

ДОПУЩЕН К ЗАЩИТЕ

Заместитель директора
по учебной работе

_____ И.В.Малафей

"__" _____ 2019

ПРОГРАММНОЕ СРЕДСТВО ДЛЯ АВТОМАТИЗАЦИИ УЧЕТА И
МАТЕРИАЛЬНОЙ ОЦЕНКИ ЛЕСНЫХ РЕСУРСОВ

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

ДП Т.594004.401

Председатель цикловой комиссии	(Т.Г. Багласова)
Руководитель проекта	(Т.Г. Багласова)
Консультант по экономической части	(Д.А.Кошепарова)
Консультант по охране труда	(Н.К. Фоменко)
Учащийся	(И.Д. Алейчик)
Рецензент	()

Содержание

Введение	4
1 Объектно-ориентированный анализ и проектирование системы	5
1.1 Сущность задачи	5
1.2 Проектирование модели	6
2 Вычислительная система	9
2.1 Требования к аппаратным и операционным ресурсам	9
2.2 Инструменты разработки	9
3 Проектирование задачи	11
3.1 Требования к приложению	11
3.2 Концептуальный прототип	11
3.3 Организация данных	13
3.4 Функции и элементы управления	17
3.5 Проектирование справочной системы приложения	19
4 Описание программного средства	20
4.1 Общие сведения	20
4.2 Функциональное назначение	20
4.3 Входные и выходные данные	21
5 Методика испытаний	22
5.1 Технические требования	22
5.2 Функциональное тестирование	22
6 Применение	28
6.1 Назначение программы	28
6.2 Условия применения	28
6.3 Справочная система	28
7 Охрана труда и охрана окружающей среды	31
7.1 Правовые, нормативные, социально-экономические и организационные вопросы охраны труда.	31
7.2 Организация режима труда и отдыха при работе с ПЭВМ	33
7.3 Пожарная безопасность	35
7.4 Охрана окружающей среды	36
8 Экономический раздел	38
8.1 Техничко-экономическое обоснование разработки программного средства	38
8.2 Составление плана по разработке программного средства	38
8.3 Определение цены программного средства	38
8.4 Экономическая эффективность разработки	42
Заключение	44
Список используемых источников	45
Приложение А Текст программы	47
Приложение Б Формы выходных документов	53

					ДП Т.594004.401 ПЗ				
Изм.	Лист	№ докум.	Подпись	Дата	Программное средство для автоматизации учета и материальной оценки лесных ресурсов	Лит.		Лист	Листов
Разраб.		Алейчик И.Д.				у		3	53
Провер.		Багласова Т.Г.				КБП			
Т.контр.		Багласова Т.Г.							
Н. контр.		Ржеутская Н.В.							
Утверд.		Багласова Т.Г.							

Введение

Методика учета и материальной оценки лесных ресурсов предоставляет явные преимущества организации. Для облегчения работы сотрудникам, ведущим контроль за лесными ресурсами, целесообразно пользоваться компьютерными программами, а также утвержденными методическими рекомендациями или технологическими картами на предприятии. При этом, очевидно, компьютерная программа обладает неоспоримыми преимуществами перед бумажным документооборотом.

С помощью учета лесных ресурсов решаются задачи по формированию информационной базы лесного фонда объекта хозяйствования.

Вся лесохозяйственная деятельность базируется на учете лесных ресурсов, их количественной и качественной оценке. Лесная таксация разрабатывает методы и нормативную базу оценки и учета древостоев, отдельных деревьев, а также дает оценку динамике роста древостоев и его прироста по ряду лесотаксационных показателей, что необходимо для решения лесохозяйственных задач.

Целью дипломного проектирования для автоматизации учета лесных ресурсов является разработка программного средства «AppTaxationCard.appx», которое будет осуществлять ведение базы данных, содержащей информацию о выделах и их типах, категориях, генерирующей карточку таксации по объекту хозяйствования.

Пояснительная записка состоит из семи разделов, содержащих необходимую информацию по организации эксплуатации программного средства.

В первом разделе «Объектно-ориентированный анализ и проектирование системы» раскрывается организационная сущность задачи, описывается предметная область и круг задач, которые должны быть автоматизированы. Описывается задача, перечисляются основные функции программы.

Во втором разделе «Вычислительная система» перечисляются требования к аппаратному и программному обеспечению компьютеров, приводится характеристика операционной системы, обоснование выбранных инструментов разработки программы.

В третьем разделе «Проектирование задачи» приводится объектно-ориентированный анализ задачи, строится концептуальный прототип диалоговых окон и элементов управления, приводится организация данных в контексте выбранной системы управления базами данных.

В четвертом разделе «Описание программного средства» представлены общие сведения о программе и функциональном назначении, описываются входные и выходные данные.

В пятом разделе «Методика испытаний» описываются требования к аппаратному и программному обеспечению компьютера для проведения испытаний, требования к характеристикам программы применительно к условиям эксплуатации. Представляются результаты функционального тестирования.

Шестой раздел «Применение» предназначен для описания сведений о назначении программного средства и области его применения. В этом разделе приводится структура справочной системы, а также методика ее использования.

Седьмой раздел «Охрана труда и охрана окружающей среды» предназначен для описания сведений о назначении охраны труда и окружающей среды на предприятии и ее необходимости. В этом разделе описываются основные правила и требования, которые необходимо соблюдать во избежание преждевременной утомляемости при работе с ПЭВМ.

Восьмой раздел «Экономический раздел» в данном разделе представлено экономическое обоснование дипломного проекта.

В заключении анализируется созданное программное средство, определяется степень соответствия поставленной задачи и выполненной работы.

Приложение А содержит текст программы.

Приложение Б содержит примеры выходных документов.

В графической части представлены диаграммы вариантов использования, классов, деятельности, последовательности и компонентов.

1 Объектно-ориентированный анализ и проектирование системы

1.1 Сущность задачи


Предметной областью решаемой задачи является организация РУП «Белгослес». Предприятие осуществляет инвентаризацию лесного фонда, проектирование лесохозяйственных мероприятий и изготовление планово-картографических материалов.

Клиентами предприятия могут быть как обычные клиенты, так и корпоративные.

В настоящее время в подобных организациях осуществляют оценку и учет лесных ресурсов, с чем решаются задачи по формированию информационной базы лесного фонда объекта хозяйствования. Она оценивает состояние и динамику лесных ресурсов, дает разностороннюю оценку лесным и парковым ландшафтам.

Контроль за лесными ресурсами осуществляется карточкой таксации, пример которой приведен на рисунке 1.1.

Отчет о выделе



Макет 1

Дата и время создания документа:	24.05.2019 18:42:00
Номер квартала:	2
Лесничество:	Держинское
Номер выдела:	2
Площадь:	21
Вид земель:	Квартальные просеки
ОРЛ:	Участки леса в поймах рек
Крутизна:	23
Экспозиция:	В

Рисунок 1.1

Исходя из исследования предметной области основными задачами, подлежащими автоматизации, будут являться:

- ведение базы данных, содержащей информацию об кварталах, выделах и их типах, категориях.
- формирование списка выделов;
- визуальное представление кварталов на реальной карте;
- поиск, сортировка и фильтрация данных;
- создание отчётов по интересующей пользователя информации, карточки таксации;
- экспорт в файлы и вывод на печать.

Аналогами разрабатываемой программы является программа «Мобильный таксатор». Однако нужна программа для персонального компьютера связи с необходимостью печати карточки.

Программа будет обладать простым, удобным интерфейсом и легкой в использовании.

1.2 Проектирование модели

Цель моделирования данных состоит в обеспечении разработчика информационной системы концептуальной схемой базы данных в форме одной модели или нескольких локальных моделей, которые относительно легко могут быть отображены в любую систему баз данных.

Наиболее распространенным средством моделирования данных являются диаграммы «Сущность-связь» (ERD). С их помощью определяются важные для предметной области объекты (сущности), их свойства (атрибуты) и отношения друг с другом (связи). ERD непосредственно используются для проектирования реляционных баз данных. Нотация ERD была впервые введена П. Ченном и получила дальнейшее развитие в работах Баркера. Диаграмма «Сущность-связь» представлена на рисунке 1.2.

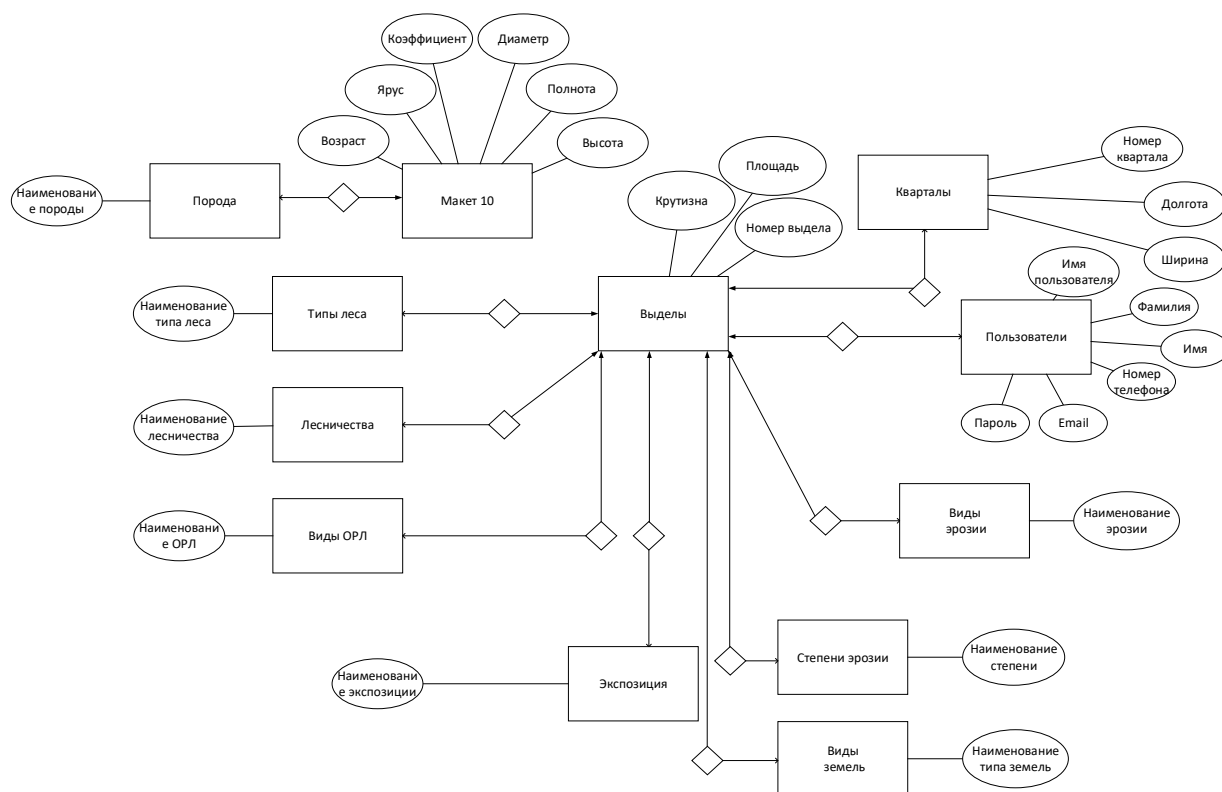


Рисунок 1.2

Исходя из исследования предметной области, можно выделить следующие сущности разработки: «Кварталы», «Пользователи», «Выделы», «Макет 10», «Виды эрозии», «Степени эрозии», «Экспозиция», «Виды ОРЛ», «Виды земель», «Лесничества», «Типы леса», «Порода».

Для сущности «Кварталы» атрибутами будут являться:

- «Номер квартала»;
- «Долгота»;
- «Ширина».

Для сущности «Виды эрозии» атрибутом будет являться «Наименование эрозии».

Для сущности «Степени эрозии» атрибутом будет являться «Наименование степени».

Для сущности «Экспозиция» атрибутом будет являться «Наименование экспозиции».

Для сущности «Виды ОРЛ» атрибутом будет являться «Наименование вида ОЗУ».

Для сущности «Виды земель» атрибутом будет являться «Наименование типа земель».

Для сущности «Лесничества» атрибутом будет являться «Наименование лесничества».

Для сущности «Типы леса» атрибутом будет являться «Наименование типа леса».

Для сущности «Порода» атрибутом будет являться «Наименование породы».

Для сущности «Пользователи» можно выделить следующие атрибуты:

- «Имя пользователя»;
- «Имя»;
- «Фамилия»;
- «Email»;
- «Номер телефона»;
- «Пароль».

Для сущности «Выделы» атрибутами будут являться:

- «Номер выдела»;
- «Площадь»;
- «Крутизна».

Для сущности «Макет 10» атрибутами будут являться:

- «Ярус»;
- «Коэффициент»;
- «Возраст»;
- «Высота»;
- «Диаметр»;
- «Полнота».

Суть диаграммы вариантов использования состоит в том, что проектируемая система представляется в виде множества сущностей или актёров, взаимодействующих с системой с помощью, так называемых, вариантов использования.

Варианты использования описывают не только взаимодействия между пользователями и сущностью, но также реакции сущности на получение отдельных сообщений от пользователей и восприятие этих сообщений за пределами сущности. Варианты использования могут включать в себя описание особенностей способов реализации сервиса и различных исключительных ситуаций, таких как корректная обработка ошибок системы. Множество вариантов использования в целом должно определять все возможные стороны ожидаемого поведения системы.

Актёр представляет собой внешнюю по отношению к моделируемой системе сущность, которая взаимодействует с системой и использует её функциональные возможности для достижения определённых целей или решения частных задач. При этом актёры служат для обозначения согласованного множества ролей, которые могут играть пользователи в процессе взаимодействия с проектируемой системой. Каждый актёр может рассматриваться как некоторая отдельная роль относительно конкретного варианта использования.

Диаграмма вариантов использования представлена в графической части на листе 1.

Диаграмма классов служит для представления статической структуры модели системы в терминологии классов объектно-ориентированного программирования. Диаграмма классов может отражать, в частности, различные взаимосвязи между отдельными сущностями

предметной области, такими как объекты и подсистемы, а также описывает их внутреннюю структуру и типы отношений. На данной диаграмме не указывается информация о временных аспектах функционирования системы. С этой точки зрения диаграмма классов является дальнейшим развитием концептуальной модели проектируемой системы.

Диаграмма классов для проектируемой системы представлена в графической части на листе 2.

При моделировании поведения проектируемой или анализируемой системы возникает необходимость детализировать особенности алгоритмической и логической реализации выполняемых системой операций. Для моделирования процесса выполнения операций в языке UML используются так называемые диаграммы деятельности. Каждое состояние на диаграмме деятельности соответствует выполнению некоторой элементарной операции, переход в следующее состояние срабатывает только при завершении этой операции. Графически диаграмма деятельности представляется в форме графа, вершинами которого являются состояния действия, а дугами - переходы от одного состояния действия к другому.

Основная цель использования диаграмм деятельности - визуализация особенностей реализации операций классов, когда необходимо представить алгоритмы их выполнения.

Диаграмма деятельности представлена в графической части на листе 3.

Для моделирования взаимодействия объектов в UML используются соответствующие диаграммы взаимодействия. Если рассматривать взаимодействия объектов во времени, тогда для представления временных особенностей передачи и приема сообщений между объектами используется диаграмма последовательности.

Временной аспект поведения имеет существенное значение при моделировании синхронных процессов, описывающих взаимодействия объектов. Именно для этой цели и используются диаграммы последовательности, в которых ключевым моментом является динамика взаимодействия объектов во времени. При этом диаграмма последовательности имеет как бы два измерения: одно - слева направо в виде вертикальных линий, каждая из которых изображает линию жизни отдельного объекта, участвующего во взаимодействии; второе - вертикальная временная ось, направленная сверху вниз, на которой начальному моменту времени соответствует самая верхняя часть диаграммы.

Диаграмма последовательности для проектируемой системы представлена в графической части на листе 4.

Рассмотренные ранее диаграммы отражали концептуальные аспекты построения модели системы и относились к логическому уровню представления. Особенность логического представления заключается в том, что оно оперирует понятиями, которые не имеют самостоятельного материального воплощения. Другими словами, различные элементы логического представления, такие как классы, ассоциации, состояния, сообщения, не существуют материально или физически. Они лишь отражают наше понимание структуры физической системы или аспекты ее поведения.

