

Министерство науки и высшего образования Российской Федерации  
Филиал Федерального государственного автономного образовательного учреждения  
высшего образования  
«Южно-Уральский государственный университет  
(национальный исследовательский университет)»  
в г. Нижневартовске

Кафедра «Гуманитарные, естественно-научные и технические дисциплины»

ДОПУСТИТЬ К ЗАЩИТЕ  
Зав. кафедрой «ГЕНТД»  
к.филос.н., доцент  
\_\_\_\_\_/И.Г. Рябова/  
« \_\_\_\_ » \_\_\_\_\_ 2021 г.

## Разработка программного приложения «Планировщик обучения»

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА  
К ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЕ  
ЮУрГУ-09.03.04.2021.049.ПЗ ВКР

Консультанты

Экономическая часть

к.э.н., доцент  
\_\_\_\_\_/С.В. Данилова/  
« \_\_\_\_ » \_\_\_\_\_ 2021 г.

Руководитель работы

к.п.н., доцент  
\_\_\_\_\_/Е.З. Никонова/  
« \_\_\_\_ » \_\_\_\_\_ 2021 г.

Автор работы

Студент группы НвФл-422  
\_\_\_\_\_/Г.Р. Эминов/  
« \_\_\_\_ » \_\_\_\_\_ 2021 г.

Нормоконтролер

Старший преподаватель  
\_\_\_\_\_/Л.Н. Буйлушкина/  
« \_\_\_\_ » \_\_\_\_\_ 2021 г.

## АННОТАЦИЯ

Эминов Г.Р. Разработка программного приложения «Планировщик обучения» - Нижневартовск: филиал ЮУрГУ, Программная инженерия: 2021, 71 с., 32 ил., 6 табл., библиогр. список – 20 наим., 3 прил.

Данная выпускная квалификационная работа выполнена в соответствии с техническим заданием и представляет собой описание разработки программного обеспечения для автоматизации процесса планирования значимых видов работ по охране труда, в частности обучения сотрудников по необходимым программам и учета медицинских осмотров.

Изучена предметная область, разработан проект программы с интуитивно понятным интерфейсом. Выполнен расчет технико-экономической эффективности от внедрения разработанного программного продукта. Проведен литературный обзор.

					<b>ЮУрГУ-09.03.04.2021.049.ПЗ ВКР</b>					
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>	<b>Разработка программного приложения «Планировщик задач»</b>	<i>Лит.</i>			<i>Лист</i>	<i>Листов</i>
<i>Разработал</i>	<i>Эминов Г.Р.</i>					<i>В</i>	<i>К</i>	<i>Р</i>	6	71
<i>Проверил</i>	<i>Никонова Е.З.</i>					<i>Филиал ФГАОУ ВО «ЮУрГУ (НИУ)» в г. Нижневартовске</i>				
<i>Н.контр.</i>	<i>Буйлушкина Л.Н.</i>					<i>кафедра «ГЕНТД»</i>				
<i>Утвердил</i>	<i>Рябова И.Г.</i>									

## ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ.....	8
1 АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ .....	10
1.1 Обзор и анализ области внедрения .....	10
1.2 Обзор аналогов разработки.....	11
1.3 Выбор инструментов для разработки .....	13
2 РАЗРАБОТКА ПРИЛОЖЕНИЯ .....	16
2.1 Проектирование базы данных .....	16
2.2 Конструирование базы данных .....	20
2.3 Реализации фильтра списка сотрудников .....	21
2.4 Использование механизма отправки и получения сообщений .....	23
2.5 Использование глобального класса приложения .....	24
2.6 Главное окно.....	26
2.7 Окно редактирования информации о сотруднике .....	30
2.8 Вывод текущего списка сотрудников в файл.....	36
2.9 Формирование направлений на осмотры .....	38
2.10 Окно настроек приложения.....	46
2.11 Использование Material Design в интерфейсе.....	47
3 ОРГАНИЗАЦИОННО-ЭКОНОМИЧЕСКИЙ РАЗДЕЛ .....	51
3.1 Организационно-экономическая характеристика деятельности предприятия.....	51
3.2 Анализ финансовых показателей деятельности предприятия .....	55
3.3 Расчет затрат на разработку программного обеспечения.....	56
3.4 Расчет доходов и финансовых результатов.....	57
ЗАКЛЮЧЕНИЕ .....	61
БИБЛИОГРАФИЧЕСКИЙ СПИСОК .....	62

## ПРИЛОЖЕНИЯ

ПРИЛОЖЕНИЕ А. ТЕХНИЧЕСКОЕ ЗАДАНИЕ НА СОЗДАНИЕ АВТОМАТИЗИРОВАННОЙ СИСТЕМЫ «ПЛАНИРОВЩИК ОБУЧЕНИЯ» .....	65
ПРИЛОЖЕНИЕ Б. ЛИСТИНГ МОДУЛЯ ПРОГРАММЫ.....	68
ПРИЛОЖЕНИЕ В. КОМПАКТ-ДИСК.....	70

## ВВЕДЕНИЕ

В настоящее время разработка программного обеспечения является важной составляющей в работе многих компаний и предприятий. От качества и функциональности программного обеспечения зависят многие факторы, от денежной прибыли и до человеческого здоровья. Поэтому важно выбрать надежное программное обеспечение, и чтобы его функциональность соответствовала поставленным задачам.

Объектом исследования данной работы является работа специалиста, отвечающего за охрану труда в филиале ФГАОУ ВО «ЮУрГУ (НИУ)» в г. Нижневартовске.

Предметом исследования является автоматизация процессов по учету обучения сотрудников по программам охраны труда и сопутствующим им, а также прохождение различных медицинских осмотров.

Целью работы является готовый программный продукт, позволяющий автоматизировать учет обучения сотрудников охране труда и отвечающий всем требованиям технического задания.

Задачи, которые предстоит выполнить в ходе работы:

- Настройка проекта. Подключение необходимых библиотек.
- Проектирование базы данных.
- Проектирование архитектуры приложения.
- Конструирование приложения и разработка интерфейса приложения.

Теоретической основой исследования послужили существующие аналоги, их функционал, принцип работы и устройство пользовательского интерфейса. Методологической основой исследования являются межгосударственные стандарты, определяющие эргономику взаимодействия человека и системы и общие требования к разработке и документированию программного обеспечения.

Новизной исследования является новый подход к разработке, при котором используются современные инструменты и методологии, позволяющие значительно

сократить временные затраты на разработку программного обеспечения, что в свою очередь снизит общие денежные затраты на разработку.

# 1 АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ

## 1.1 Обзор и анализ области внедрения

Охрана труда — это система сохранения жизни и здоровья работников в процессе трудовой деятельности, включающая в себя правовые, социально-экономические, организационно-технические, санитарно-гигиенические, лечебно-профилактические, реабилитационные и иные мероприятия, образующие механизм реализации конституционного права граждан на труд (ст. 37 Конституции РФ) в условиях, отвечающих требованиям безопасности и гигиены.

Создание и применение таковых мероприятий означает переход от низкоэффективного метода реагирования на многочисленные случаи производственного травматизма и профессиональных заболеваний к методу современного устранения их причин - нейтрализации и предотвращению профессиональных рисков, т.е. к управлению профессиональными рисками.

В целях создания механизма управления профессиональными рисками в последние годы проводится значительная работа по приведению Трудового кодекса РФ и иных нормативных правовых актов в соответствие с новым порядком организации охраны труда в производственных организациях, вызванных к жизни переходом к управлению профессиональными рисками. В частности, только в 2011 году в раздел «Охрана труда» ТК РФ федеральными законами было внесено 15 изменений и дополнений. С января 2021 года изменились требования к перечню документов. Кроме того, значительно возросли штрафы за нарушение правил хранения, учета, ведения документации.

В последние годы разработано значительное количество нормативных правовых актов: утверждены новые правила специальной оценки рабочих мест, условий труда, прохождения работниками обязательных медицинских осмотров, аккредитации организаций, оказывающих услуги в области охраны труда и др.

Всеми этими процессами необходимо управлять, и за это отвечает ответственный по охране труда.

Для автоматизации работы со всеми необходимыми процессами и документами необходимо программное обеспечение, удовлетворяющее следующим потребностям:

- Ведение списка сотрудников.
- Возможность изменение информации о сотрудниках.
- Фильтрация сотрудников по необходимым параметрам.
- Создание отчетной документации по выбранному сотруднику.
- Сохранение отфильтрованного списка в файл.
- Отслеживание обучения сотрудников по выбранным программам обучения
- Возможность создания, редактирования и удаления программ обучения
- Отслеживание актуальности медицинских документов и осмотров у сотрудников

Создание подобного программного продукта требует тщательного проектирования для достижения лучшего результата, поэтому следует выбрать инструменты, максимально подходящие для решения вышеперечисленных задач.

## 1.2 Обзор аналогов разработки

Для контроля за состоянием учета обучения сотрудников на предприятиях были разработаны некоторые решения, позволяющие в какой-то мере автоматизировать сопутствующие процессы. Одни из самых часто используемых решений это «1С:Производственная безопасность. Охрана труда» и «АКИО 2.0». Рассмотрим каждое решение подробнее.

«1С:Производственная безопасность. Охрана труда» – решение для автоматизации задач охраны труда на предприятиях различных отраслей, обеспечивающее процессы учета, планирования, контроля и формирования аналитической отчетности по охране труда. Стоимость решения – от 117000 руб за клиентскую лицензию на 20 рабочих мест. При этом решение имеет функционал, не



требующийся для работы на предприятии. То есть при использовании этого программного обеспечения предприятие будет переплачивать за ненужный функционал.

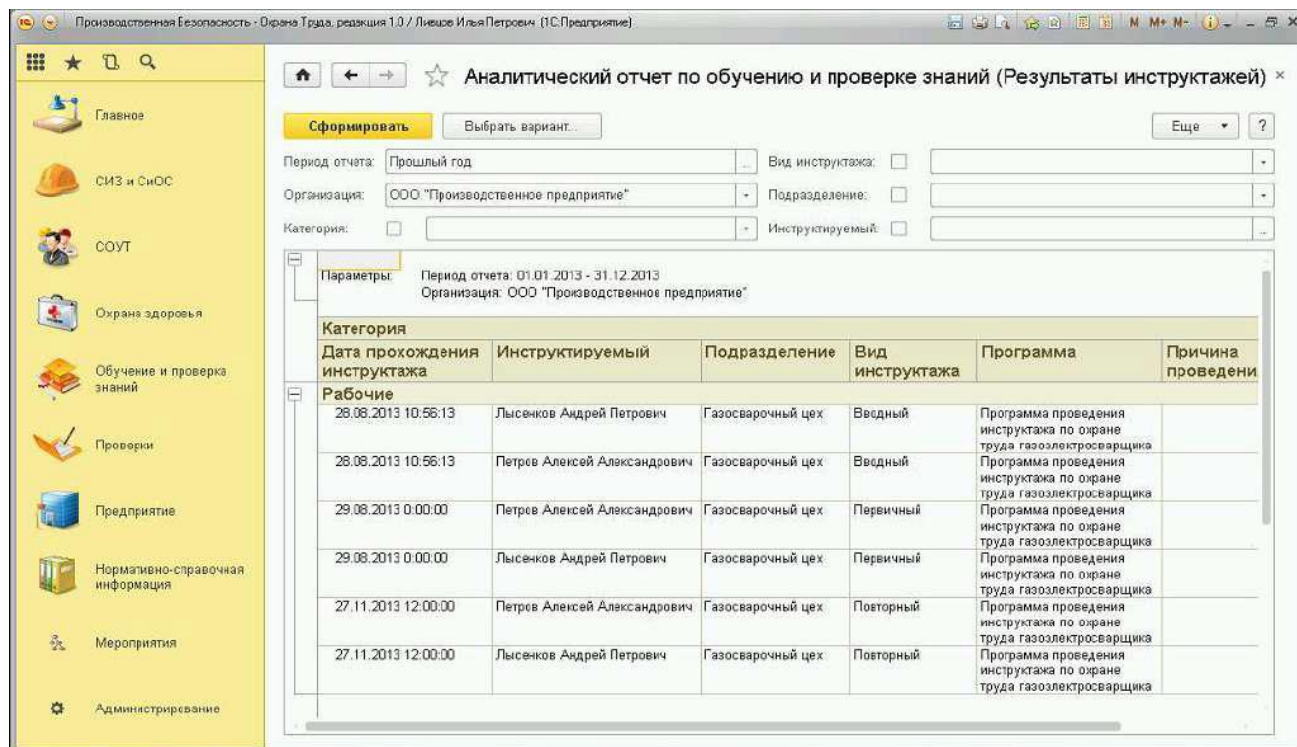


Рисунок 1.1 – Пример интерфейса программы «1С: Производственная безопасность. Охрана труда»

АКИО 2.0 – Программа, предназначенная для автоматического контроля сроков проведения инструктажей и обучения в областях охраны труда, пожарной безопасности, электробезопасности и т. п. Распространяется на бесплатной основе. При этом программа реализована в среде Microsoft Excel, что подразумевает использование таблиц в качестве элемента управления, а это в свою очередь не является самым эффективным методом работы. Также программа имеет минимальные функции автоматизации учета, т.е. фильтрация списка, сортировка и вывод на печать конкретного списка будет требовать больших временных затрат, чем если бы сотрудник использовал при работе программное обеспечение, разработанное с оглядкой на принцип работы отдела и сотрудников предприятия.

### 1.3 Выбор инструментов для разработки

Для написания логики программы существует несколько языков программирования, такие как C++, C# и Python и т.д.

Язык программирования C++ является мощным инструментом при разработке программного обеспечения, так как позволяет управлять всеми ресурсами системы, на которой запускается. Минусами использования данного языка является высокий порог вхождения для написания относительно сложных программных решений, а также ручное управление памятью. Язык программирования Python является менее мощным инструментом, так как требует большей вычислительной мощности при написании относительно сложных программных решений, при этом сложно найти достаточно функциональную среду программирования для этого языка, в которой можно эффективно разрабатывать программные решения.

Основным языком программирования при разработке приложения был выбран C#. Это объектно-ориентированный язык программирования, особенностью которого является кроссплатформенность, которая достигается за счет платформы .NET Framework, то есть программа, написанная на этом языке, может быть запущена на различных устройствах, главное, чтобы на нем поддерживалась платформа .NET Framework, на данный момент это Windows, MacOS и Linux. Дополнительными преимуществами данного языка программирования являются более низкий порог вхождения чем у C++, позволяющий писать относительно сложные решения, прикладывая при этом меньшие усилия, а также автоматический сборщик мусора, который следит за использованием памяти и высвобождает ненужные ресурсы при необходимости.

