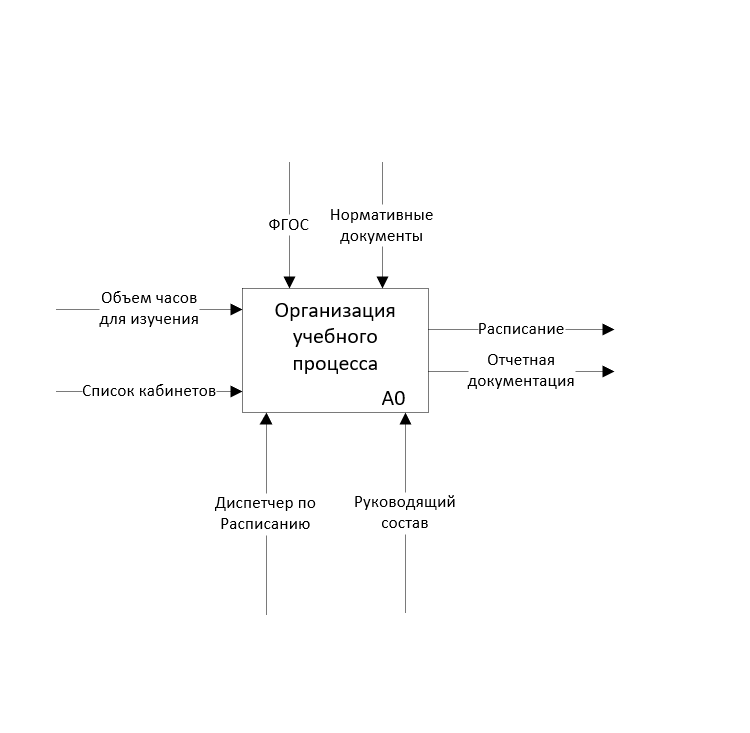


Рисунок 1 – Контекстная диаграмма процесса составления расписания

Задачу составления расписания не стоит рассматривать только как некую программу, реализующую функцию механического распределения занятий в начале учебного года, на экономический эффект от более эффективного использования трудовых ресурсов может быть достигнут только в результате кропотливой работы по управлению этими трудовыми ресурсами. Расписание здесь является инструментом управления, которая содержит в себе средства для составления оптимального распределения учебной нагрузки. Кроме этого, оптимальное управление такой сложной системой невозможно без накопления некоей статистической информации о процессах, происходящих в системе. Потому сама задача составления оптимального расписания является лишь частью сложной системы управления учебным процессом. Ниже, на рисунке 2, представлена диаграмма декомпозиции процесса составления расписания как есть сейчас.

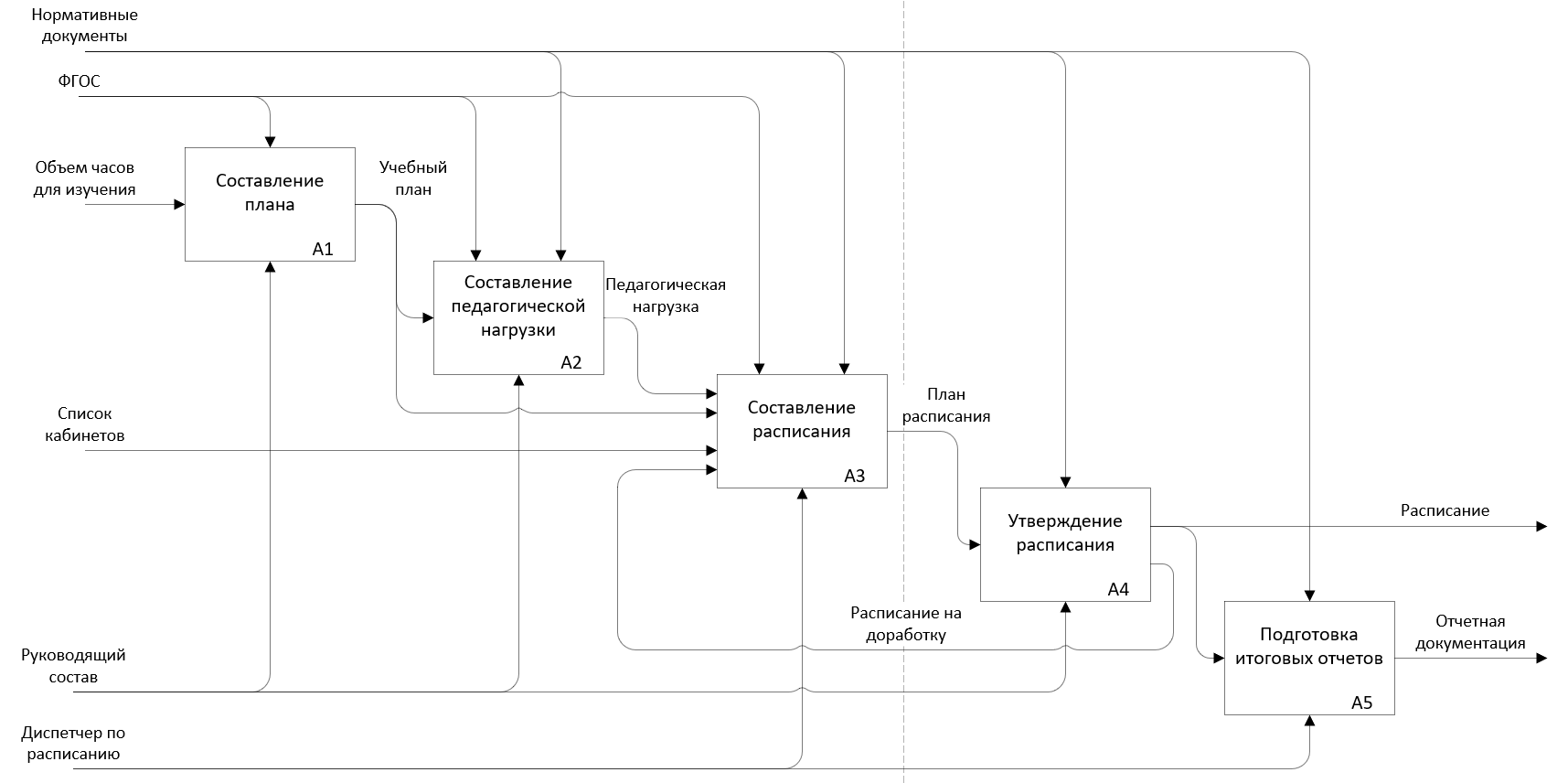


Рисунок 2 – Диаграмма декомпозиции процесса составления расписания «AS-IS» (Как есть)

## Анализ технического задания на программный продукт

Согласно техническому заданию, необходимо разработать программу для составления расписания МОУ «Гимназия – школа с. Ивантеевка Саратовкой области». Она предназначена для проектирования расписания и распределения учебной нагрузки образовательного учреждения.

Требования, предъявляемые к программному продукту, согласно техническому заданию:

* Простой и понятный интерфейс;
* Хранение данных об учебном плане, классах, педагогах, дисциплинах, кабинетах;
* Возможность составления и экспортирования расписания;
* Возможность формирования отчета.

Программу для составления расписания можно сделать несколькими способами:

1. С помощью инструментов Microsoft Office (Microsoft Access и Excel):

Создание программы для составления расписания с помощью стандартных функций и макросов.

Достоинства:

* Простота в разработке;
* Возможность реализации хранения данных;
* Не требуется глубоких знаний в области программирования.

Недостатки:

* Необходим опыт работы с инструментами Microsoft Office;
* Отсутствие возможности взаимодействия с продуктами других компаний;
* Трудность реализации узконаправленных задач;
* Обязательная установка Microsoft Office;

1. При помощи языков программирования. С их помощью можно решить любые узконаправленные задачи, которые могут возникнуть при разработке и реализовать множество возможностей и функций.

Достоинства:

* Реализация простого и понятного интерфейса;
* Большой функционал среды программирования;
* Возможность реализации хранения данных;
* Возможность реализации экспортирования;
* Возможность реализации узконаправленных задач;
* Возможность взаимодействия с продуктами компании.

Недостатки:

* Знание языков программирования;
* Установка необходимой среды программирования.

В результате исследования способов реализации цели был выбран вариант создания программного продукта с помощью языков программирования высокого уровня. Программа для составления расписания, созданная с помощью языков программирования, удовлетворяет все требования технического задания.

Ниже, на рисунке 3, представлена диаграмма декомпозиции составления расписания как будет после внедрения программы.

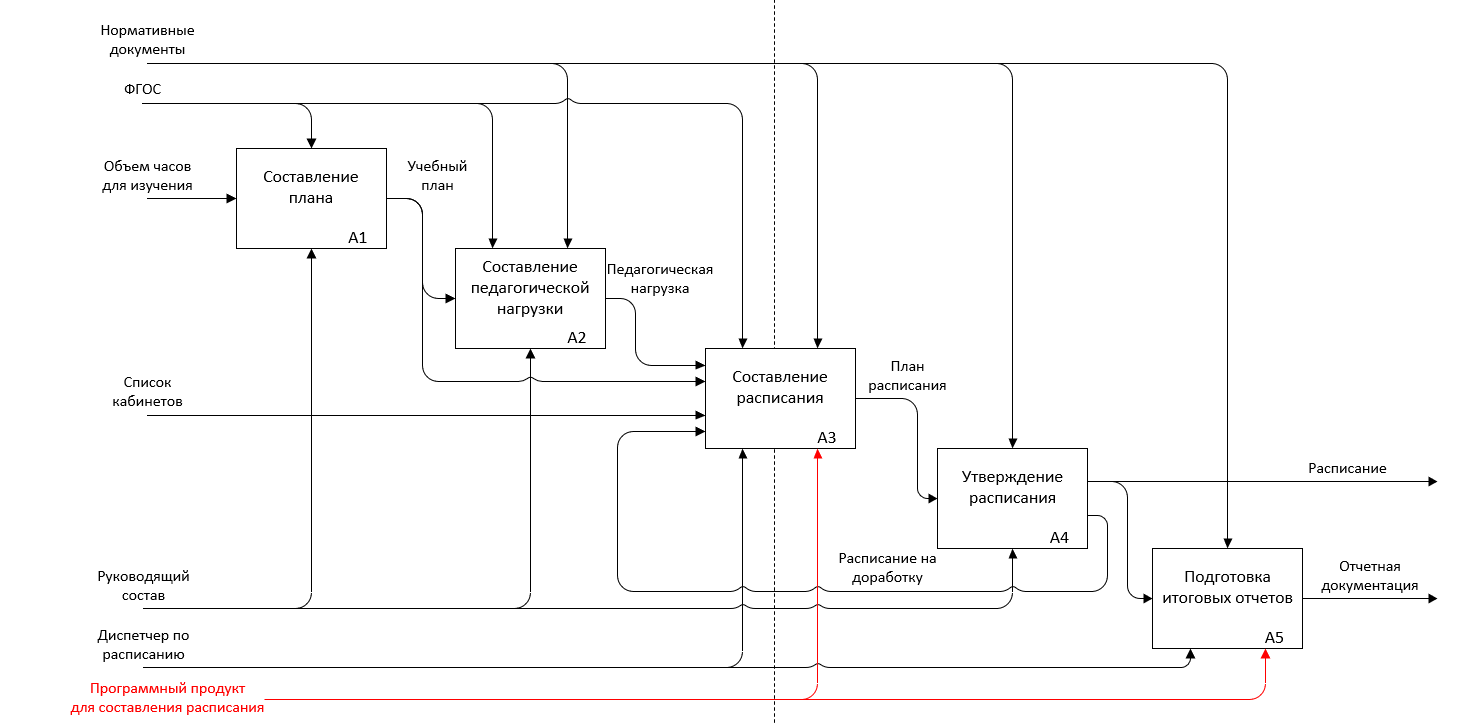


Рисунок 3 - Диаграмма декомпозиции процесса составления расписания «TO-BE» (Как будет)

## Выбор инструментальных программных средств

Существует огромное количество разнообразных программных средств, предназначенных для создания базы данных и программ. Рассмотрим некоторые из них:

1. Microsoft Visual Studio

Линейка продуктов компании Microsoft, включающих интегрированную среду разработки программного обеспечения и ряд других инструментов. Данные продукты позволяют разрабатывать как консольные приложения, так и игры и приложения с графическим интерфейсом, в том числе с поддержкой технологии Windows Forms, а также веб-сайты, веб-приложения, веб-службы как в родном, так и в управляемом кодах для всех платформ, поддерживаемых Windows, Windows Mobile, Windows CE, .NET Framework, Xbox, Windows Phone .NET Compact Framework и Silverlight.

1. Microsoft SQL Server

Система управления базами данных обрабатывает команды на языке запросов Transact-SQL (Transact Structured Query Language), который разработан совместно с Microsoft и Sybase.

1. Microsoft SQL Server Management Studio

Утилита для конфигурирования, управления и администрирования всех компонентов Microsoft SQL Server. Утилита включает скриптовый редактор и графическую программу, которая работает с объектами и настройками сервера.

1. Embarcadero Delphi

Интегрированная среда разработки ПО для Microsoft Windows, Mac OS, iOS и Android на языке Delphi (ранее носившем название Object Pascal), созданная первоначально фирмой Borland и на данный момент принадлежащая и разрабатываемая Embarcadero Technologies. Embarcadero Delphi является частью пакета Embarcadero RAD Studio и поставляется в четырёх редакциях: Community (распространяется бесплатно и имеет ограниченную лицензию на использование в коммерческих целях), Professional, Enterprise и Architect.

1. PyCharm

Интегрированная среда разработки для языка программирования Python. Предоставляет средства для анализа кода, графический отладчик, инструмент для запуска юнит-тестов и поддерживает веб-разработку на Django. PyCharm разработана компанией JetBrains на основе IntelliJ IDEA.

1. DB Browser for SQLite

Высококачественный визуальный инструмент с открытым исходным кодом для создания, проектирования и редактирования файлов базы данных, совместимых с SQLite, для пользователей и разработчиков, желающих создавать базы данных, искать и редактировать данные. Использует знакомый интерфейс, похожий на электронную таблицу, и не нужно изучать сложные команды SQL.

1. Eclipse

Eclipse является бесплатной программной платформой с открытым исходным кодом, контролируется организацией Eclipse Foundation. Написана на языке программирования Java и основной целью её создания является повышение продуктивности процесса разработки программного обеспечения. Претендует на статус наиболее популярной Java IDE и является единственным конкурентом такой мощной платформы как NetBeans.

Проанализировав достоинства и недостатки различных инструментальных программных средств, можно сделать вывод что для реализации программного продукта наиболее подходящими будут Microsoft Visual Studio и DB Browser for SQLite.

## Определение системных требований к программному продукту

Системные требования определяют соответствие характеристик компьютера для корректной работы программного продукта. Требования к системе могут быть аппаратными и программными.

Для полноценной работы программного продукта необходимо соответствовать следующим аппаратным системным требованиям:

* Процессор: 2-ядерный с частотой 1,6 ГГц и выше;
* Объем оперативной памяти: 2 ГБ и более (с последующим увеличением по мере роста базы данных);
* Минимальное свободное место на жестком диске: 20 МБ и более (c последующим увеличением по мере роста базы данных);
* Видеоадаптер с поддержкой DirectX 9;
* Монитор: 1280x720, 60 Гц;
* Периферия: клавиатура и мышь;

Также необходимо соответствовать следующим программным системным требованиям:

* Операционная система: Windows 8 32-разрядная и лучше;
* Среда выполнения .NET: .NET Runtime 6.0 32-разрядная;
* Программа для работы с электронными таблицами: Microsoft Excel 2016 и новее.

# ГЛАВА 2. РАЗРАБОТКА ПРОГРАММНОГО ПРОДУКТА

## Проектирование структуры программного продукта

Изучив предметную область организации и техническое задание, можно составить диаграмму вариантов использования.

Диаграмма вариантов использования в UML — диаграмма, отражающая отношения между актерами и прецедентами и являющаяся составной частью модели прецедентов, позволяющей описать систему на концептуальном уровне.[[2]](#footnote-2)

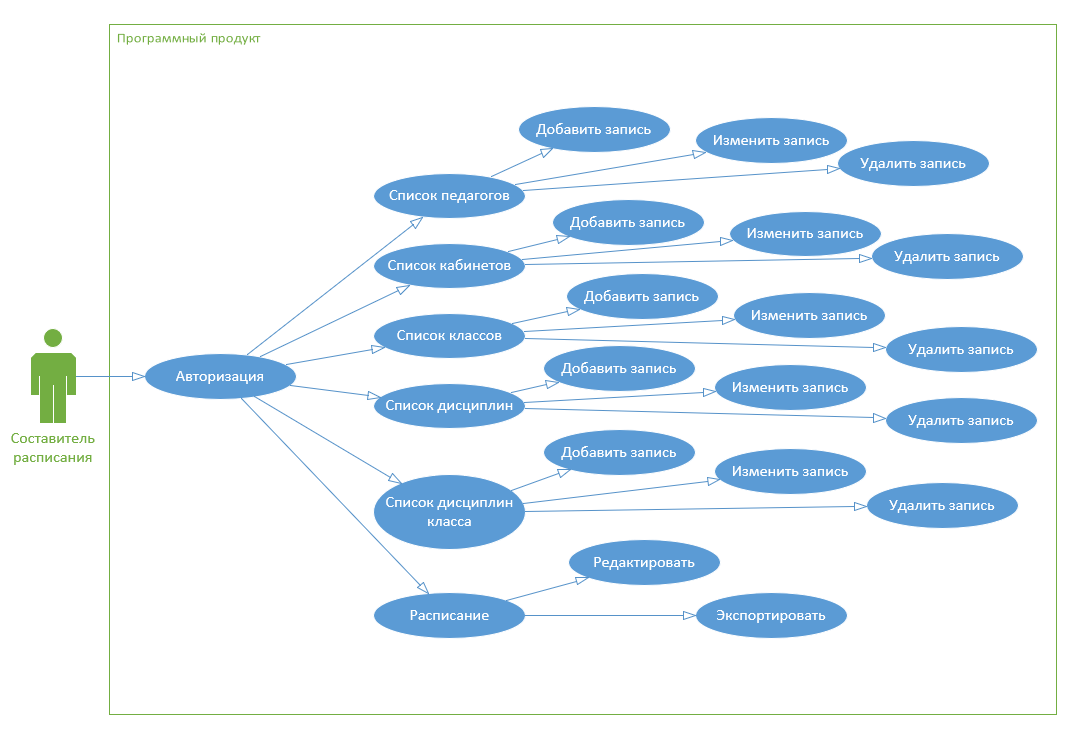


Рисунок 4 – Диаграмма вариантов использования

Из диаграммы вариантов использования (рисунок 4) видно, что необходимо организовать хранение данных. Для этого необходимо спроектировать базу данных.

База данных – набор взаимосвязанных сведений (фактов), относящихся к определенной предметной области, организованных по определенным правилам, которые могут предусматривать их клиаративное представление, хранение и манипулирование ими.[[3]](#footnote-3)

Сущность – это любой объект в базе данных, который можно выделить исходя из сути предметной области для которой разрабатывается эта база данных.[[4]](#footnote-4)

Для хранения данных в базе данных было спроектированы следующие сущности: «Учителя», «Кабинеты», «Классы», «Дисциплины», «Дисциплины классов», «Расписание».