Основное назначение логического представления состоит в анализе структурных и функциональных отношений между элементами модели системы. Однако для создания конкретной физической системы необходимо некоторым образом реализовать все элементы логического представления в конкретные материальные сущности. Для описания таких реальных сущностей предназначен другой аспект модельного представления, а именно физическое представление модели.

Диаграмма компонентов описывает объекты реального мира - компоненты программного обеспечения. Эта диаграмма позволяет определить архитектуру разрабатываемой системы, установив зависимости между программными компонентами.

Вид диаграммы компонентов для данной проектируемой системы представлен в графической части на листе 5. Она содержит следующие компоненты:

- программные компоненты: «.xaml», «.cs», «.appxmanifest»;
- файл проекта Visual Studio 2019: «AppTaxationCard.sln»;
- файл базы данных «LocalDataBase.db»;
- файл справочной системы «index.html».

2 Вычислительная система

2.1 Требования к аппаратным и операционным ресурсам

Конфигурация компьютера, на котором будет разрабатываться программное приложение:

- процессор Intel Core i5 2500 МГц;
- оперативная память DDR3 8 Гбайт;
- жёсткий диск SSD 120 Гбайт;
- видеокарта nVidia GeForce GT 740m 2048 Мбайт;
- материнская плата Lenovo.

Для удобной работы с программой необходимо наличие клавиатуры и мыши.

2.2 Инструменты разработки

Инструментами разработки будут являться:

- операционная система Windows 10 Pro Версия 1809 (Сборка ОС 17763.503);
- среда разработки Microsoft Visual Studio 2019;
- система управления базами данных SQLite;
- среда для управления SQLite Studio, Valentina Studio 9;
- язык программирования C#;
- графический редактор, редактор блок схем Microsoft Visio;
- офисный пакет Microsoft Office.

Операционная система Windows 10 появилась относительно недавно – она стала доступной с 29 июля 2015 года. Компания Microsoft при разработке продолжала свой путь, направленный на унификацию. Допускается установка на компьютеры, ноутбуки, планшеты, а также смартфоны и консоли [18].

Microsoft Visual Studio — линейка продуктов компании Microsoft, включающих интегрированную среду разработки программного обеспечения и ряд других инструментальных средств. Данные продукты позволяют разрабатывать как консольные приложения, так 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 [13].

SQLite – это встраиваемая реляционная база данных, поставляемая с исходными кодами. Впервые выпущена в 2000 году, предназначена для предоставления привычных возможностей реляционных баз данных без присущих им накладных расходов. За время эксплуатации успела заслужить репутацию как переносимая, легкая в использовании, компактная, производительная и надежная база данных [14].

SQLite Studio - Менеджер SQLite баз данных, который позволяет просматривать и редактировать SQLite 3, SQLite 2 и SQLCipher базы данных [15].

Valentina Studio 9 - менеджер баз данных, который распространяется совершенно бесплатно. Он обладает вполне удобной графической оболочкой и умеет работать с всеми популярными типами СУБД. Как и прочие подобные программные решения, данное ПО позволяет как редактировать, так и создавать базы данных с нуля. Менеджер автоматически определяет все установленные и связи между элементами [16].

C# — это универсальный объектно-ориентированный язык программирования, обеспечивающий безопасность преобразования типов. Целью его создания было повышение производительности труда программистов. Поэтому в языке сбалансированы - простота, выразительность и эффективность [10].

Microsoft Visio — 2D программное обеспечение редактор диаграмм, блок-схем для Windows [11].

Microsoft Office — это офисный пакет приложений, созданных корпорацией Microsoft для операционных систем Microsoft Windows, Windows Phone, Android, OS X, iOS. В состав этого пакета входит программное обеспечение для работы с различными типами документов: текстами, электронными таблицами, базами данных. Microsoft Office является сервером OLE-объектов и его функции могут использоваться другими приложениями, а также самими приложениями Microsoft Office [12].

3 Проектирование задачи

3.1 Требования к приложению

Программное средство предполагает дальнейшее использование пользователями, не обладающими специальной профессиональной компетенцией по работе с современными информационными технологиями, что определяет концептуальный прототип программного приложения, методы работы с приложением, способ его инсталляции на персональном компьютере. А значит приложение должно иметь удобный интерфейс для простоты использования его пользователем.

Программное приложение должно:

- содержать привычные и понятные пользователю пункты меню, соответствующие функциям обработки;
- использовать карту с реальными координатами;
- осуществлять печать карточек согласно установленного образца;
- ориентироваться на пользователя, который общается с программой на внешнем уровне взаимодействия.

Пользователю предлагается выбор альтернативных функций обработки из фиксированного перечня, меню может содержать вложенные подменю.

Диалоговые окна должны содержать элементы управления:

- поля ввода и вывода информации;
- списки возможных альтернатив для выбора;
- кнопки.

3.2 Концептуальный прототип

Концептуальный прототип представляет собой описание внешнего пользовательского интерфейса – системы меню, диалоговых окон и элементов управления.

Предполагается создание формы меню для демонстрации функций модуля. Данная форма будет состоять из фреймов для просмотра страниц, области просмотра работы функций и управляющих кнопок.

На форме будут располагаться различные элементы управления, предназначенные для определенных задач. За компонентом Button будут закреплены определенные действия для выполнения функций.

При проектировании концептуального прототипа предполагается, что при загрузке программы первой будет загружаться «Форма входа».

Макет формы «Форма входа» представлен на рисунке 3.1

Diagram of a login form. It features a registration link in the top right corner. The main area contains two input fields for 'Логин' (Login) and 'Пароль' (Password), followed by a 'Войти' (Login) button and a link for 'забыли пароль?' (forgot password?).

Рисунок 3.1

После входа в систему будет загружаться форма «Рабочая панель». Макет формы «Рабочая панель» представлен на рисунке 3.2.

Diagram of the 'Рабочая панель' (Dashboard) form. It includes a sidebar with 'Рабочая панель' and 'Управление базой'. The main content area shows user information, navigation buttons, and a section for 'Кварталы' (Quarters) with four 'Номер выдела' (Plot Number) boxes. A search bar and filters are also present.

Рисунок 3.2

Для работы с базой данных необходимо будет отыскать страницу «Управление базой». Макет формы «Управление базой» представлен на рисунке 3.3.

Diagram of the 'Управление базой' (Database Management) form. It features a sidebar with 'Управление базой' and a search bar. The main content area is divided into a 'Кварталы' (Quarters) section with a list of 'Выделы' (Plots) and a 'Список таблиц' (Table List) section. A large 'Таблица' (Table) area is also shown.

Рисунок 3.3

Для просмотра данных о выделе необходимо будет перейти на страницу «Выделы». Макет формы «Выделы» представлен на рисунке 3.4.

Рисунок 3.4

3.3 Организация данных

Реляционная модель основана на математическом понятии отношения, представлением которого является таблица. В реляционной модели отношения используются для хранения информации об объектах, представленных в базе данных. Отношение имеет вид двухмерной таблицы, в которой строки соответствуют записям, а столбцы – атрибутам. Каждая запись должна однозначно характеризоваться в таблице. Для этого используют первичные и вторичные ключи. Достоинством реляционной модели является простота и удобство физической реализации.

Реляционная модель базы данных подразумевает нормализацию всех таблиц данных. Нормализация – это формальный метод анализа отношений на основе их первичного ключа и функциональных зависимостей, существующих между их атрибутами.

Исходя из анализа предметной области, выделяем главные сущности:

- менеджер учетных записей;
- журнал активности;
- кварталы;
- выделения.

База данных соответствует реляционной модели данных, где каждый выделенный в ходе проектирования сущности соответствует таблице.

Структура базы данных разрабатываемого программного средства на данный момент включает двенадцать таблиц.

Структура таблиц и их краткое описание приводится в таблицах 3.1-3.12.

Таблица «Accounts» хранит информацию о пользователях.

Таблица 3.1 – Структура таблицы «Accounts»

Имя поля	Тип данных	Размер, байт	Описание
Id	int	4	Идентификатор пользователя
Username	nvarchar	16	Уникальное имя пользователя

Продолжение таблицы 3.1

Имя поля	Тип данных	Размер, байт	Описание
Password	nvarchar	16	Пароль пользователя
CreateTime	datetime	8	Время создания учетной записи пользователя
Email	nvarchar	16	Электронная почта
Name	nvarchar	16	Настоящее имя пользователя
Surname	nvarchar	16	Настоящая фамилия пользователя
Phone	nvarchar	16	Номер телефона
ProfileImage	nvarchar	16	Путь к изображению профиля

Таблица «Videls» хранит информацию о выделах.

Таблица 3.2 – Структура таблицы «Videls»

Имя поля	Тип данных	Размер, байт	Описание
Id	int	4	Идентификатор выдела
NumVidel	int	4	Идентификатор номер квартала
Area	int	4	Площадь земли
CreateDateVidel	nvarchar	16	Дата создания документа
Krut	int	4	Крутизна
AccountId	int	4	Идентификатор пользователя
DegreeErrosionId	int	4	Идентификатор степени эрозии
ExpositionSlopeId	int	4	Идентификатор экспозиции склона
ForestryId	int	4	Идентификатор лесничества
KvartalId	int	4	Идентификатор квартала
TypeEarthId	int	4	Идентификатор вида земли
TypeErrosionId	int	4	Идентификатор вида эрозии
TypeOrlId	int	4	Идентификатор ОРЛ

Таблица «SessionUser» хранит информацию о текущих пользователях, вошедших в систему.

Таблица 3.3 – Структура таблицы «SessionUser»

Имя поля	Тип данных	Размер, байт	Описание
Id	int	4	Идентификатор сессии

Продолжение таблицы 3.3

Имя поля	Тип данных	Размер, байт	Описание
SessionTime	datetime	8	Дата и время входа в программу
SId	int	4	Идентификатор пользователя
Username	nvarchar	16	Имя пользователя

Таблица «Kvartals» хранит информацию о кварталах.

Таблица 3.4 – Структура таблицы «Kvartals»

Имя поля	Тип данных	Размер, байт	Описание
Id	int	4	Идентификатор квартала
CreateAcc	int	4	Идентификатор пользователя
CreateDateKvartal	datetime	8	Дата и время создания
NumKvartal	int	4	Номер квартала
Latitude	double	8	Широта
Longitude	double	8	Долгота

Таблица «Forestries» хранит информацию о лесничествах.

Таблица 3.5 – Структура таблицы «Forestries»

Имя поля	Тип данных	Размер, байт	Описание
Id	int	4	Идентификатор лесничества
NameForestry	nvarchar	16	Наименование лесничества

Таблица «TypeOrl» хранит информацию о типах ОРЛ.

Таблица 3.6 – Структура таблицы «TypeOrl»

Имя поля	Тип данных	Размер, байт	Описание
Id	int	4	Идентификатор ОРЛ (ОЗУ)
NameOrl	nvarchar	16	Наименование ОРЛ (ОЗУ)

Таблица «TypeErrosion» хранит информацию о типах эрозии.

Таблица 3.7 – Структура таблицы «TypeErrosion»

Имя поля	Тип данных	Размер, байт	Описание
Id	int	4	Идентификатор эрозии
NameErrosion	nvarchar	16	Наименование эрозии

Таблица «TypeEarth» хранит информацию о типах земли.

Таблица 3.8 – Структура таблицы «TypeEarth»

Имя поля	Тип данных	Размер, байт	Описание
Id	int	4	Идентификатор вида земли
NameEarth	nvarchar	16	Наименование вида земли

Таблица «ExpositionSlope» хранит информацию о типах экспозиции.

Таблица 3.9 – Структура таблицы «ExpositionSlope»

Имя поля	Тип данных	Размер, байт	Описание
Id	int	4	Идентификатор экспозиции
NameExpositionSlope	nvarchar	16	Наименование экспозиции

Таблица «DegreeErrosion» хранит информацию о степенях эрозии.

Таблица 3.10 – Структура таблицы «DegreeErrosion»

Имя поля	Тип данных	Размер, байт	Описание
Id	int	4	Идентификатор степени эрозии
NameDegreeErrosion	nvarchar	16	Наименование степени эрозии

Таблица «MaketThens» хранит информацию о ярусах выдела.

Таблица 3.11 – Структура таблицы «MaketThens»

Имя поля	Тип данных	Размер, байт	Описание
Id	int	4	Идентификатор макета
Koeficent	inr	16	Коэффициент яруса
Polnota	double	8	Полнота
Poroda	nvarchar	16	Порода
Visota	int	4	Высота
Vozrast	int	4	Возраст
Yarus	int	4	Ярус
Diametr	int	4	Диаметр
CurrentKvartalId	int	4	Идентификатор квартала
CurrentAccountId	int	4	Идентификатор пользователя
CurrentVidelId	int	4	Идентификатор выдела

Таблица «Porodas» хранит информацию о породах.

Таблица 3.12 – Структура таблицы «Porodas»

Имя поля	Тип данных	Размер, байт	Описание
Id	int	4	Идентификатор породы
NamePoroda	nvarchar	16	Наименование породы

Выходной информацией данного приложения будут данные в формате документа docx, xlsx, json.

Физическая структура базы данных представлена схемой данных на рисунке 3.5.

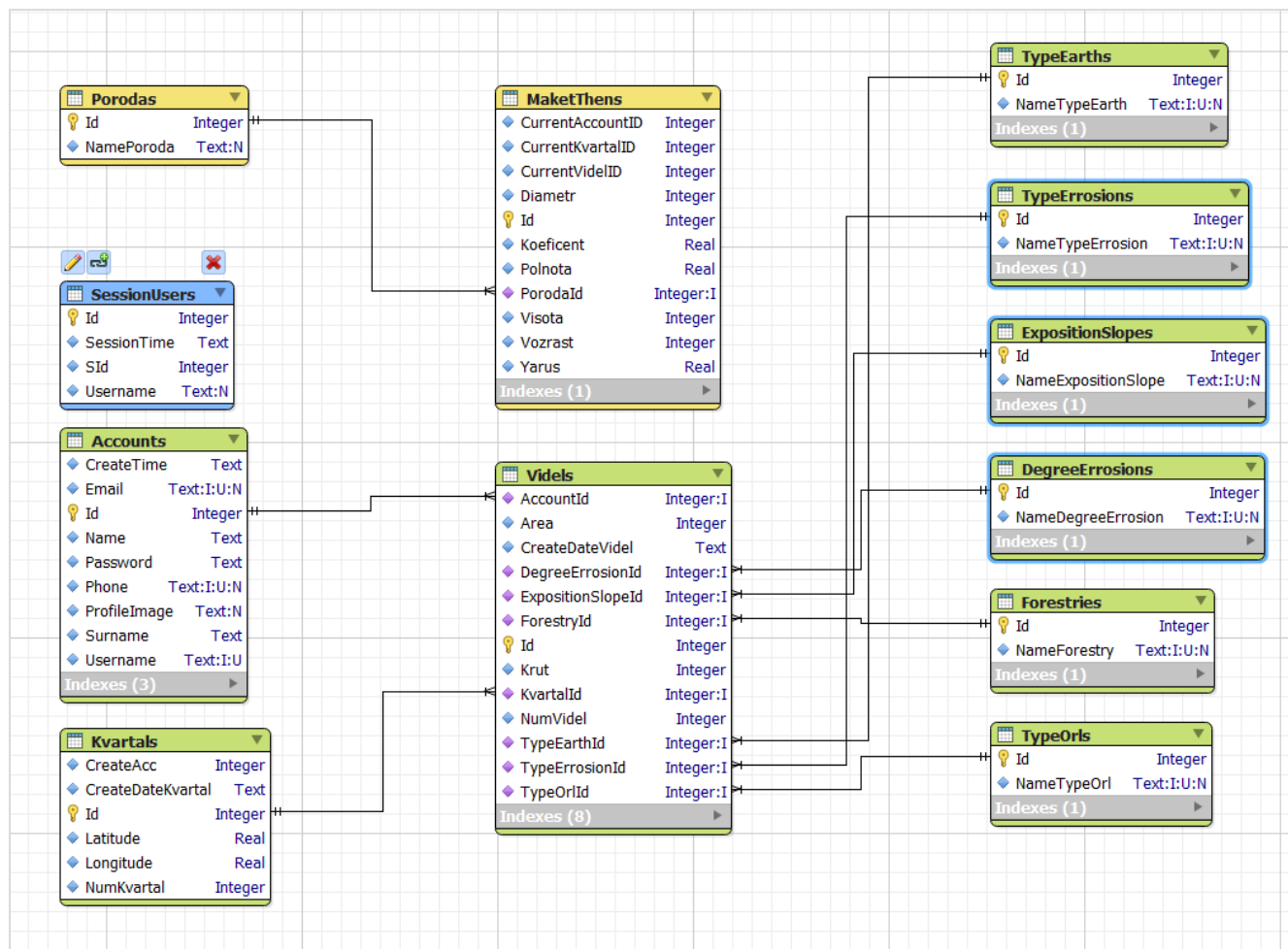


Рисунок 3.5

3.4 Функции и элементы управления

Функция «Вход в программу» предназначена для проверки данных и входа в программу.