Для конструирования программного обеспечения требуется также интегрированная среда программирования (далее ИСП), для данной работы в качестве ИСП была выбрана Microsoft Visual Studio. Данная ИСП является самым мощным инструментом при разработке среди таких аналогов как Rider, Eclipse, MonoDevelop, так как она позволяет писать с помощью нее .NET приложения с

использованием системы построения клиентских приложений Windows Presentation Foundation (далее WPF), которая была выбрана для разработки интерфейсной части приложения.

WPF — усовершенствованный аналог WinForms, система для построения клиентских приложений Windows с визуально привлекательными возможностями взаимодействия с пользователем, графическая подсистема в составе .NET Framework (начиная с версии 3.0), использующая язык XAML.

Так как от программы требовалась возможность вывода в файл упорядоченной информации о сотрудниках, то необходимо было выбрать инструмент, который бы позволил работать с документами Microsoft Word и с таблицами Microsoft Excel. В качестве инструмента для решения данной задачи была выбрана бесплатная версия библиотек Spire.DOC и Spire.XLS. Данные библиотеки являются единственными целесообразными решениями, так как позволяют в несколько строк кода осуществить запись, чтение и сохранение документов.

Для того чтобы хранить информацию о сотрудниках необходимо подключить к проекту базу данных. Так как подключение существующих решений вроде SQL Server или Microsoft Access хоть и позволяло работать с чтением и записью информации с очень высокой скоростью, но могло потенциально осложнить дальнейшую поддержку кода, поэтому было принято решение написать собственный вариант хранения информации, а именно хранение в виде JSON строки.

JSON – это текстовый формат хранения информации. Его особенностью является то, что в этот формат очень легко можно привести практически любой объект языка C# и сохранить его в виде текстового файла. Также формат позволяет создать из него новый объект языка C# и заполнить его поля в соответствии с сохраненной информацией. Для реализации этой возможности была выбрана библиотека Newtonsoft.JSON, использующая механизм сериализации, который и переводит объекты C# в формат JSON. Данная библиотека является наиболее

функциональной среди таких аналогов как Json.NET, Jil и встроенной библиотеки .NET Framework System.Text.Json.

Выводы по разделу один:

В ходе анализа предметной области была выявлена необходимость в использовании программного обеспечения, позволяющего повысить эффективность работы специалистов предприятия, так как с каждым годом меняются стандарты ведения документации и необходимо программное обеспечение, способное реагировать на изменение требований. Было проведен анализ существующих аналогов разработки, среди которых не оказалось такого, которое удовлетворяло бы всем требованиям, в связи с чем принято решение о разработке программы в соответствии с составленным техническим заданием. Определены инструменты для разработки программного продукта.

## 2 РАЗРАБОТКА ПРИЛОЖЕНИЯ

### 2.1 Проектирование базы данных

Проектирование базы данных будет являться одним из важнейших этапов разработки программного обеспечения, так как от ее результатов зависит правильность выполнения программы и сложность дальнейшей разработки.

Перед конструированием программного обеспечения необходимо составить техническое задание (Приложение А) и продумать логику взаимодействия между различными компонентами программы. Автоматизация учета охраны труда подразумевает под собой хранение информации о сотрудниках предприятия, результатов их обучения по различным программам обучения, информации о необходимых документах, а также автоматическое уведомление об истечении срока действия каких-либо документов сотрудников.

Перед тем как проектировать базу данных нужно выяснить какие таблицы необходимы для хранения необходимой информации. В данном случае потребуются следующие таблицы:

- Employee – таблица, которая содержит в себе всю информацию, относящуюся к сотрудникам предприятия.
- TrainProgram – таблица, хранящая в себе тип программы обучения и всех сотрудников, обучающихся по этой программе.
- AdditionalExamination – таблица, содержащая все дополнительные осмотры сотрудника.
- AppSettings – таблица, хранящая в себе настройки приложения.

После определения всех таблиц базы данных необходимо указать все атрибуты каждой таблицы и их типы:

Employee:

- ID – уникальный ключ, числовое значение.
- Name – имя, строковое значение.

- Surname – фамилия, строковое значение.
- Patronymic – отчество, строковое значение.
- Sex – пол, строковое значение.
- BirtDate – дата рождения, значение даты и времени.
- Position – должность, строковое значение.
- Fired – уволен ли сотрудник, числовое значение.
- LastTrainingDate – дата последнего обучения, значение даты и времени.
- NextTrainingDate – дата следующего обучения, значение даты и времени.
- TrainingOrganization – название организации, проводившей обучение, строковое значение.

- TrainingOrganizationProtocolNum – номер протокола обучающей организации, строковое значение.

- TrainingOrganizationInfo – дополнительная информация об организации, проводившей обучение, строковое значение.

- AdditionalExaminations – информация о прохождении дополнительных осмотров, список ключей.

- Commentary – дополнительная информация о сотруднике, строковое значение.

- NowTraining – обучается ли сейчас сотрудник, числовое значение.

- TrainPrograms – ключи всех программ обучения, по которым обучается сотрудник, список ключей.

- MedicalExam – информация о прохождении медосмотра.

#### TrainProgram:

- ID – уникальный ключ, числовое значение.

- Name – название программы обучения, строковое значение.

#### AdditionalExamination:

- ID – уникальный ключ, числовое значение.

- Name – название осмотра, строковое значение.
- Date – дата осмотра, значение даты и времени.
- ValidityDays – срок действия осмотра.
- NotifyDays – количество дней перед окончанием срока действия осмотра.

AppSettings:

- TrainNotifyDays – количество дней перед окончанием срока действия обучения.
- MedExpiredNotifyDays – количество дней перед окончанием срока действия медосмотра.
- MedValidityDays – срок действия медосмотра.

После определения всех атрибутов таблиц, можно наглядно увидеть взаимодействие между таблицами базы данных:

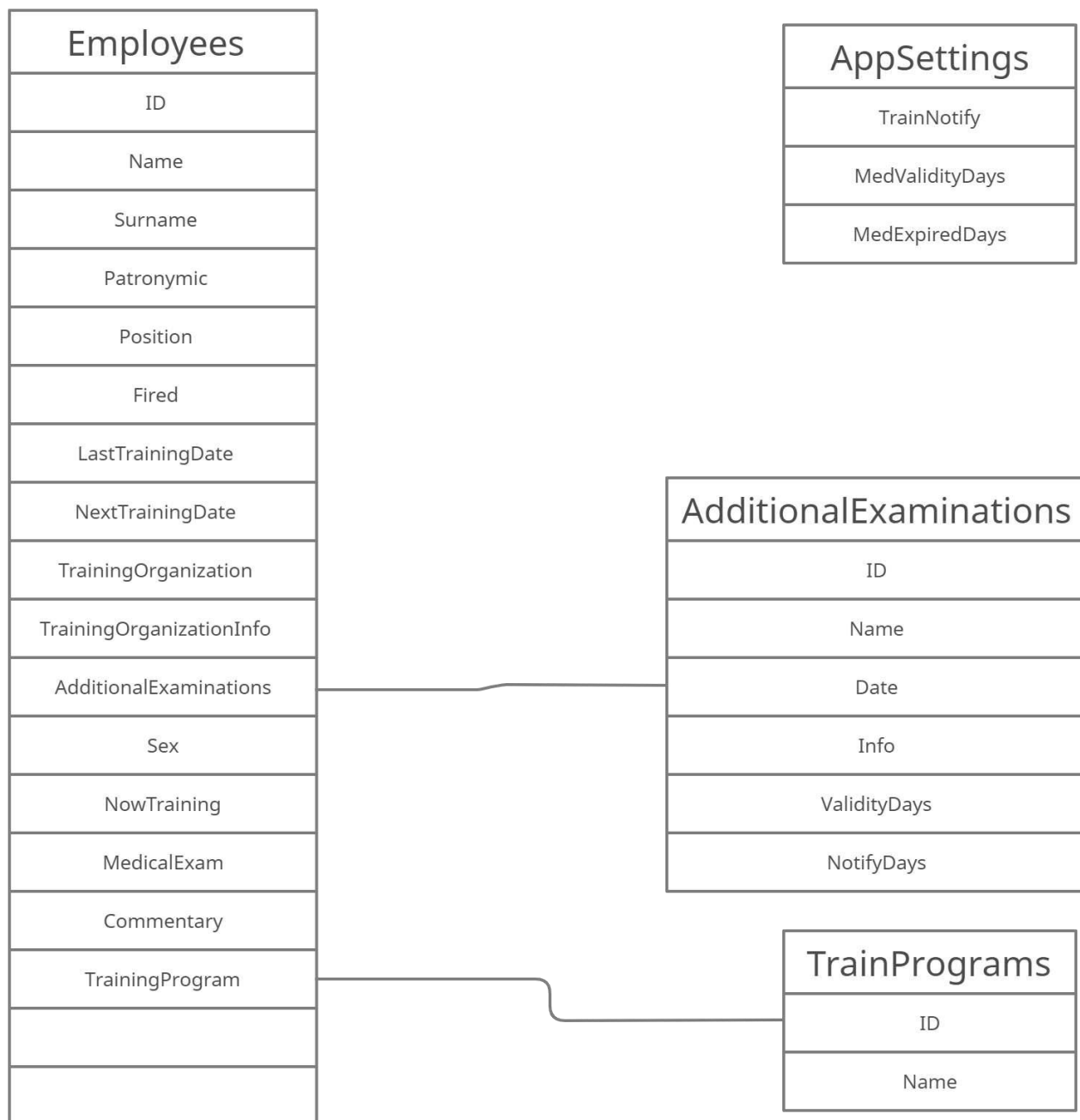


Рисунок 2.1 – Блок схема базы данных



## 2.2 Конструирование базы данных

Основой для базы данных будет являться класс, в котором содержатся таблицы базы данных, и который отвечает за сохранение данных из таблицы в файл и за загрузку данных из файла в таблицу.

Таблица 2.1 – Описание класса базы данных

<b>DataBase</b>
+Load()
+Save()
-AppData

Таблица 2.2 – Описание класса данных приложения

<b>AppData</b>
-List<Employee>
-List<TrainProgram>
-AppSettings
-MedDocumentTemplate
-PsychoDocumentTemplate

Поля MedDocumentTemplate и PsychoDocumentTemplate – это классы, содержащие информацию о том, какие поля при формировании направления необходимо сохранить и заполнить при открытии.

Метод Load() отвечает за загрузку данных из файла, он использует с указанный при создании объекта базы данных путь к файлу данных, загружает значения из него и переводит их в объект AppData.

Метод Save() отвечает за сохранение данных в файл, он также использует указанный путь к файлу, переводит объект AppData в строку типа JSON и сохраняет по указанному пути.

Оба этих метода проверяют наличие файла по указанному пути, и если его нет, то он создается.

### 2.3 Реализации фильтра списка сотрудников

Для улучшения пользовательского опыта в программе используется фильтр списка сотрудников, посредством которого появляется возможность отображать только тех сотрудников, которые удовлетворяют условиям фильтра.

Объектно-ориентированный стиль языка C# позволяет использовать некоторые готовые приемы, которые полезны при проектировании различных частей программы. Один из таких приемов – паттерн «Декоратор». Суть данного паттерна заключается в том, что во время исполнения программы объекты класса, унаследованного от другого абстрактного класса, могут оборачиваться в другие классы, так же унаследованные от того же абстрактного класса. В связи с этим объект может получить новый функционал, при этом нет необходимости писать для каждой реализации основной функционал – механизмы наследования позволяют использовать уже написанный код.

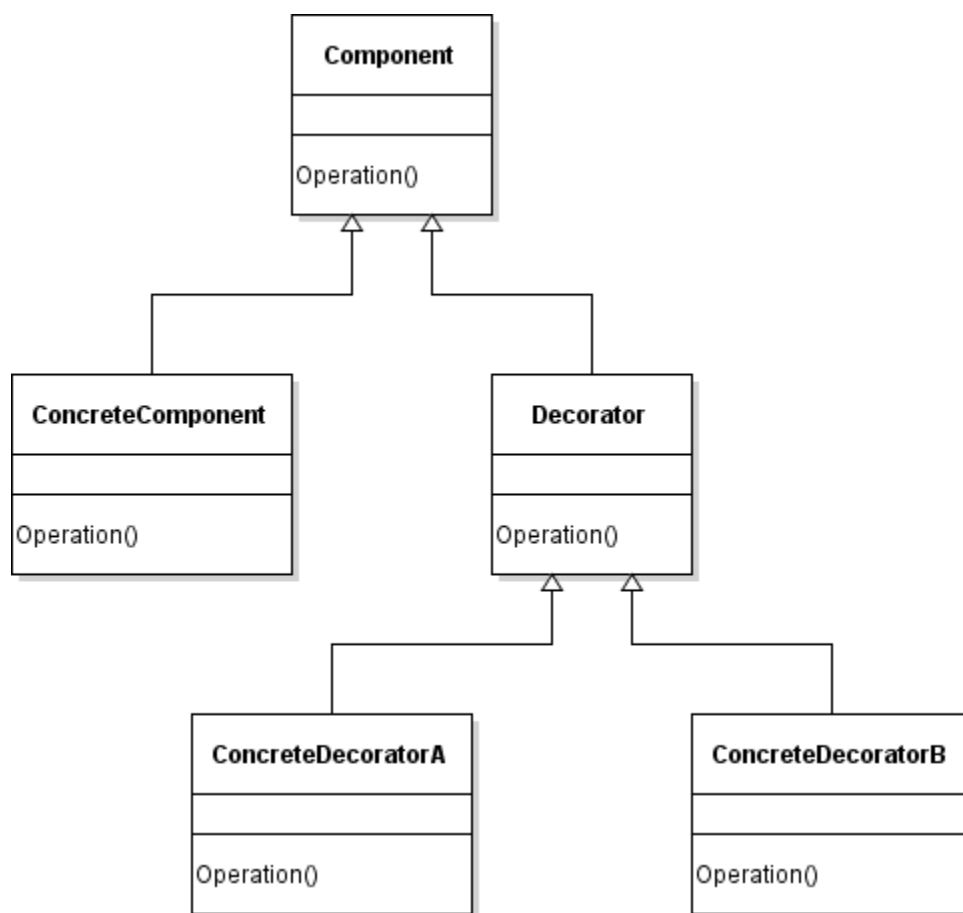


Рисунок 2.2 – Принцип работы паттерна «Декоратор»

В программе в качестве фильтра выступает объект, которому можно через публичный метод указать список сотрудников, и, обернув его в реализацию необходимого фильтра, выполнить метод, в котором описан алгоритм фильтрации, и получить обратно отфильтрованный список сотрудников.