Таблица 1 – Сущность «Учитель»

|  |  |  |  |
| --- | --- | --- | --- |
| Ключ | Имя столбца | Тип данных | Допуск Null значения |
| PK | TeacherId | INTEGER | Нет |
|  | Surname | TEXT | Да |
|  | Name | TEXT | Да |
|  | Patronymic | TEXT | Да |
|  | Birthday | TEXT | Да |
|  | Passport | TEXT | Да |
|  | PhoneNumber | TEXT | Да |
|  | Photo | IMAGE | Да |
|  | Education | TEXT | Да |

Таблица 2 – Сущность «Кабинет»

|  |  |  |  |
| --- | --- | --- | --- |
| Ключ | Имя столбца | Тип данных | Допуск Null значения |
| PK | CabinetId | INTEGER | Нет |
| FK | TeacherId | INTEGER | Да |
|  | Name | TEXT | Да |
|  | CountPlaces | INTEGER | Да |
|  | Photo | IMAGE | Да |
|  | Description | TEXT | Да |

Таблица 3 – Сущность «Класс»

|  |  |  |  |
| --- | --- | --- | --- |
| Ключ | Имя столбца | Тип данных | Допуск Null значения |
| PK | ClassId | INTEGER | Нет |
| FK | TeacherId | INTEGER | Да |
| FK | CabinetId | INTEGER | Да |
|  | Name | TEXT | Да |
|  | CountPupils | INTEGER | Да |
|  | MaxDifficulty | INTEGER | Да |
|  | Photo | IMAGE | Да |

Таблица 4 – Сущность «Дисциплина»

|  |  |  |  |
| --- | --- | --- | --- |
| Ключ | Имя столбца | Тип данных | Допуск Null значения |
| PK | LessonId | INTEGER | Нет |
|  | Name | TEXT | Да |
|  | Description | TEXT | Да |

Таблица 5 – Сущность «Дисциплина класса»

|  |  |  |  |
| --- | --- | --- | --- |
| Ключ | Имя столбца | Тип данных | Допуск Null значения |
| PK | ClassLessonId | INTEGER | Нет |
| FK | ClassId | INTEGER | Да |
| FK | LessonId | INTEGER | Да |
| FK | TeacherId | INTEGER | Да |
| FK | PairClassLessonId | INTEGER | Да |
| FK | DefaultCabinetId | INTEGER | Да |
|  | CountLesson | INTEGER | Да |
|  | Difficulty | INTEGER | Да |

Таблица 6 – Сущность «Расписание»

|  |  |  |  |
| --- | --- | --- | --- |
| Ключ | Имя столбца | Тип данных | Допуск Null значения |
| PK | Date | TEXT | Нет |
| PK | NumberLesson | INTEGER | Нет |
| PK, FK | ClassLessonId | INTEGER | Нет |
|  | CabinetId | INTEGER | Да |
|  | PairCabinetId | INTEGER | Да |
|  | IsBold | INTEGER | Нет |

Также добавлена сущность «Пароль» (таблица 7), которая будет хранить доступ к программе.

Таблица 7 – Сущность «Пароль»

|  |  |  |  |
| --- | --- | --- | --- |
| Ключ | Имя столбца | Тип данных | Допуск Null значения |
| PK | PasswordId | INTEGER | Нет |
|  | Hash | TEXT | Нет |

По описанным сущностям была построена ER-диаграмма (рисунок 5), которая покажет наглядное взаимодействие сущностей базы данных.

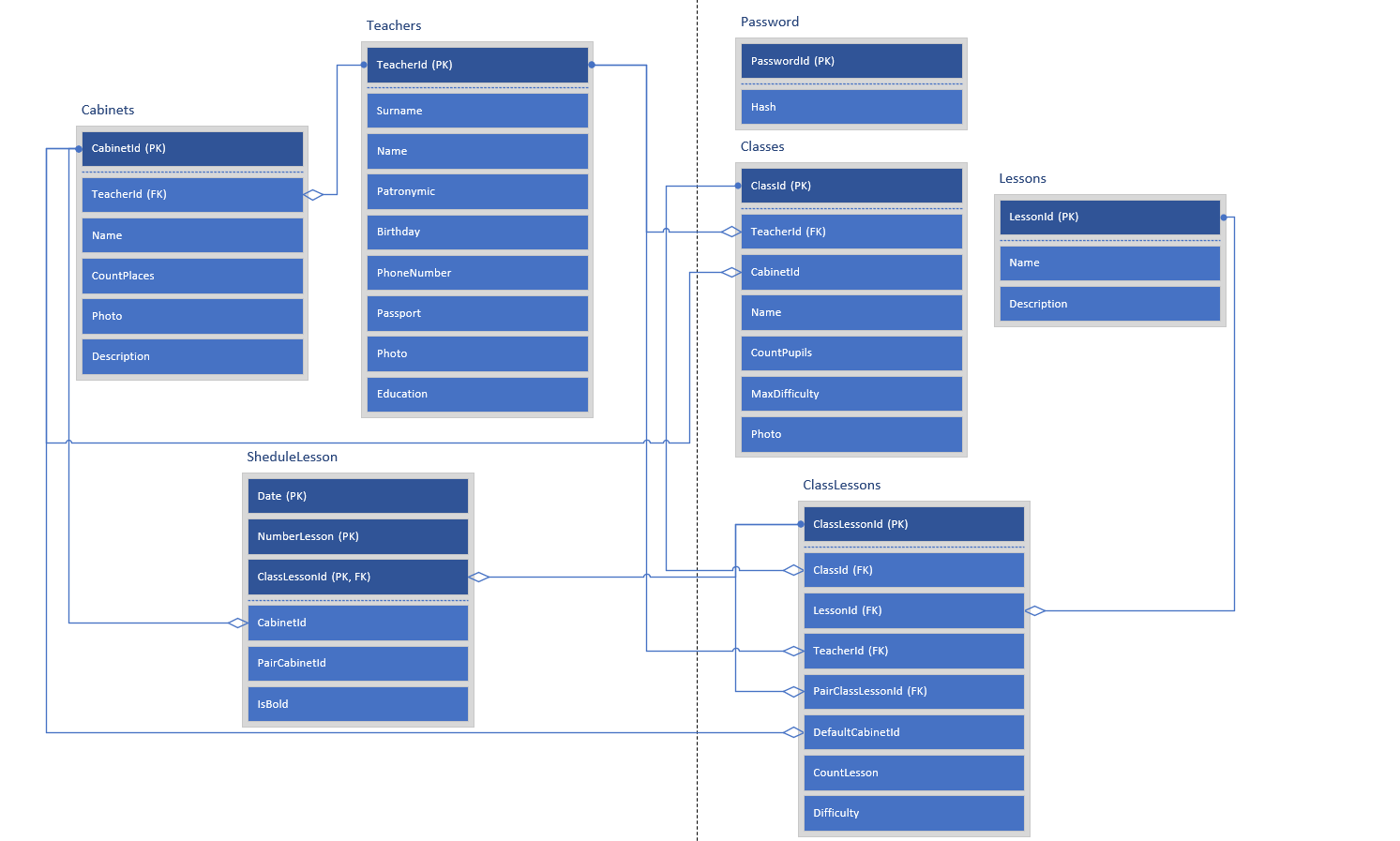


Рисунок 5 – ER-диаграмма хранения данных

Для написания программы необходима платформа разработки. Для этого была выбрана .NET Core платформа от компании Microsoft на языке программирования C# с технологией создания форм WPF.

Язык C# является объектно-ориентированным языком (ООП) программирования. Объектно-ориентированное программирование – это методология программирования, основанная на представлении программы в виде совокупности взаимодействующих объектов, каждый из которых является экземпляром определённого класса, а классы образуют иерархию наследования.[[5]](#footnote-5)

Платформа .NET – это модульная платформа для разработки программного обеспечения с открытым исходным кодом. Поддерживает следующие языки программирования: C#, Visual Basic .NET (частично) и F#.

Windows Presentation Foundation (WPF) – система для построения клиентских приложений Windows с визуально привлекательными возможностями взаимодействия с пользователем, графическая (презентационная) подсистема в составе .NET Framework (начиная с версии 3.0), использующая язык XAML.[[6]](#footnote-6)

Для хранения данных выбрана система управления базами данных SQLite. Для взаимодействия СУБД SQLite и .NET-платформой задействована библиотека от компании Microsoft Entity Framework Core.

Система управления базами данных (СУБД) – это комплекс программно-языковых средств, позволяющих создать базы данных и управлять данными.[[7]](#footnote-7)

Для экспортирования данных в документ была выбрана программа Microsoft Office Excel, для этого существует библиотека Microsoft Office Interop Excel, с помощью которой можно взаимодействовать с таблицей Excel-документа непосредственно языком программирования C#.

Ниже представлена схема взаимодействия всех компонентов программы (рисунок 6).

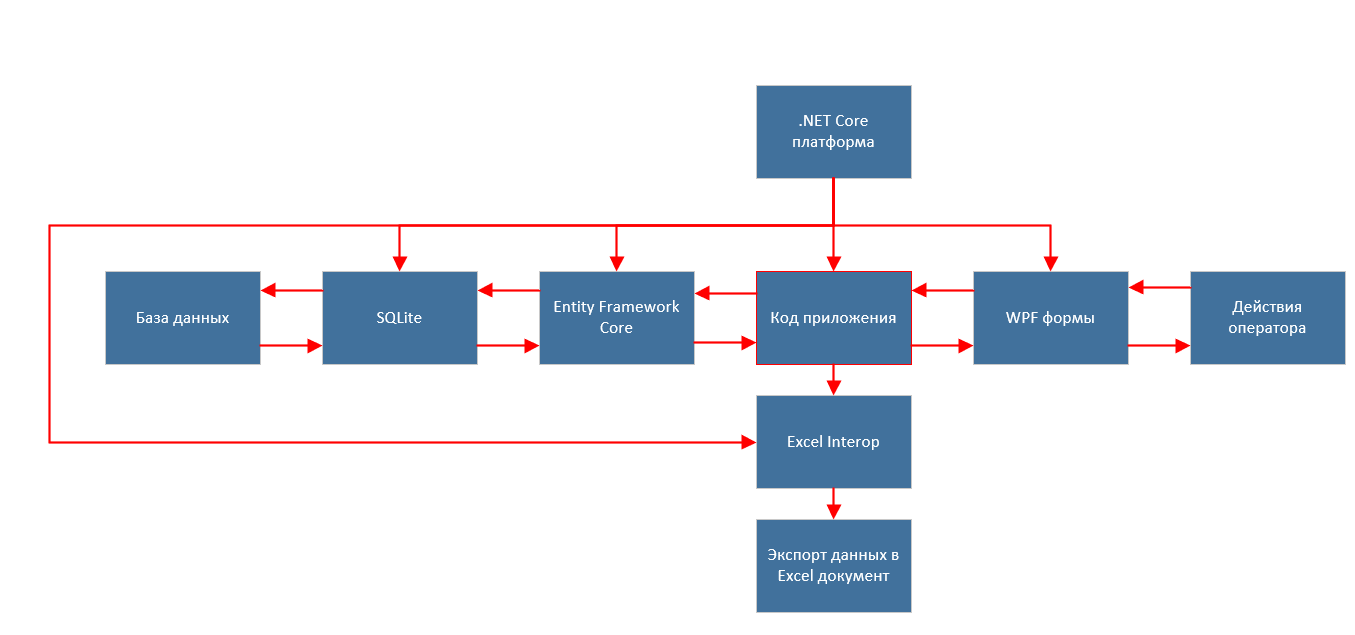


Рисунок 6 – Схема взаимодействия компонентов программы

## Создание программного продукта

Программа разработана на .NET платформе 6 версии. Для этого создан новый WPF проект (рисунок 7) в среде разработки Visual Studio.

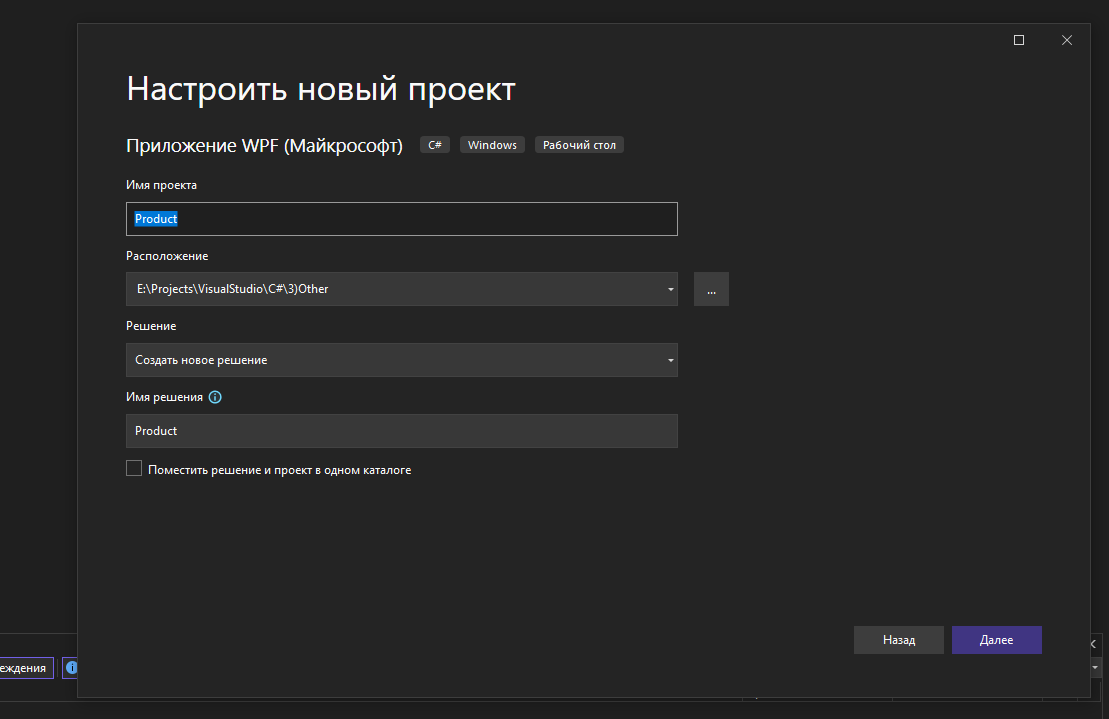


Рисунок 7 – Создание проекта

После создания проекта была создана база данных. Создание базы данных производилось при помощи библиотеки Entity Framework Core, которую необходимо установить в проект. Скачиваем библиотеку с помощью каталога пакетов NuGet (рисунок 8). Также необходимо скачать библиотеку-провайдер SQLite для Entity Framework Core.

Nuget – это механизм, с помощью которого разработчики могут создавать, передавать друг другу и использовать полезный код.[[8]](#footnote-8)

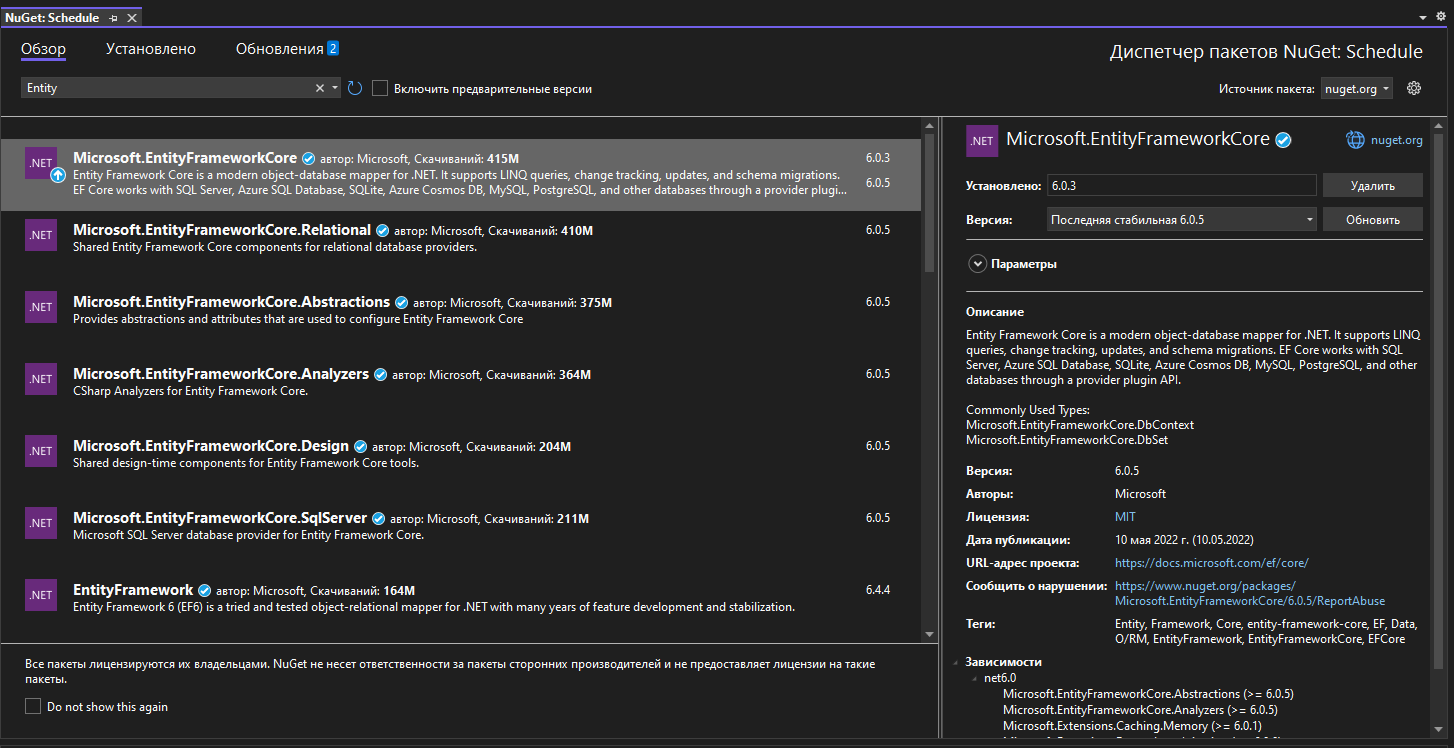


Рисунок 8 – Скачивание библиотек в NuGet

Создать базу данных можно двумя способами:[[9]](#footnote-9)

* Database First (при помощи инструментов СУБД SQLite);
* Code First (при помощи кода C#).

База данных создана по схеме «Code First». Для этого создан каталог в проекте с названием «Database» и создаем файл DatabaseContext.cs. В данном файле описаны сущности, которые были спроектированы ранее и созданы ключи в сущностях и связи между сущностями в методе OnModelCreating() (рисунок 9).