Функция «Назад» предназначена для возврата на предыдущую страницу.

Функция «Удаление» предназначена для удаления выбранных записей.

При загрузке приложения открывается форма входа, после ввода соответствующих данных при клике на кнопку «Войти» будет происходить переход на главную страницу. Код реализации данной функции представлен ниже.

```
private async void Login_Click(object sender, RoutedEventArgs e)
{
    using (ModelContext db = new ModelContext())
    {
        // получение первой строки таблицы из базы данных;
        var account = db.Accounts.FirstOrDefault(u => u.Username == usernameBox.Text);
        if (account != null)
        {
            // проверка пароля на соответствие
            if (account.Password == passwordBox.Password)
            {
                db.SessionUsers.Add(new SessionUser { SId = account.Id, Username =
usernameBox.Text, SessionTime = DateTime.Now, });
                db.SaveChanges();
                Frame.Navigate(typeof(MainPage), new DrillInNavigationTransitionInfo());
            }
            else
            {
                // открытие диалогового окна с сообщением об ошибке
                ContentDialog deleteFileDialog = new ContentDialog()
                {
                    Title = "Сообщение об ошибке",
                    Content = "Не верный пароль, желаете продолжить? ",
                    PrimaryButtonText = "ОК",
                    SecondaryButtonText = "Отмена"
                };
                ContentDialogResult result = await deleteFileDialog.ShowAsync();
            }
        }
        else
        {
            ContentDialog deleteFileDialog = new ContentDialog()
            {
                Title = "Сообщение об ошибке",
                Content = "Пользователь не найден ",
                PrimaryButtonText = "ОК"
            };
            ContentDialogResult result = await deleteFileDialog.ShowAsync();
        }
    }
}
```

После перехода на следующую страницу может потребоваться вернуться на главную страницу приложения, при клике на кнопку «Назад» будет происходить переход на главную страницу. Код функции для осуществления переходов между страницами представлен ниже.

```
private void GoToMainPage()
{
    if (Frame.CanGoBack) //если фрейм может перейти на предыдущую страницу
        Frame.GoBack() // переходим на предыдущую страницу
    else
        Frame.Navigate(typeof(DashboardPage)); // иначе переходим на главную страницу}
```

При работе с картами таксации может потребоваться в любое время удалить запись, при клике на кнопку «Удалить» будет происходить удаление выбранной записи. Код функции для удаления выдела представлен ниже.

```
private void Delete_Click(object sender, RoutedEventArgs e)
{
    if (newItemGridView.SelectedItem != null) // если выделенный элемент не пуст
    {
        // Kvartal kvv = newItemGridView.SelectedItem as Kvartal;
        if (kvv != null) // таблица не равна пустой строке
        {
            using (ModelContext db = new ModelContext())
            {
                SessionUser current = db.SessionUsers.LastOrDefault();
                // db.Kvartals.Remove(kvv); // удаляем выделенную запись
                db.SaveChanges(); // сохраняем результат удаления
                newItemGridView.ItemsSource = db.Kvartals.Where(x => x.CreateAcc ==
current.SId).ToList();
            }
        }
    }
}
```

3.5 Проектирование справочной системы приложения

Для корректной работы с приложением требуется обеспечить пользователя справочной системой, в которой будут приведены приемы работы с приложением, включающие данные о том, что произойдет после нажатия на определенную кнопку или при выборе вкладки.

Справочная система необходима для ознакомления с программой. В ней должна присутствовать информация, которая поможет в решении проблемы с приложением, а также может напомнить, как пользоваться программным средством.

Система справки данного приложения будет содержать следующие разделы:

- «Инструкция по применению»;
- «Условия использования»;
- «Технические требования»;
- «О программе».

Справочная система будет создана на языке гипертекстовой разметки «HTML».

(HyperText Markup Language) HTML — это стандартизированный язык разметки документов во всемирной паутине. Большинство веб-страниц содержат описание разметки на языке HTML. Язык HTML интерпретируется браузерами; полученный в результате интерпретации форматированный текст отображается на экране монитора компьютера или мобильного устройства. [5].

Справка будет находится в настройках приложения во вкладке «Справка».

4 Описание программного средства

4.1 Общие сведения

Программное средство «AppTaxationCard.appx» предназначено снизить затраты времени на обработку информации о лесных ресурсах, формировать отчеты по определённым видам и категориям.

Программное средство создано в среде разработки Visual Studio 2019 на языке программирования C#. Программа может работать исключительно под управлением операционной системы Windows 10. К проекту подключена база «LocalDataBase», размером в 8 Мбайт, представленная одним файлом с расширением «.db». Само приложение имеет объем в 50 Мбайт. Программа не требовательна к системным ресурсам.

Для успешного использования программы, на компьютере должно быть свободного места на диске не меньше 100 Мбайт, оперативная память должна быть от 200 Мбайт. Приложение использует стандартные библиотеки такие как:

- System.Collections.Generic;
- System.Data.SqlClient;
- System.IO;
- System.Linq;
- System.Runtime.InteropServices.WindowsRuntime;
- Windows.Foundation;
- Windows.Foundation.Collections;
- Windows.UI.Xaml;
- Windows.UI.Xaml.Controls;
- Windows.UI.Xaml.Controls.Primitives;
- Windows.UI.Xaml.Data;
- Windows.UI.Xaml.Input;
- Windows.UI.Xaml.Media;
- Windows.UI.Xaml.Media.Imaging;
- Windows.UI.Xaml.Navigation.

Важное место играет занимаемый объем приложения на жестком диске. Должна быть предусмотрена совместимость программного приложения с другими программными ресурсами. Установка программы производится с установочного пакета. Для входа в программу требуется ввод логина и пароля.

4.2 Функциональное назначение

Программа предназначена для эффективного ведения учета лесных ресурсов и их материальной оценки, вывода соответствующих документов на печать. Без информационной системы весь процесс учета происходит вручную, путем заполнения документов.

На данный момент в приложении предусмотрен ввод логина и пароля для доступа к данным конкретного пользователя.

Назначение программного средства «AppTaxationCard» является автоматизация работы организации, которая занимается учетом леса и лесных ресурсов. В данной программе классом решаемых задач являются:

- возможность добавлять, изменять и удалять данные карточек;
- возможность сортировать, фильтровать и искать данные по различным критериям;
- возможность создавать отчеты по шаблону;
- возможность добавлять местоположение выдела на реальной карте.

4.3 Входные и выходные данные

Исходными для работы информационной системы будут являться данные, вводимые в поля ввода.

Входными является данные, вводимые пользователем в таблицы базы данных: «Учетная запись», «Карточка выдела», «Кварталы».

Кроме того, входными являются данные, вводимые пользователем в поля для поиска и фильтрации данных.

При добавлении новой информации в таблицу «Кварталы» необходимо ввести следующие данные:

- «Номер квартала»;
- «Ширина»;
- «Долгота».

При добавлении новой информации в таблицу «Пользователи» необходимо ввести следующие данные:

- «Имя пользователя»;
- «Имя»;
- «Фамилия»;
- «Электронный адрес почты»;
- «Номер телефона»;
- «Пароль».

При добавлении новой информации в таблицу «Выделы» необходимо ввести следующие данные:

- «Номер выдела»;
- «Лесничество»;
- «Площадь»;
- «Вид земли»;
- «ОРЛ»;
- «Экспозиция склона»;
- «Крутизна склона»;
- «Вид эрозии»;
- «Степень эрозии».

Выходная информация будет представлена в виде отчётов, экспортируемые в Word или Excel.

Примеры сформированных отчётов представлены в приложении Б, на рисунках Б.1 – Б.3.

5 Методика испытаний

5.1 Технические требования

Минимальные системные требования персонального компьютера для оптимальной работы программного средства:

- операционная система Windows 10 32-64бит;
- процессор AMD Athlon / Intel Pentium 2500 МГц или совместимый аналог;
- оперативная память: 4 Гбайт
- 30 Мбайт свободного места на винчестере
- видеокарта с объемом памяти не менее 256 Мбайт.
- для удобства использования клавиатура и мышь.

Наиболее удобной операционной системой для проведения испытаний является Windows 10, так как она ориентирована на максимальное использование всех возможностей персонального компьютера, сетевых ресурсов, и обеспечение комфортных условий работы.

5.2 Функциональное тестирование

Данное тестирование проводится для выявления неполадок и недочетов программы на этапе ее сдачи в эксплуатацию. При функциональном тестировании осуществляется проверка каждого пункта меню, каждой операции, с целью проверки выполнения всех функций, определенных на этапе объектно-ориентированного анализа и проектирования. Функциональное тестирование должно гарантировать работу всех элементов управления в автономном режиме. Тестирование программы будет производиться последовательно, переходя из одной части программы в другую. Во время теста будут проверяться все действия с программой, навигация по пунктам меню, которые может произвести пользователь. После чего, все собранные и найденные ошибки будут исправлены.

При запуске программы открывается окно загрузки после, форма входа программного средства. Окно загрузки и форма входа представлена на рисунках 5.1, 5.2.

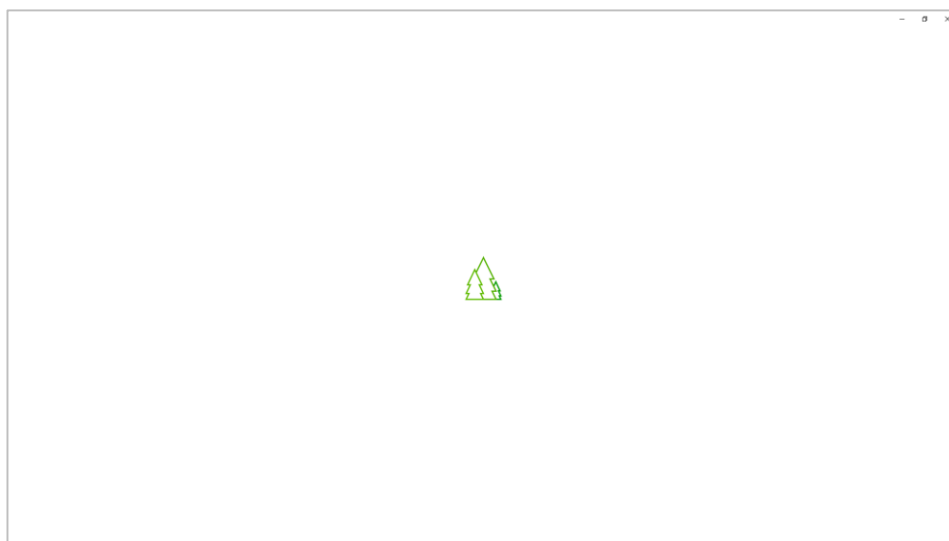


Рисунок 5.1

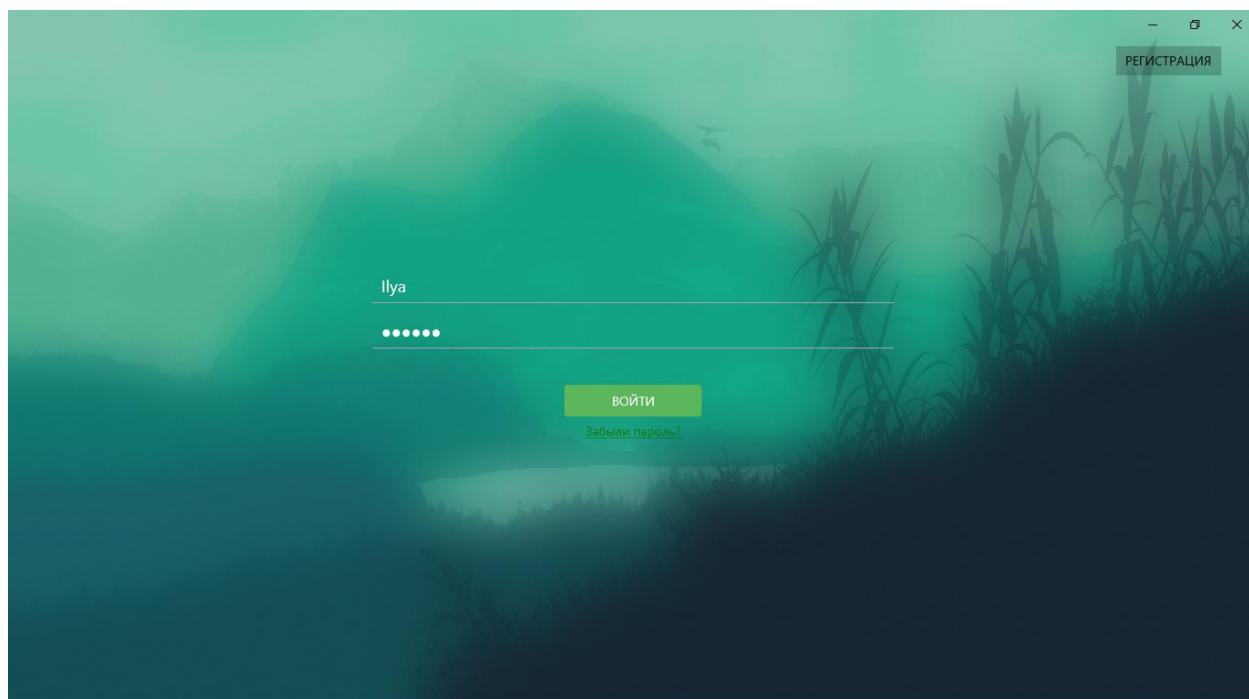


Рисунок 5.2

В таблице 5.1 представлены тест-кейсы, подготовленные для проведения функционального тестирования.

Таблица 5.1 – Тест-кейсы для проведения функционального тестирования

№ тест-кейса	Модуль	Описание	Ожидаемый результат	Реальный результат
001	Авторизация	1. Заполнить требуемые поля данными - "Логин" - Пуя; - "Пароль" – 121790; 2. Нажать на кнопку «Войти»	1. Обязательные поля заполнены 2. Осуществляется переход на главную страницу пользователя.	Результат соответствует ожидаемому. Все условия соблюдены. Результат входа в программу представлен на рисунке 5.5.
002	Авторизация	1. Заполнить требуемые поля данными - "Логин" - Пуяс; - "Пароль" – 121793; 2. Нажать на кнопку «Войти»	1. Обязательные поля заполнены 2. Осуществляется переход на главную страницу пользователя.	Результат не соответствует ожидаемому. Все условия соблюдены. Результат вывода сообщения представлен на рисунке 5.3.
003	Авторизация	1. Заполнить требуемые поля данными - "Логин" - Пуя; - "Пароль" – 434; 2. Нажать на кнопку «Войти»	1. Обязательные поля заполнены 2. Отображается сообщение об ошибке «Пользователь не найден».	Результат не соответствует ожидаемому. Все условия соблюдены. Результат вывода сообщения представлен на рисунке 5.4.
004	Регистрация	1. Заполнить требуемые поля данными - "Логин" - Пуя; - "Имя" – Илья; - "Фамилия" – Алейчик; - "Email" – ilya.alejchi@outlook.com; - "Телефон" – +375333343156; - "Пароль" – 121790; - "Повторение пароля" – 121790; 2. Нажать на кнопку «Зарегистрироваться»	1. Обязательные поля заполнены 2. Отображается сообщение «Регистрация успешно завершена». 3. Автоматический переход на исходную страницу.	Результат частично соответствует ожидаемому. Все условия соблюдены. Регистрация проходит, сообщение не отображается.