## 2.4 Использование механизма отправки и получения сообщений

Паттерны проектирования позволяют добиться уменьшения времени, требуемого для реализации функций приложения. Паттерн «Наблюдатель» позволяет решить проблему, возникающую, когда код становится сильно связанным, то есть каждому объекту в приложении необходимо знать о состоянии другого объекта, чтобы продолжить работу без ошибок.

Суть работы паттерна заключается в том, что есть некий посредник, объект, который берет на себя ответственность за обработку сообщений, поступающих от других объектов. Для подписки на событие необходимо обратиться к экземпляру класса, предоставляющего подписку, вызвать у него метод `Subscribe` и передать в него ключ-идентификатор события и делегат, в котором хранятся действия, совершаемые при получении сообщения. После проделанных действий, каждый раз, когда в приложении будет обращение к классу, реализующему паттерн и вызван у него метод `Send` с ключом-идентификатором сообщения, у всех объектов, подписавшихся на получение сообщений по заданному ключу-идентификатору, будет вызван метод, указанный при подписке.

В разрабатываемом приложении паттерн «Наблюдатель» используется для:

- оповещения об изменении выделенного объекта в списке сотрудников, что позволяет во всех необходимых объектах обновлять информацию о выбранном сотруднике, не обращаясь к объектам напрямую. Это позволяет легко изменять отдельные части приложения, не меняя при этом каждый раз класс, в котором обрабатывается изменение выделения в списке сотрудников.

- оповещения для сохранения базы данных, что позволяет при необходимости замены механизма сохранения данных не изменять при этом остальные классы.

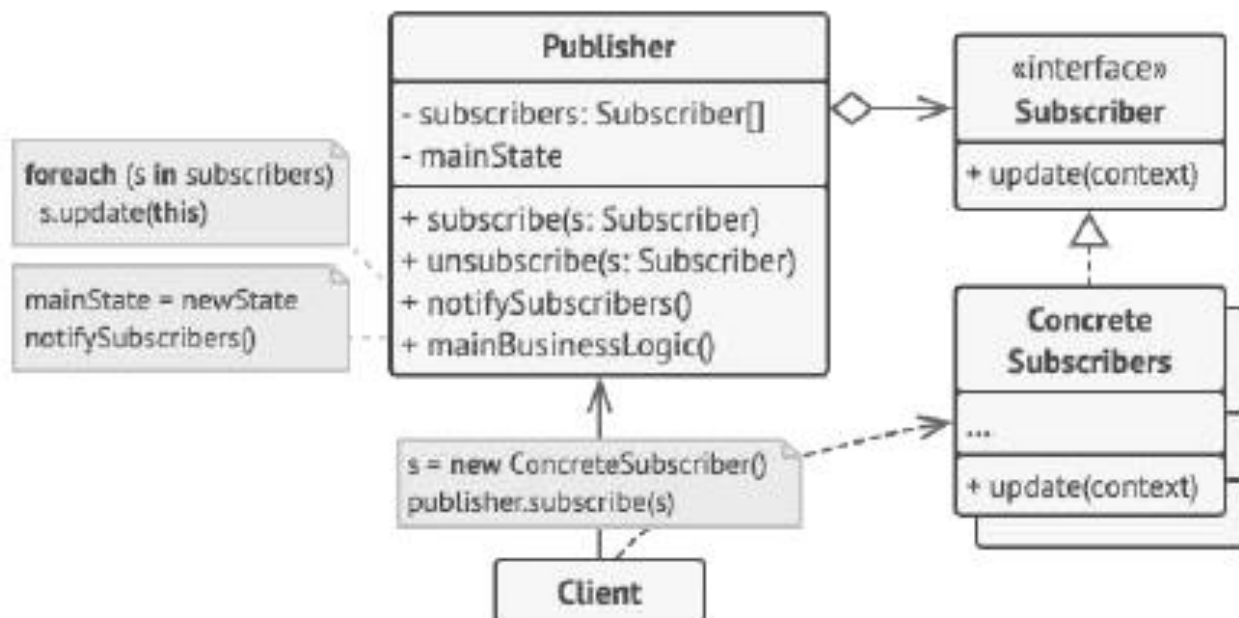


Рисунок 2.3 – Принцип работы паттерна «Наблюдатель»

## 2.5 Использование глобального класса приложения

Во время работы приложения, объектам классов часто требуется обратиться к некоторому другому глобальному классу, для изменения его состояния или для получения данных. Один из способов обращения к такому классу это передача этого класса в необходимый объект при его создании, однако такой подход требует больших временных затрат, так как при создании каждого объекта нужно следить за тем, чтобы в нем присутствовала ссылка на глобальный класс.

Эту проблему решает паттерн «Одиночка», он предоставляет статическую ссылку на экземпляр глобального класса, и при этом гарантирует что этот экземпляр является единственным во всем приложении. Это позволяет избежать случаев, когда приложение аварийно завершает свою работу, так как объект обратился к пустой ссылке на другой объект.

Суть работы паттерна заключается в том, что глобальный класс хранит в себе ссылку на самого себя и предоставляет другим объектам статичное публичное поле того же типа, что и сам класс. При обращении других объектов к этому полю происходит проверка ссылки, хранящейся в классе, если она пустая, то создается

новый экземпляр класса, и ссылка на него сохраняется, если же ссылка уже существует, то она возвращается объекту, который обратился к полю. Тем самым, данный алгоритм гарантирует, что экземпляр глобального класса является единственным во всем приложении, и что каждый объект будет обращаться к одному и тому же экземпляру, что в свою очередь исключает возникновение ошибок, возникающих, когда объекты получают разные данные.

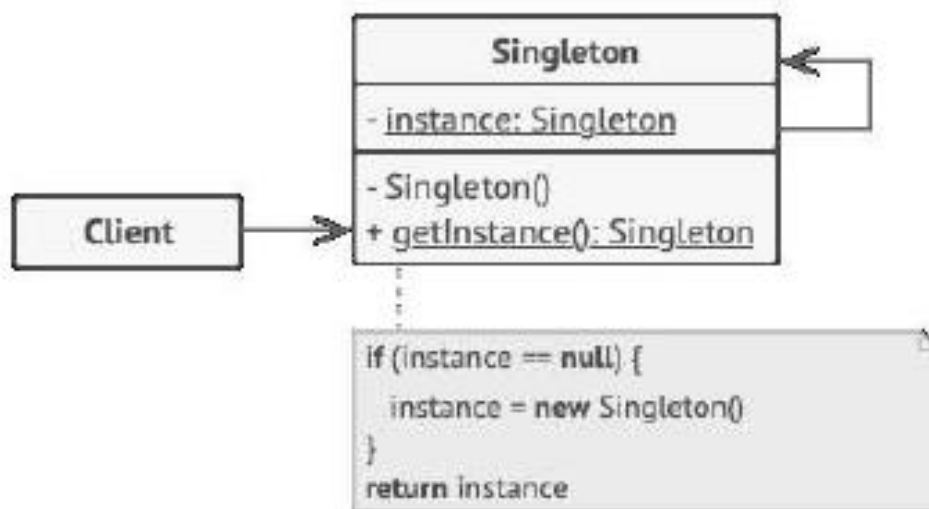


Рисунок 2.4 – Принцип работы паттерна «Одиночка»

В разрабатываемом приложении паттерн «Одиночка» используется для реализации класса, отвечающего за контекст приложения, хранящего в себе ссылки на базу данных, на текущего выбранного сотрудника и т.д.

## 2.6 Главное окно

Главное окно должно отображать следующую информацию:

- Список сотрудников.
- Элементы управления для редактирования информации о сотрудниках.
- Элементы управления для фильтрации отображаемого списка.
- Элементы управления для сортировка отображаемого списка.
- Элементы управления для создания отчета о сотруднике.
- Элементы управления для изменения настроек приложения.
- Элементы управления для вывода текущего отображаемого списка в файл.

Для создание всех необходимых элементов в программе используются заранее заготовленные элементы управления, такие как поле для ввода текста, изменяемый счетчик целочисленных значений, выпадающее меню и т.п.

Чтобы отобразить информацию о сотрудниках используется элемент управления ListView.

Имя	Дата рождения	Пол	Должность	Уволен	Обучался	Последнее обучение
Петров Геннадий Васильевич	9 июня 1977 г.	М	Механик	Нет	Да	10 сентября 2000 г.
Горбанев Богдан Дмитриевич	14 июля 1988 г.	М	Методист	Да	Нет	11 февраля 2021 г.
Коновалова Оксана Ивановна	16 июня 2021 г.	Ж	Инженер	Нет	Нет	19 февраля 2021 г.
Бондарчук Михаил Евстафьев	16 июня 2021 г.	М	Уборщик	Нет	Нет	16 июня 2021 г.
Рябова Ирина Геннадьевна	18 июня 2021 г.	Ж	Зав. кафедры "ТЕНТД"	нет	нет	17 февраля 2021 г.

Рисунок 2.5 – Элемент управления ListView

Как видно на рисунке, данный элемент управления имеет возможность создания нескольких колонок для разделения и упорядочивания информации.

Для правильного сопоставления информации о сотрудниках с колонками в списке используется механизм привязки данных в WPF.

С помощью языка разметки XAML можно указать какие публичные поля объекта C# необходимо привязать к указанному визуальному элементу, тем самым при отрисовке элемента будет указана соответствующая информация.

Для фильтрации списка используются различные элементы управления, такие как:

- Поле для ввода текста.
- Флажок.
- Выпадающий список.
- Поле для ввода целочисленных значений.

A screenshot of a text input field with the placeholder text "Имя" (Name) and a horizontal line for text entry.

Рисунок 2.6 – Поле для ввода текста для фильтрации по имени сотрудника

A screenshot of two dropdown menus. The first menu is labeled "Программа обучения" (Training program) and the second is labeled "Пол" (Gender) with "Все" (All) selected.

Рисунок 2.7 – Выпадающие списки для фильтрации по программам обучения и полу сотрудника

A screenshot of two numeric input fields. The first is labeled "Возраст" (Age) and "От" (From) with the value "0". The second is labeled "До" (To) with the value "100". Both fields have up and down arrow icons.

Рисунок 2.8 – Поля для ввода целочисленных значений для фильтрации по возрасту сотрудника

Для сортировки списка сотрудников по различным критериям необходимо щелкнуть левой кнопкой мыши по необходимому заголовку колонки в списке, при этом при повторном нажатии сортировка будет выполнена в обратном порядке.

Для редактирования информации о сотрудниках используются кнопки



Добавить

Удалить

Изменить

Рисунок 2.9 – Кнопки для добавления, изменения и удаления информации о сотруднике

При нажатии на кнопку «Добавить» открывается окно с пустыми полями, в котором необходимо заполнить всю соответствующую информацию. Кнопка «Изменить» активна только при наличии выбранного сотрудника и открывает такое же окно, но при этом все поля заполнены в соответствии с выбранным сотрудником. При нажатии на кнопку «Удалить» открывается диалоговое окно с подтверждением удаления выбранного сотрудника.

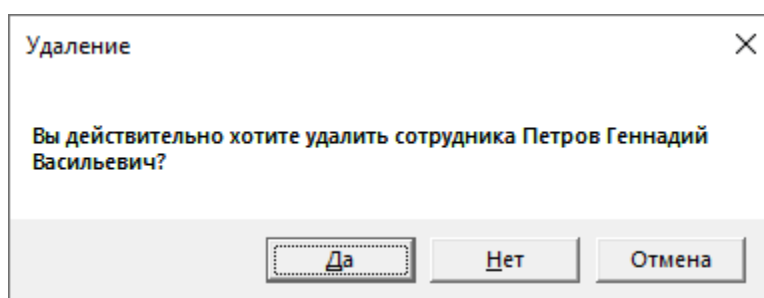


Рисунок 2.10 – Диалоговое окно с подтверждением удаления сотрудника

Для вывода текущего списка сотрудников в файл также используется кнопка, открывающая окно для настройки информации, записываемой в файл.

Сохранить текущий список...

Рисунок 2.11 – Кнопка для открытия окна сохранения текущего списка в файл

В главном окне также используется меню, через которое можно получить доступ к настройкам приложения и окну формирования направления на осмотр для выбранного сотрудника.

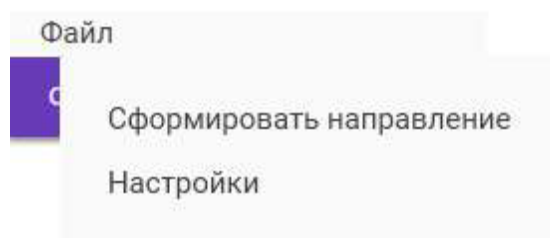


Рисунок 2.12 – Выпадающее меню «Файл»

Для просмотра уведомлений об окончании срока действия документов необходимо перейти на нужную вкладку в верхней части окна.

При нажатии на вкладку, к списку сотрудников применяется соответствующий фильтр, который отображает только тех сотрудников, чья разница окончания срока действия документов и текущей даты меньше или равна значению, указанному в настройках.