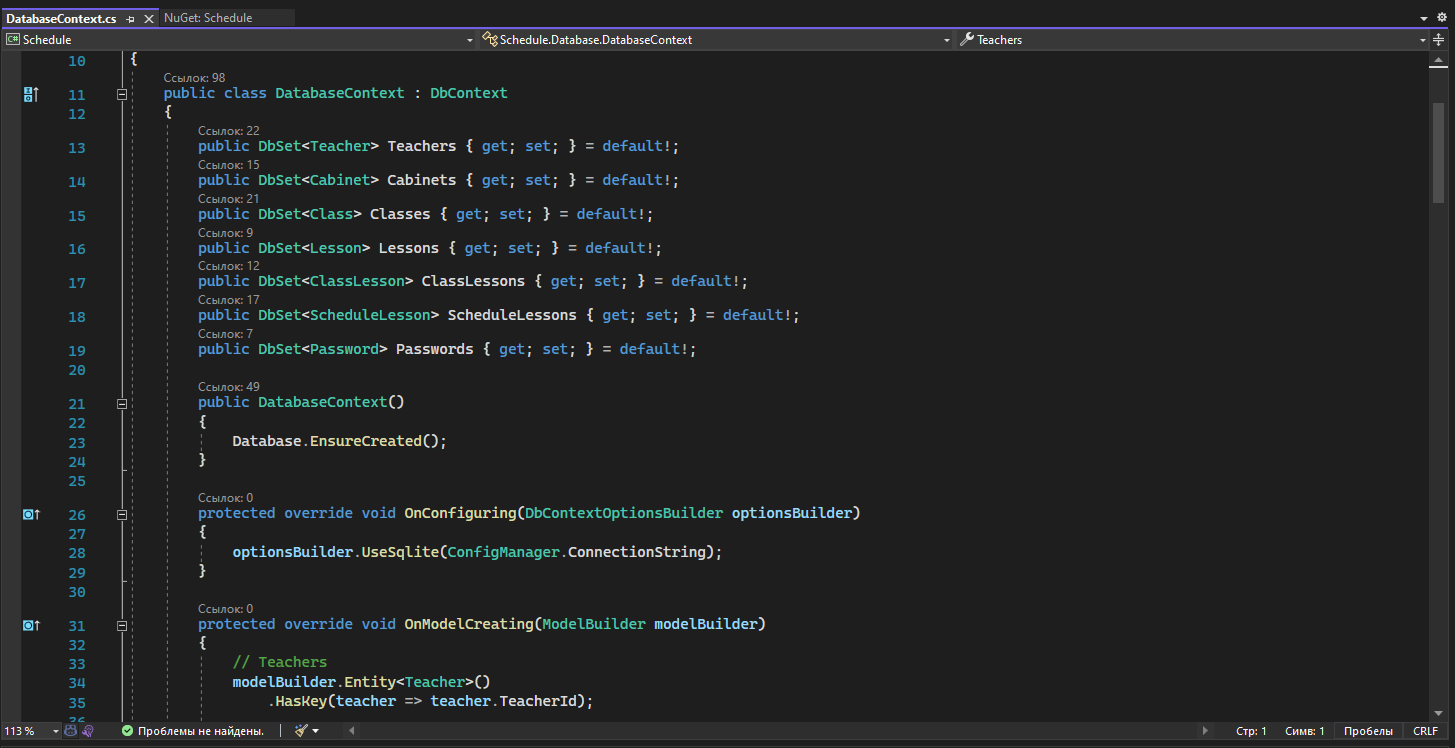


Рисунок 9 – Содержимое файла DatabaseContext.cs

Метод – это именованный блок кода, который выполняет некоторые действия. Класс – шаблон, который определяет содержимое объекта.[[10]](#footnote-10)

Параллельно с созданием контекста базы данных созданы сущности (рисунок 10), в которых описаны поля и связанные сущности. Таким образом позволяя обращаться к различным полям по связям.

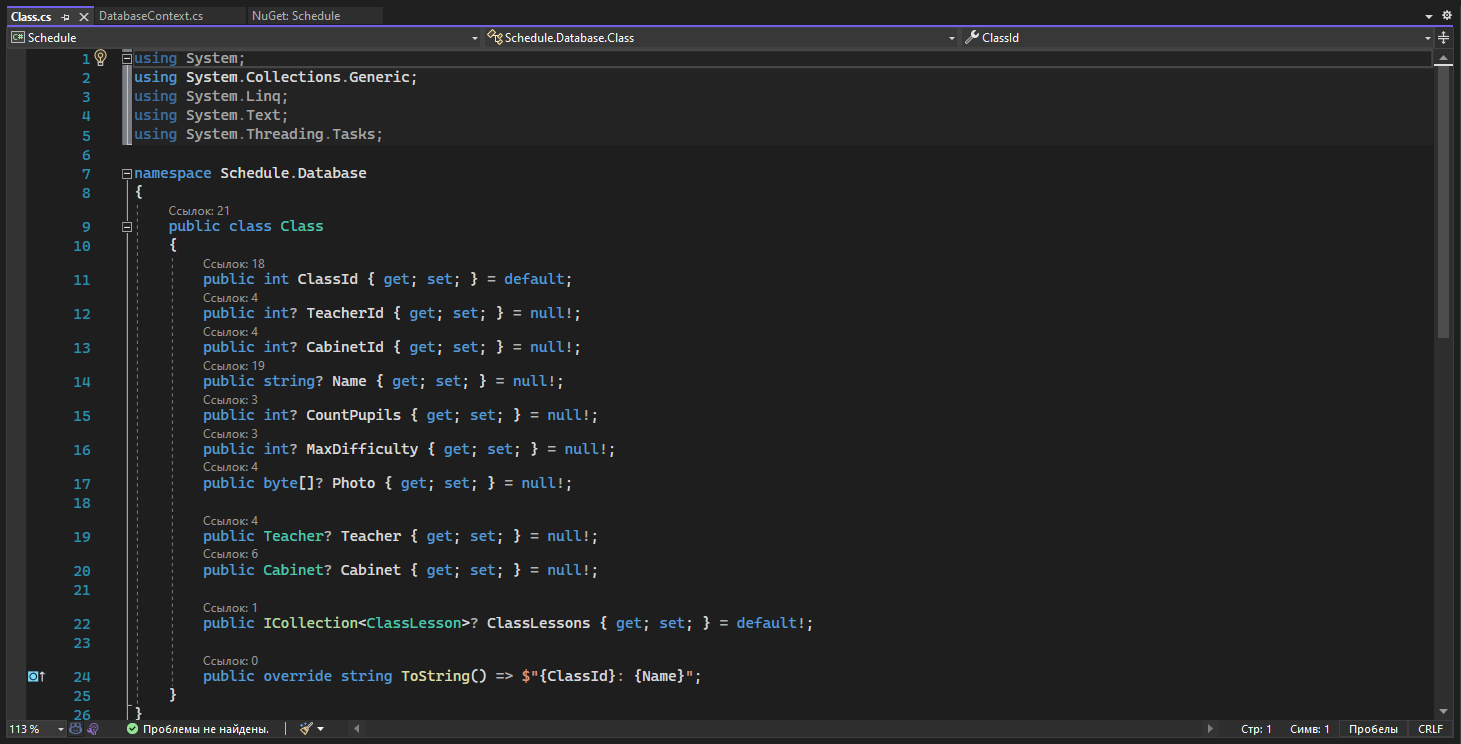


Рисунок 10 – Сущность «Класс», описанный кодом C#

В конечном итоге получился код, описывающий базу данных. Такой подход будет генерировать базу данных при работе программы в случае отсутствия последнего.

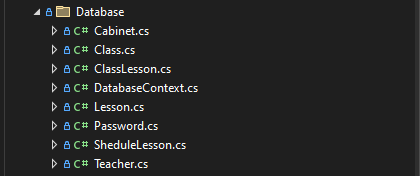


Рисунок 11 – Структура базы данных в проекте

Для разработки визуальной части программы необходимо работать с окнами и страницами при помощи WPF технологии. Взаимодействие (рисунок 12) между окнами и страницами реализовывался при помощи элемента Frame. В параметр Content можно встраивать страницы. Таким образом, окна как контейнера для страниц с элементами. Соответственно не нужно создавать множество окон и постоянно наблюдать на переход между окнами впоследствии.

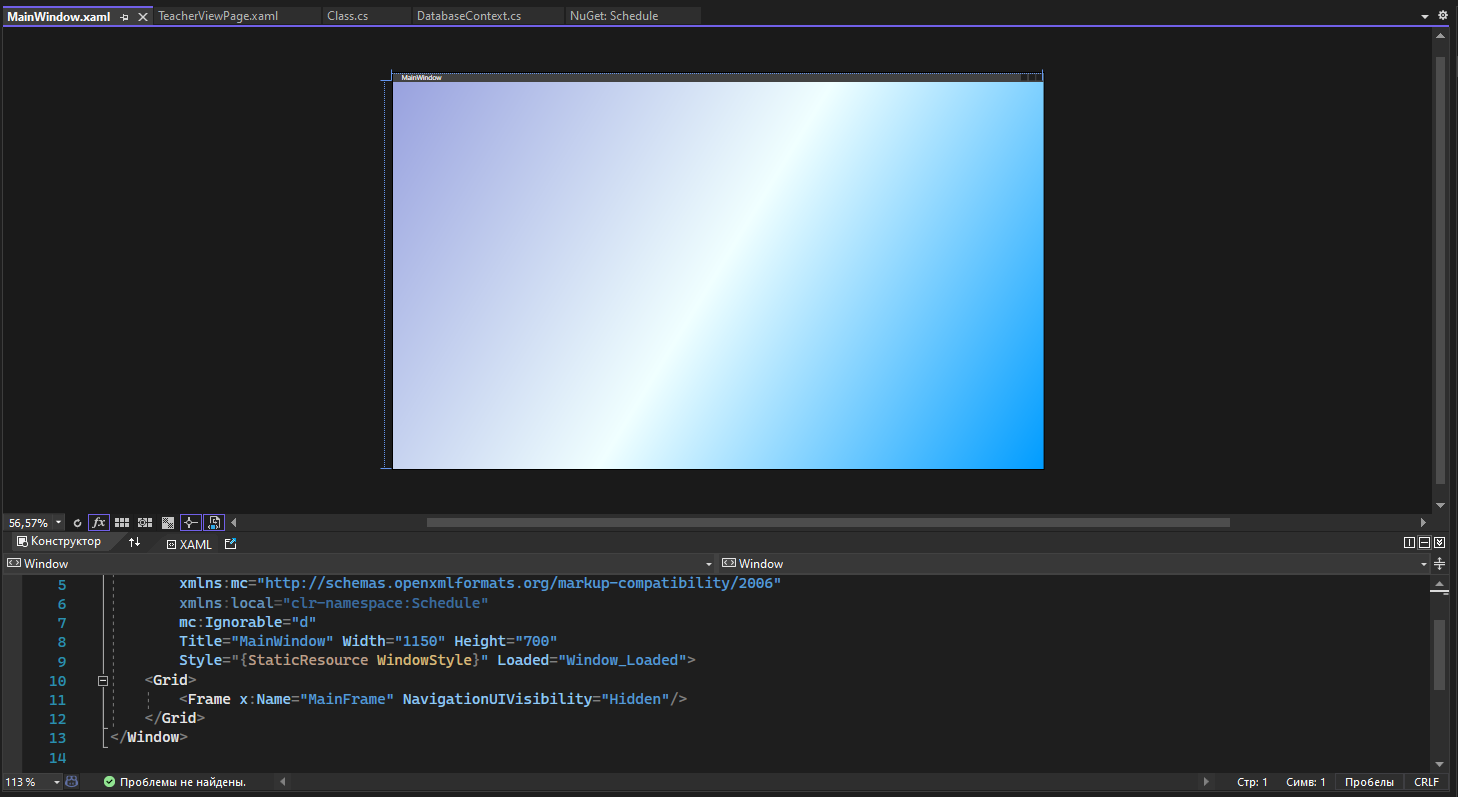


Рисунок 12 – Дизайн окна MainWindow

На рисунке 13, представленный ниже, видно, что в параметр Content элемента Frame присваивается стартовая страница.

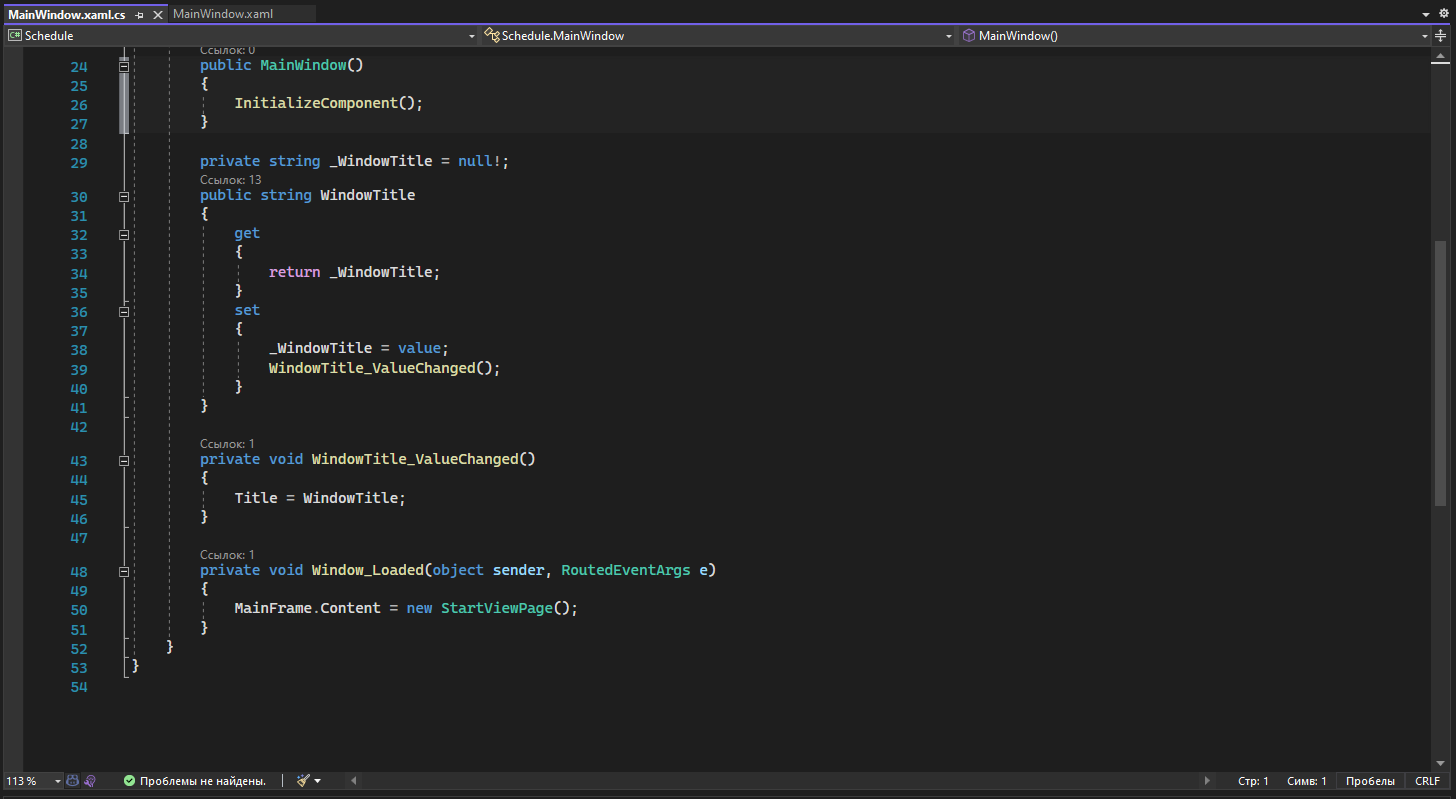


Рисунок 13 – Функциональная часть окна MainWindow

В самих страницах находиться основной функционал программы. Для взаимодействия с сущностями в базе данных, создано по странице для каждой сущности.

Для создания дизайна страницы «Педагоги» (рисунок 14) необходимо было использовать следующие элементы WPF: Button, ListBox, TextBox, DatePicker, ComboBox, TabControl и многие другие.

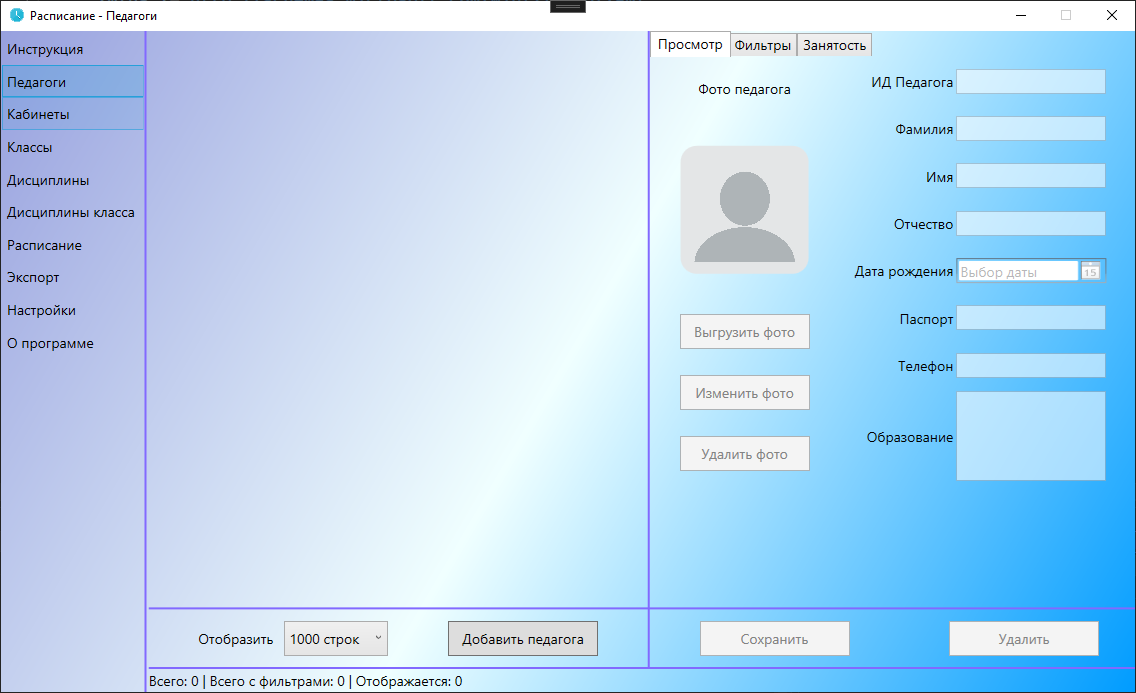


Рисунок 14 – Дизайн страницы «Педагоги»

Для хранения записей в странице используется элемент ListBox. Чтобы его заполнить записями, был разработан метод UpdateListBox() (рисунок 15), который будет подгружать данные из базы данных.