005	Добавление	1. Нажать на кнопку «Добавить». 2. Заполнить требуемые поля данными - "Номер квартала" - 1; - "Номер выдела" – 1; - "Лесничество" – Дзержинское; - "Площадь" – 220; - "Вид земель" – Квартальные просеки; - "ОРЛ" – Леса генетических резерватов; - "Экспозиция" – СЗ; - "Крутизна" – 23; - "Вид" – Ветровая; - "Степень" – Сильная; 3. Нажать на кнопку «Сохранить»	1. Обязательные поля заполнены 2. Осуществляется переход на главную страницу.	Результат не соответствует ожидаемому. Все условия соблюдены. Результат вывода представлен на рисунке 5.5
006	Удаление	1. Выделить нужный квартал. 2. Нажать на кнопку «Удалить»	1. Условия выполнены. 2. Осуществляется удаление выделенного квартала.	Результат соответствует ожидаемому. Все условия соблюдены. Результат вывода представлен на рисунке 5.6
007	Режим карты	1. Нажать на кнопку «Карты»	1. Условия выполнены. 2. Осуществляется переход в режим карты.	Результат соответствует ожидаемому. Все условия соблюдены. Результат вывода представлен на рисунке 5.7
008	Режим списка	1. Нажать на кнопку «Список»	1. Условия выполнены. 2. Осуществляется переход в режим списка.	Результат соответствует ожидаемому. Все условия соблюдены. Результат вывода представлен на рисунке 5.8
009	Режим сетки	1. Нажать на кнопку «Сетка»	1. Условия выполнены. 2. Осуществляется переход в режим сетки.	Результат соответствует ожидаемому. Все условия соблюдены. Результат вывода представлен на рисунке 5.9