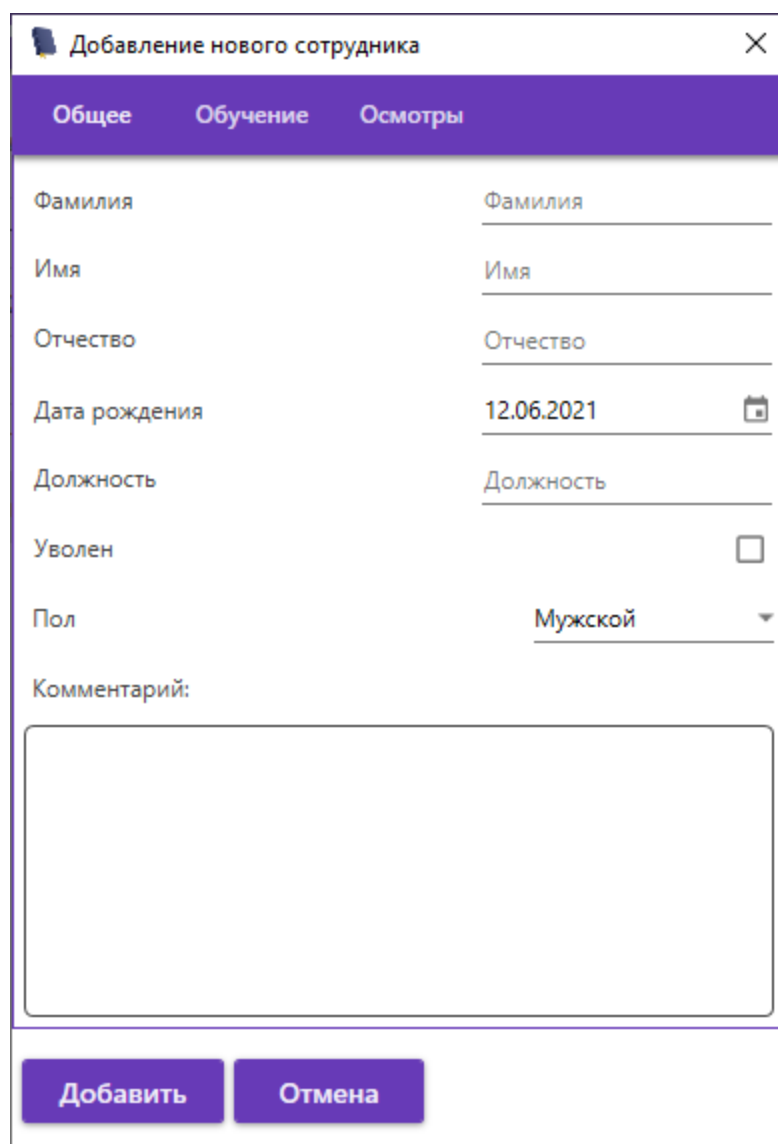
Рисунок 2.13 – Вкладки уведомлений

## 2.7 Окно редактирования информации о сотруднике

Окно редактирования информации о сотруднике открывается при нажатии кнопки «Добавить» или «Изменить» на главном окне.

Окно редактирования имеет три вкладки:

- Общее.
- Обучение.
- Осмотры.

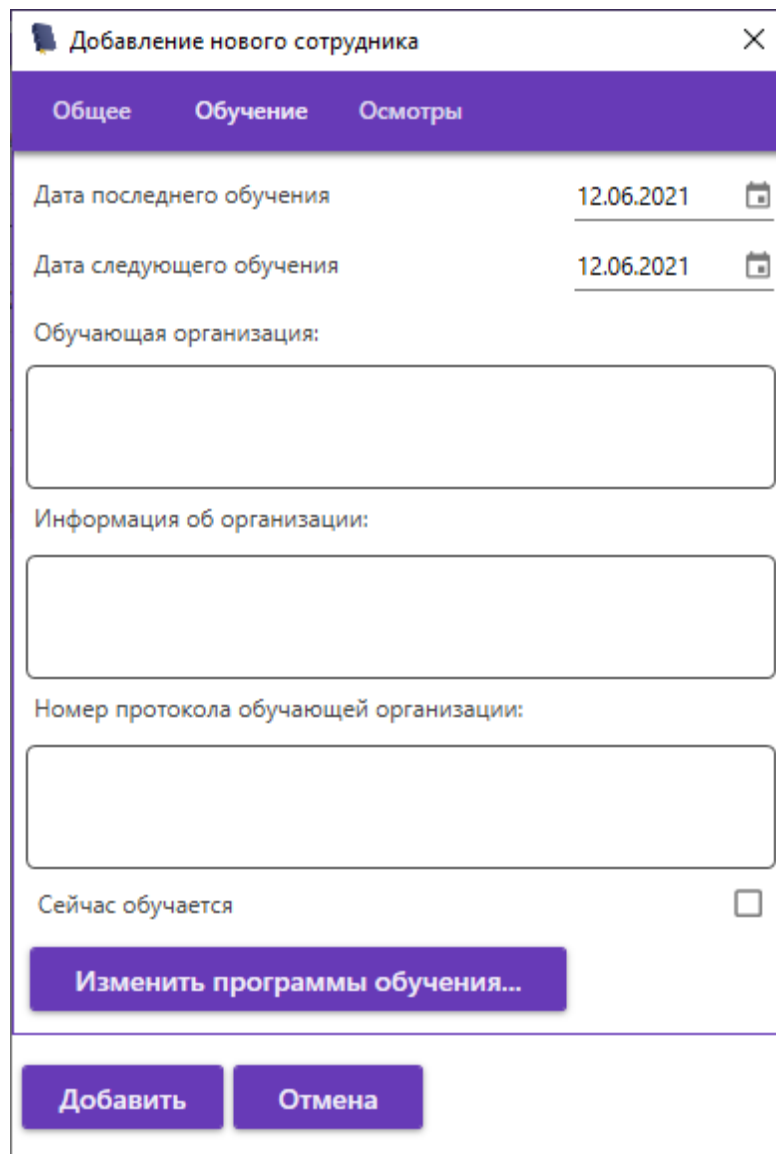


The screenshot shows a window titled "Добавление нового сотрудника" (Add new employee) with a close button (X) in the top right corner. Below the title bar is a purple tabbed interface with three tabs: "Общее" (General), "Обучение" (Education), and "Осмотры" (Examinations). The "Общее" tab is currently selected and highlighted. The form contains the following fields:

- Фамилия (Surname): Text input field with placeholder "Фамилия".
- Имя (Name): Text input field with placeholder "Имя".
- Отчество (Patronymic): Text input field with placeholder "Отчество".
- Дата рождения (Date of birth): Date picker showing "12.06.2021".
- Должность (Position): Text input field with placeholder "Должность".
- Уволен (Dismissed): Checkmark input field, currently unchecked.
- Пол (Gender): Dropdown menu showing "Мужской" (Male).
- Комментарий (Comment): Large text area for notes.

At the bottom of the window are two buttons: "Добавить" (Add) and "Отмена" (Cancel).

Рисунок 2.14 – Окно добавления нового сотрудника с открытой вкладкой «Общее»



Добавление нового сотрудника

Общее    **Обучение**    Осмотры

Дата последнего обучения    12.06.2021

Дата следующего обучения    12.06.2021

Обучающая организация:

Информация об организации:

Номер протокола обучающей организации:

Сейчас обучается

Изменить программы обучения...

Добавить    Отмена

Рисунок 2.15 – Окно добавления нового сотрудника с открытой вкладкой «Обучение»

Добавление нового сотрудника

Общее    Обучение    **Осмотры**

Тип медосмотра: Периодический

Дата медосмотра: 12.06.2021

Дополнительные осмотры:

Название	Дата	Доп. Инф.
Осмотр 1	6/12/2021 3:20:01 AM	
Осмотр 2	6/12/2021 3:20:15 AM	
Осмотр 3	6/12/2021 3:22:14 AM	
Осмотр 4	6/12/2021 3:22:20 AM	
Осмотр 5	6/12/2021 3:22:25 AM	

Добавить    Изменить    Удалить

Добавить    Отмена

Рисунок 2.16 – Окно добавления нового сотрудника с открытой вкладкой «Осмотры»

Вкладка «Общее» содержит общую информацию о сотруднике, такую как фамилия, имя, отчество, пол, дата рождения, должность, уволен ли сотрудник и комментарий.

Вкладка «Обучение» содержит информацию о программах обучения у сотрудников, датах текущего и следующего обучения и об обучающей организации.

Для настройки программ обучения необходимо нажать на кнопку «Изменить программы обучения...», при этом откроется окно, которое имеет два списка. В левом списке указаны текущие программы обучения, по которым обучается

сотрудник, в правом же указаны доступные программы обучения. При помощи кнопок «>>>» и «<<<<» можно перемещать выбранную программу обучения между списками.

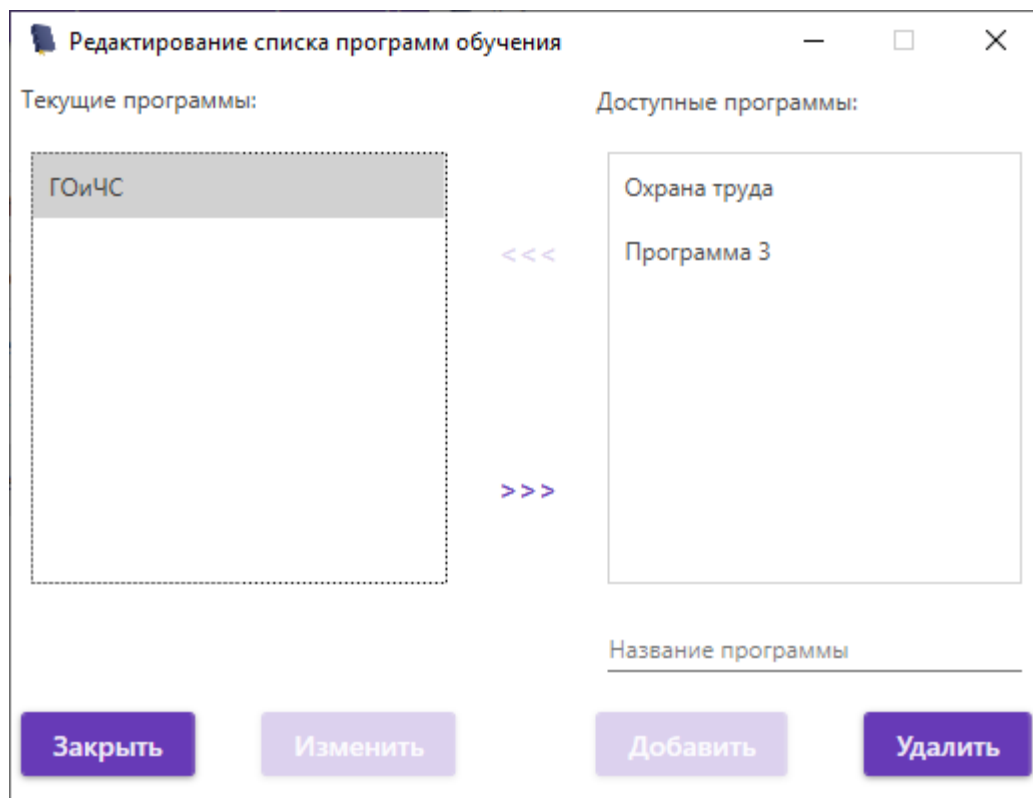


Рисунок 2.17 – Окно редактирования списка программ обучения

Для добавления новой программы обучения необходимо ввести ее название в поле для ввода и нажать кнопку добавить. Для изменения названия программы обучения необходимо выбрать программу, ввести новое название в поле ввода и нажать кнопку «Изменить», при этом откроется диалоговое окно с подтверждением изменения.

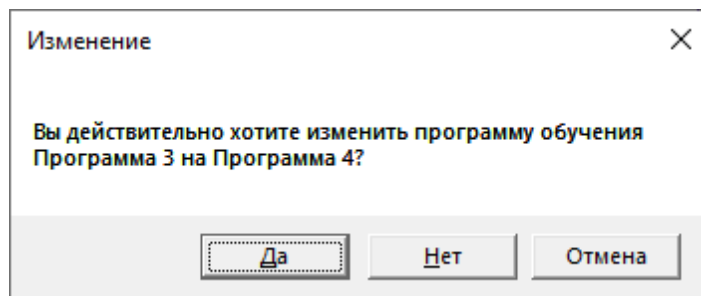


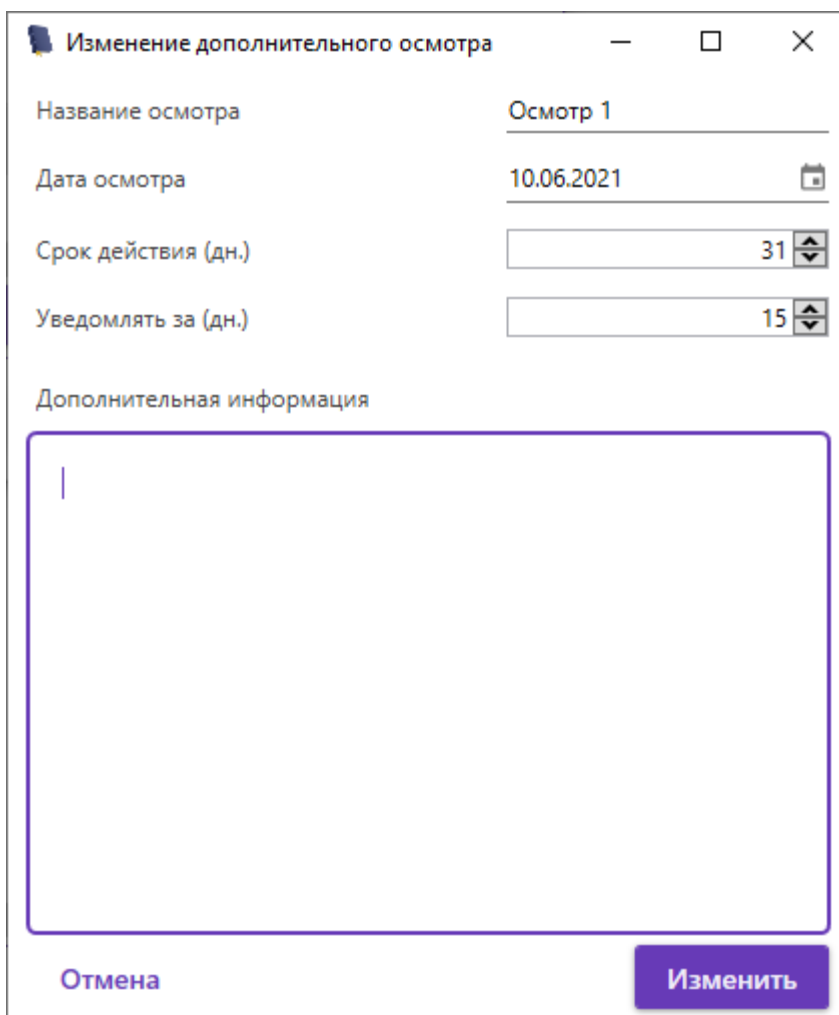
Рисунок 2.18 – Диалоговое окно с подтверждением изменения названия программы обучения

Вкладка «Осмотры» содержит информацию о медицинских осмотрах сотрудника и сроках их действия.