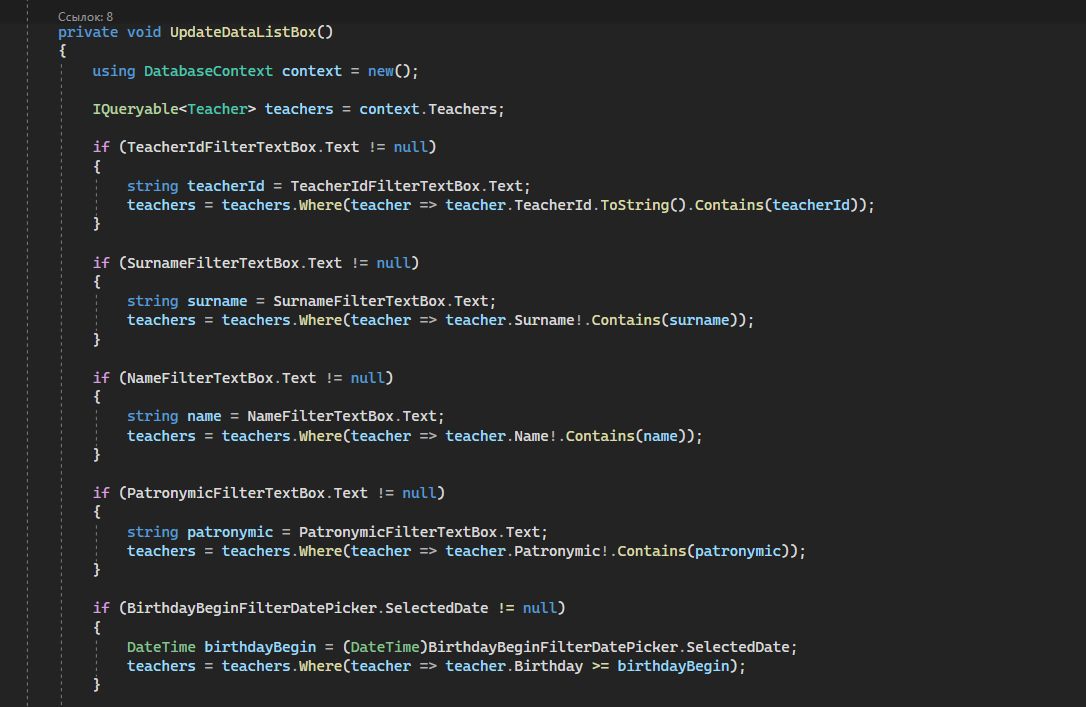


Рисунок 15 – Метод UpdateListBox()

При выборе записи из ListBox данные из него подгружается в элемент TabControl, который находиться справа от ListBox. Для заполнения элемента TabControl, который хранит в себе поля показа и редактирования данных записи сущности, написан метод UpdateViewTabControl() (рисунок 16), который срабатывает при событии SelectionChanged в ListBox.

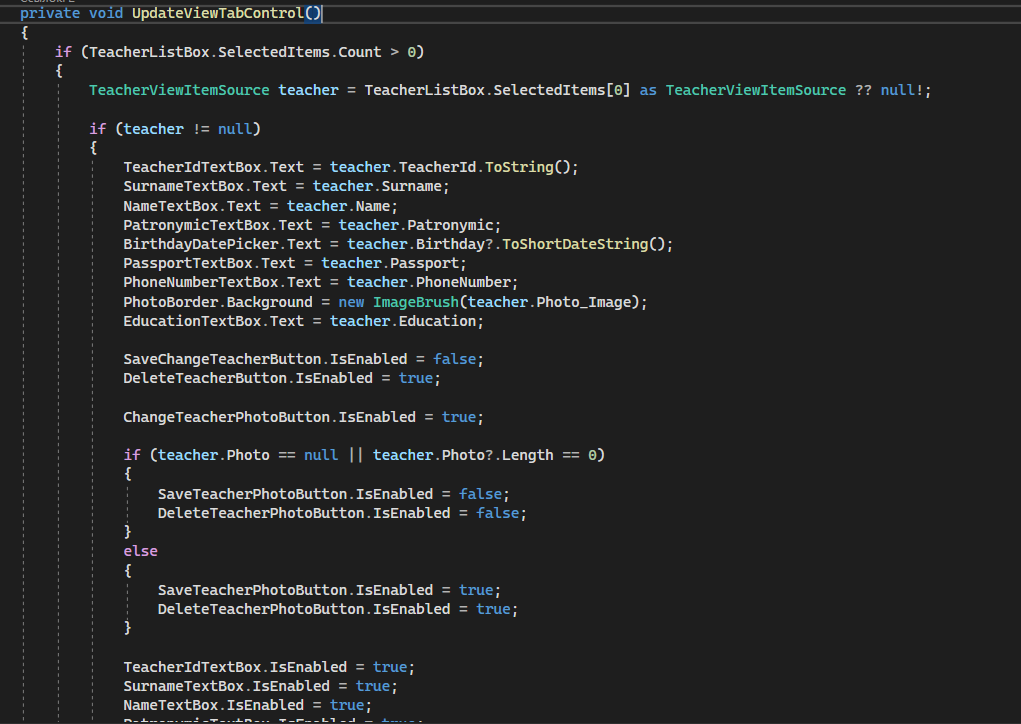


Рисунок 16 – Метод UpdateViewTabControl()

Для добавления записи была созадана кнопка AddTeacherButton. При нажатии на AddTeacherButton, срабатывает событие Click, которая вызывает метод AddTeacherButton\_Click() (рисунок 17), что в свою очередь создает запись в базе данных и обновляет список ListBox.

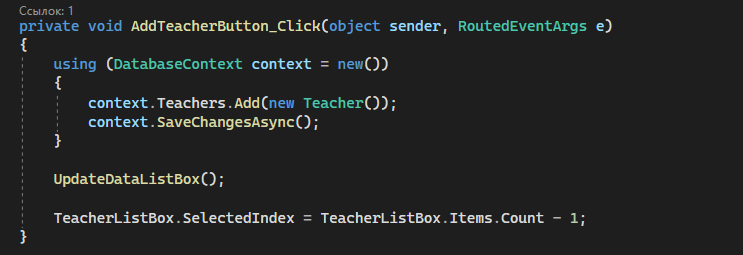


Рисунок 17 – Метод AddTeacherButton\_Click()

Для редактирования записи была размещена кнопка SaveChangeTeacherButton. При нажатии вызывается метод SaveChangeTeacherButton\_Click() (рисунок 18), который собирает все изменения с полей, которые находятся в TabControl, и редактирует данные в записи в соответствии с значениями последних и в конце обновляет ListBox.

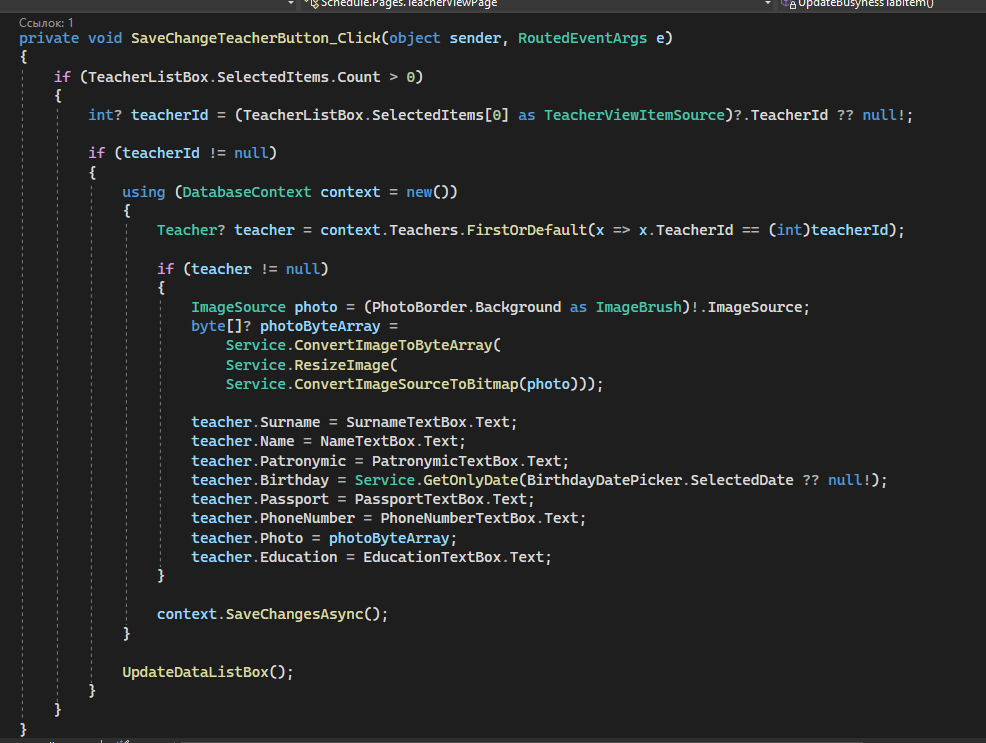


Рисунок 18 – Метод SaveChangeTeacherButton\_Click()

Для удаления записи была размещена кнопка DeleteTeacherButton. При нажатии на вызывается метод DeleteTeacherButton\_Click() (рисунок 19), который удаляет выбранную запись в ListBox из базы данных.

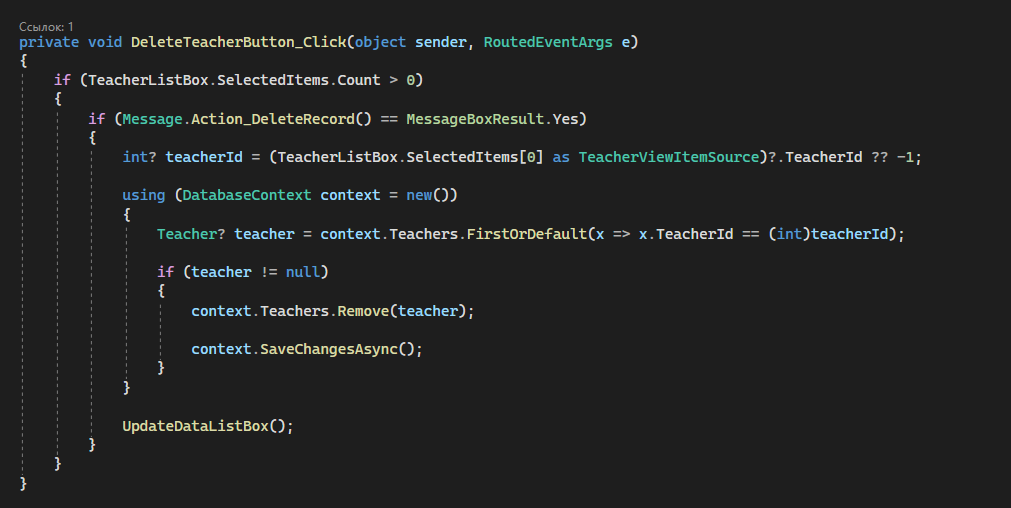


Рисунок 19 – Метод DeleteTeacherButton\_Click()

По такой схеме были созданы следующие страницы: «Кабинеты», «Классы», «Дисциплины».

Для создания страницы «Дисциплины класса» необходимо было создать уже 2 элемента ListBox. В одном храниться список классов, в другом – список дисциплин. Соответственно для редактирования записи необходимо выбрать по одной записи из каждого ListBox. Дизайн страницы представлен ниже (рисунок 20).

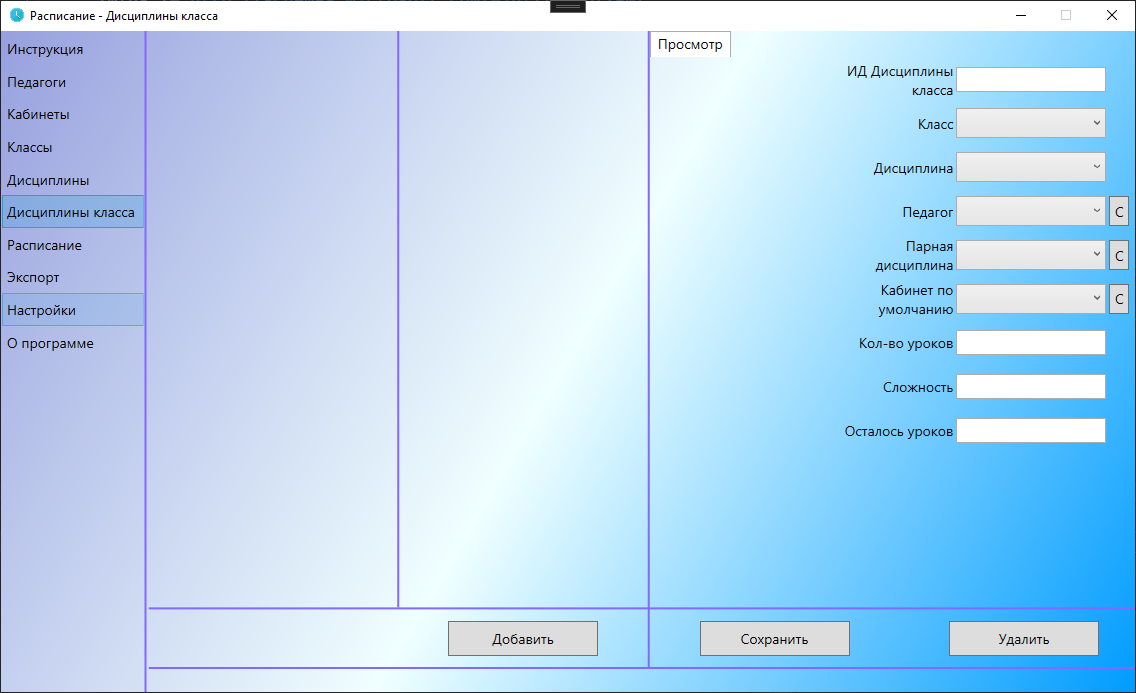


Рисунок 20 – Дизайн страницы «Дисциплины класса»

Для создания страницы «Расписание» (рисунок 21) использовался список ItemsControl, в котором можно создавать объект. В качестве объекта для списка ItemsControl использовался DataGrid, в который помещается расписание класса на неделю.

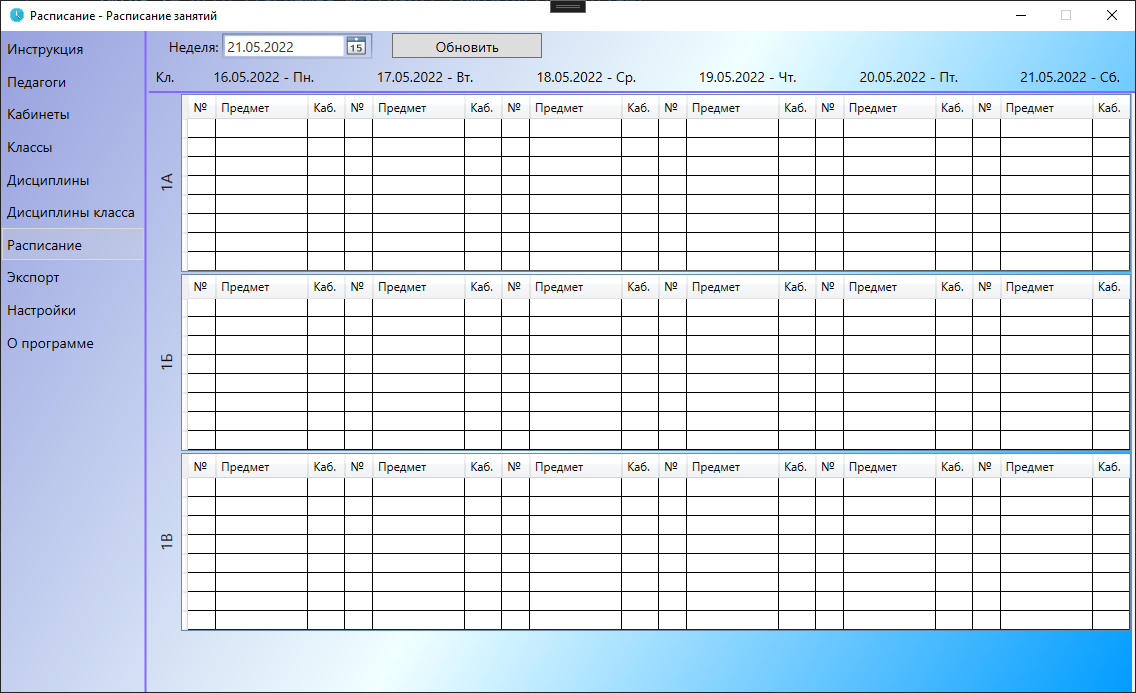


Рисунок 21 – Дизайн страницы «Расписание»

Дизайн страницы способен отобразить расписание на неделю. Для выбора отображаемой недели создаем элемент DatePicker.

Для отображения расписания в ItemsControl было создано 3 класса:

* WeekScheduleNumberLessonClass – содержит в себе недельное расписание определенного номера урока определенного класса;

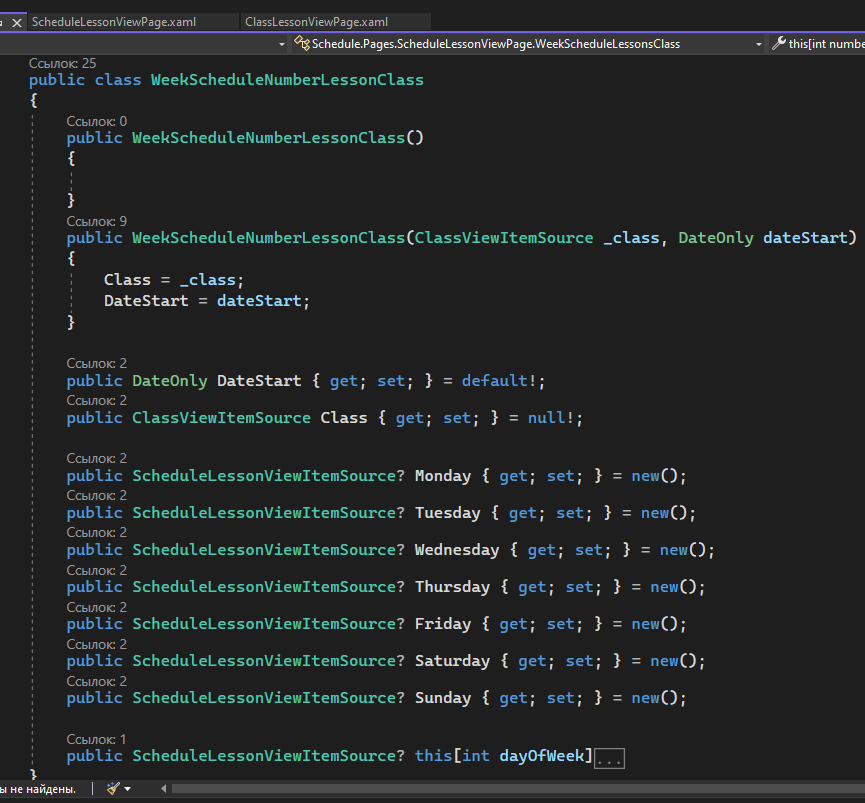


Рисунок 22 – Класс WeekScheduleNumberLessonClass

* WeekScheduleLessonsClass – содержит в себе расписание всех уроков на неделю определенного класса;

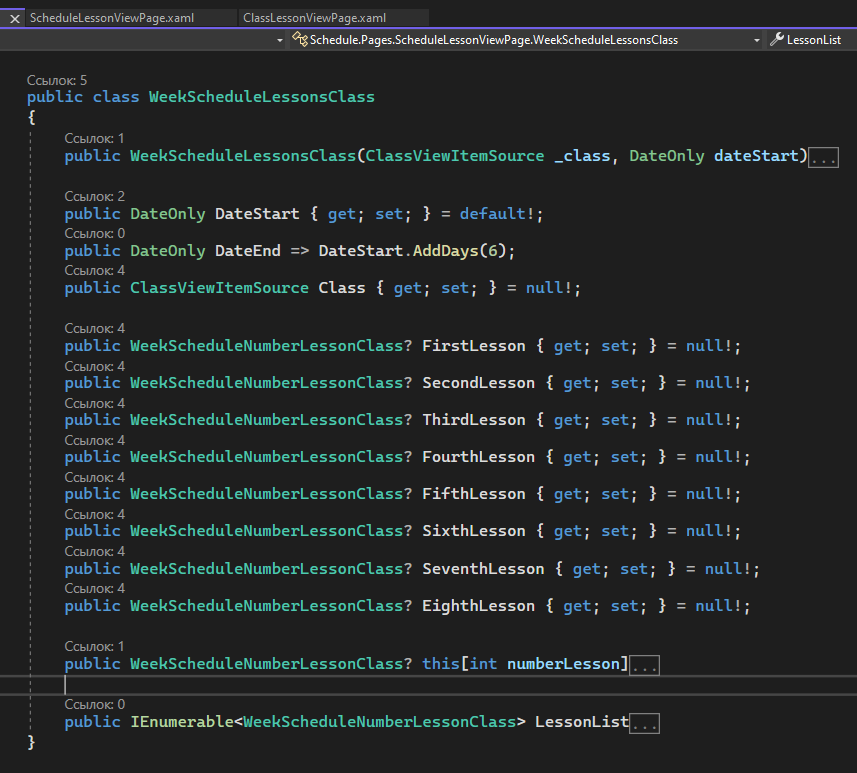


Рисунок 23 – Класс WeekScheduleLessonsClass

* WeekScheduleLessonsClasses - содержит в себе расписание всех уроков на неделю всех классов.