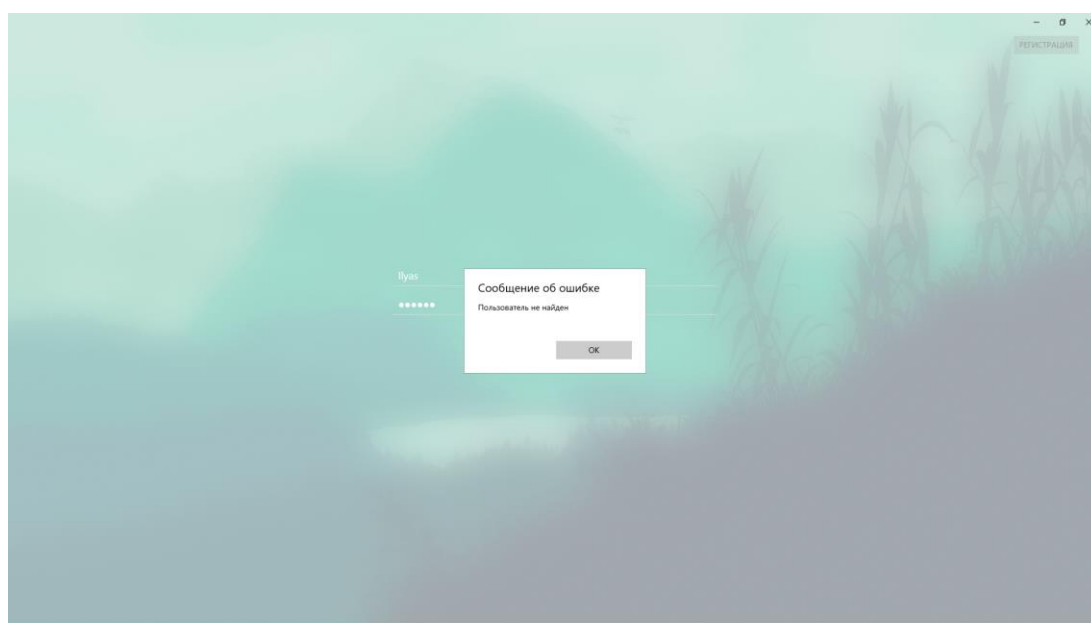


Рисунок 5.3

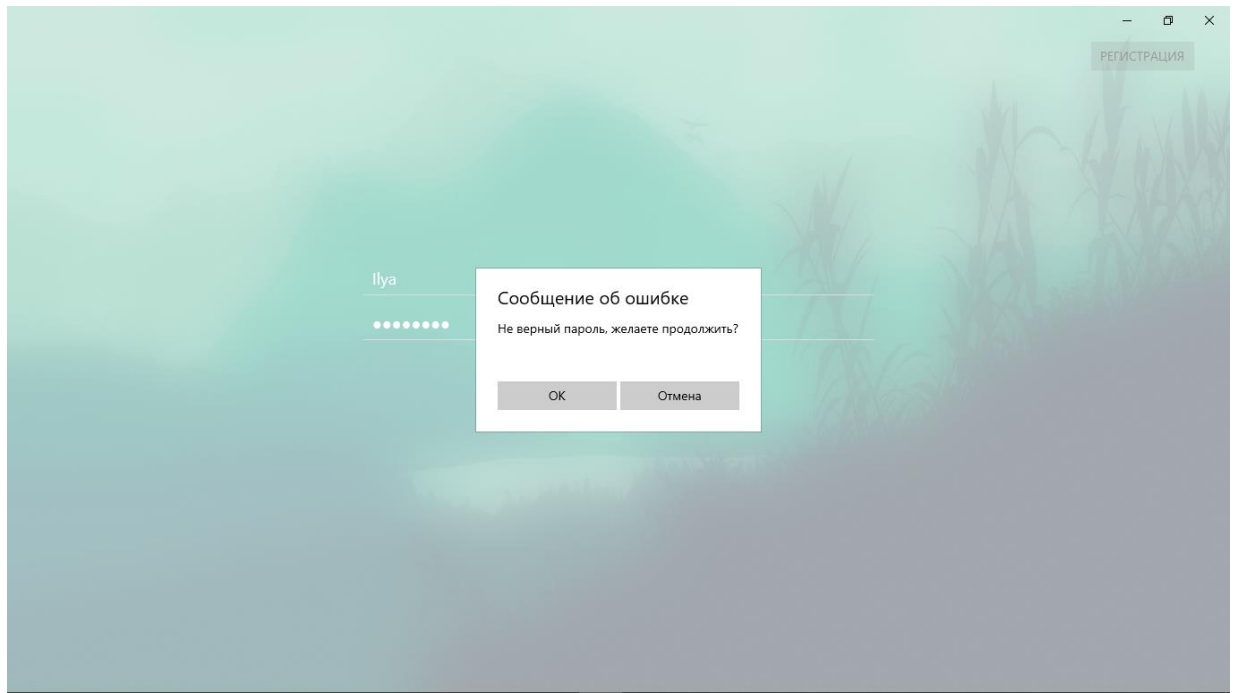


Рисунок 5.4

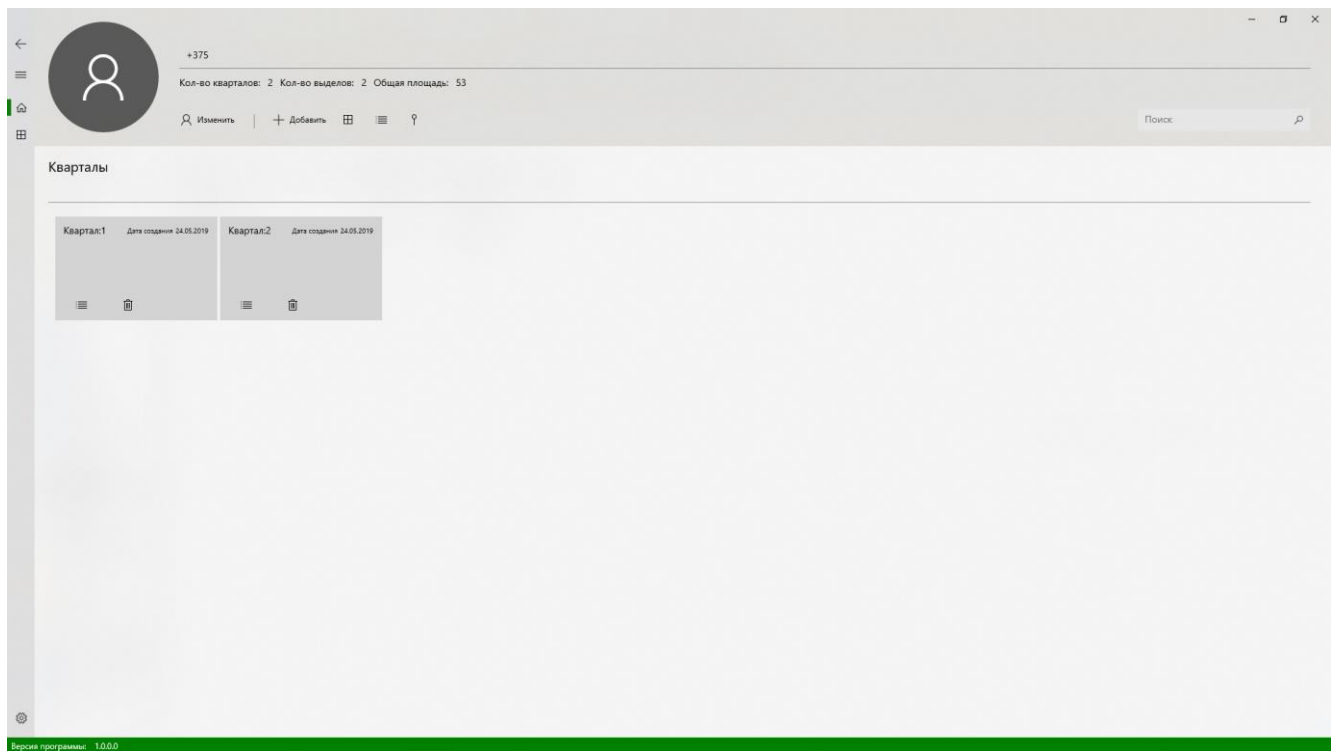


Рисунок 5.5

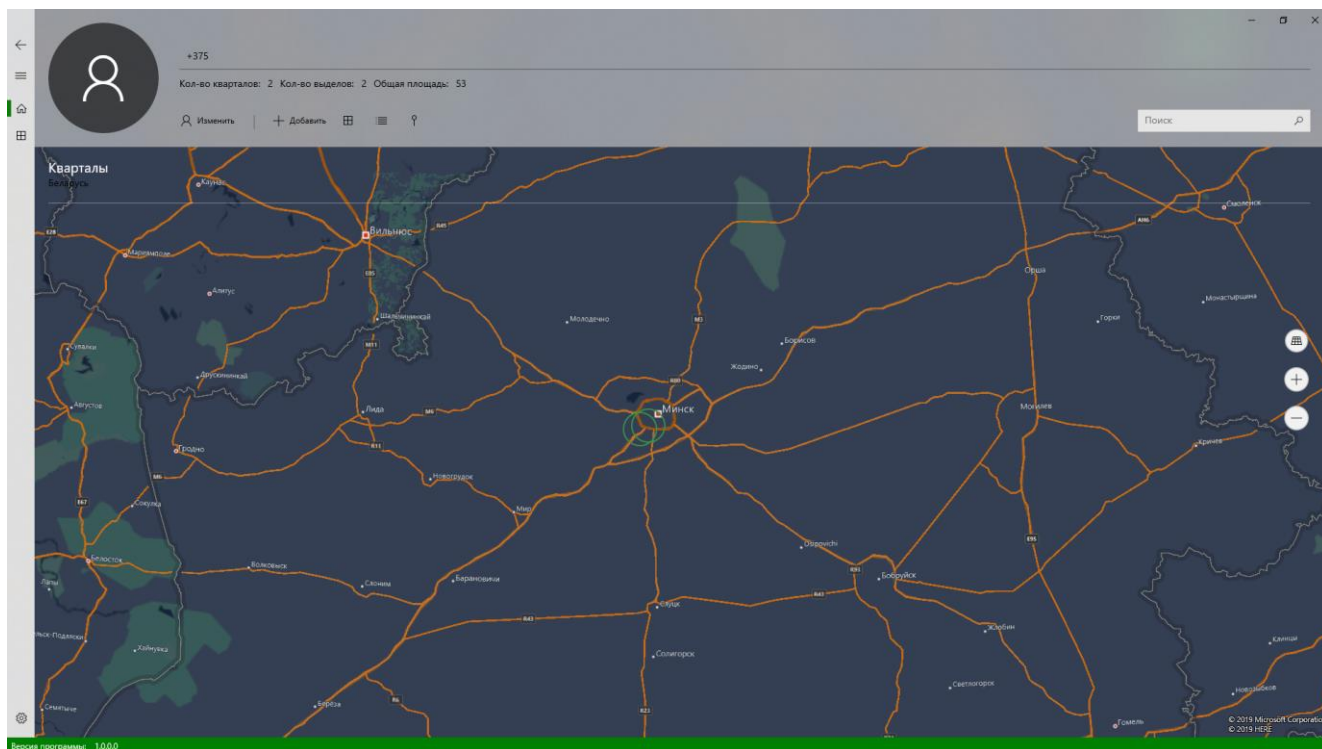


Рисунок 5.6

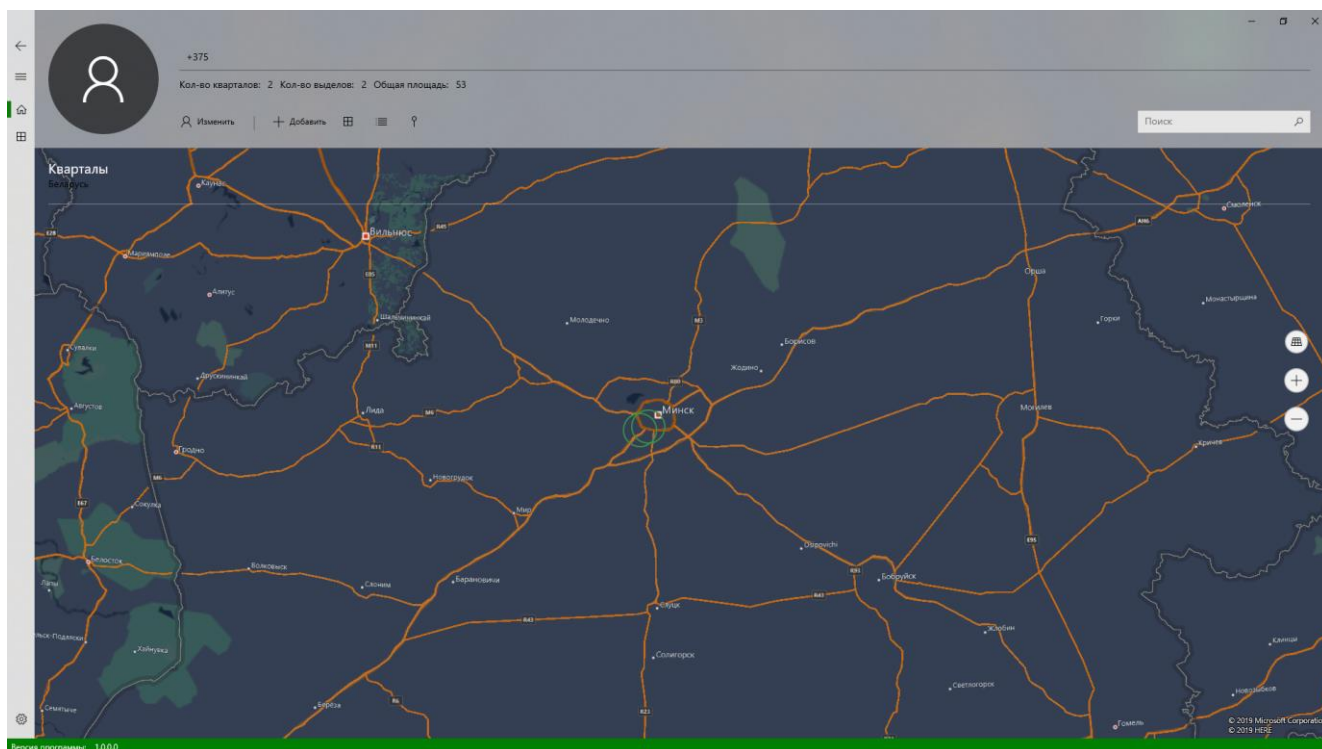


Рисунок 5.7

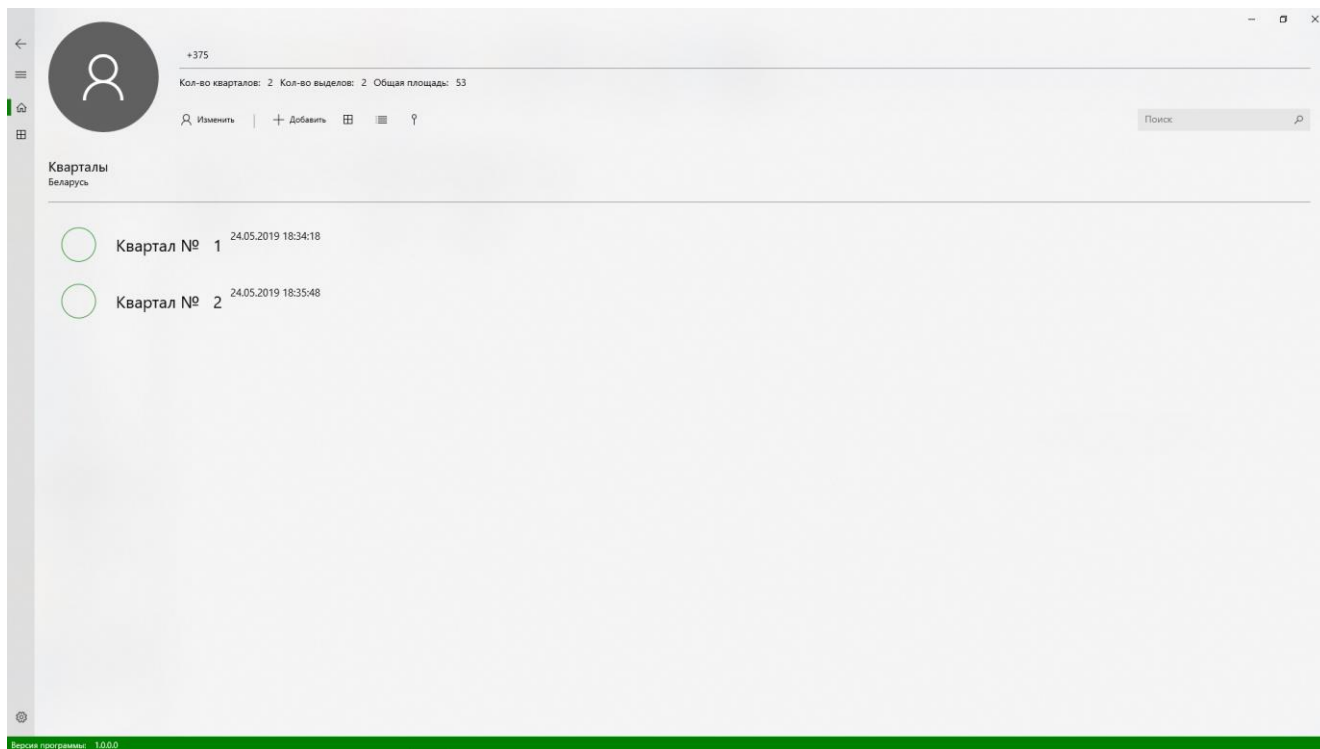


Рисунок 5.8

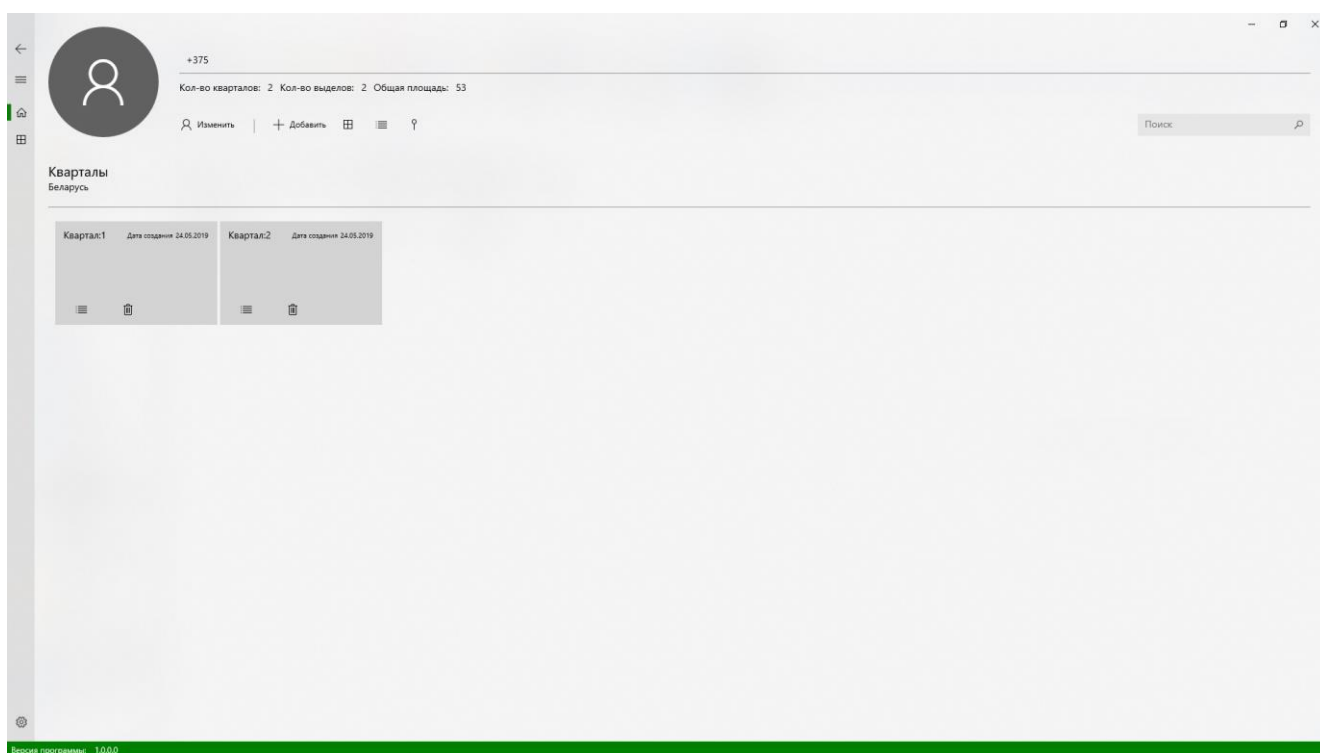


Рисунок 5.9

6 Применение

6.1 Назначение программы

Программа «AppTaxationCard.appx» предназначена для сотрудников организаций, которые занимаются учетом лесных ресурсов. В этой программе сотрудник ведёт учёт лесных ресурсов, может создавать отчёты по интересующей его информации: карточку таксации, отчет о выделах.

6.2 Условия применения

Для применения данного программного средства необходимы следующие технические требования:

- операционная система Windows 10;
- процессор Intel Core 2 Duo или выше;
- минимальный объем оперативной памяти — 500 Мбайт;
- жесткий диск – 10 Гбайт;
- видеокарта – 256 Мбайт.

6.3 Справочная система

Справочная система программного средства представляет собой отдельный файл «index.html», который открывается при выборе подпункта меню «Справка». Просмотр справки выполняется встроенным браузером программы. В справочной системе даны ответы на типичные вопросы, возникающие при работе с приложением, что должно помочь при освоении программного средства.

Справка имеет следующие разделы:

- «Инструкция по применению»;
- «Условия использования».
- «Технические требования»;
- «О программе»;

Структура справочной системы представлена на рисунке 6.1.

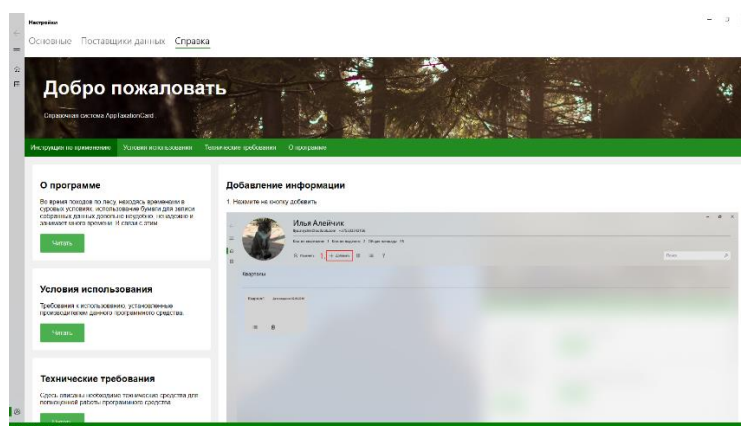


Рисунок 6.1

В разделе «Инструкция по применению» рассказывается о том, как следует работать с программой. Содержимое раздела представлено на рисунке 6.2.

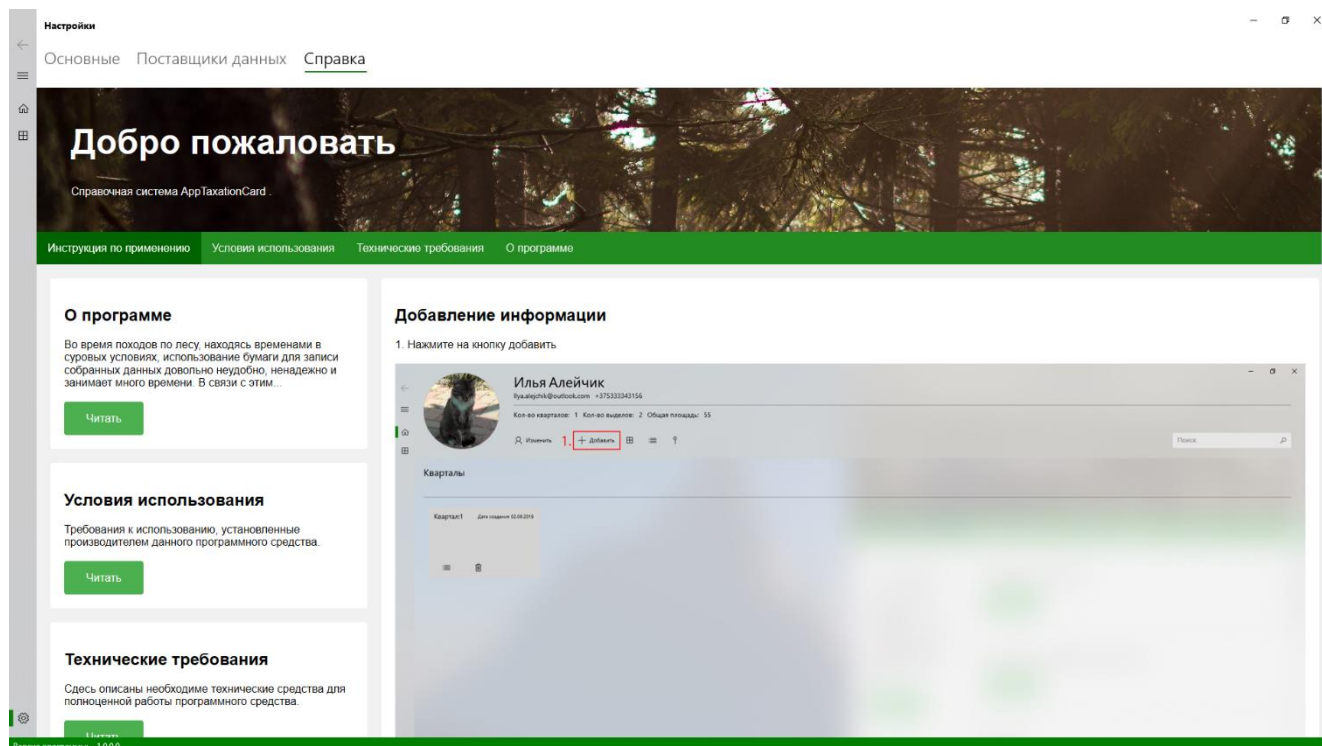


Рисунок 6.2

В разделе «Технические требования» рассказывается о требованиях программы к аппаратным ресурсам. Содержимое раздела представлено на рисунке 6.3.

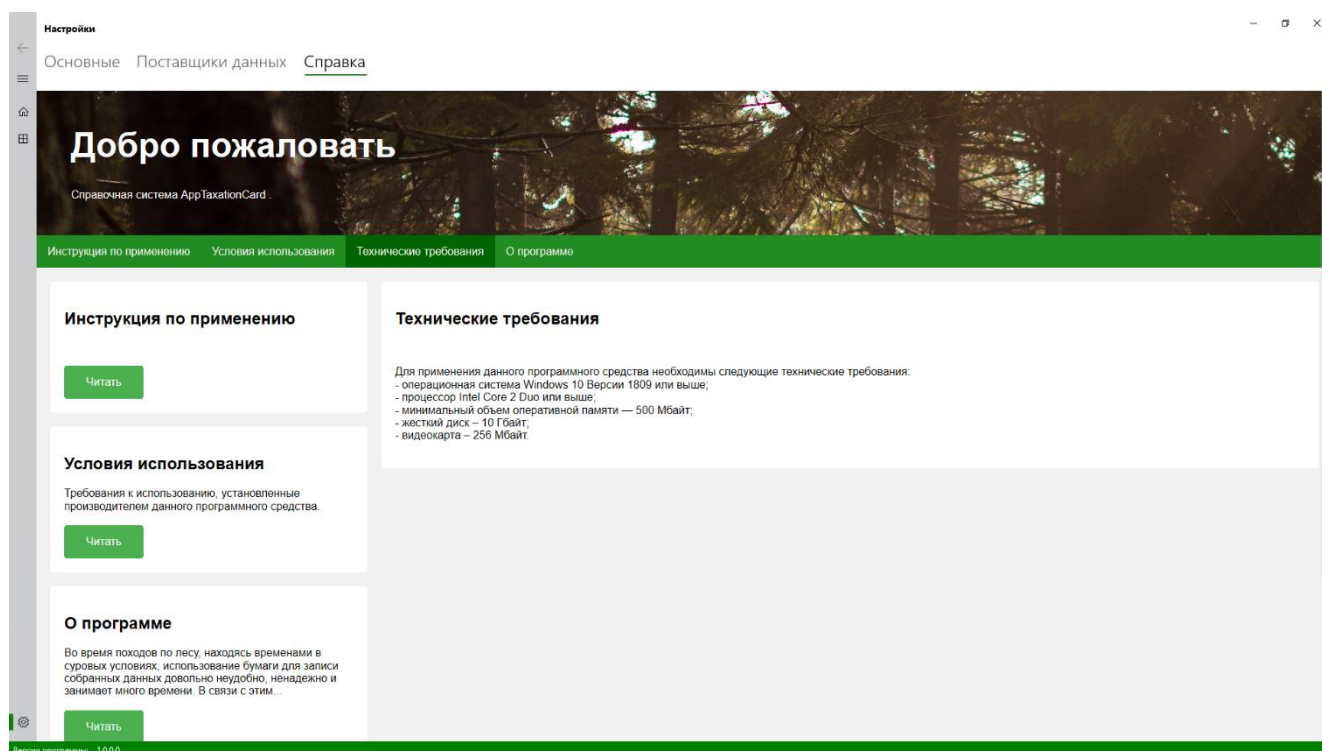


Рисунок 6.3

В разделе «Условия использования» отображена информация об условиях использования данной программы, содержит информацию о наложенных ограничениях во время использования. Содержание раздела представлено на рисунке 6.4.