Для добавления осмотра необходимо нажать кнопку «Добавить», при этом откроется окно с полями для ввода информации об осмотре.

Рисунок 2.19 – Окно добавления нового осмотра

Для изменения информации об осмотре, необходимо выбрать нужный осмотр и нажать кнопку «Изменить», при этом откроется окно с заполненными полями, соответствующими выбранному осмотру.



Изменение дополнительного осмотра

Название осмотра: Осмотр 1

Дата осмотра: 10.06.2021

Срок действия (дн.): 31

Уведомлять за (дн.): 15

Дополнительная информация

Отмена Изменить

Рисунок 2.20 – Окно изменения информации об осмотре

Для удаления осмотра необходимо выбрать осмотр подлежащий удалению и нажать на кнопку «Удалить», при этом откроется диалоговое окно с подтверждением удаления.



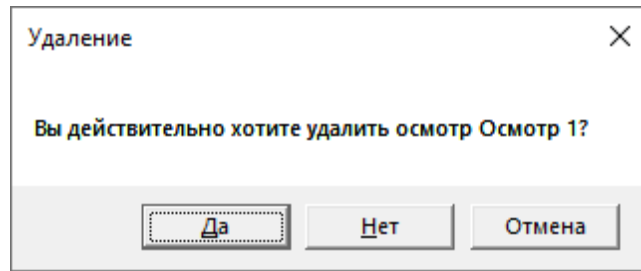


Рисунок 2.21 – Диалоговое окно с подтверждением удаления осмотра

## 2.8 Вывод текущего списка сотрудников в файл

Для реализации функции вывода текущего списка сотрудников в файл используется библиотека Spire.XLS, которая позволяет работать с таблицами Microsoft Excel. Так как класс сотрудника содержит в себе большое количество полей с информацией, то необходимо добавить возможность настройка выводимой информации, для этого используется окно с элементами управления – переключателями. Каждый переключатель подписан и отвечает за отдельное поле с информацией.

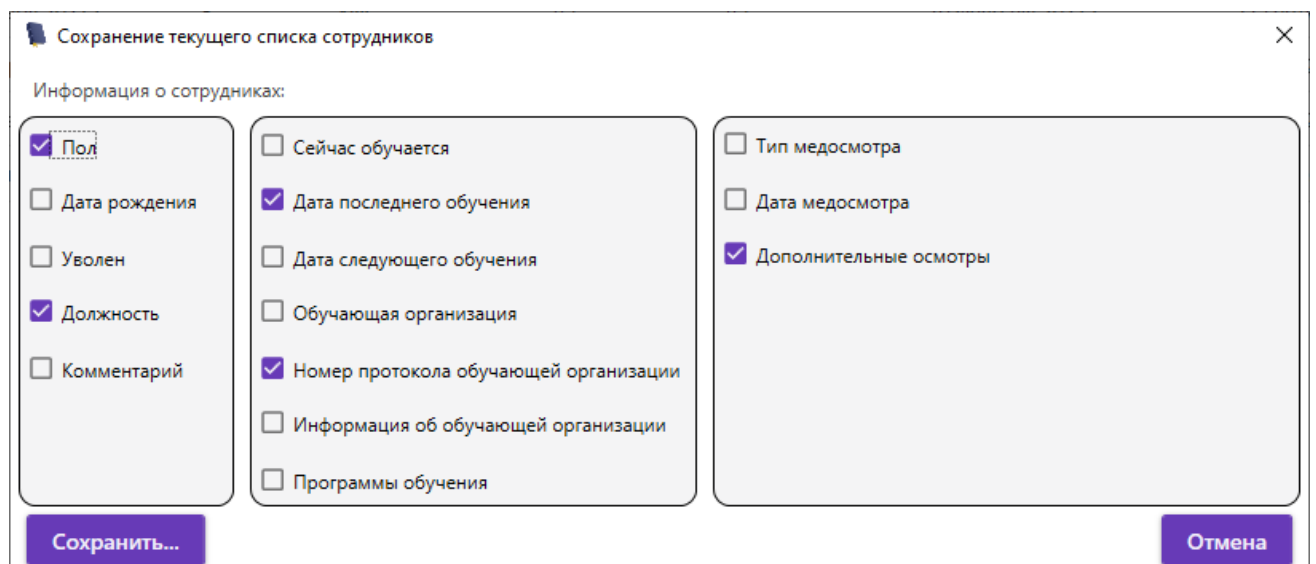


Рисунок 2.22 – Окно сохранения текущего списка сотрудников

После выбора необходимых полей и нажатия кнопки «Сохранить» будет открыто диалоговое окно с предложением выбрать место расположения нового файла. Выбранное место используется в алгоритме для создания нового файла с расширением .xls, затем, в цикле, происходит проверка каждого поля, если оно отмечено то создается отдельная колонка в таблице с названием этого поля и для каждого сотрудника записывается значение.

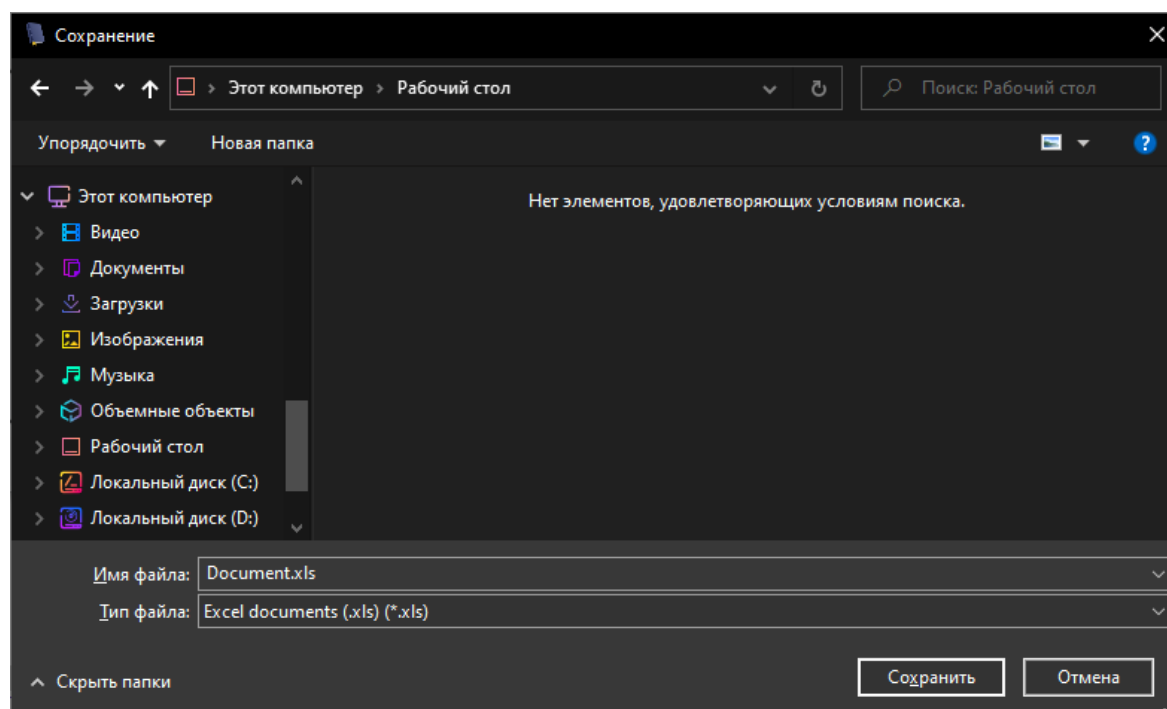


Рисунок 2.23 – Диалоговое окно выбора места расположения нового файла

После записи всех значений в таблицу, она сохраняется в указанном файле и затем ее можно открыть при помощи Microsoft Excel и дальнейшие операции проводить уже в нем.

	A	B	C	D	E	F
	<b>Имя</b>	<b>Дата рождения</b>	<b>Должность</b>	<b>Дата последнего обучения</b>	<b>Дата следующего обучения</b>	<b>Номер протокола обучающей организации</b>
1						
2	Петров Геннадий Васильевич	9 июня 1977 г.	Механик	10 сентября 2020 г.	7 января 2022 г.	1324134
3	Горбачев Богдан Дмитриевич	14 июля 1966 г.	Методист	11 февраля 2021 г.	24 сентября 2021 г.	857657657
4	Коновалова Оксана Ивановна	16 июня 2021 г.	Инженер	19 февраля 2021 г.	12 ноября 2021 г.	
5	Бондарчук Михаил Евстафьевич	16 июня 2021 г.	Уборщик	16 июня 2021 г.	16 июня 2021 г.	
6	Рябова Ирина Геннадьевна	16 июня 2021 г.	Зав. кафедры "ГЕНТД"	17 февраля 2021 г.	12 ноября 2021 г.	
7						
8						
9						

Рисунок 2.24 – Пример созданной таблицы

## 2.9 Формирование направлений на осмотры

Сотрудники предприятия с некоторой периодичностью должны проходить медицинские осмотры. Для прохождения осмотра необходимо направление, в котором указаны данные сотрудника.

Так как на предприятии вышеописанные направления заполнялись вручную, было принято решение реализовать в разрабатываемом приложении функционал автоматического формирования направлений

Для предприятия необходимо два вида направлений: направление на медосмотр и направление на психологическое освидетельствование. Каждое направление должно содержать различную информацию как о сотруднике, так и о предприятии в целом.

Для формирования направлений было принято решение использовать библиотеку Spire.DOC, которая позволяет проводить различные манипуляции над


документами Microsoft Word, но интересен именно функционал, позволяющий искать ключевые слова в документе и заменять их на другие.

Чтобы сформировать отчет необходимо выбрать сотрудника, а затем в контекстном меню «Файл» выбрать пункт «Сформировать направление», при этом откроется окно, в котором будут указаны поля с недостающей информацией для формирования направления. Так как часто некоторые поля направления повторяются, в приложении учтена возможность сохранения этих полей, и при следующем формировании направления будет открыто окно с уже заполненными полями.

Создание отчета


**Сотрудник: Петров Геннадий Васильевич**

Медосмотр    Псих. осв.

Дата  

Номер

Наименование структурного подразделения

Тип направления  

Содержание

Вредные и (или) опасные производственные факторы. Вид работы

Специалист по кадрам

Рисунок 2.25 – Окно формирования направления на медицинский осмотр

Рисунок 2.26 – Окно формирования направления на психиатрическое освидетельствование

В открытом окне располагаются две вкладки: «Медосмотр» и «Псих. Осв.». В зависимости от выбранной вкладки будет сформировано соответствующее направление. Вкладка «Медосмотр» отвечает за всю информацию, необходимую для формирования направления на медицинский осмотр, вкладка «Псих. Осв.» в свою очередь отвечает за информацию для направления на психологическое освидетельствование.

Шаблоны для формирования направлений составлены в соответствии с приказом Министерства Здравоохранения Российской Федерации от 28 января 2021 г. № 29н об утверждении порядка проведения обязательных предварительных и периодических медицинских осмотров работников, предусмотренных частью

четвертой статьи 213 трудового кодекса российской федерации, перечня медицинских противопоказаний к осуществлению работ с вредными и (или) опасными производственными факторами, а также работам, при выполнении которых проводятся обязательные предварительные и периодические медицинские осмотры.

В качестве ключей в документе используется следующий шаблон: <НАЗВАНИЕ\_КЛЮЧА>. В том месте, где необходимо автоматически подставить данные нужно указать соответствующий ключ, который будет использоваться в дальнейшем для поиска. Документ с заполненными ключами должен располагаться в папке с программой.



МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ  
ФИЛИАЛ ЮЖНО-УРАЛЬСКОГО ГОСУДАРСТВЕННОГО УНИВЕРСИТЕТА  
В г. НИЖНЕВАРТОВСКЕ

НАПРАВЛЕНИЕ  
НА <TYPE> МЕДИЦИНСКИЙ ОСМОТР

<<DATE\_DAY>> <DATE\_MONTH> <DATE\_YEAR>

N <NUM>

<BODY>

Фамилия, имя, отчество работника	<FULL_NAME>
Дата рождения работника	<BIRTH_DATE>
Наименование структурного подразделения, в котором занят работник	<STRUCT>
Наименование должности (профессии) или вида работы	<POSITION>
Вредные и (или) опасные производственные факторы. Вид работы	<DANGER>

Специалист по кадрам

<SPECIALIST>

Рисунок 2.27 – Шаблон для формирования направления на медицинский осмотр



Ф.И.О. \_\_\_\_\_ <FULL\_NAME>  
Дата рождения \_\_\_\_\_ <BIRTH\_DATE>  
Место работы \_\_\_\_\_ <JOB>  
Адрес регистрации \_\_\_\_\_ <ADDRESS>

в соответствии со ст. 213 Трудового кодекса Российской Федерации направляется на обязательное психиатрическое освидетельствование с целью определения соответствия состояния здоровья поручаемой ему (ей) работе в должности

\_\_\_\_\_ <POSITION>

(перечень выполняемых работ и вредных и (или) опасных производственных факторов)

\_\_\_\_\_ <ORDER>

\_\_\_\_\_ <WORK\_SCROLL>

Решение врачебной психиатрической комиссии прошу выдать на руки обследуемому.

\_\_\_\_\_ *Директор филиала*

(должностное лицо)

\_\_\_\_\_ (подпись)

\_\_\_\_\_ <DIRECTOR>

(фамилия, инициалы)

" \_\_\_ " \_\_\_\_\_ 20 \_\_\_ г.

### Рисунок 2.28 – Шаблон для формирования направления на психиатрическое освидетельствование

После заполнения всех полей и нажатия кнопки «Создать документ...» будет открыто диалоговое окно с предложением выбрать путь сохранения документа со сформированным направлением. Затем программа открывает файл, расположенный в корневой папке и происходит поиск и замена значений в открытом документе по каждому ключу. Сформированный документ затем сохраняется по пути, указанном в диалоговом окне.