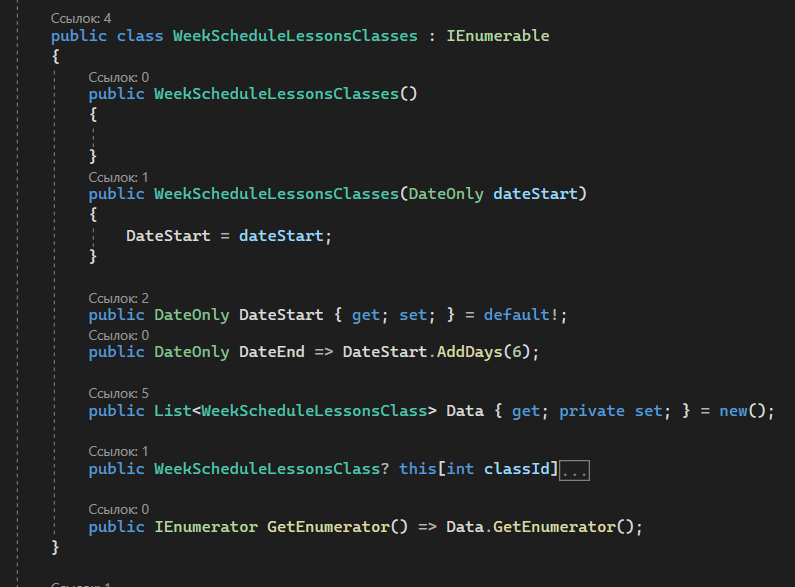


Рисунок 24 – Класс WeekScheduleLessonsClasses

Ниже, на рисунке 25, представлена схема временного хранения расписания уроков на неделю для показа в ItemsSource.

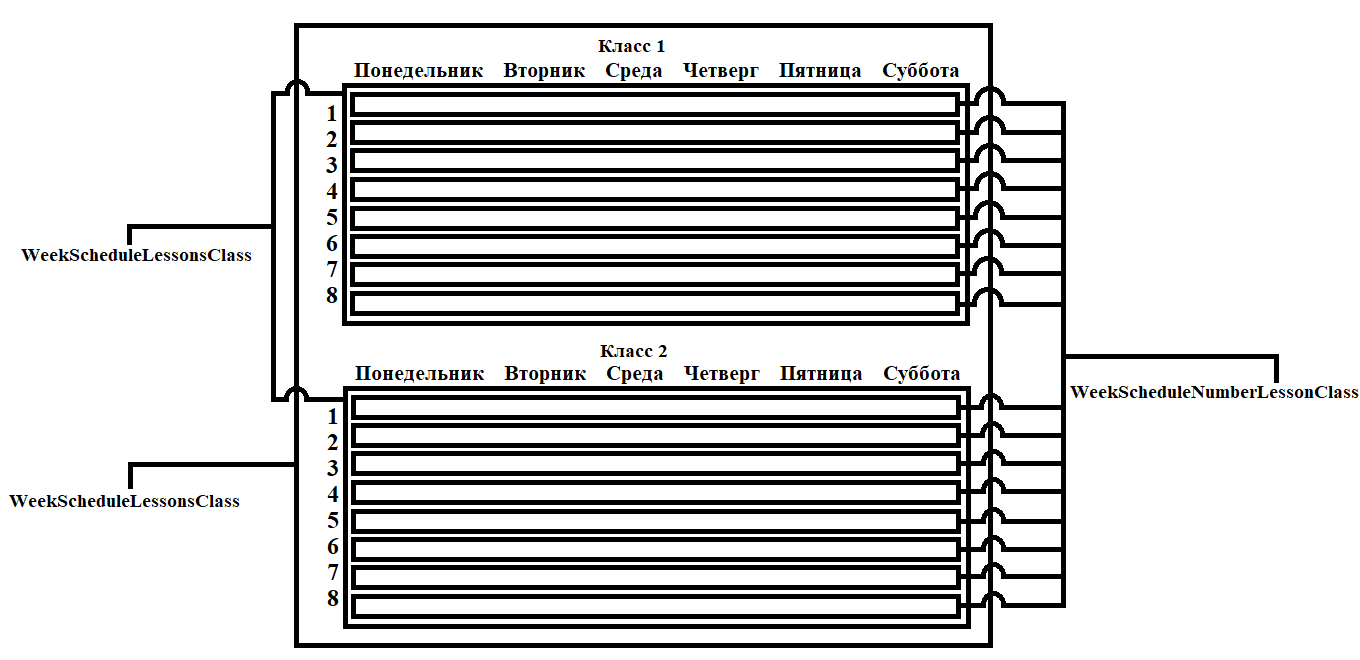


Рисунок 25 – Схема хранения расписания на неделю в программе

Для заполнения ItemsControl расписанием создаем асинхронный метод UpdateScheduleItemsControlAsync() (рисунок 26), который срабатывает при изменении даты в DatePicker и при изменении непосредственно расписания. Инициализирует класс WeekScheduleNumberLessonClass и заполняет его данными из базы данных.

Асинхронный метод позволяет вынести отдельные задачи из основного потока в специальные асинхронные методы и при этом более экономно использовать потоки. Асинхронные методы выполняются в отдельных потоках. Однако при выполнении продолжительной операции поток асинхронного метода возвратится в пул потоков и будет использоваться для других задач. А когда продолжительная операция завершит свое выполнение, для асинхронного метода опять выделяется поток из пула потоков, и асинхронный метод продолжает свою работу.[[11]](#footnote-11)

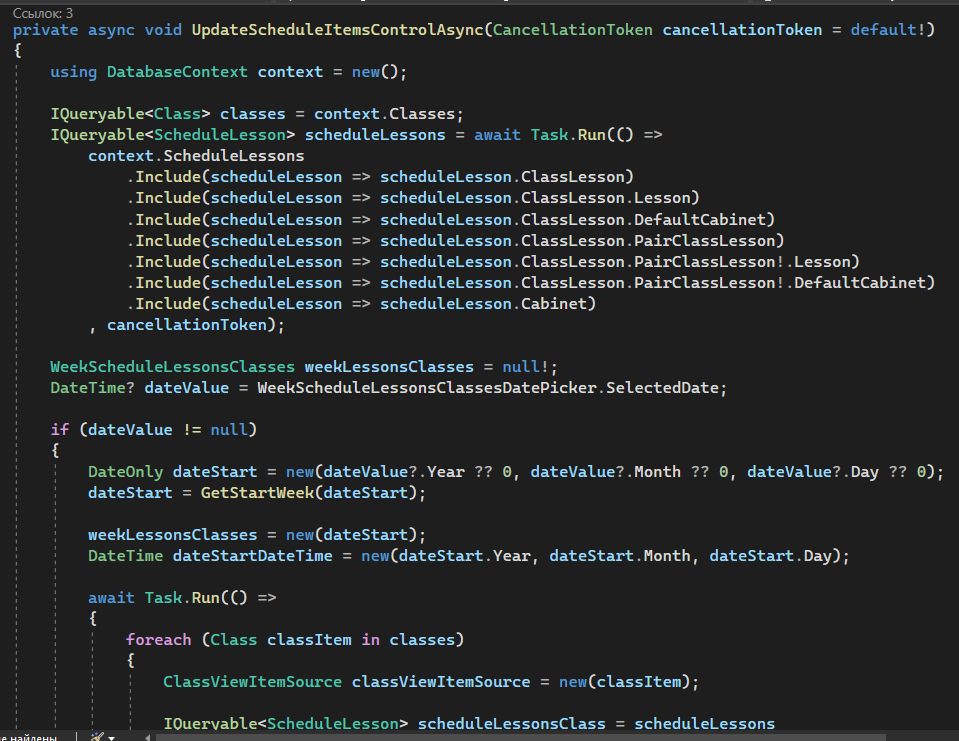


Рисунок 26 – Метод UpdateScheduleItemsControlAsync()

Для редактирования расписания было создано окно «Редактор расписания» (рисунок 27). Открывается при нажатии на определенную сетку в ItemsControl, расположенная на странице «Расписание».



Рисунок 27 – Дизайн окна «Редактор расписания»

Для загрузки текущего расписания в окно был разработан метод LoadDataComboBox() (рисунок 28). Чтобы отредактировать расписание необходимо указывать дисциплину и кабинет проведения. Также можно указать дополнительный кабинет для уроков проводимые по подгруппам и можно указать то, что урок является зачетом, если необходимо.

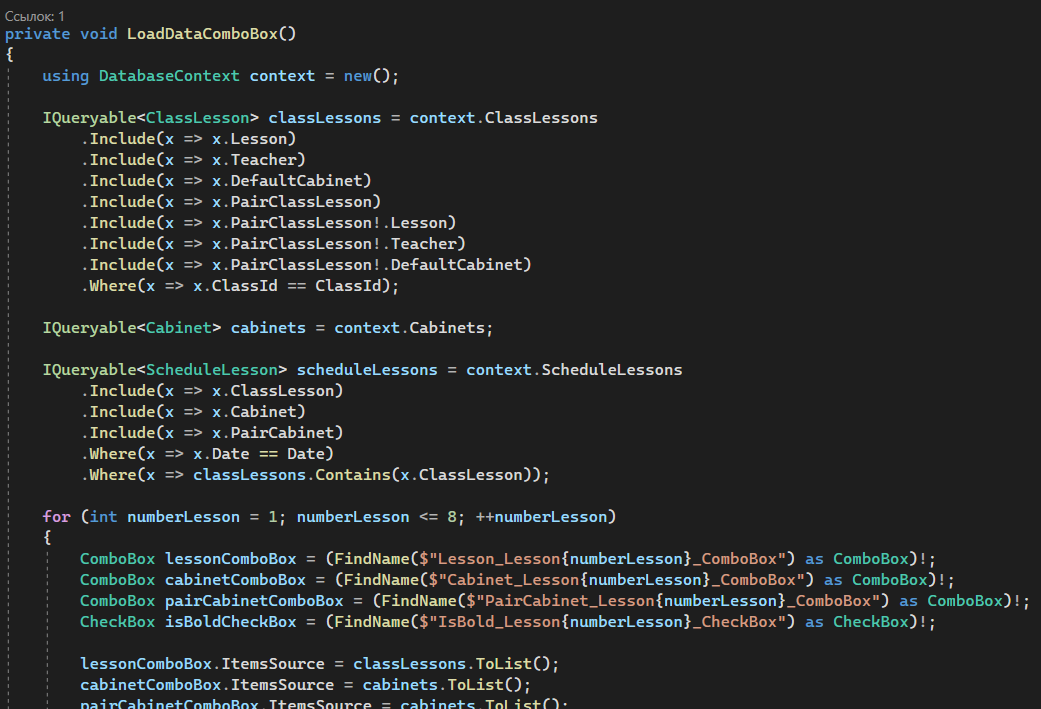


Рисунок 28 – Метод LoadDataComboBox()

Для сохранения изменений в расписании был разработан метод SaveButton\_Click() (рисунок 29). При нажатии на кнопку сохранения возвращает обратно на страницу расписания.

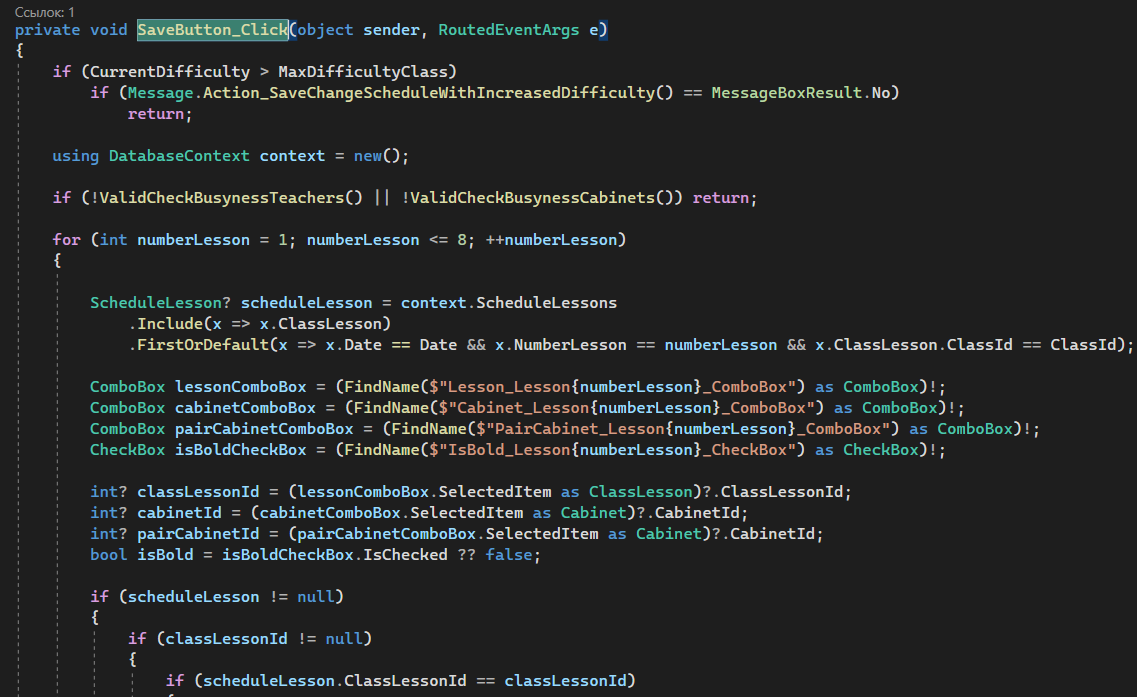


Рисунок 29 – Метод SaveButton\_Click()

Далее необходимо разработать функцию экспортирования расписания и формирования отчета в Excel-документ. Для этого создана новая страница с названием «Экспорт» (рисунок 30), в котором расположены 2 блока с выбором диапазона даты и кнопка для экспортирования.

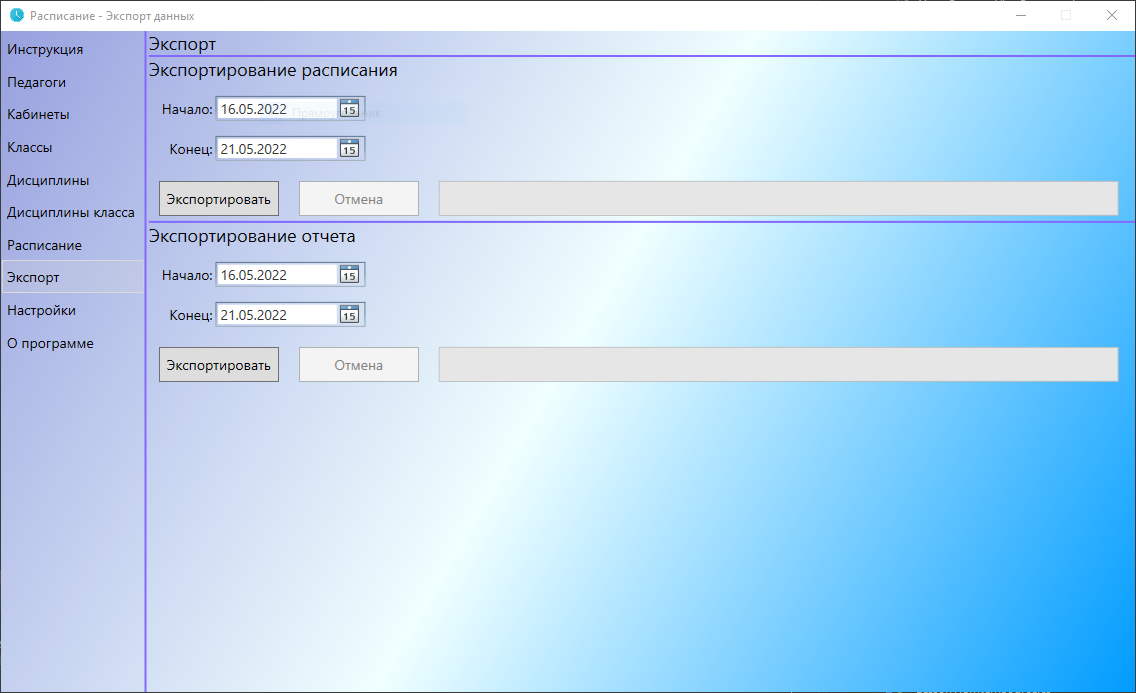


Рисунок 30 – Дизайн страницы «Экспорт»

При нажатии на кнопку экспортирования в блоке «Экспортирование расписания» выполняется метод ScheduleExportAsync() (рисунок 31), который загружает расписание в диапазоне указанной даты, инициализирует новый Excel-документ и заполняет его информацией по созданному шаблону.