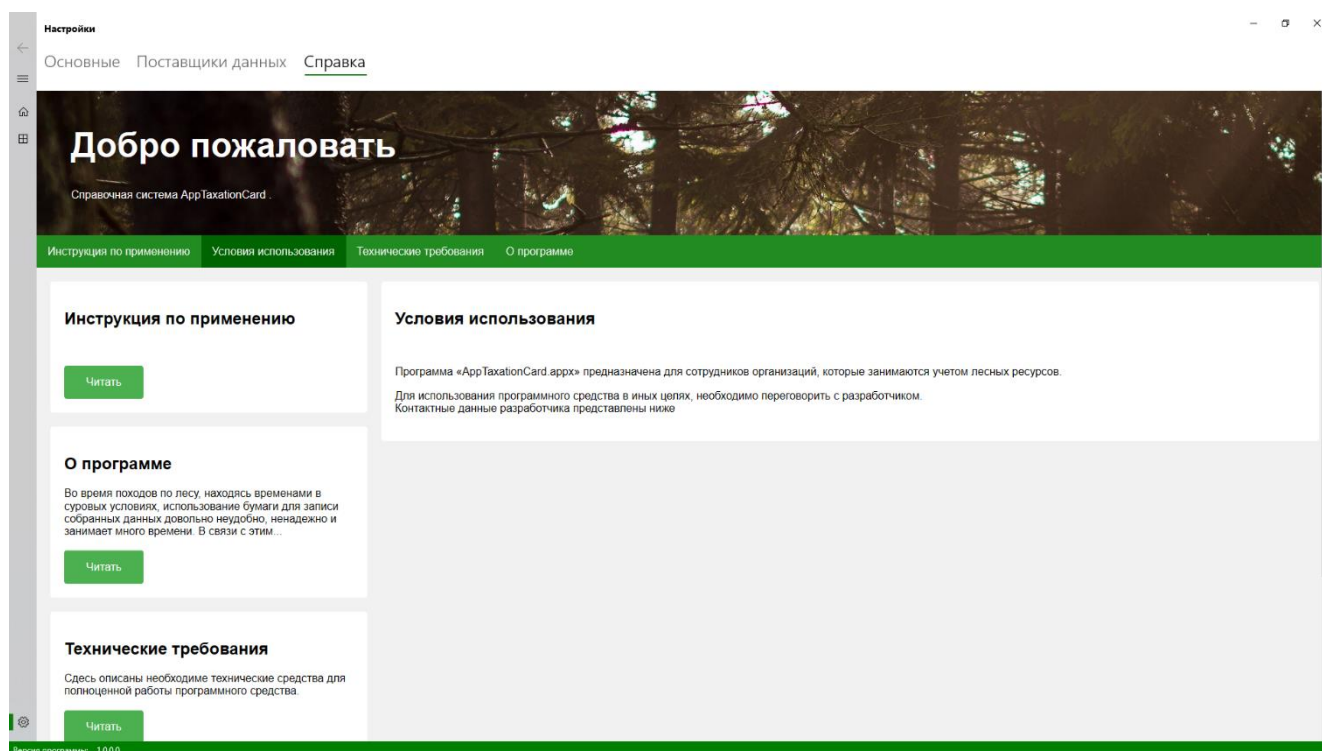


Рисунок 6.4

В разделе «О программе» рассказывается о приложении и о разработчике. Содержимое раздела представлено на рисунке 6.5.

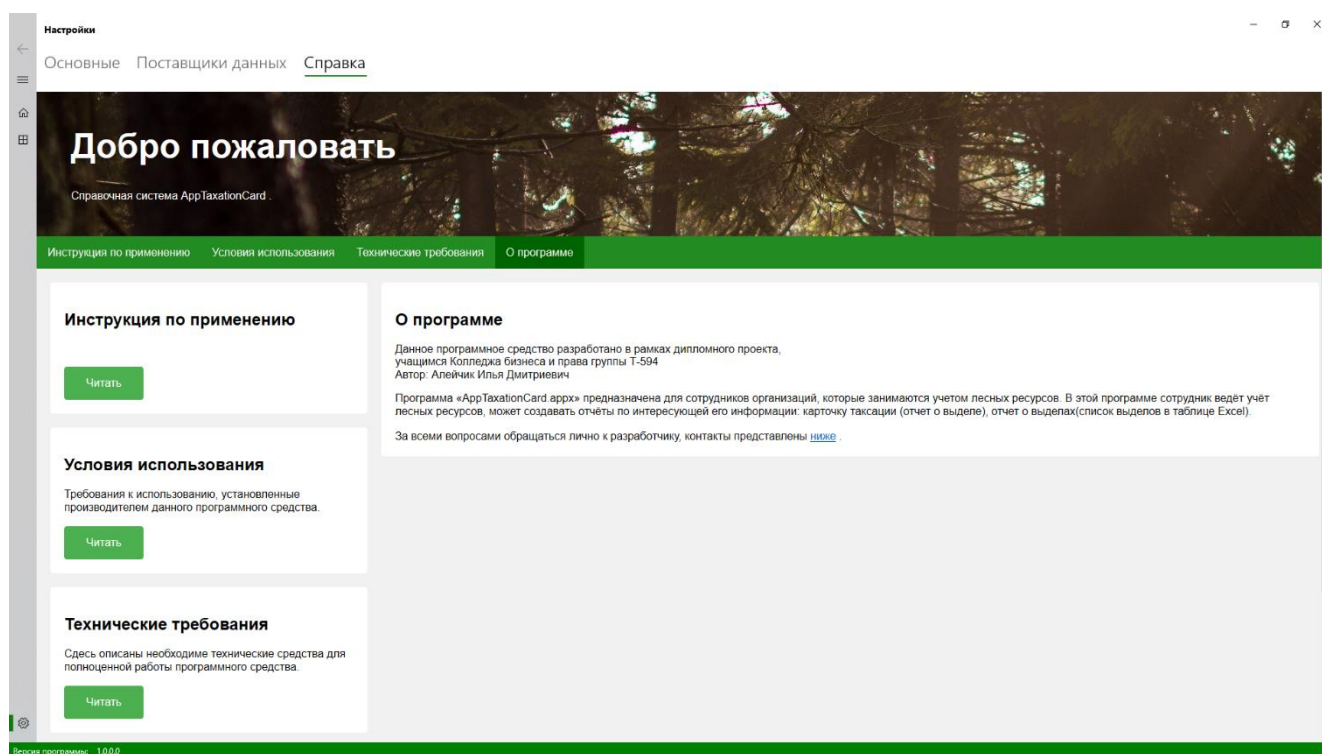


Рисунок 6.5

7 Охрана труда и охрана окружающей среды

7.1 Правовые, нормативные, социально-экономические и организационные вопросы охраны труда.

В РУП «Белгослес» разработано и функционирует Система управления охраной труда (СУОТ). СУОТ разработан в соответствии с требованиями нормативных правовых актов Республики Беларусь в области охраны труда и СТБ 18001-2009 «Системы управления охраной труда. Требования» и направлено на создание на предприятии условий, обеспечивающих сохранение жизни и здоровья работников в процессе трудовой деятельности, соблюдение законодательства в области охраны труда.

Для организации работы и осуществление контроля по охране труда на предприятии создана служба охраны труда в лице инженера по охране труда.

Основными задачами службы охраны труда являются:

- координация деятельности структурных подразделений организации по обеспечению здоровых и безопасных условий труда;
- совершенствование СУОТ;
- внедрение передового опыта и научных разработок по безопасности и гигиене труда, пропаганда охраны труда;
- информирование и консультирование работников организации, в том числе ее руководителя, по вопросам охраны труда;
- осуществление контроля за соблюдением требований актов законодательства, технических нормативных правовых актов и локальных нормативных правовых актов, содержащих требования охраны труда.

Главный инженер предприятия организует выполнение мероприятий по устранению недостатков и нарушений по охране труда, выявленных в ходе проведения ежемесячного контроля. О нарушениях, которые не могут быть устранены работниками структурного подразделения, сообщается генеральному директору предприятия.

Организация, оперативное руководство и координация работы по охране труда, общее руководство управлением охраной труда осуществляется генеральным директором РУП "Белгослес". Ответственным за реализацию функций СУОТ являются: главный инженер, заместитель генерального директора, начальники экспедиций, отделов и структурных подразделений, главные инженеры экспедиций.

Инженер по охране труда осуществляет организационно-методическую работу по управлению охраной труда, подготовку управленческих решений, контроль за их реализацией под руководством главного инженера.

Функции инженера по охране труда:

- анализ причин производственного травматизма и профессиональной заболеваемости в организациях, расположенных на подведомственной территории, разработка мер по их профилактике;
- участие в организации разработки и реализации территориальных целевых программ улучшения условий и охраны труда;
- организация обучения, повышения квалификации и проверки знаний по вопросам охраны труда руководителей и специалистов организаций, расположенных на подведомственной территории, которые не находятся в подчинении республиканских органов государственного управления, иных государственных организаций, подчиненных Правительству Республики Беларусь, и других организаций;
- организация и руководство работой кабинета охраны труда в соответствии с Типовым положением о кабинете охраны труда, утвержденным постановлением Министерства труда Республики Беларусь от 8 ноября 1999 г;

- подготовка государственной статистической отчетности по охране труда;
- пропаганда и распространение передового опыта в области охраны труда в организациях, расположенных на подведомственной территории, в том числе через средства массовой информации, организацию выставок, смотров-конкурсов, семинаров, совещаний и других мероприятий;
- участие в установленном законодательством порядке в расследовании несчастных случаев на производстве;
- информационное обеспечение организаций, расположенных на подведомственной территории, по вопросам охраны труда;
- консультирование, подготовка ответов на запросы организаций, рассмотрение в установленном порядке писем, обращений граждан по вопросам охраны труда.

В соответствии с требованиями ст.17 Закона от 23.06.2008 №356-З «Об охране труда» и ст.226 Трудового кодекса работодатель обязан осуществлять обучение, стажировку, инструктаж и проверку знаний, работающих по вопросам охраны труда.

Обучение по охране труда работников предприятия производится в соответствии с действующими Государственными стандартами, межотраслевыми положениями, типовыми инструкциями и программами.

Обучение руководителей и специалистов осуществляется по учебным планам и программам, составленным на основании типового перечня вопросов для обучения и проверки знаний по вопросам охраны труда руководителей и специалистов в соответствующих учреждениях.

Руководитель организации утверждает перечень должностей специалистов, которые должны проводить стажировку и определяет ее продолжительность.

Не позднее месяца со дня назначения на должность и периодически в соответствии с требованиями нормативных правовых актов, но не реже одного раза в три года, руководители и специалисты проходят проверку знаний по вопросам охраны труда и комиссиях для проверки знаний по вопросам охраны труда.

Ответственность за организацию своевременного и качественного проведения обучения, инструктажа, стажировки и проверки знаний работников по вопросам охраны труда в предприятии возлагается на генерального директора, в структурных подразделениях – на их руководителей.

Вводный инструктаж с вновь принятыми работниками всех специальностей проводит инженер по охране труда, при его отсутствии, лицо, назначенное приказом предприятия в оборудованном кабинете.

Первичный, повторный, целевой и внеплановый инструктажи проводятся главными инженерами экспедиций, начальниками отделов и подразделений.

Расследование и учет несчастных случаев, профессиональных заболеваний в РУП «Белгослес» осуществляется в соответствии с «Правилами расследования и учета несчастных случаев на производстве и профессиональных заболеваний», утвержденными Постановлением Совета Министров Республики Беларусь от 15.01.2004 г. №30.

Расследование несчастного случая, профессионального заболевания, на производстве проводится с целью выявления вызвавших их причин и принятия соответствующих мер для предотвращения повторения подобных ситуаций, а именно:

- установления причин несчастного случая и определение мер по экстренному устранению причин травм.
- установления должностных и иных лиц, виновных в нарушениях, которые привели к несчастному случаю; какие требования норм, правил были нарушены;
- привлечения к ответственности виновных лиц на основании законодательства и иных нормативных правовых актов.

Основными видами контроля на предприятии являются:

– контроль за соблюдением законодательства об охране труда, осуществляемый руководителями и специалистами предприятия в соответствии с их должностными обязанностями;

– контроль по охране труда, осуществляемый службой охраны труда предприятия в соответствии с Типовым положением о службе охраны труда организации, утвержденным постановлением Министерства труда и социальной защиты Республики Беларусь от 24 мая 2002 г. N82 (Национальный реестр правовых актов Республики Беларусь, 2002 г., N89, 8/8286);

– периодический контроль за соблюдением законодательства об охране труда, осуществляемый представителями нанимателя с участием общественных инспекторов профсоюзов по охране труда;

В РУП «Белгослес» периодический контроль осуществляется представителями нанимателя с участием общественных инспекторов профсоюзов по охране труда, проводится:

- ежедневно;
- ежемесячно;
- ежеквартально.

Ежедневно контроль за состоянием охраны труда осуществляется руководителем структурного подразделения с участием общественного инспектора профсоюза по охране труда в отделе материально технического обеспечения, участке полиграфических работ и в производственном отделе.

Ежемесячно контроль проводится на полевых работах. Проводится главным инженером предприятия с участием общественного инспектора профсоюза по охране труда, ведущим инженером по охране труда, на полевых работах – главным инженером экспедиции.

Ежеквартально контроль осуществляется на предприятии в целом генеральным директором предприятия с участием руководителей структурных подразделений, общественных инспекторов по охране труда [9].

7.2 Организация режима труда и отдыха при работе с ПЭВМ

Виды трудовой деятельности разделяются на 3 группы:

– группа А - работа по считыванию информации с экрана ВДТ, ЭВМ или ПЭВМ с предварительным запросом;

– группа Б - работа по вводу информации;

– группа В - творческая работа в режиме диалога с ЭВМ.

Виды трудовой деятельности делятся на 3 категории по тяжести и напряженности работы с ВДТ, ЭВМ и ПЭВМ, которые определяются:

– для группы А - по суммарному числу считываемых знаков за рабочий день (смену), но не более 60 000 знаков за смену;

– для группы Б - по суммарному числу считываемых или вводимых знаков за рабочий день (смену), но не более 40 000 знаков за смену;

– для группы В - по суммарному времени непосредственной работы с ВДТ, ЭВМ и ПЭВМ за рабочий день (смену), но не более 6 часов за рабочий день (смену).

Продолжительность непрерывной работы с ВДТ, ЭВМ и ПЭВМ без регламентированного перерыва не должна превышать двух часов.

При организации в учреждениях дошкольного образования, игровых занятий с использованием ВДТ, ЭВМ и ПЭВМ для воспитанников 5 – 6 лет должны соблюдаться следующие требования:

– не используются портативные ПЭВМ (нетбуки, ноутбуки и другое) и ВДТ (планшеты, электронные книги и другое);

– занятия проводятся не более 2 раз в неделю;

– продолжительность непрерывного занятия, связанного с фиксацией взгляда непосредственно на экране видеомонитора ПЭВМ, составляет не более 10 минут;

- одновременное использование одного ПЭВМ для двух и более детей не допускается;
- игровые занятия проводятся в присутствии педагогического работника;
- использование компьютерных игр с напряженным темпом и жестоким содержанием запрещается.

При организации занятий в учреждениях общего среднего, специального образования, должны соблюдаться следующие требования:

Количество занятий в течение учебного дня должно составлять:

- для обучающихся I-IV классов – не более одного занятия;
- для обучающихся V-IX классов – не более двух занятий;
- для обучающихся X-XI классов – не более трех занятий;

Продолжительность непрерывного занятия, связанного с фиксацией взгляда непосредственно на экране видеомонитора ПЭВМ (за исключением портативных), не должна превышать:

- для обучающихся I класса – 10 минут;
- для обучающихся II-IV классов – 15 минут;
- для обучающихся V-VII классов – 20 минут;
- для обучающихся VIII-IX классов – 25 минут;
- для обучающихся X-XI классов – 30 минут на первом занятии в расписании занятий и по 20 минут – на двух последующих занятиях;

Продолжительность непрерывного занятия, связанного с фиксацией взгляда непосредственно на ВДТ (планшеты, электронные книги и другое), экране портативного ПЭВМ, не должна превышать:

- для обучающихся I класса – 7 минут;
- для обучающихся II-IV классов – 10 минут;
- для обучающихся V-VII классов – 15 минут;
- для обучающихся VIII-XI классов – 20 минут;

Оптимальная плотность учебного занятия с использованием ПЭВМ, в том числе портативных, не должна превышать:

- для обучающихся I-IV классов – 55 %;
- для обучающихся V-IX классов – 70 %;
- для обучающихся X-XI классов – 80 %;

В учреждениях профессионально-технического, среднего специального и высшего образования:

Для обучающихся 1-го и 2-го курсов учреждений профессионально-технического, среднего специального образования, обучающихся на основе общего базового образования, режим занятий с ВДТ, ЭВМ и ПЭВМ устанавливается как для учащихся X – XI классов учреждений общего среднего образования.