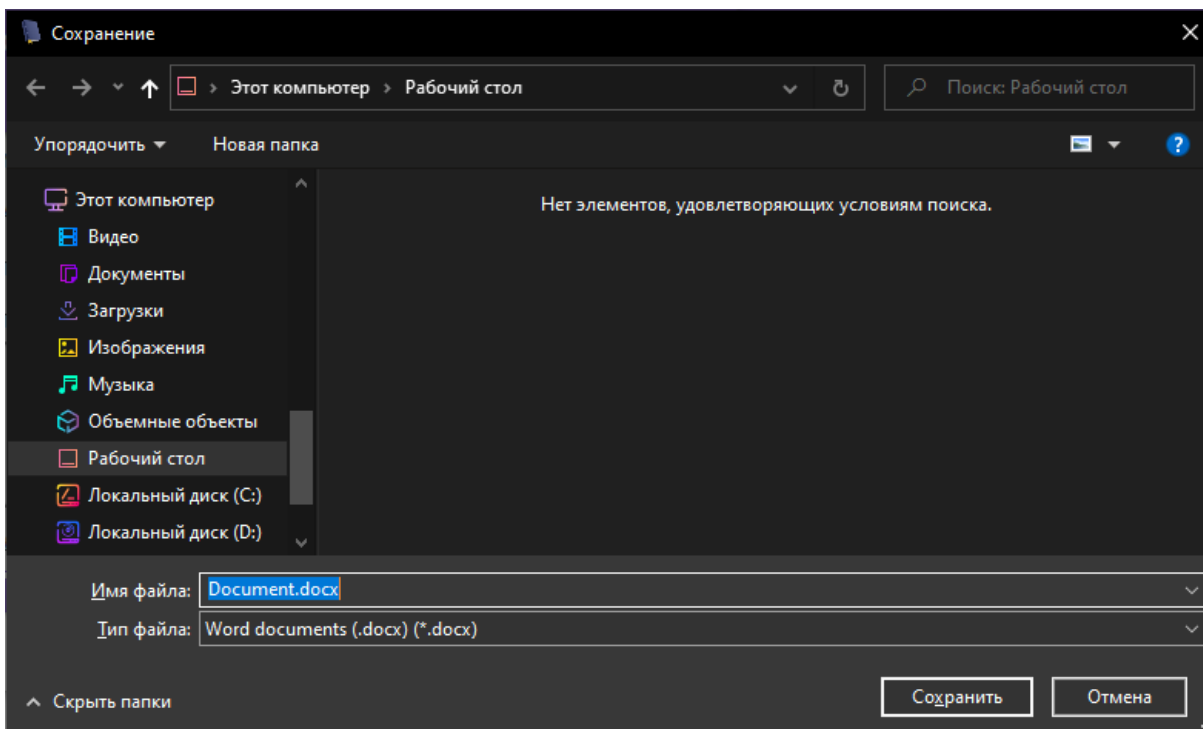


Рисунок 2.29 – Диалоговое окно выбора пути сохранения документа

## 2.10 Окно настроек приложения

Так как для работы уведомлений об окончании срока действия обучения или медосмотра требуется указать количество дней, за которое будет выведено уведомление и срок действия медосмотра, то было принято решения создать отдельное окно, содержащее в себе поля, в которых указываются количество дней, за которое будет отображаться уведомление об окончании срока действия обучения, срок действия медосмотра и колчество дней, за которое будет отображаться уведомление об окончании срока действия медосмотра.

Для изменения значений используется поля для ввода целочисленных значений. После изменения необходимых полей пользователь может сохранить выбранные настройки нажав кнопку «Сохранить».

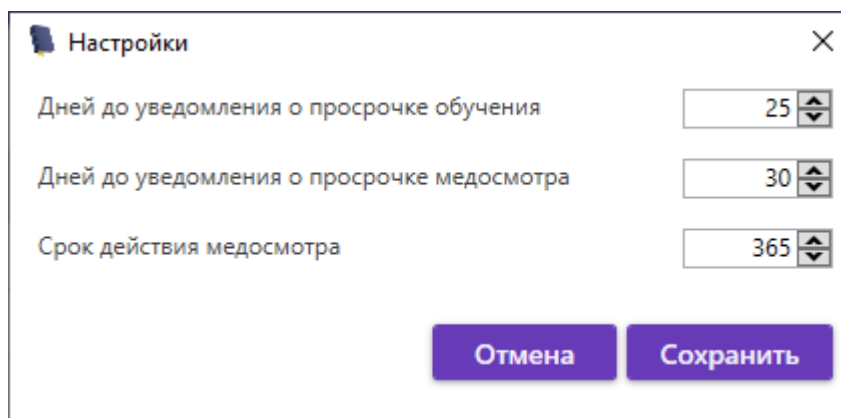


Рисунок 2.30 – Окно настроек приложения

## 2.11 Использование Material Design в интерфейсе

Одной из главных составляющих программного продукта является пользовательский опыт. Опыт пользователя, восприятие пользователя, опыт взаимодействия (англ. User eXperience, UX) — это восприятие и ответные действия пользователя, возникающие в результате использования и/или предстоящего использования продукции, системы или услуги (стандарт ISO 9241-210).

Опыт пользователя включает все эмоции, убеждения, предпочтения, ощущения, физические и психологические реакции пользователя, поведение и достижения, которые возникают до, во время и после использования системы. Опыт пользователя сочетает образ торговой марки, способ представления, функциональность, производительность системы, интерактивное поведение и вспомогательные возможности интерактивной системы, физическое и психологическое состояние пользователя, являющееся результатом предшествующего опыта, привычек, навыков и индивидуальности.

Для улучшения пользовательского опыта создано множество решений, одним из таких решений является стиль графического дизайна интерфейсов программного обеспечения и приложений, разработанный компанией Google. Впервые представлен на конференции Google I/O 25 июня 2014 года. Стиль расширяет идею «карточек», появившуюся в Google Now, более широким применением строгих макетов, анимаций и переходов, отступов и эффектов глубины (света и тени). По идее графических дизайнеров Google, у приложений не должно быть острых углов, карточки должны переключаться между собой плавно и практически незаметно.

В приложении Material Design реализован посредством C# библиотек MaterialDesignThemes и MaterialDesignColors, которые позволяют через язык разметки XAML встраивать элементы управления, использующие Material Design.

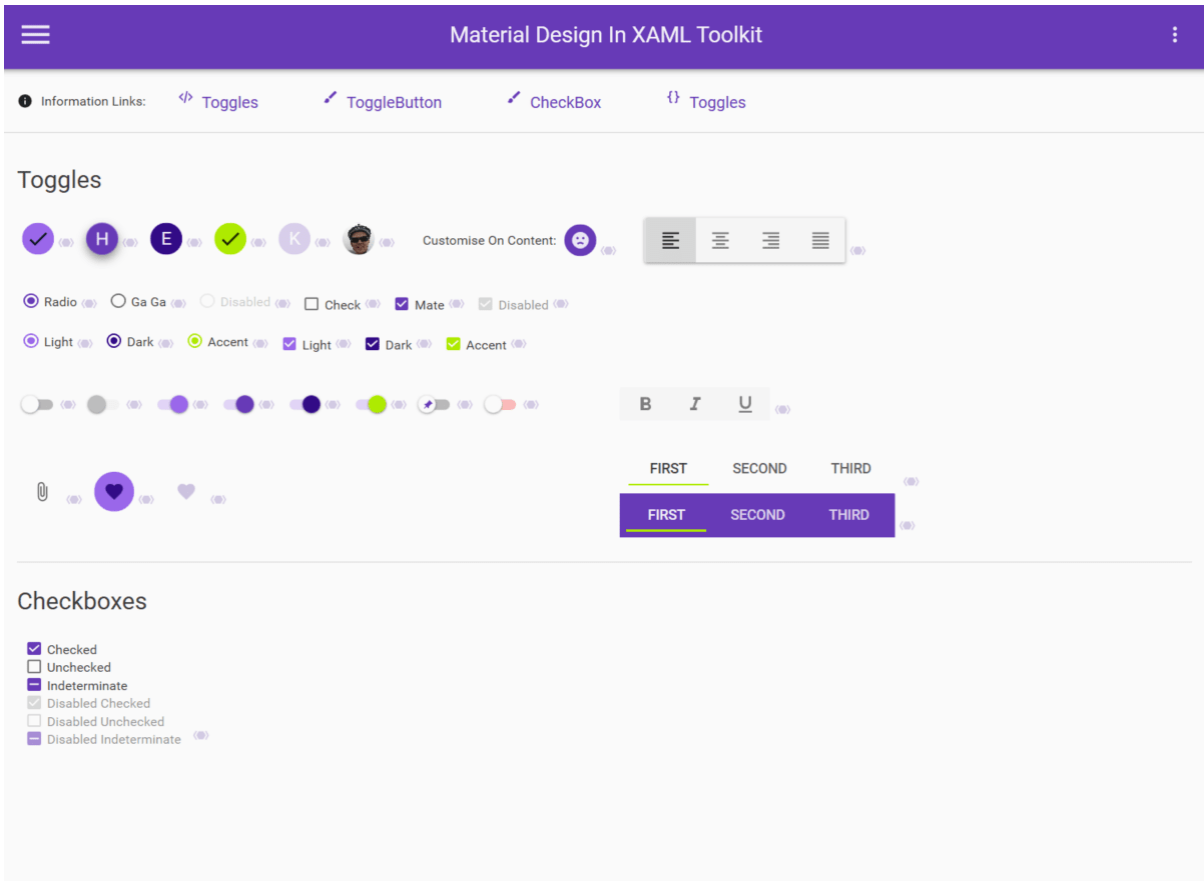


Рисунок 2.31 – Пример элементов управления Material Design

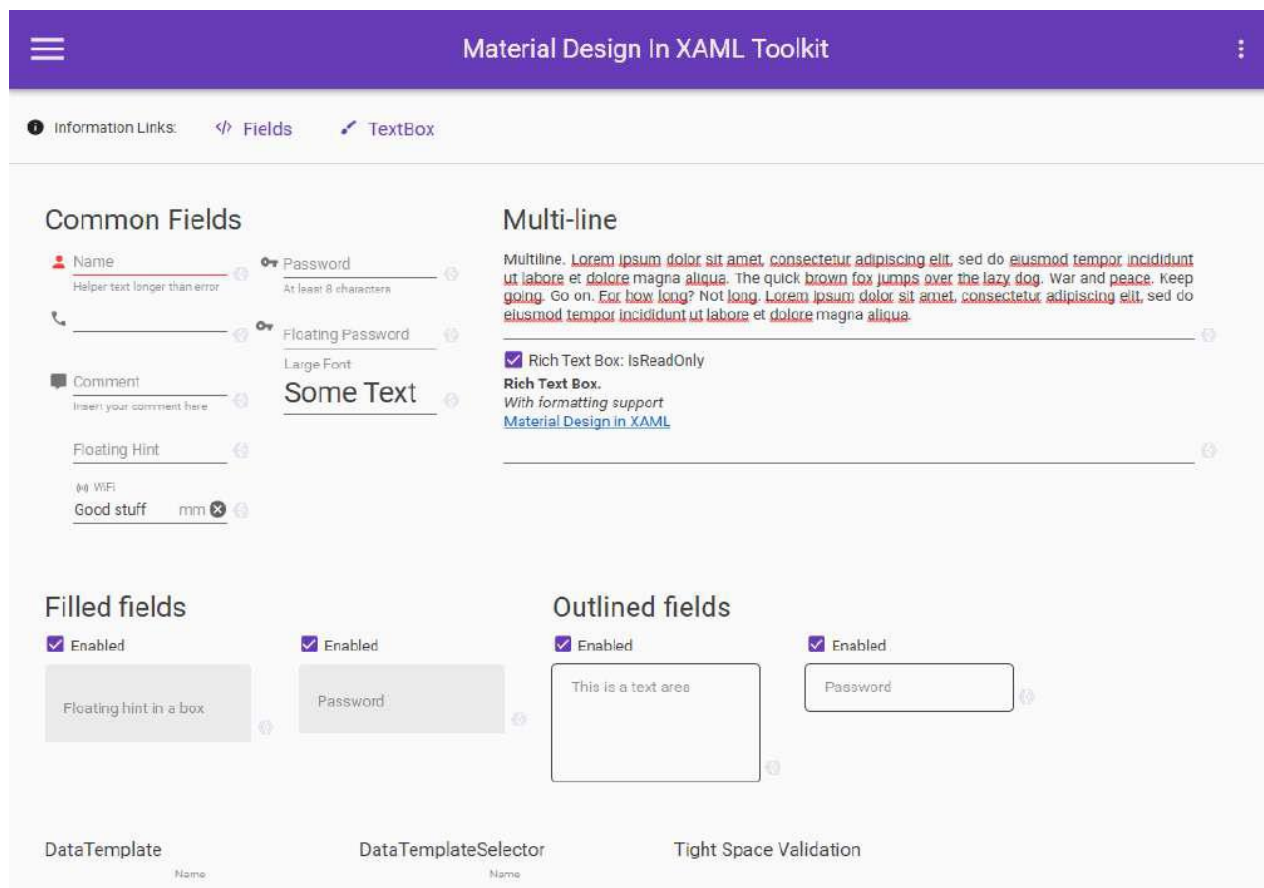


Рисунок 2.32 – Пример полей для ввода данных Material Design

Как видно из иллюстраций, элементы управления Material Design используют красочную расцветку, что позволит пользователю проще находить нужный элемент управления на экране, а также используется принцип What You See Is What You Get (То, что ты видишь, то и получишь), который подразумевает, что вид каждого элемента управления является самодостаточным для определения пользователем его функциональности.

Вывод по разделу два:

В соответствии с обозначенными требованиями, приведенными в техническом задании, разработан программный продукт, удовлетворяющий потребностям заказчика. При разработке были использованы современные инструменты, позволяющие снизить временные и материальные затраты на реализацию

необходимого функционала, кроме того, упрощается дальнейшая поддержка продукта за счет профессиональных подходов к проектированию структуры программного обеспечения.

### 3 ОРГАНИЗАЦИОННО-ЭКОНОМИЧЕСКИЙ РАЗДЕЛ

#### 3.1 Организационно-экономическая характеристика деятельности предприятия

Филиал ФГАОУ ВО «Южно-Уральский государственный университет» (НИУ) в г. Нижневартовске был создан 04 декабря 1995 года приказом ректора Южно-Уральского государственного университета (г. Челябинск) Вяткина Г.П. как консультационный пункт в г.Нижневартовске для оказания помощи студентам заочного отделения. 23 февраля 1998 года учреждение получило статус филиала.