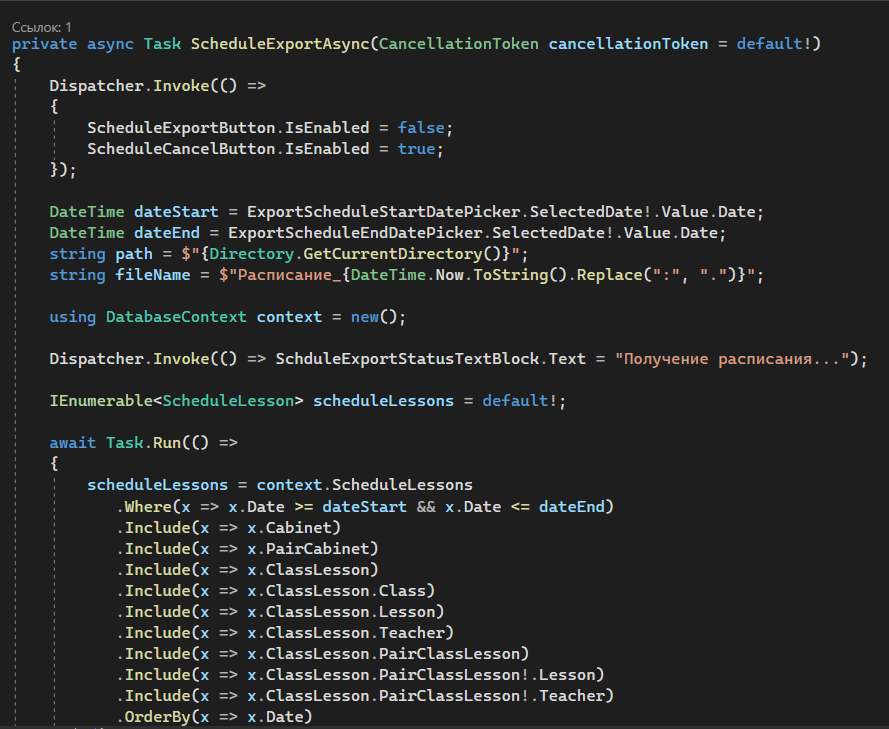


Рисунок 31 – Метод ScheduleExportAsync()

Ниже, на рисунке 32, представлен сформированный Excel-документ с расписанием.

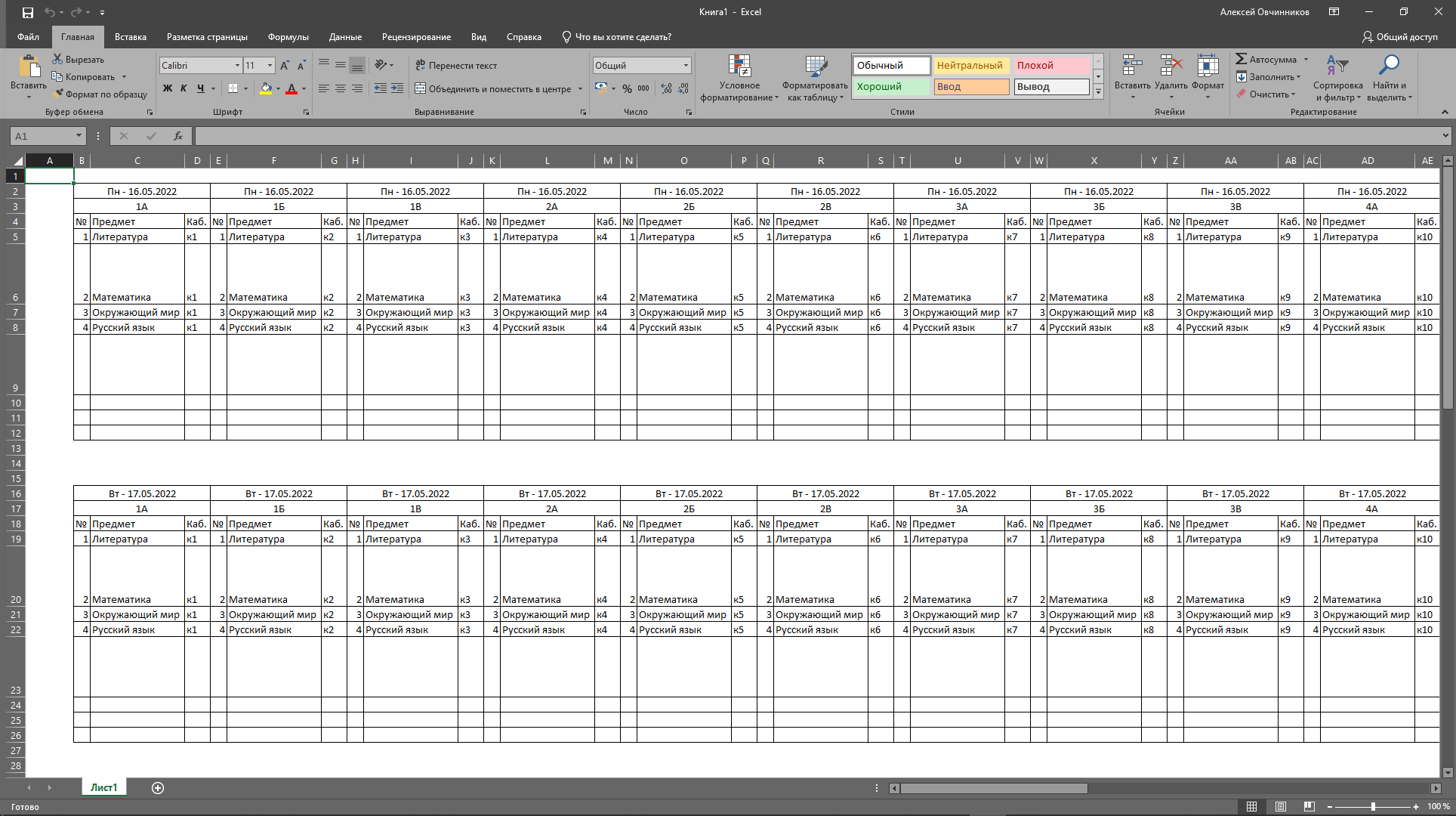


Рисунок 32 – Excel-документ с расписанием

При нажатии на кнопку экспортирования в блоке «Экспортирование отчета» выполняется метод ReportExportAsync() (рисунок 33). Он загружает расписание в диапазоне указанной даты и выполняет расчеты для подведения итогов по категориям, инициализирует документ и заполняет его информацией по созданному шаблону.

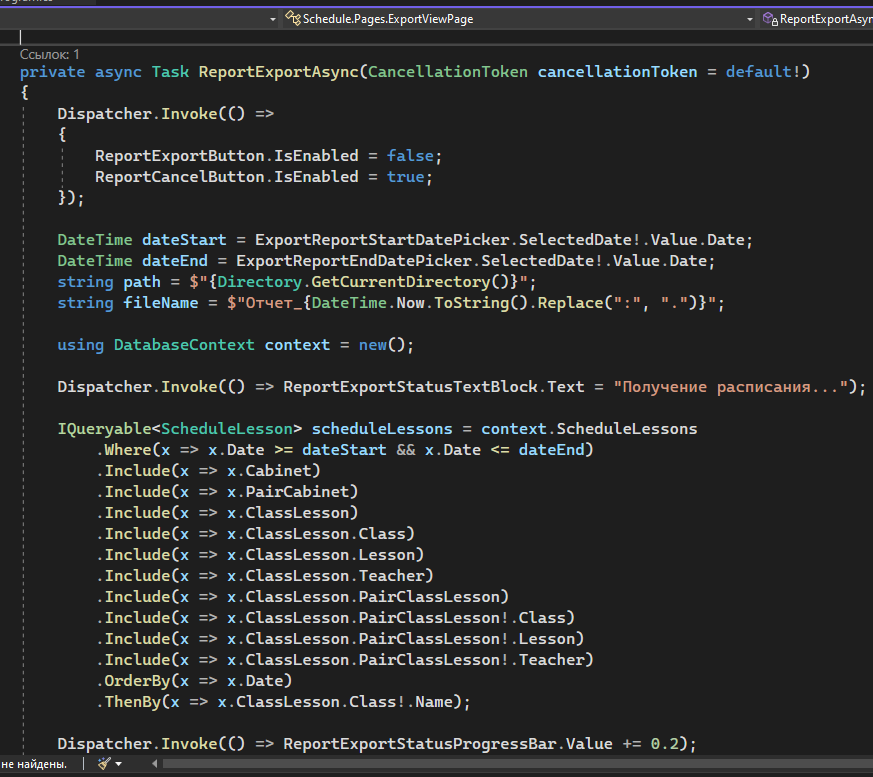


Рисунок 33 – Метод ReportExportAsync()

Ниже, на рисунке 34, представлен сформированный Excel-документ с отчетом.

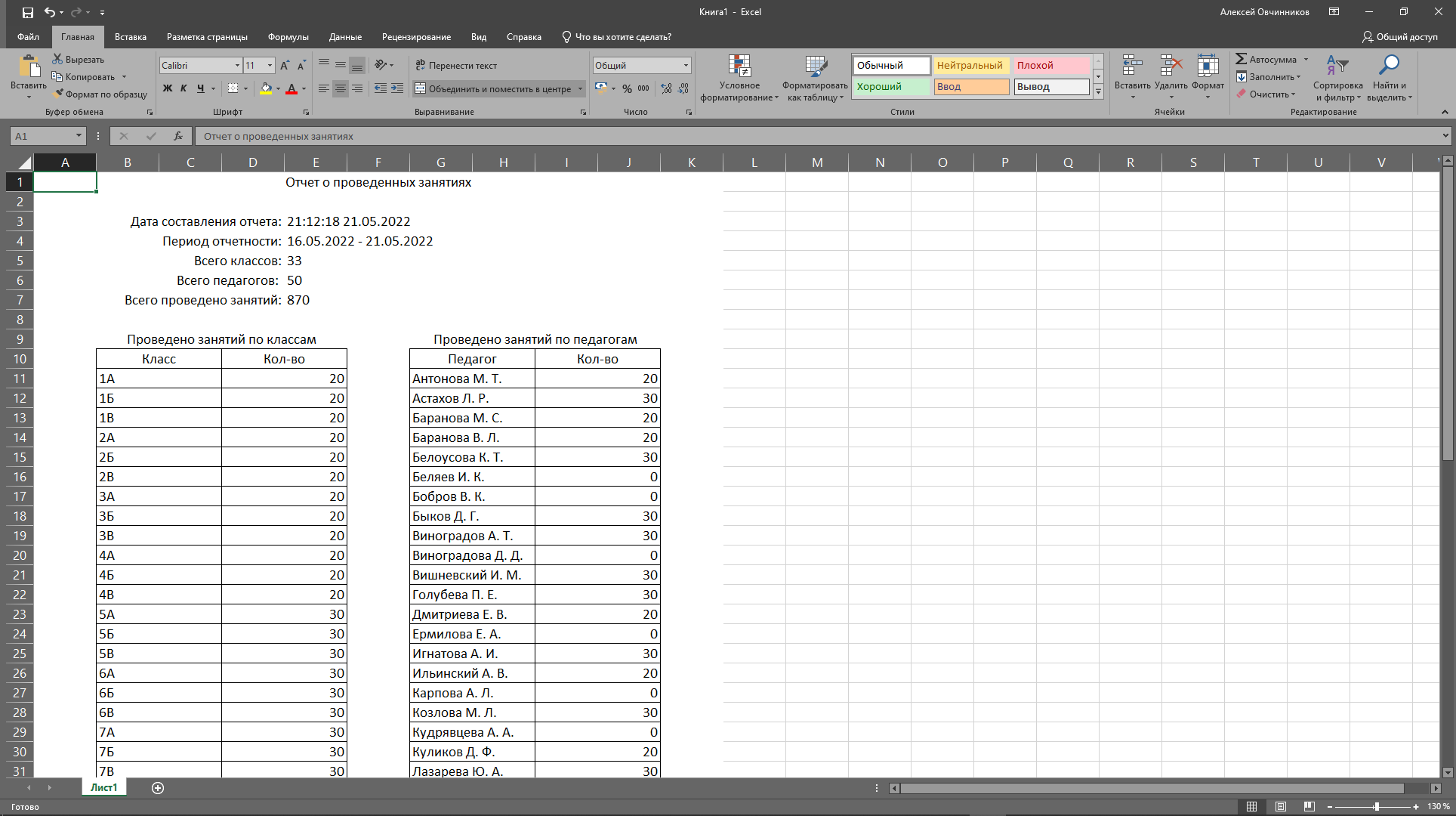


Рисунок 34 – Excel-документ с отчетом

Таким образом, разработан основной функционал приложения. Далее для удобства использования и эстетики необходимо разработать следующие дополнительные функции:

* Анимированный логотип при запуске программы;
* Форма авторизации с возможностью настройки;
* Возможность быстрой очистки данных;
* Краткое руководство по использованию.

Для создания анимированного логотипа при запуске программы была создана стартовая страница (рисунок 35). При запуске программы проходит открывается стартовая страница и воспроизводиться анимация, после этого программа делает переход на страницу, которая идет следующей (форма авторизации, если пароль установлен или сразу главная страница).



Рисунок 35 – Стартовая страница с анимацией

Для реализации анимации текста на стартовой странице разработан метод StartAnimationAsync() (рисунок 36), который постепенно уменьшает прозрачность логотипа при помощи постоянной смены кисти в надписи с новым значением.

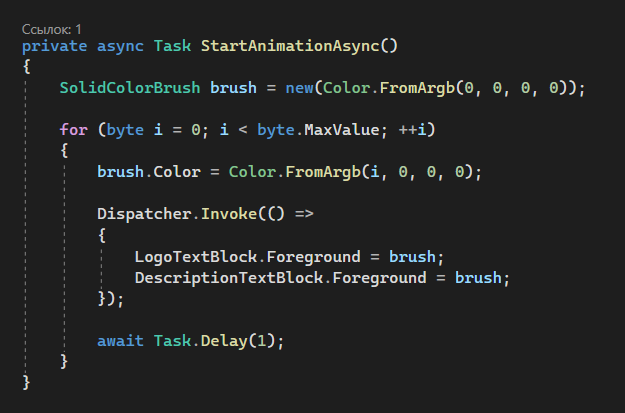


Рисунок 36 – Метод StartAnimationAsync()

Для функции авторизации разработана страница (Рисунок 37), с полем ввода пароля для получения доступа к программе.



Рисунок 37 – Дизайн страницы авторизации

Для шифрования пароля используется криптографический метод SHA512. В базу данных добавлена сущность «Пароль» (таблица 7), который хранит в себе хэш зашифрованного пароля. Для работы с паролем разработан класс PasswordManager (рисунок 38), который шифрует пароль, записывает хэш в базу данных, считывает его и сравнивает с введенным паролем.

Криптография – это методики обеспечения конфиденциальности (невозможности прочтения информации посторонним), целостности данных (невозможности незаметного изменения информации), аутентификации (проверки подлинности авторства или иных свойств объекта), шифрования (кодировка данных).[[12]](#footnote-12)

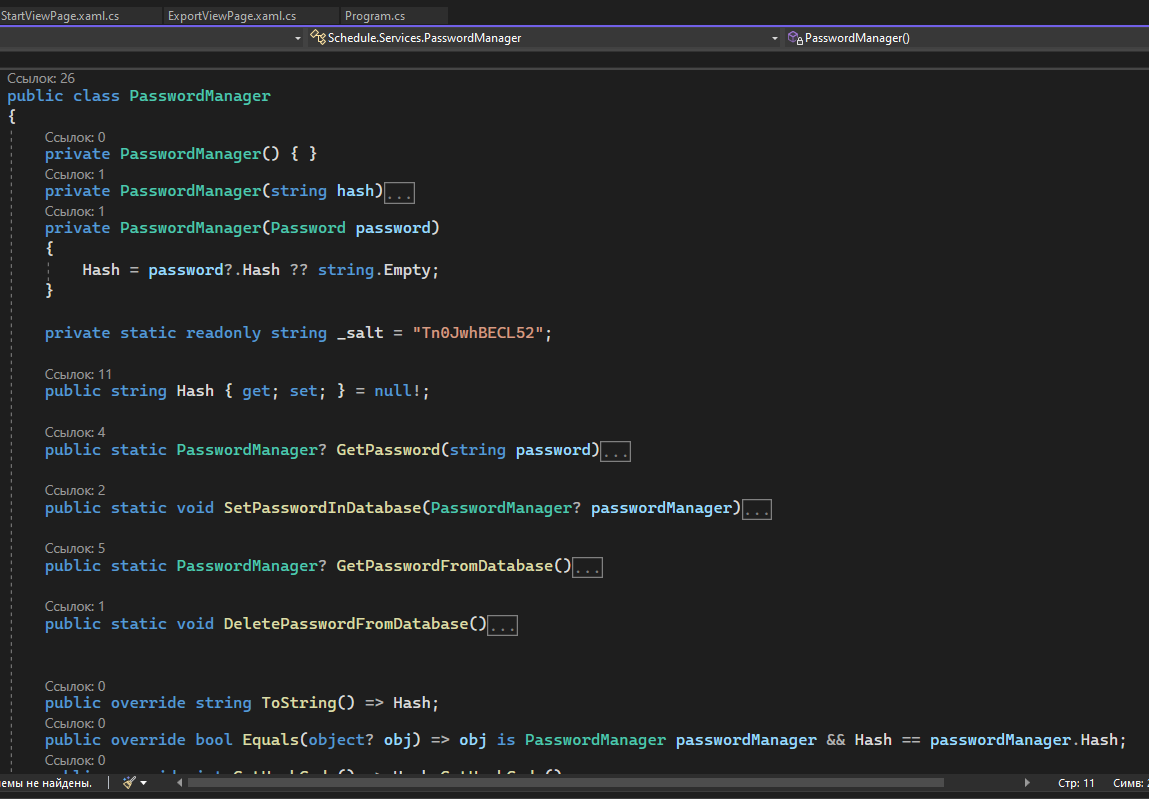


Рисунок 38 – Класс PasswordManager

Далее разработана страница «Настройки» (Рисунок 39), в которой можно устанавливать пароль, отключать стартовую анимацию и безопасно удалять базу данных.

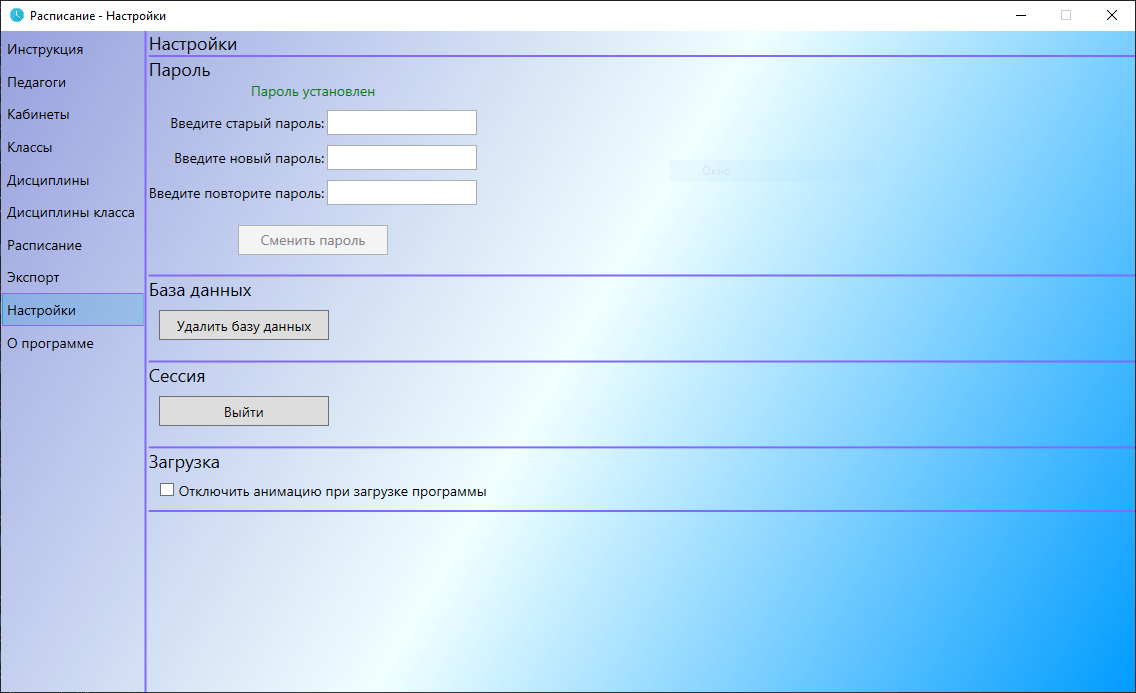


Рисунок 39 – Дизайн страницы «Настройки»

Для редактирования пароля написан метод ChangePasswordButton\_Click() (рисунок 40), который через класс PasswordManager редактирует пароль в базе данных.