При составлении расписания занятий для обучающихся учреждений высшего образования с учетом использования ВДТ, ЭВМ и ПЭВМ необходимо соблюдать следующие требования:

- в середине сдвоенного учебного занятия (90 минут) предусматривается перемена продолжительностью не менее 10 минут;
- не допускается объединение третьего и четвертого сдвоенных учебных занятий.

При использовании обучающимися ВДТ, ЭВМ и ПЭВМ, необходимо:

- чередовать теоретическую и практическую работу на протяжении занятия;
- соблюдать перерывы длительностью не менее 10 минут после каждого занятия;
- во время перерыва проводить в отсутствие обучающихся сквозное проветривание помещения с ВДТ, ЭВМ и ПЭВМ;
- централизованно отключать видеомониторы с целью обеспечения нормируемого времени работы;
- выполнять на занятиях упражнения для глаз, физкультурные минутки (в течение 1-2 минут), физкультурные паузы (в течение 3-4 минут).

Лица, работающие с ВДТ, ЭВМ и ПЭВМ более 50 % рабочего времени должны проходить обязательные медицинские осмотры [8].

7.3 Пожарная безопасность

По взрывопожарной и пожарной опасности помещения и здания РУП «Белгослес» относятся по ТКП 474-2013 к категории В1-В4, так как в помещениях обращаются горючие вещества и материалы, способные при взаимодействии с кислородом воспламенятся.

По классификации пожароопасных зон ПУЭ, организация относится к зоне П-Па, так как имеют зоны, расположенные в помещениях, в которых обращаются твердые горючие вещества.

Здания для ВЦ и части зданий другого назначения, в которых предусмотрено размещение электронных вычислительных машин, относятся к II степени огнестойкости, поскольку здание имеет несущие и ограждающие конструкции из естественных или искусственных каменных материалов, бетона или железобетона с применением листовых и плитных негорючих материалов.

Строительные конструкции здания имеют следующий предел огнестойкости и класс пожарной опасности:

- несущие элементы здания (R45-K1) – предел огнестойкости 45 минут – по потере несущей способности, мало пожароопасные;

- самонесущие стены (RE 30-K1) – предел огнестойкости 30 минут – по потере несущей способности и потере целостности независимо от того, какое из двух предельных состояний наступит ранее, мало пожароопасные;

- наружные несущие стены (E 15-K2) – предел огнестойкости 15 минут – по потере целостности, умеренно пожароопасные;

- перекрытия междуэтажные (RE1 45-K1) – предел огнестойкости 45 минут – по потере несущей способности, целостности и теплоизолирующей способности независимо от того, какое из трех предельных состояний наступит ранее, мало пожароопасные;

- элементы бесчердачных покрытий – настилы, в том числе с утеплителем (RE 15-K1) – предел огнестойкости 15 минут – по потере несущей способности и потере целостности независимо от того, какое из двух предельных состояний наступит ранее, мало пожароопасные;

- элементы бесчердачных покрытий – фермы, балки, прогоны (R 15-K1) – предел огнестойкости 15 минут – по потере несущей способности, мало пожароопасные;

- лестничные клетки – внутренние стены (REI 60-KO) – предел огнестойкости 60 минут – по потере несущей способности, целостности и теплоизолирующей способности независимо от того, какое из трех предельных состояний наступит ранее, не пожароопасные;

- лестничные клетки – марши и площадки лестниц (R 45-KO) – предел огнестойкости 45 минут – по потере несущей способности, не пожароопасные.

Согласно требованиям пожарной безопасности, предприятие обеспечено первичными средствами пожаротушения:

- огнетушители (водные (ОВ), углекислотные (ОУ-2));

- пожарные краны (ПК).

Первичные средства пожаротушения расположены в коридорах, проходах, не препятствуют безопасной эвакуации людей. Их располагают на видных местах вблизи от выходов из помещений на высоте не более 1,5 м.

Расстояние от возможного очага возгорания до места размещения огнетушителя не должно превышать 20 метров.

В каждом кабинете имеется один огнетушитель ОУ-1 с диоксидом углерода, расположен рядом с выходной дверью на высоте метра от уровня пола.

Так же во всех помещениях установлены тепловые пожарные извещатели ИП 101-01-A2MS. Извещатель предназначен для обнаружения загораний, сопровождающихся появлением дыма в закрытых помещениях зданий и сооружений, и формирования электрического сигнала о возникшем пожаре и передачи его на приемно-контрольные приборы.

В РУП «Белгослес» для проведения профилактических мероприятий по предупреждению и тушению пожаров создана добровольная пожарная дружина в составе:

- командир добровольной пожарной дружины – 1 чел;

- командир расчета – 4 чел.;
- старший боевого расчета – 4 чел.;
- член боевого расчета – 12 чел.

На добровольную пожарную дружину возложены обязанности по контролю за соблюдением противопожарного режима в помещениях, проведение разъяснительной работы среди работников предприятия по соблюдению противопожарного режима на рабочих местах, вызов пожарной службы в случае возникновения пожара, принятие мер по его тушению имеющимися средствами пожаротушения.

Выход из здания осуществляется поэтапно:

- на первом этапе люди покидают внутренние помещения.
- второй этап — выбор пути движения к выходу.

В высотных зданиях принят следующий порядок действий:

- первым эвакуируется этаж, на котором произошло возгорание, затем выводят людей с этажа выше и ниже, потом с оставшихся этажей, начиная с верхних.
- третий этап относится к этажам, расположенным выше первого.
- четвёртый этап — перемещение людей на безопасное расстояние от горящего здания.

7.4 Охрана окружающей среды

В организации РУП «Белгослес» за охрану окружающей среды отвечает инженер по охране окружающей среды, который ведет постоянный контроль деятельности всех служб на соблюдение требований инструкции по охране окружающей среды. Ответственность природопользователей, направленная в первую очередь на рациональное использование природных ресурсов, модернизацию оборудования, а также разработка и выполнение ими мероприятий по охране окружающей среды помогут снизить вредное воздействие производства на окружающую среду и улучшить экологическую обстановку в зоне влияния их хозяйственной деятельности.

Основные направления разработки мероприятий по охране окружающей среды на предприятии:

Охрана атмосферного воздуха, защита озонового слоя:

- создание автоматизированных систем контроля выбросов загрязняющих веществ в атмосферный воздух, приобретение оборудования, приборов и программных средств;
- проведение осмотров и эксплуатацию механических частей систем вентиляции.

Рациональное использование и охрана водных ресурсов:

- строительство, реконструкция, модернизация, капитальный ремонт сетей и сооружений для сбора, транспортировки, очистки, отведения и выпуска сточных вод, приобретение оборудования для этих целей;
- внедрение наилучших доступных технических методов, обеспечивающих предотвращение и снижение сбросов загрязняющих веществ в составе сточных вод в окружающую среду, а также направленных на сокращение потребления водных ресурсов.

Мероприятия, направленные на предотвращение и снижение вредного воздействия отходов:

- внедрение современных технологий по использованию и обезвреживанию отходов, технологических процессов, направленных на уменьшение объемов и (или) предотвращение образования отходов производства;
- приобретение оборудования для использования и обезвреживания отходов;
- приобретение, изготовление контейнеров для сбора отходов и вторичных материальных ресурсов.

Сточные воды подвергают очистке различными способами: механической, химической, механохимической, физико-химической и биохимической (биологической).

Механическую очистку применяют для удаления из сточных вод взвешенных веществ и частично загрязнений, находящихся в коллоидном состоянии. Для механической очистки используют решетки, песколовки, отстойники, жироловки, нефтеловушки, маслоотделители, гидроциклоны, фильтры и другие сооружения. Под действием силы тяжести частицы, удельный вес которых больше удельного веса воды, выпадают на дно сооружений, образуя осадок. В то же время частицы, удельный вес которых меньше удельного веса воды (жиры, масла, нефть), всплывают на поверхность. Осадок и всплывшие загрязнения удаляют из сооружений и направляют на обработку.

Для очистки производственных сточных вод от большого количества жиров, нефти и масел используют жироловки, нефтеловушки и маслоотделители. Эти сооружения аналогичны отстойникам, но имеют оборудование для удаления большого количества всплывающих загрязнений. Одновременно они служат и для очистки воды от оседающих веществ.

Химическая очистка заключается в выделении из сточных вод загрязнений путем проведения реакций между ними и вводимыми в воду реагентами. Такими реакциями являются реакции окисления и восстановления, реакции образования соединений, выпадающих в осадок, и реакции, сопровождающиеся газовыделением. Химическая очистка применяется для очистки только некоторых производственных сточных вод.

Механохимическую очистку применяют для выделения из сточных вод нерастворенных загрязнений. Сущность ее состоит в том, что в воду добавляют коагулянты, которые способствуют удалению из нее загрязнений в процессе ее механической очистки.

К физико-химическим методам очистки сточных вод относятся сорбция, экстракция, эвапорация, коагуляция, флотация, электролиз, ионный обмен, кристаллизация и др.

Биохимическая (биологическая) очистка заключается в окислении остающихся в воде после механической очистки органических загрязнений с помощью микроорганизмов, способных в процессе своей жизнедеятельности осуществлять минерализацию органических веществ.

Обезвреживание и захоронение токсичных отходов проводятся на специальных полигонах.

Выбор места для полигона, проектирование, строительство и функционирование строго регламентированы. Полигон размещается в обособленных, свободных от застройки, хорошо проветриваемых местах, которые допускают осуществление мероприятий и инженерных решений, исключающих загрязнение окружающей среды, населенных пунктов, зон отдыха трудящихся и источников питьевого водоснабжения.

8 Экономический раздел

8.1 Техничко-экономическое обоснование разработки программного средства

Необходимо рассчитать экономическую эффективность разработки программного средства.

В технико-экономическом обосновании будут рассмотрены следующие вопросы:

- составление плана по разработке программного средства;
- расчет стоимости разработки;
- экономическая эффективность разработки.

В первой части технико-экономического обоснования необходимо рассчитать срок разработки по созданию программного средства «Программное средство для автоматизации учета и материальной оценки лесных ресурсов».

Во второй части технико-экономического обоснования необходимо рассчитать стоимость разработки программного средства с момента получения первого варианта технического задания и заканчивая оформлением документации и сдачей разработки.

В третьей части необходимо оценить экономическую эффективность программного средства.

8.2 Составление плана по разработке программного средства

План разработки программного средства представлен в таблице 8.1.

Таблица 8.1 - План разработки программного средства

Наименование этапов и видов работ	Исполнитель	Количество исполнителей	Трудоемкость, человеко-дни
Подготовительный	техник-программист	1	2
Выбор методов и средств	техник-программист	1	5
Разработка алгоритмов и программ	техник-программист	1	20
Отладка программ и анализ результатов	техник-программист	1	10
Оформление документации и подготовка к сдаче разработки	техник-программист	1	3

8.3 Определение цены программного средства

В условиях рыночных отношений научно-техническая продукция также является товаром. Поэтому узловым вопросом технико-экономического обоснования выступает определение цены основного результата дипломного проекта.

Вместе с тем следует иметь в виду, что в отраслевых рекомендациях по установлению цены на научно-техническую продукцию фактически реализован вариант механической аналогии с материальным производством (издержек или сметной стоимости) и учет минимального уровня рентабельности.

Расчет цены основного результата дипломного проекта осуществляется в определенной последовательности.

Определяются материальные затраты на выполнение работ по теме, включая стоимость покупных комплектующих изделий и полуфабрикатов на изготовление макетов и опытных образцов.

Расчет осуществляется по формуле (8.1).

$$P_m = K_{\text{Тр}} \sum_{i=1}^n (H_{\text{pi}} C_i - O_{\text{di}} C_{\text{di}}), \quad (8.1)$$

где $K_{\text{Тр}}$ - коэффициент, учитывающий транспортно-заготовительные расходы $K_{\text{Тр}} \approx$ от 1,05 до 1,10;

H_{pi} - норма расхода i -го вида материалов на макет или опытный образец (кг, м, и так далее);

C_i - действующая отпускная цена за единицу i -го вида материала, руб.;

O_{di} - возвратные отходы i -го вида материала (кг, м, и так далее);

C_{di} - цена за единицу возвращенных отходов i -го вида материала, руб.;

n – количество применяемых видов материалов.

Расчет целесообразно представить в таблице 8.2.

Таблица 8.2 - Расчет затрат на материалы, покупные полуфабрикаты и комплектующие изделия

Наименование материалов покупных полуфабрикатов и комплектующих изделий	Единица измерения	Количество	Цена приобретения без НДС, руб.	НДС, руб.	Цена с НДС, руб.
Бумага	пачка	1	8,00	1,60	9,60
Ручка шариковая	штука	1	1,10	0,22	1,32
Папка-скоросшиватель	штука	1	0,30	0,06	0,36
Диск	штука	1	0,70	0,14	0,84
Всего расходов					12,12
Всего с транспортно-заготовительными расходами					12,73

Затраты на электроэнергию находятся исходя из продолжительности периода разработки программного обеспечения, количества кВт/ч, затраченных на его проектирование и тарифа за 1 кВт/ч. по формуле (8.2):

$$P_{\text{э}} = K_{\text{э}} \times T_{\text{р}} \quad (8.2)$$

где $K_{\text{э}}$ – стоимость одного кВт/ч, руб.;

$T_{\text{р}}$ – количество кВт/ч.

Тарифы на электроэнергию применяются согласно приложения к Декларации «Об уровне тарифов на электроэнергию, отпускаемую РУП Электроэнергетики ГПО «Белэнерго» для юридических лиц и ИП» на соответствующий период времени, когда разрабатывается программное обеспечение.

Базовый тариф для прочих потребителей с 01.01.2019 г. составляет 0,3199 руб. за 1 кВт/ч. Время реализации проекта 40 дней, среднее потребление энергии в месяц составило 75 кВт/ч, то есть было потреблено 143 кВт/ч. Исходя из вышеизложенного получаем, что на электроэнергию было затрачено:

$$P_{\text{э}} = 0,3199 \times 143 = 45,75 \text{ руб.}$$

Определяется основная заработная плата научно-технического персонала, непосредственно занятого выполнением работ.

К этой статье относятся основная заработная плата работников, а также премии, входящие в фонд заработной платы. Среднее количество рабочих дней в месяце равно 21, а средняя продолжительность рабочего дня составляет восемь часов. Следовательно, часовая заработная плата определяется делением размера оклада на количество рабочих часов в месяце (то есть на 168 часов).

Тарифная ставка седьмого разряда на предприятии составляет 65 руб. Трудоемкость определяется исходя из данных, представленных в таблице 8.1.

Произведение трудоемкости на сумму часовой заработной платы определяет затраты по зарплате для работника на все время разработки.

Расчет основной заработной платы по теме приведен в таблице 8.3.