Наименование учредителя: Министерство образования и науки Российской Федерации

Место нахождения образовательной организации: Филиал ФГАОУ ВО «Южно-Уральский государственный университет» (НИУ) в г. Нижневартовске находится по адресу: 628616, Тюменская область, Ханты-Мансийский АО-Югра, г. Нижневартовск, ул. Мира, д. 9

Таблица 3.1 – Структура и органы управления образовательной организацией

Наименование структурного подразделения	ФИО руководителя структурного подразделения	Место нахождения структурного подразделения	Адрес электронной почты структурного подразделения
Директор	Борщенок Вера Николаевна	628616, ХМАО-Югра, г.Нижневартовск, ул.Мира, д.9, каб. 108 (приёмная) тел: 27-25-30	<a href="mailto:comm-nv@susu.ru">comm-nv@susu.ru</a>
Центр досуга и творчества студентов	Товт Руслан, руководитель Центра	628616, ХМАО-Югра, г.Нижневартовск, ул.Мира, д.9, каб.110 тел: 27-24-54	<a href="mailto:comm-nv@susu.ru">comm-nv@susu.ru</a>
Бухгалтерия	Байтимилова Разиля Гатиятовна, главный бухгалтер	628616, ХМАО-Югра, г.Нижневартовск, ул.Мира, д.9, каб.103, 106 тел: 27-16-52	<a href="mailto:comm-nv@susu.ru">comm-nv@susu.ru</a>
Ведущий инженер	Качановский Александр Владимирович	628616, ХМАО-Югра, г.Нижневартовск, ул.Мира, д.9, каб.252 тел: 27-16-06	<a href="mailto:comm-nv@susu.ru">comm-nv@susu.ru</a>



Окончание таблицы 3.1.

Специалист по работе с персоналом	Дайнеко Лилия Сергеевна	628616, ХМАО-Югра, г.Нижневартовск, ул.Мира, д.9, каб.254 тел: 27-17-31	<a href="mailto:comm-nv@susu.ru">comm-nv@susu.ru</a>
Юридическая клиника «Центр правовой помощи малоимущим категориям граждан»	Семерьянова Нина Анатольевна, руководитель	628616, ХМАО-Югра, г.Нижневартовск, ул.Мира, д.9, каб.250 тел: 27-17-95	<a href="mailto:comm-nv@susu.ru">comm-nv@susu.ru</a>
Административно-хозяйственный отдел	Зиновьева Юлия Владимировна, начальник АХО	628616, ХМАО-Югра, г.Нижневартовск, ул.Мира, д.9, каб.101 тел:27-16-06	<a href="mailto:comm-nv@susu.ru">comm-nv@susu.ru</a>
Учебно-методический отдел	Креймер Елена Ивановна, начальник учебно-методического отдела	628616, ХМАО-Югра, г.Нижневартовск, ул.Мира, д.9, каб.129, 127 тел: 27-35-99	<a href="mailto:kreimeri@susu-ac.ru">kreimeri@susu-ac.ru</a>
Ответственный за научную деятельность	Семерьянова Нина Анатольевна	628616, ХМАО-Югра, г.Нижневартовск, ул.Мира, д.9, каб.250 тел: 27-17-95	<a href="mailto:comm-nv@susu.ru">comm-nv@susu.ru</a>
Библиотека	Шаповалова Гульнара Габдуллоевна, заведующий библиотекой	628616, ХМАО-Югра, г.Нижневартовск, ул.Мира, д.9, каб.220 тел: 27-19-32	<a href="mailto:comm-nv@susu.ru">comm-nv@susu.ru</a>
Кафедра «Экономика, менеджмент и право»	Зяблицкая Наталья Викторовна, заведующий кафедрой	628616, ХМАО-Югра, г.Нижневартовск, ул.Мира, д.9, каб.112 тел: 27-14-74	<a href="mailto:econom-nv@susu.ru">econom-nv@susu.ru</a>
Кафедра «Гуманитарные, естественно-научные и технические дисциплины»	Рябова Ирина Геннадьевна, заведующий кафедрой	628616, ХМАО-Югра, г.Нижневартовск, ул.Мира, д.9, каб.234 тел: 27-02-01	<a href="mailto:giend@susu.ru">giend@susu.ru</a>
Центр по подготовке, переподготовке и повышению квалификации кадров	Хаметова Гульнара Мьявлютовна (методист)	628616, ХМАО-Югра, г.Нижневартовск, ул.Мира, д.9, каб. 111 тел:27-36-09	<a href="mailto:comm-nv@susu.ru">comm-nv@susu.ru</a>
Научный испытательный центр	Латвина Ольга Владимировна, заведующий НИЦ	628616, ХМАО-Югра, г.Нижневартовск, ул.Мира, д.9, каб.5 (цокольный этаж) тел:27-19-32	<a href="mailto:comm-nv@susu.ru">comm-nv@susu.ru</a>

Должностная инструкция инженера программиста I категории научного испытательного центра филиала ФГАОУ ВО «ЮУрГУ (НИУ)» в г. Нижневартовске:

#### I Общие положения

1.1 Инженер-программист Научного испытательного центра ФГАОУ ВО «ЮУрГУ (НИУ)» в г. Нижневартовске (далее инженера программиста I категории) относится к I категории специалистов.

1.2 На должность инженера-программиста I категории назначается лицо, имеющее высшее профессиональное (техническое или инженерно-экономическое) образование и стаж работы в должности инженера-программиста не менее 3 лет.

1.3 Назначение на должность инженера-программиста I категории и освобождение от нее производится приказом директора филиала по представлению начальника научного испытательного центра.

1.4 Инженер-программист I категории должен знать:

1.4.1 Руководящие и нормативные материалы, регламентирующие методы разработки алгоритмов и программ и использования вычислительной техники при обработке информации.

1.4.2 Основные принципы структурного программирования.

1.4.3 Виды программного обеспечения.

1.4.4 Техничко-эксплуатационные характеристики, конструктивные особенности, назначение и режимы работы ЭВМ, правила ее технической эксплуатации.

1.4.5 Технологию автоматической обработки и кодирования информации.

1.4.6 Формализованные языки программирования.

1.4.7 Действующие стандарты, системы счислений, шифров и кодов.

1.4.8 Порядок оформления технической документации.

1.4.9 Передовой отечественный и зарубежный опыт программирования и использования вычислительной техники.

1.4.10 Основы экономики, организации производства, труда и управления.

1.4.11 Основы трудового законодательства.

1.4.12 Правила внутреннего трудового распорядка.

1.4.13 Правила и нормы охраны труда.

1.5 Инженер программист 1 категории подчиняется непосредственно начальнику Научного испытательного центра

1.6 На время отсутствия инженера программиста 1 категории (отпуск, болезнь, пр.) его обязанности исполняет лицо, назначенное в установленном порядке, которое несет ответственность за качественное и своевременное исполнение возложенных на него обязанностей

1.7 За ненадлежащее исполнение или неисполнение своих должностных обязанностей, предусмотренных настоящей должностной инструкцией, — в пределах, определенных действующим трудовым законодательством Российской Федерации.

1.8 За правонарушения, совершенные в процессе осуществления своей деятельности, — в пределах, определенных действующим административным, уголовным законодательством Российской Федерации.

1.9 За причинение материального ущерба — в пределах, определенных действующим трудовым законодательством Российской Федерации.

1.10 За сохранение конфиденциальности информации в связи с возможностью доступа к ней при выполнении должностных обязанностей.

Должностная инструкция инженера программиста 1 категории Научного испытательного центра разработана в соответствии с требованиями Приказа от 17.10.2005 г. № 129 «Об утверждении инструкции по делопроизводству в Южно-Уральском государственном университете».

Заработная плата инженера-программиста составляет 30000 руб.

### 3.2 Анализ финансовых показателей деятельности предприятия

В данном пункте рассматриваются финансовые показатели университета за 2020 год.

Таблица 3.2 – Общие результаты деятельности учреждения

№ п/п	Наименование показателя	На начало отчетного года (тыс. руб.)	На конец отчетного года (тыс. руб.)	Изменение, %
1	2	3	4	5
1	Нефинансовые активы, всего:	5480958.0000	4187362.0000	-23.6%
1.1	Недвижимое имущество, всего	3486004.0000	4265806.0000	22.37%
1.2	Остаточная стоимость	2171827.0000	2909253.0000	33.95%
1.3	Особо ценное движимое имущество, всего	2962265.0000	2994955.0000	1.1%
1.4.	Остаточная стоимость	353542.0000	279361.0000	-20.98%
2	Финансовые активы, всего:	5869349.0000	5155048.0000	-12.17%
2.1.	Денежные средства учреждения, всего	622981.0000	712456.0000	14.36%
2.2.	Денежные средства учреждения на счетах	622699.0000	712319.0000	14.39%
2.3.	Денежные средства учреждения, размещенные на депозиты в кредитной организации	177100.0000	181200.0000	2.32%
2.4.	Иные финансовые инструменты	192.0000	261.0000	35.94%
2.5.	Дебиторская задолженность по доходам	5213377.0000	4403009.0000	-15.54%
2.6.	Дебиторская задолженность по расходам	26773.0000	36026.0000	34.56%
2.7.	Дебиторская задолженность, нереальная к взысканию	0.0000	0.0000	0%
3	Обязательства, всего:	13652276.0000	12224192.0000	-10.46%
3.1.	Долговые обязательства	0.0000	0.0000	0%
3.2.	Кредиторская задолженность	263038.0000	176134.0000	-33.04%

### 3.3 Расчет затрат на разработку программного обеспечения

Смета разработки программного обеспечения содержит следующие пункты:

- затраты на материалы и электроэнергию;
- контрагентные расходы.

Выполним расчёт стоимости материалов и электроэнергии, затраченных при разработке программного обеспечения.

При разработке использовался стационарный компьютер. Потребляемая мощность компьютера составляет  $180 \text{ Вт} + \text{монитор } 40 \text{ Вт} = 220 \text{ Вт/час} = 0,22 \text{ кВт/час}$ . В среднем во время разработки ПО компьютер работал 3 часа  $0,22 \times 3 = 0,66 \text{ кВт}$  в день

Разработка велась ежедневно, в течении 42 дней, отсюда, на разработку потрачено  $42 \times 0,66 = 27,72 \text{ кВт}$  в сумме. Приняв стоимость  $1 \text{ кВт} = 6,63 \text{ руб.}$ , стоимость затраченной электроэнергии составила  $27,72 \times 6,63 = 183,79 \text{ руб.}$

Для записи итоговой версии ПО был использован CD-диск CD-RTDK 700 Мб стоимостью 85 руб.

Выполним расчёт контрагентных расходов.

В данную категорию расходов можно отнести затраты на доступ в интернет. Разработка велась на протяжении 3 месяцев. Приняв стоимость безлимитного доступа в интернет равную 1000 руб. в месяц, затраты на интернет составили  $1000 \times 3 = 3000 \text{ руб.}$

Для расчета амортизации оборудования необходимо установить срок полезного использования этого оборудования. Персональный компьютер относится ко второй амортизационной группе основных средств (имущество со сроком полезного использования свыше 2 лет до 3 лет включительно). Соответственно, срок полезного использования Персонального компьютера устанавливается в интервале от 2 лет и 1 месяца и до 3 лет, в данном случае, для компьютера, используемого при разработке программного продукта, срок полезного использования достигает 2 лет и 6 месяцев, то есть 30 месяцев. Ежемесячная амортизация для данного компьютера составляет

$1 / 30 \times 100\% = 3\%$  Так как суммарная стоимость всех компонентов компьютера 50000 руб., то итоговая амортизация за все  $42 / 30 = 1,4$  месяца работы будет составлять  $50000 \times (0,03 \times 1,4) = 2100$  руб.

Подведем итоги всех затрат на созданный продукт в одну таблицу.

Таблица 3.3 – Затраты на разработанный программный продукт

Затраты, связанные с разработкой, внедрением и эксплуатацией	Сумма, руб.
Электроэнергия	183,79
CD-диск	85
Интернет-услуги	3000
Амортизация оборудования	2100
Итого:	5368,79

Таким образом, сумма затрат на разработанное программное обеспечение составила 5368,79 руб.

### 3.4 Расчет доходов и финансовых результатов

Целью внедрения разработанного программного обеспечения было заменить аналогичные продукты, стоимость пользования которых намного превышает стоимость пользования данным продуктом или чья функциональность не являлась достаточной для эффективной работы отдела.

Экономический эффект от внедрения средств автоматизации заключается в улучшении экономических и хозяйственных показателей работы предприятия, в первую очередь за счет повышения оперативности управления и снижения трудозатрат на реализацию процесса управления, то есть сокращения расходов на управление. Для большинства предприятий экономический эффект выступает в виде экономии трудовых и финансовых ресурсов, получаемой от:

- снижения трудоемкости расчетов.

- снижение трудозатрат на поиск и подготовку документов.
- экономии на расходных материалах.
- сокращения служащих предприятия.

Проведём расчёт экономии средств за счёт увеличения производительности труда сотрудника.

До внедрения программного продукта отслеживание состояния обучения сотрудников велось в Microsoft Excel. Дополнительные отчеты и необходимые списки заполнялись вручную, основываясь на данных из таблиц, что могло привести к ошибкам при заполнении, так как в данном случае важную роль играет человеческий фактор.

Программный продукт снижает время на выполнение отдельных операций, необходимых для ведения учета охраны труда. Для наглядности ниже приведена таблица возможных операций, которые оптимизируются за счет внедрения программного продукта.

Таблица 3.4 – Виды процессов

Вид процесса	Время выполнения вручную, мин.	Время выполнения при использовании программного продукта, мин.
Создание, изменение, удаление информации о сотрудниках, видах их обучения и осмотров	30	5
Поиск необходимых сотрудников по различным параметрам фильтра и сортировке	10	1
Вывод на печать списка сотрудников с необходимыми параметрами	60	3
Формирование отчета на основании информации о сотруднике	30	5

Рассчитаем часовую тарифную ставку сотрудника, с окладом 40000 руб. на 2021 год.

Исходя из того, что в 2021 году 247 рабочих дней, а рабочий день на предприятии составляет в среднем 10 часов, то годовое количество рабочего времени для данного сотрудника составит 2470 часов. Отсюда рабочее время равно  $2470/12= 205$  часов в месяц. Часовая тарифная ставка сотрудника составит  $40000/205 = 195$  руб. Отсюда несложно вывести стоимость одной минуты сотрудника, которая составит  $195/60=3,25$  руб.

Предположив, что сотрудник составляет список сотрудников с указанными параметрами хотя бы раз в день, то за 247 рабочих дней сотрудник проведет данную операцию, соответственно, 247 раз. Рассчитаем время, затраченное сотрудником на проведение операций за год до и после внедрения программного продукта.