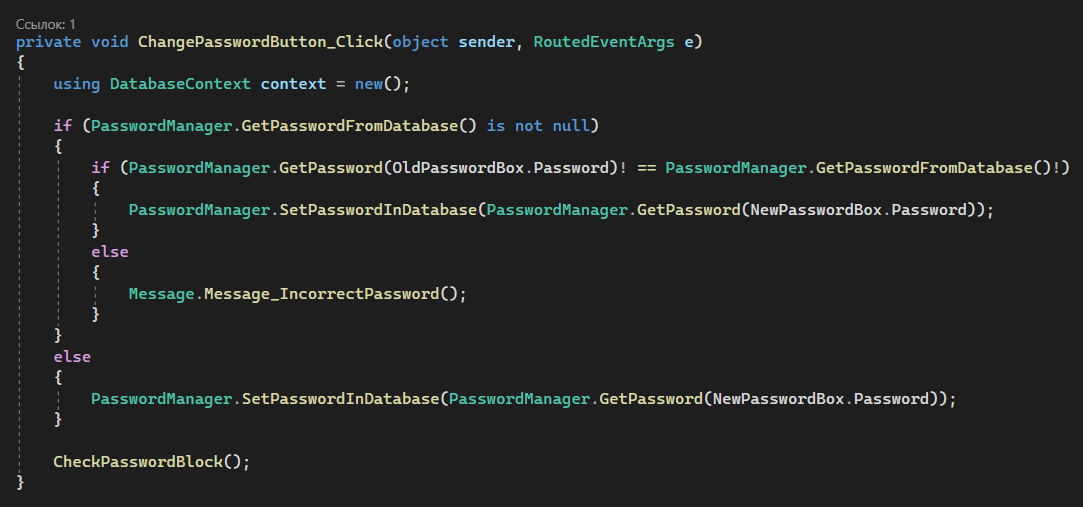


Рисунок 40 – Метод ChangePasswordButton\_Click()

Для хранения состоянии отключения стартовой анимации разработан файл конфигурации (рисунок 41), который генерируется под названием Config.json.

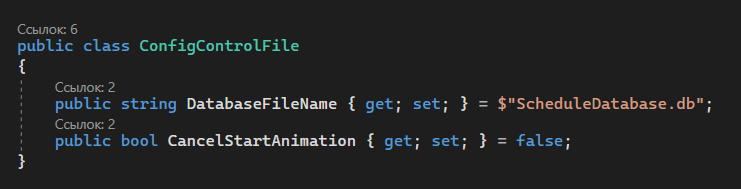


Рисунок 41 – Класс файла конфигурации

Для управления файлом конфигурации разработан класс ConfigManager (рисунок 42), который генерирует файл и управляет его содержимым.

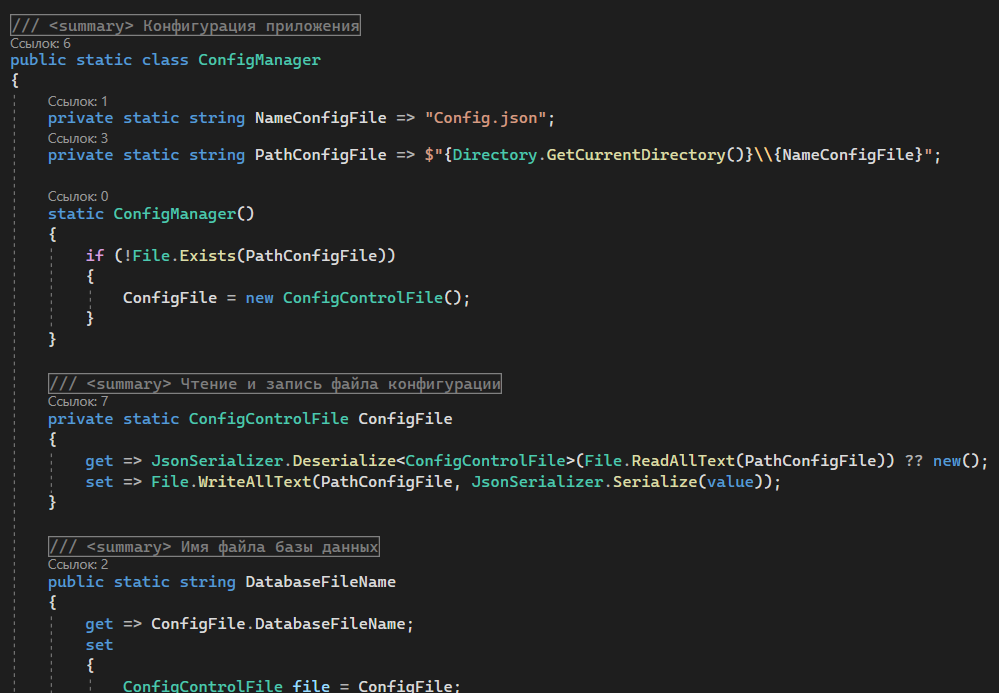


Рисунок 42 – Класс ConfigManager

Для безопасного удаления базы данных разработан метод DeleteDatabaseButton\_Click() (рисунок 43). Для его выполнения также необходимо подтвердить запрос на удаление в диалоговом окне, в избежание случайного нажатия.

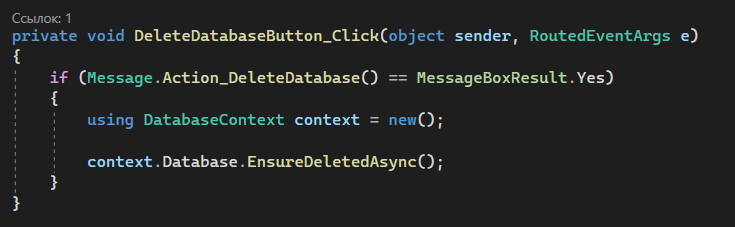


Рисунок 43 – Метод DeleteDatabaseButton\_Click()

Для создания краткого руководства по использованию программы необходимо создать новую страницу (рисунок 44). В нем будет показано как правильно создавать записи в таблице и в какой последовательности заполнять страницы данными.

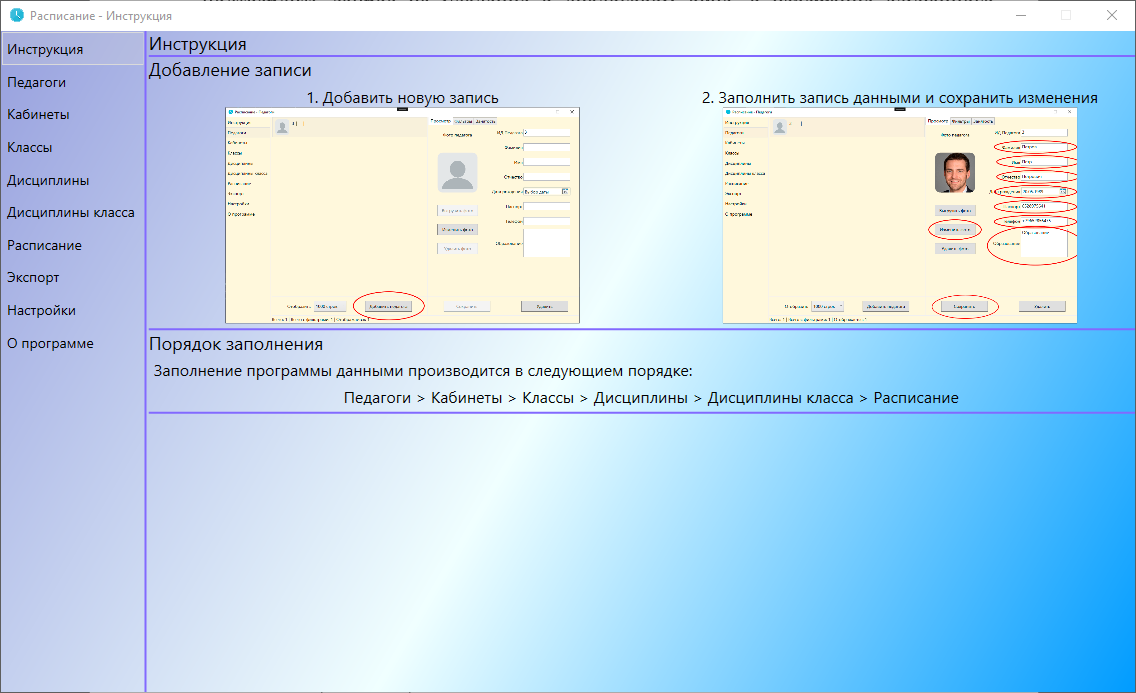


Рисунок 44 – Дизайн страницы «Инструкция»

Программа готова к использованию. Была разработана база данных в виде её сущностей, разработаны страницы для их управления, страница для просмотра и редактирования расписания, экспортирование расписания и отчета в excel-документ и разработана возможность ограничения доступа с применением криптографических методов шифрования данных. Далее необходимо протестировать на предмет ошибок и багов.

## Тестирование программного продукта

Тестирование программного продукта представляет процесс испытания программного продукта, целью которого является проверка соответствия ожидаемого поведения с реальным.

В ходе тестирования ошибок в работе программы не обнаружено.

# ГЛАВА 3. ЭКОНОМИЧЕСКАЯ ЧАСТЬ

Целью данного раздела является расчет затрат на создание программного продукта, создаваемого в ходе выполнения практической части ВКР.

Для достижения поставленной цели необходимо решить следующие задачи:

* Определить общую продолжительность работ по созданию ПП
* Произвести расчет суммарных затрат на разработку ПП
* Определить цену реализации ПП
* Произвести расчет затрат на внедрение ПП

Определение общей продолжительности работ по созданию ПП

Общая продолжительность процесса разработки определяется количеством и длительностью отдельных этапов, которые обеспечивают создание ПП требуемого качества. Состав этапов работ по созданию программного продукта:

1. Подготовительный этап.

2. Составление технического задания (ТЗ).

3. Основной этап.

4. Тестирование.

5. Документирование.

6. Сдача темы.

Данные по длительности этапов представим в табл.1.

Таблица 8 – Длительность этапов работ на этапе проектирования

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | | Этапы работ | Исполнитель | Длительность этапа  (дни) |
| 1. | | Подготовительный этап | Программист | 1 |
| 2 | | Составление технического задания | Программист | 3 |
| 3 | | Основной этап | Программист | 13 |
| 4 | | Тестирование | Программист | 3 |
| 5 | | Документирование | Программист | 2 |
| 6 | Сдача темы | | Программист | 1 |
| Итого | | | | 23 |

Для иллюстрации общей продолжительности проектирования (Тп), по данным таблицы 8 построим график организации работ во времени (рисунок 44).

25.04.22

26.04.22-28.04.22

29.04.22-14.05.22

16.05.22-18.05.22

21.05.22

19.05.22-20.05.22

1д.

3д

1

2

3

4

5

6

13д

3д

2д

1д

Рисунок 44 – Ленточный график организации работ на этапе проектирования

Расчет суммарных затрат на разработку ПП

К затратам на разработку ПП (Зс) относятся:

* Материальные затраты (Зм);
* Затраты на электроэнергию (Зэ);
* Затраты на оплату труда (Зт);
* Отчисления на социальные нужды (Зо);
* Амортизация оборудования (За).
* Прочие затраты (Зп).

Зс = Зм +Зэ+Зт+Зо+За+Зпр (1)

Расчет материальных затрат на разработку ПП

Материальные затраты – затраты на приобретение основных и вспомогательных материалов (Зм), используемых в процессе проектирования и разработки ПП затраты на покупные комплектующие изделия, полуфабрикаты и услуги производственного характера сторонних организаций. В таблице 9 представлен расчет количества и стоимости основных и вспомогательных материалов.

Таблица 9 – Расчет затрат на основные и вспомогательные материалы, комплектующие изделия, используемые в разработке ПП

|  |  |  |  |
| --- | --- | --- | --- |
| Наименование | Цена за единицу, без НДС, руб. | Кол-во | Стоимость, руб. |
| 1. Бумага | 370,00 | 1 | 370,00 |
| 1. Flash-память | 270,00 | 1 | 270,00 |
| 1. Тонер | 320,00 | 1 | 320,00 |
| Итого | | | 960,00 |

Расчет затрат на оплату электроэнергии

Для разработки ПП используется оборудование, потребляющее электроэнергию: компьютеры, принтеры, серверы и т.п.

В соответствии с Санитарно-эпидемиологическими правилами и нормами, программист должен работать за компьютером не более 6 часов за 1 рабочую смену, от сюда общее время работы оборудования:

Tобщ = 23 × 6 = 138 ч.

Расход электропотребления одного компьютера

Рэл = 138 × 150 = 20700 Вт = 20,7 КВт.

Стоимость 1 КВт для юридических лиц 5,9 р. (для бюджетных потребителей по Саратовской области)

Зэл = 20,7 × 5,9 = 122,13 р.

На искусственное освещение израсходовано 40 Вт/ч.

Эиосв = 138 × 40 = 5520 Вт = 5,52 КВт.

Зосв = 5,52 × 5,9 = 32,57 р.

Общие затраты на электроэнергию находим по формуле:

Зосн = Зосв + Зэлпк, (2)

Зэл = 122,13+32,57 = 154,7 р.

Расчет затрат на оплату труда

Расчет затрат на оплату труда (Зт) ведется в соответствии с указанным ранее количеством и квалификационным составом специалистов, занятых в проектировании и создании ПП, исходя из действующих окладов специалистов конкретных предприятий, на которых производится разработка (либо как средних по отрасли, городу, региону, стране).

Основной фонд оплаты труда складывается из оплаты труда разработчика программы (студент-программист):

Общее время работы над проектом разработчика составляет – 26 дней. Учитывая, что оклад составляет 13890 рубль в месяц и 24 рабочих дней в месяц, получаем:

Зосн = 13890×23/24=13311,25 (руб).

Принимаем дополнительный фонд оплаты труда равным 15% от основного фонда оплаты труда.

Здоп = 11624,58×0,15=1996,69 (руб).

Затраты на оплату труда составят:

Зопт = 13311,25+1996,69=15307,94 (руб).

Отчисления на социальные нужды

В статью «Отчисления на социальные нужды» включаются суммы страховых взносов в Пенсионный фонд РФ, Фонд социального страхования РФ и фонды обязательного медицинского страхования (федеральный и территориальный). На 2022 г. установлены следующие тарифы страховых взносов – ПФР - 22%, ФСС - 2,9%, ФФОМС - 5,1%, что в сумме составляет 30%.

Суммарную величину страховых взносов (Зо) рассчитываем по формуле

, (3)

где Зопт – основная заработная плата.

(руб).

Расчет затрат на амортизацию

Рассчитаем сумму амортизационных отчислений.

Общая величина амортизационных отчислений (За) определяется по формуле

, (4)

где Фi – стоимость i-го оборудования, руб.;

НАi – годовая норма амортизации i-го оборудования, %;

ТНИРi – время работы i-го оборудования за весь период разработки ПП, ч;

Тэфi – эффективный фонд времени работы i-го оборудования за год, ч/год;

i – вид оборудования;

n – количество оборудования i-го вида.

Все данные сведем в таблицу 10.

Таблица 10 – Расчет амортизационных отчислений

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Наимено­вание обо­рудования | Стоимость оборудования, руб. | Годовая норма  амортизации, % | Эффективный фонд времени работы обору­дования ч/год | Время работы оборудования для разработки ПП, ч | Сумма, руб. |
| Системный блок | 29990 | 20 | 1314 | 138 | 629,93 |
| Монитор | 8290 | 20 | 1314 | 138 | 174,13 |
| Клавиатура | 1590 | 20 | 1314 | 138 | 33,40 |
| Мышь | 1190 | 20 | 1314 | 138 | 25,00 |
| МФУ | 10990 | 20 | 219 | 1 | 10,04 |
| ИТОГО: | | | | | 872,48 |

Расчет прочих затрат

К прочим затратам (Зпр) относят оплату интернет-трафика, оплату процентов по банковским кредитам, прочие накладные расходы.

Накладные расходы учитывают затраты на управление и хозяйственное обслуживание проводимых работ:

 (5)

где ηн – коэффициент накладных расходов.

Коэффициент накладных расходов ηн составляет 0,21 основной заработной платы.

Расчет сметы затрат на разработку ПО

На основании полученных данных по отдельным статьям составляем смету затрат на разработку ПП (таблица 11).

Таблица 11 – Смета затрат

|  |  |
| --- | --- |
| Статьи затрат | Сумма, руб. |
| 1. Материальные затраты, в том числе:  * материалы * электроэнергия  1. Затраты на оплату труда 2. Отчисления на социальные нужды   4. Амортизация оборудования  5. Прочие затраты | 960,00  154,70  15307,94  4592,38  872,48  2795,36 |
| ИТОГО | 24682,86 |

Определение цены реализации ПП

Для нахождения цены программного продукта, используем таблицу 12.

Таблица 12 – Определение цены реализации 1 копии ПП

|  |  |
| --- | --- |
| Наименование статьи калькуляции | Сумма, руб. |
| Суммарные затраты на ПП | 24682,86 |
| Закладываемая прибыль (20 %) | 4936,57 |
| Итого, продажная цена без НДС | 29619,43 |
| НДС 20% | 5923,89 |
| Итого, продажная цена с НДС (Ц) | 35543,32 |

# ГЛАВА 4. ОХРАНА ТРУДА И БЕЗОПАСНОСТЬ ЖИЗНЕДЕЯТЕЛЬНОСТИ

Эксплуатация ЭВМ

Включение ЭВМ (ПК) производить согласно инструкции по включению и выключению на данную ЭВМ. Перед включением ЭВМ (ПК) необходимо ознакомиться с аппаратным журналом и убедиться, что все устройства ЭВМ были исправны при работе ранее. Если на устройствах ЭВМ проводились ремонтные работы необходимо проверить: Наличие и исправность заземления отдельных блоков; Исправность кабелей и их подключения, включения вентиляторов.

Запрещается включать устройства, к работе которых сотрудник не имеет доступа. Запрещается включать ЭВМ (отдельное устройство) при неисправной защите электропитания. Запрещается снимать крышки и щиты, закрывающие доступ к токоведущим частям. Запрещается пользоваться неисправной аппаратурой инструментом. Запрещается пользоваться электрическим паяльником с напряжением более 36В и незаземленном корпусом. При ремонте электропитания необходимо вывешивать плакаты "НЕ ВКЛЮЧАТЬ! РАБОТА НА ЛИНИИ!" Запрещается заменять съемные элементы и поводить пайку под напряжением. Запрещается включать и отключать разъемы кабелей электропитания и блоков вентиляторов под напряжением. Запрещается менять предохранители под напряжением, при замене руководствами их номиналами. Измерение напряжения на токоведущих частях с напряжением свыше 42В проводить, пользуясь резиновыми ковриками и изолированными щупами, при этом соблюдать особую предосторожность во избежание короткого замыкания смежных клемм и проводов. При техническом обслуживании съемных блоков системы электропитания их корпуса необходимо заземлить. Металлические корпуса измерительной аппаратуры должны быть заземлены. При прозвонке электрических цепей необходимо предварительно эти цепи обесточить и проверить отсутствие напряжения. При проведении всех работ необходимо присутствие не менее 2-х человек, допущенных к работе с электроустановками и имеющих соответствующую квалификационную группу для установок напряжением до 1000В. Все приведенные ремонтные работы необходимо записать в журнал учета ремонтных работ.

Противопожарные мероприятия

При техническом обслуживании и эксплуатации ЭВМ (ПК) необходимо строго соблюдать правила противопожарной безопасности. Запрещается хранить в машинном зале смазочные материалы. Включенные паяльники необходимо класть на специальные подставки. Рабочие места, проходы и выход не должны загромождаться посторонними предметами. По окончании работы все электроприборы должны быть выключены. Курение, пользование электронагревательными приборами, открытым огнем в данных классах ЗАПРЕЩАЕТСЯ! При возникновении пожара отключить оборудование (электроустановку), принять меры к ликвидации пожара, сообщить дежурному персоналу, руководству ЦНИТ. При ликвидации пожара применять средства тушения, гасящее вещество которых не проводит электрический ток (огнетушители углекислотные, порошковые). Лица, работающие в классах, должны быть обучены приемам освобождения пострадавшего от электрического тока, приемам искусственного дыхания, правилам оказания первой помощи и способам тушения пожара в производственном помещении. Лица, допустившие нарушения, несут дисциплинарную, административную, уголовную ответственность!

Меры предосторожности

К самостоятельной работе на ПК допускаются лица не моложе 18-ти лет, прошедшие медицинское освидетельствование, специальное обучение, инструктаж по охране труда на рабочем месте, изучившие “Руководство по эксплуатации” и усвоившие безопасные методы и приемы выполнения работы.

ПК должен подключаться к однофазной сети с нормальным напряжением 220 (120) В, частотой 50 (60) Гц и заземленной нейтрально. Заземляющие контакты розеток должны быть надежно соединены с контуром защитного заземления помещения. В помещении должен быть установлен автомат аварийного или рубильник общего отключения питания.

Запрещается самостоятельно производить ремонт ПК (его блоков), если это не входит в круг ваших обязанностей.

При эксплуатации ПК должны выполняться следующие требования, правила:

* не подключать и не отключать разъемы и кабели электрического питания при поданном напряжении сети;
* не оставлять ПК включенным без наблюдения;
* не оставлять ПК включенным во время грозы;
* по окончании работы отключить ПК от сети;
* устройства должны быть расположены на расстоянии 1 м от нагревательных приборов; рабочие места должны располагаться между собой на расстоянии не менее 1,5 метров;

- устройства не должны подвергаться воздействию прямых солнечных лучей;

- непрерывная продолжительность работы при вводе данных на ПК не должна превышать 4 часов при 8-часовом рабочем дне, через каждый час работы необходимо делать перерыв 5-10 минут, через 2 часа на 15 минут;

- в помещении, где расположена компьютерная техника, должен быть оборудован уголок пожаротушения.

Санитарно-гигиенические требования при работе с компьютером

В соответствии с СанПиН: 2.2.2.542-96 "Гигиенические требования к ВДТ и ПЭВМ. Организация работы"

Все вредности, возникающие при работе ВДТ и ПЭВМ можно разделить на три группы:

1. Параметры рабочего места и рабочей зоны.
2. Визуальные факторы (яркость, контрастность, мерцание изображения, блики).
3. Излучения (рентгеновское, электромагнитное излучение ВЧ и СВЧ диапазона, гамма-излучение, электростатические поля).

Внедрение ЭВМ имеет как положительные, так и отрицательные моменты. С одной стороны, это обеспечение более высокой эффективности производства за счет совершенствования технологического процесса и повышение производительности труда, а с другой - увеличение нагрузки на работающих в связи с интенсификацией производственной деятельности и специфическими условиями труда.

Условия труда работающих с ЭВМ характеризуются возможностью воздействия на них следующих производственных факторов: шума, тепловыделений, вредных веществ, статического электричества, ионизирующих и неионизирующих излучений, недостаточной освещенности, параметров технологического оборудования и рабочего места. Основными источниками шума в помещениях, оборудованных вычислительной техникой, являются принтеры, плоттеры, множительная техника и оборудование для кондиционирования воздуха, вентиляторы систем охлаждения, трансформаторы. Для снижения шума и вибрации в помещениях вычислительных центров оборудование, аппараты необходимо устанавливать на специальные фундаменты и амортизирующие прокладки, предусмотренные нормативными документами.

Уровень шума на рабочих местах не должен превышать 50 дБА. Нормируемые уровни шума обеспечиваются путем использования малошумного оборудования, применением звукопоглощающих материалов (специальные перфорированные плиты, панели, минераловатные плиты). Кроме того, необходимо использовать подвесные акустические потолки.

В помещениях с избытком тепла необходимо предусматривать регулирование подачи теплоносителя для соблюдения нормативных параметров микроклимата. Микроклиматические условия на рабочих местах в помещениях с вычислительной техникой должны соответствовать требованиям. Воздух, поступающий в рабочие помещения операторов ЭВМ, должен быть очищен от загрязнений, в том числе от пыли и микроорганизмов. Патогенной микрофлоры быть не должно.

При профессиональном отборе работников ЭВМ основное внимание обращается на состояние органов зрения: состояния мышечного равновесия глаз, положительный запас аккомодации, цветовую чувствительность, остроту зрения, рефракционную способность глаз, контрастную чувствительность и поле зрения.

Организация рабочего места

Приступая к работе на компьютере желательно:

Осмотреть рабочее место (расположение блоков и их состояние...).

Подобрать по высоте стул.

Монитор должен располагаться на уровне глаз и перпендикулярно углу зрения.

Экран монитора и защитный экран (с обеих сторон) должны быть чистыми.

Освещение должно соответствовать нормам СанПиН.

Не рекомендуется располагать монитор около яркого источника света, т.к. приходится повышать яркость и контрастность, что влечет за собой: увеличение нагрузки на глаза, излучения, выгорает люминофор экрана, сокращается срок службы монитора.

На мониторе не должно быть бликов, сильного контраста с внешним освещением.

Мышь располагается так, чтобы было удобно работать с ней. Провод должен лежать свободно. При работе с мышью по периметру коврика должно оставаться пространство не менее 2-5 сантиметров.

Клавиатуру следует располагать прямо перед пользователем, работающим на компьютере. По периметру оставляется свободное место 2-5 сантиметров.

Десять шагов по обеспечению безопасности и комфорта домашнего компьютера:

1. Системный блок. Устанавливать системный блок следует устойчиво на ровной поверхности таким образом, чтобы углы системного блока не выступали за края стола. Лучше установить его на полу (специальной подставке) под столом.
2. Стол. Лучше всего использовать специальный компьютерный стол со столешницей достаточной глубины, местом под системный блок и выдвижной доской для клавиатуры и мышки. Это позволит использовать стол и для других целей.
3. Питание. Чтобы надежно обезопасить компьютер от неожиданностей и неприятностей, имеет смысл потратить средства на источник бесперебойного питания или, как минимум, установить заземленную розетку с фильтрами.
4. Чистота. Содержите вентиляционные отверстия системного блока и монитора открытыми, не захламляйте стол бумагами, журналами и книгами. Ухудшение условий охлаждения может существенно сократить срок жизни монитора и компьютера. Если на столе скапливается много документов, имеет смысл установить монитор на специальном кронштейне (подставке), чтобы приподнять его.
5. Мышь. Даже в работе с мышкой следует обзавестись хорошими привычками. Есть правила, которые стоит неукоснительно выполнять:

* Запястье должно быть прямым. Не стоит сжимать мышку с силой. Это вызывает ненужное напряжение мышц, нарушает кровообращение и затрудняет движения.
* Нельзя работать с мышкой полностью вытянутой рукой. Подбирая рабочий стол, необходимо выбрать такой, чтобы за мышкой не приходилось тянуться слишком далеко. Для движения мышкой должно быть достаточно свободного места.
* Различные дополнительные приспособления, такие как коврики для мыший, подушки и подпорки для отдыха запястий рук, помогают только в том случае, если правильно используются.

Некоторые предпочитают обычной мыши трэкбол. Этот манипулятор имеет свои достоинства и недостатки: для него требуется меньше пространства на столе и меньше приходиться двигать рукой, зато гораздо больше приходиться работать пальцами.

1. Клавиатура. Оптимальная высота стола или выдвижной полки для клавиатуры 68 – 73 см. над полом. Высоту стула и стола следует подобрать так, чтобы минимально напрягать мышцы плечей, рук и запястий. Запястья могут касаться стола перед клавиатурой. Но ни в коем случае нельзя переносить на них хотя бы часть веса тела. Клавиатура регулируется по высоте наклона. Угол наклона подбирается индивидуально. Некоторые клавиатуры, например, Microsoft Natural Keyboard, имеют большие возможности для регулировки. Такие клавиатуры имеют расщепление в середине буквенной части и особую форму, предназначенную для более естественного положения запястий над клавишами. Однако такой клавиатурой имеет смысл обзаводиться только в том случае, если пользователь много пишет и владеет слепым десятипальцевым методом печати. В остальных случаях от такой клавиатуры нет никакого эргономического выигрыша. В компьютерных салонах можно найти специальные подпорки и подушки для установки перед клавиатурой, предназначенные для отдыха запястий и предупреждения туннельного синдрома – острых болей из-за перегрузки и повреждения сухожилий запястий.
2. Кабели. У компьютера довольно много кабелей и проводов. Их лучше всего собрать сзади стола. Но стол необходимо отодвинуть от стены, чтобы в этом пространстве кабели не запутались между собой.
3. Освещение. Чтобы уменьшить отражение и блики на экране, монитор размещается перпендикулярно окну. Обеспечивается достаточное местное освещение, направленное на документ, чтобы можно было выключать общее освещение в помещении и снижать яркость бликов.
4. Стул. Не стоить экономить на хорошем стуле или удобном кресле, регулируемом по высоте и специально предназначенном для работы за компьютером. Перед компьютером очень важно соблюдать правильную позу. А для младших членов семьи под столом необходимо установить стул, чтобы ноги стояли на твердой устойчивой поверхности.
5. Средства ухода. В компьютерном уголке следует предусмотреть место, где будут храниться средства ухода за машиной: минимум инструментов, аэрозоль со сжатым воздухом, бутылочка спирта для очистки мыши и CD-ROM, а также фланелевый лоскут для протирки компьютера.

# ЗАКЛЮЧЕНИЕ

В ходе выполнения дипломной работы была изучена предметная область составления расписания образовательной организации, в которой выяснилось, что оно состоит из 5 действий (составления плана, составление педагогической нагрузки, составление и утверждение расписания, и подготовка итоговых отчетов), и техническое задание, котором были описаны требования к программе.

Далее необходимо было выбрать из множества инструментальных средств разработки программ подходящие под задание, в ходе которого были выбраны VisualStudio и DB Browser for SQLite. Также были составлены системные требования к программному продукту.

После был спроектирована структура программы, в которой была составлена диаграмма вариантов использования и ER-диаграмма хранения данных, также был выбран язык программирования C# и технология создания форм программы WPF.

Следующим шагом была разработана непосредственно программа, которая хранит информацию о педагогах, классах и кабинетах, также существующие дисциплины, и в каких классах они изучаются. И непосредственно форма для составления расписания и экспортирования в Excel-документ, а также формирование отчета о результатах проведения за определенный период.

Далее необходимо было провести тестирование программы. Результат которой показал, что ошибки в программе отсутствуют.

После были рассчитаны затраты на программу, итоговая стоимость которой составила 35 543 рублей в том числе с НДС 20%, из которой прибыль составила 4 936 рублей.

Следующим шагом было необходимо составить документацию, а именно спецификацию, руководство оператора и руководство программиста.

Цель и задачи, поставленные в данной работе, выполнены.

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Гордеев, С. И. Организация баз данных в 2 ч. Часть 1: учебник для среднего профессионального образования / С. И. Гордеев, В. Н. Волошина. — 2-е изд., испр. и доп. — М.: Издательство Юрайт, 2021. — 310 с.
2. Лаврищева, Е. М. Программная инженерия. Парадигмы, технологии и CASE-средства: учебник для вузов / Е. М. Лаврищева. — 2-е изд., испр. — Москва: Издательство Юрайт, 2022. — 280 с.
3. Гордеев, С. И. Организация баз данных в 2 ч. Часть 2: учебник для среднего профессионального образования / С. И. Гордеев, В. Н. Волошина. — 2-е изд., испр. и доп. — М.: Издательство Юрайт, 2021. — 513 с.
4. Маркин, А.В. Программирование на SQL в 2 ч. Часть 1: учебник и практикум для вузов / А. В. Маркин. — 2-е изд., перераб. и доп. — М.: Издательство Юрайт, 2021. — 403 с.
5. Мамонова, Т. Е. Информационные технологии. Лабораторный практикум: учеб. пособие для СПО / Т. Е. Мамонова. — М.: Издательство Юрайт, 2019. — 178 с.
6. Трофимов, В. В. Алгоритмизация и программирование : учебник для вузов / В. В. Трофимов, Т. А. Павловская ; под редакцией В. В. Трофимова. — Москва: Издательство Юрайт, 2022. — 137 с.
7. Моделирование процессов и систем: учебник и практикум для вузов / Е. В. Стельмашонок, В. Л. Стельмашонок, Л. А. Еникеева, С. А. Соколовская; под редакцией Е. В. Стельмашонок. — М.: Издательство Юрайт, 2021
8. Нестеров, С. А. Базы данных: учебник и практикум для вузов / С. А. Нестеров. — М.: Издательство Юрайт, 2021. — 230 с.
9. Кудрина, Е. В. Основы алгоритмизации и программирования на языке C#: учебное пособие для среднего профессионального образования / Е. В. Кудрина, М. В. Огнева. — Москва: Издательство Юрайт, 2022. — 322 с.
10. Нестеров, С. А. Базы данных: учебник и практикум для среднего профессионального образования / С. А. Нестеров. — М.: Издательство Юрайт, 2021. — 230 с.
11. Советов, Б. Я. Информационные технологии: учебник для вузов / Б. Я. Советов, В. В. Цехановский. — 7-е изд., перераб. и доп. — М.: Издательство Юрайт, 2021. — 327 с.
12. Стружкин, Н. П. Базы данных: проектирование. Практикум: учебное пособие для среднего профессионального образования / Н. П. Стружки, В. В. Годин. — М.: Издательство Юрайт, 2021. — 291 с.
13. Методы оптимизации: учебник и практикум для вузов / Ф. П. Васильев, М. М. Потапов, Б. А. Будак, Л. А. Артемьева; под редакцией Ф. П. Васильева. — Москва: Издательство Юрайт, 2022. — 375 с.
14. Стружкин, Н. П. Базы данных: проектирование. Практикум: учебное пособие для вузов / Н. П. Стружки, В. В. Годин. — М.: Издательство Юрайт, 2021. — 291 с.
15. Черпаков, И. В. Основы программирования: учебник и практикум для вузов / И. В. Черпаков. — Москва: Издательство Юрайт, 2022. — 219 с.
16. Тузовский, А. Ф. Объектно-ориентированное программирование: учебное пособие для вузов / А. Ф. Тузовский. — Москва: Издательство Юрайт, 2022. — 206 с.

РЕСУРСЫ УДАЛЕННОГО ДОСТУПА

1. Документация Microsoft, <https://docs.microsoft.com/ru-ru/>
2. Онлайн-пособие Metanit, [https://metanit.com/](https://metanit.com/sql/)
3. Интернет-ресурс Wikipedia, <https://ru.wikipedia.org/>

1. Моделирование процессов и систем: учебник и практикум для вузов / Е. В. Стельмашонок, В. Л. Стельмашонок, Л. А. Еникеева, С. А. Соколовская; под редакцией Е. В. Стельмашонок. — М.: Издательство Юрайт, 2021 [↑](#footnote-ref-1)
2. Моделирование процессов и систем: учебник и практикум для вузов / Е. В. Стельмашонок, В. Л. Стельмашонок, Л. А. Еникеева, С. А. Соколовская; под редакцией Е. В. Стельмашонок. — М.: Издательство Юрайт, 2021 [↑](#footnote-ref-2)
3. Гордеев, С. И. Организация баз данных в 2 ч. Часть 1: учебник для среднего профессионального образования / С. И. Гордеев, В. Н. Волошина. — 2-е изд., испр. и доп. — М.: Издательство Юрайт, 2021. — 310 с. [↑](#footnote-ref-3)
4. Нестеров, С. А. Базы данных: учебник и практикум для вузов / С. А. Нестеров. — М.: Издательство Юрайт, 2021. — 230 с. [↑](#footnote-ref-4)
5. Интернет-ресурс Wikipedia, https://ru.wikipedia.org/ [↑](#footnote-ref-5)
6. Документация Microsoft, <https://docs.microsoft.com/ru-ru/> [↑](#footnote-ref-6)
7. Нестеров, С. А. Базы данных: учебник и практикум для вузов / С. А. Нестеров. — М.: Издательство Юрайт, 2021. — 230 с. [↑](#footnote-ref-7)
8. Документация Microsoft, https://docs.microsoft.com/ru-ru/ [↑](#footnote-ref-8)
9. Онлайн-пособие Metanit, [https://metanit.com/](https://metanit.com/sql/) [↑](#footnote-ref-9)
10. Кудрина, Е. В. Основы алгоритмизации и программирования на языке C#: учебное пособие для среднего профессионального образования / Е. В. Кудрина, М. В. Огнева. — Москва: Издательство Юрайт, 2022. — 322 с. [↑](#footnote-ref-10)
11. Кудрина, Е. В. Основы алгоритмизации и программирования на языке C#: учебное пособие для среднего профессионального образования / Е. В. Кудрина, М. В. Огнева. — Москва: Издательство Юрайт, 2022. — 322 с. [↑](#footnote-ref-11)
12. Интернет-ресурс Wikipedia, https://ru.wikipedia.org/ [↑](#footnote-ref-12)