Приняв, что на одну операцию редактирования информации сотрудник тратит примерно 60 минут, то в год на это уйдёт  $60 \times 247 = 14820$  мин. до внедрения программного продукта. Стоимость этой работы составит  $3,25 \times 14820 = 48165$  руб.

После внедрения сотрудник затратит  $3 \times 247 = 741$  мин. Стоимость этой работы составит  $3,25 \times 741 = 2408,25$  руб.



Отсюда экономия за год, при вычете стоимости годового использования программного продукта, составит  $48165 - 2408,25 = 45756,75$  руб.

Исходя из экономии при использовании программного продукта за год и затрат его разработку, продукт окупит себя за  $5368,79 / 45756,75 = 0,12$  года = 42 дня использования, что выгодно, исходя из того, что могут предложить аналогичные решения.

Следовательно, задача проектирования и конструирования программного продукта для предприятия явилась экономически целесообразной, поскольку программный продукт значительно снижает время, затрачиваемое сотрудником на выполнение одной работы, и разница во времени используется на выполнение более важной работы.

Выводы по разделу три:

В ходе анализа экономической эффективности от внедрения программного продукта на предприятии было выявлено повышение эффективности работы специалистов, снижение временных затрат на выполнение различных процессов по учету охраны труда. Использование программного продукта является выгодным для заказчика, поскольку срок окупаемости наступит через 42 дня после внедрения разработки.

## ЗАКЛЮЧЕНИЕ

В ходе выполнения выпускной квалификационной работы был проведен анализ работы отдела предприятия, в ходе которого было выяснено, что существующие на данный момент решения для автоматизации учета охраны труда не удовлетворяют требованиям для эффективной работы отдела.

В связи с этим было принято решение разработать аналогичный продукт, но с использованием современных методов разработки и при использовании технического задания, что позволило реализовать весь необходимый функционал, позволяющий улучшить качество работы отдела предприятия.

На основе анализа было составлено техническое задание, согласно которому готовый программный продукт должен реализовывать функционал, улучшающий эффективность работы специалиста по охране труда. Выполнены задачи по анализу предметной области и выбору программных средств, при помощи которых производилась разработка программного обеспечения, а также конструированию программного обеспечения с учетом межгосударственных стандартов, определяющих эргономику взаимодействия человека и системы и общие требования к разработке и документированию программного обеспечения.

Было проведено внедрение программного продукта и проведен анализ экономической эффективности, в ходе которого наблюдалось снижение временных затрат отдела на проведение различных операций, что в свою очередь позволяет окупить затраты на разработку продукта быстрее и выгоднее, чем это позволяют аналогичные решения.

Таким образом, поставленные задачи решены в полном объеме. Цель выполнения выпускной квалификационной работы достигнута.

## БИБЛИОГРАФИЧЕСКИЙ СПИСОК

- 1 Справочная информация: "Перечень правил и инструкций по охране труда" [Электронный ресурс]. – режим доступа URL: [http://www.consultant.ru/document/cons\\_doc\\_LAW\\_182373/](http://www.consultant.ru/document/cons_doc_LAW_182373/), свободный. [дата обращения – 13.11.2020]
- 2 Проектирование баз данных: методические указания к лабораторным работам для направления «Программная инженерия» / сост. Е.А. Зверева. – Нижневартовск, 2016. – 8 с.
- 3 Основы правил проектирования базы данных [Электронный ресурс]. – режим доступа URL: <https://habr.com/ru/post/514364/>, свободный. [дата обращения – 17.01.2021]
- 4 Проектирование надёжных баз данных [Электронный ресурс]. – режим доступа URL: <https://habr.com/ru/post/350084/>, свободный. [дата обращения – 17.01.2021]
- 5 Руководство по WPF [Электронный ресурс]. – режим доступа URL: <https://metanit.com/sharp/wpf/>, свободный. [дата обращения – 19.01.2021]
- 6 Создание приложения WPF [Электронный ресурс]. – режим доступа URL: <https://docs.microsoft.com/ru-ru/dotnet/desktop/wpf/get-started/create-app-visual-studio?view=netdesktop-5.0>, свободный. [дата обращения – 26.01.2021]
- 7 Элементы управления WPF [Электронный ресурс]. – режим доступа URL: [https://professorweb.ru/my/WPF/UI\\_WPF/level6/UI\\_WPF\\_index.php](https://professorweb.ru/my/WPF/UI_WPF/level6/UI_WPF_index.php), свободный. [дата обращения – 26.01.2021]
- 8 WPF – Привязка данных [Электронный ресурс]. – режим доступа URL: <https://coderlessons.com/tutorials/microsoft-technologies/vyuchit-wpf/wpf-privyazka-dannykh>, свободный. [дата обращения – 05.03.2021]
- 9 Паттерны проектирования [Электронный ресурс]. – режим доступа URL: <https://refactoring.guru/ru/design-patterns>, свободный. [дата обращения – 13.03.2021]

10 Основы паттернов проектирования [Электронный ресурс]. – режим доступа URL: <https://metanit.com/sharp/patterns/1.1.php>, свободный. [дата обращения – 13.03.2021]

11 Начало работы с .NET [Электронный ресурс]. – режим доступа URL: <https://docs.microsoft.com/ru-ru/dotnet/core/get-started>, свободный. [дата обращения – 16.03.2021]

12 Документация по C# [Электронный ресурс]. – режим доступа URL: <https://docs.microsoft.com/ru-ru/dotnet/csharp/>, свободный. [дата обращения – 17.03.2021]

13 Методические рекомендации по прохождению учебной, производственной и преддипломной практики и формированию отчетной документации для направления «Программная инженерия» / сост. Л.Н.Буйлушкина. – Нижневартовск, 2017. – 31 с.

14 Сведения об образовательной организации, филиал ЮУрГУ в г. Нижневартовске [Электронный ресурс]. – режим доступа URL: <https://nv.susu.ru/sveden/budget/>, свободный. [дата обращения – 26.03.2021]

15 CLR via C#. Программирование на платформе Microsoft.NET Framework 4.5 на языке C# / Джеффри Рихтер – Изд-во Питер, 2017. – 896 с.

16 Head First. Паттерны проектирования. / Эрик Фримен, Элизабет Робсон – Изд-во Питер СПб, 2019. – 656 с.

17 Material Design In XAML Toolkit [Электронный ресурс]. – режим доступа URL: <https://github.com/MaterialDesignInXAML/MaterialDesignInXamlToolkit>, свободный. [дата обращения – 05.04.2021]

18 Работа с библиотекой Newtonsoft.Json на реальном примере [Электронный ресурс]. – режим доступа URL: <https://habr.com/ru/post/482042/>, свободный. [дата обращения – 12.04.2021]

19 Spire.DOC Program Guide [Электронный ресурс]. – режим доступа URL: <https://www.e-iceblue.com/Tutorials/Spire.Doc/Spire.Doc-Program-Guide.html>, свободный. [дата обращения – 15.04.2021]

20 Spire.XLS Program Guide [Электронный ресурс]. – режим доступа URL: <https://www.e-iceblue.com/Tutorials/Spire.XLS/Spire.XLS-Program-Guide/Spire.XLS-Program-Guide-Content.html>, свободный. [дата обращения – 16.04.2021]

## ПРИЛОЖЕНИЯ

### ПРИЛОЖЕНИЕ А. ТЕХНИЧЕСКОЕ ЗАДАНИЕ НА СОЗДАНИЕ АВТОМАТИЗИРОВАННОЙ СИСТЕМЫ «ПЛАНИРОВЩИК ОБУЧЕНИЯ»

#### 1. ОБЩИЕ СВЕДЕНИЯ

##### 1.1. Наименование системы

Планировщик обучения «Warden».

##### 1.2. основания для проведения работ

Заказ на разработку аналога автоматического планировщика обучения

##### 1.3. наименование организаций – заказчика и разработчика

###### 1.3.1. Заказчик

Филиал ФГАОУ ВО «ЮУрГУ (НИУ)» в г. Нижневартовске.

###### 1.3.2. Разработчик

Студент филиала ФГАОУ ВО «ЮУрГУ (НИИ)» в г. Нижневартовске, группа  
НвФл-422, Эминов Гаджи Русланович

##### 1.4. плановые сроки начала и окончания работы

Начало работ: 2 ноября 2020 года

Конец работ: 17 мая 2021 года

#### 2. НАЗНАЧЕНИЕ И ЦЕЛИ СОЗДАНИЯ СИСТЕМЫ

##### 2.1. Назначение

Автоматизация деятельности отдела предприятия, отвечающего за учет обучения сотрудников охране труда.

##### 2.2. цели создания системы

Замена аналогичных систем, чей функционал или стоимость не удовлетворяет требованиям предприятия.

### 3. ХАРАКТЕРИСТИКА ОБЪЕКТОВ АВТОМАТИЗАЦИИ

Объектами автоматизации являются данные о сотрудниках, а именно сроки прохождения обучения по программам охраны труда и сопутствующим им, а также данные о медицинских осмотрах.

### 4. ТРЕБОВАНИЯ К СИСТЕМЕ

#### 4.1. требования к системе в целом

##### 4.1.1. Требования к структуре и функционированию системы

Система должна состоять из нескольких модулей:

- Модуль, отвечающий за хранение информации
- Модуль, отвечающий за отображение информации
- Модуль, отвечающий за фильтрацию и сортировку информации
- Модуль, отвечающий за расчет дат
- Модуль, отвечающий за формирование документов

##### 4.1.1.1. Требования к приспособляемости системы к изменениям

Система должна поддерживать изменяемость информации о сотрудниках предприятия. В системе должна быть реализована возможность редактирования программ обучения и типов медицинских осмотров

4.1.1.2. Требования к сохранению работоспособности системы в различных вероятных условиях

Система должна учитывать особенности файловой системы целевой операционной системы и поддерживать перехват ошибок при возникновении ситуаций, при которых она не может получить доступ к файловому потоку

#### 4.1.2. Требования к эргономике и технической эстетике

Использование современных методов построения пользовательского интерфейса при разработке. Использование межгосударственного стандарта, определяющего эргономику взаимодействия человека и интерфейса. Удовлетворение условиям свойства, при котором содержание отображается в процессе редактирования и выглядит максимально близко похожим на конечную продукцию.

#### 4.2. Требования к функциям, выполняемым системой

##### 4.2.1. Подсистема сбора, обработки и загрузки данных

Используемая система сбора, обработки и загрузки данных должна соответствовать программному интерфейсу используемого инструмента разработки

##### 4.2.1.1. Перечень функций, задач подлежащей автоматизации

- Редактирование и хранение информации о сотрудниках предприятия
- Редактирование и хранение информации о медицинских осмотрах сотрудника
- Редактирование и хранение информации о программах обучения, по которым обучается сотрудник
- Визуальное упорядоченное отображение информации о сотрудниках предприятия
- Фильтрация и сортировка отображаемой информации
- Вывод текущего состояния отображаемой информации в виде документа
- Формирование документов на основе информации о выбранном сотруднике
- Расчет окончания срока действия документов и заблаговременное уведомление об этом.



## ПРИЛОЖЕНИЕ Б. ЛИСТИНГ МОДУЛЯ ПРОГРАММЫ

```
using Newtonsoft.Json;
using System;
using System.Collections.Generic;
using System.IO;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using Warden.Utils;

namespace Warden
{
    public class DataBase
    {
        private string path;

        private ApplicationData appData;

        public List<Employee> Employees => appData.employees;
        public List<TrainProgram> TrainPrograms =>
appData.trainPrograms;
        public AppSettings Settings => appData.settings;
        public PsychoDocumentTemplate PsychoTemplate =>
appData.psychoTemplate;
        public MedDocumentTemplate MedTemplate =>
appData.medTemplate;

        public DataBase(string saveFolder)
        {
            path = saveFolder + "data.wad";
            //Load data on create instance of database
            LoadData();

            MessageBroker.Subscribe(ShareData.DB_SAVE, _ =>
SaveData());
            MessageBroker.Subscribe(ShareData.DB_LOAD, _ =>
LoadData());
            MessageBroker.Subscribe(ShareData.EMPLOYEE_ADD, x
=> Employees.Add(x as Employee));

            MessageBroker.Subscribe(ShareData.EMPLOYEE_REMOVE, x =>
Employees.RemoveAll(a => a.ID == (x as Employee).ID));

```

```
MessageBroker.Subscribe(ShareData.TRAIN_PROGRAM_ADD, x =>
TrainPrograms.Add(x as TrainProgram));

    MessageBroker.Subscribe(ShareData.TRAIN_PROGRAM_REMOVE, x
=>
        {
            //For each employee's train programs list
remove all programs that id == remove id
            Employees.ForEach(e =>
                {
                    e.TrainPrograms.RemoveAll(tr => tr == (x
as TrainProgram).ID);
                });
            //remove program
            TrainPrograms.RemoveAll(a => a.ID == (x as
TrainProgram).ID);
        });
    }

    public void SaveData()
    {
        string s = JsonConvert.SerializeObject(appData);

        try
        {
            File.WriteAllText(path, s);

            MessageBroker.Send(ShareData.FILE_WRITE_SUCCESS, path);
        }
        catch
        {
            MessageBroker.Send(ShareData.FILE_WRITE_ERROR, path);
        }
    }

    public void LoadData()
    {
        if (!File.Exists(path))
        {
            File.Create(path);
        }
        else
```

```
{
    try
    {
        string s = File.ReadAllText(path);
        appData =
JsonConvert.DeserializeObject<ApplicationData>(s);

        MessageBroker.Send(ShareData.FILE_READ_SUCCESS, path);
    }
    catch
    {

        MessageBroker.Send(ShareData.FILE_READ_ERROR, path);
    }

    if(appData == null)
    {
        appData = new ApplicationData();
        appData.employees = new List<Employee>();
        appData.trainPrograms = new
List<TrainProgram>();
        appData.settings = new AppSettings();
        appData.medTemplate = new
MedDocumentTemplate();
        appData.psychoTemplate = new
PsychoDocumentTemplate();
    }
}

public class ApplicationData
{
    public List<Employee> employees;
    public List<TrainProgram> trainPrograms;
    public AppSettings settings;
    public MedDocumentTemplate medTemplate;
    public PsychoDocumentTemplate psychoTemplate;
}
}
```

ПРИЛОЖЕНИЕ В. КОМПАКТ-ДИСК