Таблица 8.3 - Расчет затрат на основную заработную плату научно-производственного персонала

Наименование этапов работ	Исполнитель	Разряд	Тарифный коэффициент	Часовая тарифная ставка, руб.	Трудоемкость, (чел/час)	Затраты по заработной плате, руб.
Подготовительный	техник-программист	7	2,03	0,39	16	12,67
Выбор методов и средств	техник-программист	7	2,03	0,39	40	31,68
Разработка алгоритмов и программ	техник-программист	7	2,03	0,39	160	126,67
Отладка программ и анализ результатов	техник-программист	7	2,03	0,39	80	63,34
Оформление документации и подготовка к сдаче разработки	техник-программист	7	2,03	0,39	24	19,00
Всего						253,36
Всего с коэффициентом премий 25%						316,70

Определяется дополнительная заработная плата исполнителей, включающая разнообразные предусмотренные трудовым законодательством выплаты, по формуле (8.3).

$$P_{\text{дз}} = P_{\text{оз}} \frac{H_{\text{дз}}}{100}, \quad (8.3)$$

где $H_{\text{дз}}$ - норматив дополнительной заработной платы, $H_{\text{дз}} \approx$ от 10 до 25 %.

$$P_{\text{дз}} = 316,70 \cdot 25 / 100 = 79,18 \text{ руб.}$$

Рассчитываются отчисления органам социальной защиты по формуле (8.4):

$$P_{\text{ос}} = (P_{\text{оз}} + P_{\text{дз}}) \frac{H_{\text{ос}}}{100}, \quad (8.4)$$

где $H_{\text{ос}}$ - норма отчислений на социальную защиту, $H_{\text{ос}} = 34\%$.

$$P_{\text{ос}} = (316,70 + 79,18) \cdot 34 / 100 = 134,60 \text{ руб.}$$

Также рассчитываются отчисления на страхование от несчастных случаев на производстве и профессиональных заболеваний ($O_{\text{стр}}$) по ставке действующего законодательства

($H_{бгс}$ принимается равным от 0,3 до 0,9%). Для расчетов среднее значение $H_{бгс}$ принимается равным 0,6%. $O_{стр}$ рассчитывается по формуле (8.5).

$$O_{стр} = (P_{оз} + P_{дз}) \frac{H_{бгс}}{100}. \quad (8.5)$$

$$O_{стр} = (316,70 + 79,18) * 0,6 / 100 = 2,38 \text{ руб.}$$

Определяются прочие прямые расходы, связанные с амортизационными отчислениями на полное восстановление основных производственных фондов, арендная плата и лизинговые платежи, компенсация за износ (амортизацию) использованного в процессе создания научно-технической продукции оборудования по договоренности. $P_{пр}$ рассчитывается по формуле (8.6).

$$P_{пр} = P_{оз} \frac{H_{пр}}{100}, \quad (8.6)$$

где $H_{пр}$ – норматив прямых расходов, $H_{пр} \approx$ от 10 до 20 %.

$$P_{пр} = 316,70 * 10 / 100 = 31,67 \text{ руб.}$$

Исчисляются косвенные (накладные) расходы по формуле (8.7).

$$P_{кос} = P_{оз} \frac{H_{кос}}{100}, \quad (8.7)$$

где $H_{кос}$ – норматив косвенных расходов, $H_{кос} \approx$ от 50 до 100%.

$$P_{кос} = 316,70 * 50 / 100 = 158,35 \text{ руб.}$$

Определяется полная себестоимость научно-технической продукции как сумма всех групп затрат по формуле (8.8).

$$C_{п} = P_{м} + Э + P_{оз} + P_{дз} + P_{ос} + O_{стр} + P_{пр} + P_{кос}. \quad (8.8)$$

$$C_{п} = 12,73 + 45,75 + 316,70 + 79,18 + 134,6 + 2,38 + 31,67 + 158,35 = 781,36 \text{ руб.}$$

По среднему уровню рентабельности в процентах от полной себестоимости определяется плановая прибыль единицы научно-технической продукции по формуле (8.9).

$$П = C_{п} \frac{Y_p}{100}, \quad (8.9)$$

где Y_p – средний уровень рентабельности, $Y_p \approx$ от 10 до 30%.

$$П = 781,36 * 30 / 100 = 234,41 \text{ руб.}$$

Определяется приближенная (ориентировочная) отпускная цена научно-технической продукции по формуле (8.10).

$$Ц_{отп} = C_{п} + П. \quad (8.10)$$

$$Ц_{отп} = 781,36 + 234,41 = 1015,77 \text{ руб.}$$

Определяется налог на добавленную стоимость (НДС) по формуле (8.11).

$$\text{НДС} = \text{Ц}_{\text{отп}} \frac{\text{Н}_{\text{НДС}}}{100}, \quad (8.11)$$

где $\text{Н}_{\text{НДС}}$ - ставка налога НДС, $\text{Н}_{\text{НДС}}=20\%$.

$$\text{НДС}=1015,77 \cdot 20/100=203,15 \text{ руб.}$$

Определяется цена научно-технической продукции с учетом НДС по формуле (8.12).

$$\text{Ц}_{\text{отп}} = \text{Ц}_{\text{опт}} + \text{НДС} \quad (8.12)$$

$$\text{Ц}_{\text{отп}}=1015,77+203,15=1218,92 \text{ руб.}$$

Все приведенные выше расчеты целесообразно объединить в сводную таблицу 8.4.

Таблица 8.4 - Расчет ориентировочной цены научно-технической продукции

Статьи затрат	Условные обозначения	Сумма, руб.
Материалы, покупные полуфабрикаты и комплектующие изделия	P_m	12,73
Электроэнергия	$P_{\text{Э}}$	45,75
Основная заработная плата научно-производственного персонала	$P_{\text{оз}}$	316,70
Дополнительная заработная плата научно-производственного персонала	$P_{\text{дз}}$	79,18
Отчисления на социальную защиту и на страхование от несчастных случаев	$P_{\text{ос}}$ $O_{\text{стр}}$	134,60 2,38
Прочие прямые расходы	$P_{\text{пр}}$	31,67
Накладные расходы	$P_{\text{кос}}$	158,35
Полная себестоимость	$C_{\text{п}}$	781,36
Плановые накопления (прибыль)	Π	234,41
Отпускная цена (без НДС)	Ц	1015,77
Налог на добавленную стоимость	$P_{\text{НДС}}$	203,15
Отпускная цена с НДС	$\text{Ц}_{\text{отп}}$	1218,92

8.4 Экономическая эффективность разработки

В данном разделе представлено экономическое обоснование для дипломного проекта по теме «Программное средство автоматизации учета и материальной оценки лесных ресурсов».

В технико-экономическом обосновании были рассмотрены следующие вопросы:

- составление плана по разработке программы;
- расчет стоимости разработки.

В первой части технико-экономического обоснования был рассчитан срок разработки по созданию программного средства. Он составил 320 часов.

Во второй части технико-экономического обоснования была рассчитана стоимость разработки программного продукта, которая составила 1218,92 рублей.

Основное преимущество разработки программного средства в том, что программа по сравнению с аналогами проста в использовании, имеет современный и удобный интерфейс, позволяет добавлять данные в один клик и легко манипулировать ими, а также имеет карту для сохранения географических координат расположения кварталов, генерирует необходимые отчеты и выводит при необходимости на печать.

Заключение

Во время походов по лесу, находясь временами в суровых условиях, использование бумаги для записи собранных данных довольно неудобно, ненадежно и занимает много времени. В связи с этим в рамках дипломного проектирования было разработано программное средство «AppTaxationCard.appx», автоматизирующее учёт лесных ресурсов и материальную оценку лесных ресурсов.

Для достижения цели дипломного проектирования были решены следующие задачи:

- определена вычислительная система, необходимая для создания программного средства;
- разработаны логическая и физическая модели данных;
- по модели выполнена проектирование задачи;
- разработано программное средство;
- описано созданное программное средство;
- выбрана методика испытания;
- описан процесс функционального тестирования;
- приведены примеры в области применения.

В программе реализовано ведение базы данных, ее редактирование, вывод на экран, а также формирование отчетов, экспорт и вывод их на печать.

Программное средство можно модифицировать, как того пожелает заказчик.

Программа готова к использованию.

В разделе охраны труда были рассмотрены такие вопросы как: основные правовые нормативные и социально-экономические вопросы, вопросы пожарной безопасности, охраны окружающей среды, организация режима труда и отдыха при работе с ПЭВМ.

В экономическом разделе произведены расчеты экономического эффекта программного средства, расчеты заработной платы исполнителей, проведена оценка сложности и трудоемкости разработки программного средства, проведены расчеты всевозможных отчислений и расходов. Стоимость разработки программного средства составила 1218,92 руб.

Основное преимущество программного средства в том, что программа по сравнению с аналогами проста в использовании, имеет современный и удобный интерфейс, позволяет добавлять данные в один клик и легко манипулировать ими, а также имеет карту для сохранения географических координат расположения кварталов, генерирует необходимые отчеты и выводит при необходимости на печать.

Для разработки данного программного средства были применены и закреплены знания по уже изученному материалу, были отработаны навыки владения методами надёжного программирования и эффективности разработки программного обеспечения.

Данная программа может быть дополнена и модернизирована.

Список используемых источников

- 1 Багласова Т.Г. Методические указания по выполнению дипломного проекта для учащихся по специальности 2-40 01 01 «Программное обеспечение технологий» / Т.Г.Багласова. – Минск: КБП, 2017
- 2 Багласова Т.Г. Методические указания по оформлению курсовых и дипломных проектов / Т.Г.Багласова, К.О.Якимович. – Минск: КБП, 2015
- 3 Гради Буч. Объектно - ориентированный анализ и проектирование / Гради Буч. – 2-е изд. – М.: Бином, 1998
- 4 ГОСТ 2.105-95 ЕСКД. Общие требования к текстовым документам. – М.: Из-во стандартов, 1995
- 5 Крум, Э.В. Экономика предприятия. Практикум / Э. В. Крум. – Минск : Издательство Гревцова, 2009. – 360 с.
- 6 Михнюк, Т.Ф. Охрана труда / Т.Ф. Михнюк. – Минск : ИВЦ Минфина, 2009. – 365 с.
- 7 Документация по UWP [Электронный ресурс] – Документация по универсальной платформе Windows – corp. Microsoft 2019 – Режим доступа: <https://docs.microsoft.com/ru-ru/windows/uwp>
- 8 Медицинские осмотры [Электронный ресурс] – corp. ohranatruda, 2019 – Режим доступа: <http://ohranatruda.of.by/kakov-poryadok-provedeniya-medicinskix-osmotrov-rabotayushhix-na-personalnyx-evm.html>
- 9 Периодический контроль [Электронный ресурс] – corp. Охрана труда.by 2019 – Режим доступа: <http://www.ohrana-truda.by/topic/2515-periodicheskiy-kontrol-za-soblyudeniem-zakonod>
- 10 C# [Электронный ресурс] – Руководство по языку CSharp – corp. Microsoft 2019 – Режим доступа: <https://docs.microsoft.com/ru-ru/dotnet/csharp>
- 11 Microsoft Visio [Электронный ресурс] – Microsoft Visio – corp. Microsoft 2019 – Режим доступа: <https://products.office.com/ru-ru/visio/flowchart-software?rtc=1>
- 12 Microsoft Office [Электронный ресурс] – Microsoft Office – corp. Microsoft 2019 – Режим доступа: <https://products.office.com/ru-ru/?rtc=1>
- 13 Microsoft Visual Studio [Электронный ресурс] – Операционная система Windows 10 – corp. Microsoft 2019 – Режим доступа: <https://docs.microsoft.com/ru-ru/windows/uwp>
- 14 SQLite [Электронный ресурс] – SQLite – corp. sqlite 2000 - 2019 – Режим доступа: <https://www.sqlite.org/prosupport.html>
- 15 SQLite Studio [Электронный ресурс] – SQLite Studio – corp. 2007 – 2019 SalSoft Pawel – Режим доступа: <https://sqlitestudio.pl/index.rvt>
- 16 Valentina Studio 9 [Электронный ресурс] – Valentina Studio – corp. Wikipedeia – Режим доступа: <https://www.valentina-db.com/ru/>

17 Web-технологии [Электронный ресурс]. – Web-технологии: HTML – сор. 2008-2013
– Режим доступа: <http://htmlweb.ru/>

18 Windows 10 [Электронный ресурс] – Операционная система Windows 10 – сор.
Microsoft 2019 – Режим доступа: <https://www.microsoft.com/en-us/software-download/windows10>

Приложение А (обязательное) Текст программы

```
using AppTaxationCard.Models;
using System;
using System.Collections.Generic;
using System.IO;
using System.Linq;
using System.Runtime.InteropServices.WindowsRuntime;
using Windows.ApplicationModel;
using Windows.ApplicationModel.Activation;
using Windows.ApplicationModel.Core;
using Windows.Foundation;
using Windows.Foundation.Collections;
using Windows.UI;
using Windows.UI.ViewManagement;
using Windows.UI.Xaml;
using Windows.UI.Xaml.Controls;
using Windows.UI.Xaml.Controls.Primitives;
using Windows.UI.Xaml.Data;
using Windows.UI.Xaml.Input;
using Windows.UI.Xaml.Media;
using Windows.UI.Xaml.Navigation;
using Microsoft.EntityFrameworkCore;
using AppTaxationCard.Views;

namespace AppTaxationCard
{
    /// <summary>
    /// Обеспечивает зависящее от конкретного приложения поведение, дополняющее класс Application по умолчанию.
    /// </summary>
    sealed partial class App : Application
    {
        /// <summary>
        /// Инициализирует одноэлементный объект приложения. Это первая выполняемая строка разрабатываемого
        /// кода; поэтому она является логическим эквивалентом main() или WinMain().
        /// </summary>
        public App()
        {
            this.InitializeComponent();
            this.Suspending += OnSuspending;

            using (var db = new ModelContext())
            {
                db.Database.Migrate();
            }
        }
        /// <summary>
        /// Вызывается при обычном запуске приложения пользователем. Будут использоваться другие точки входа,
        /// например, если приложение запускается для открытия конкретного файла.
        /// </summary>
        /// <param name="e">Сведения о запросе и обработке запуска.</param>
        protected override void OnLaunched(LaunchActivatedEventArgs e)
        {
            Frame rootFrame = Window.Current.Content as Frame;

            // Не повторяйте инициализацию приложения, если в окне уже имеется содержимое,
            // только обеспечьте активность окна
            if (rootFrame == null)
            {
                // Создание фрейма, который станет контекстом навигации, и переход к первой странице
                rootFrame = new Frame();
            }
        }
    }
}
```

```

rootFrame.NavigationFailed += OnNavigationFailed;

if (e.PreviousExecutionState == ApplicationExecutionState.Terminated)
{
    //TODO: Загрузить состояние из ранее приостановленного приложения
}
// Размещение фрейма в текущем окне
Window.Current.Content = rootFrame;
}
if (e.PrelaunchActivated == false)
{
    if (rootFrame.Content == null)
    {
        // Если стек навигации не восстанавливается для перехода к первой странице,
        // настройка новой страницы путем передачи необходимой информации в качестве параметра
        // параметр
        rootFrame.Navigate(typeof(LoginPage), e.Arguments);
    }
    // Обеспечение активности текущего окна

    Window.Current.Activate();
    ExtendAcrylicIntoTitleBar();
}
}

private void ExtendAcrylicIntoTitleBar()
{
    var coreTitleBar = CoreApplication.GetCurrentView().TitleBar;
    coreTitleBar.ExtendViewIntoTitleBar = true;

    var viewTitleBar = ApplicationView.GetForCurrentView().TitleBar;
    viewTitleBar.ButtonBackgroundColor = Colors.Transparent;
    viewTitleBar.ButtonInactiveBackgroundColor = Colors.Transparent;
    viewTitleBar.ButtonForegroundColor = (Color)Resources["SystemBaseHighColor"];
}
/// <summary>
/// Вызывается в случае сбоя навигации на определенную страницу
/// </summary>
/// <param name="sender">Фрейм, для которого произошел сбой навигации</param>
/// <param name="e">Сведения о сбое навигации</param>

void OnNavigationFailed(object sender, NavigationFailedEventArgs e)
{
    throw new Exception("Failed to load Page " + e.SourcePageType.FullName);
}

/// <summary>
/// Вызывается при приостановке выполнения приложения. Состояние приложения сохраняется
/// без учета информации о том, будет ли оно завершено или возобновлено с неизменным
/// содержимым памяти.
/// </summary>
/// <param name="sender">Источник запроса приостановки.</param>
/// <param name="e">Сведения о запросе приостановки.</param>
private void OnSuspending(object sender, SuspendingEventArgs e)
{
    {
        var deferral = e.SuspendingOperation.GetDeferral();
        //TODO: Сохранить состояние приложения и остановить все фоновые операции
        deferral.Complete();
    }
}
}
}

```

```

using AppTaxationCard.Views;
using System;
using System.Collections.Generic;
using System.IO;
using System.Linq;
using System.Runtime.InteropServices.WindowsRuntime;
using Windows.ApplicationModel.Core;
using Windows.Foundation;
using Windows.Foundation.Collections;
using Windows.UI.Xaml;
using Windows.UI.Xaml.Controls;
using Windows.UI.Xaml.Controls.Primitives;
using Windows.UI.Xaml.Data;
using Windows.UI.Xaml.Input;
using Windows.UI.Xaml.Media;
using Windows.UI.Xaml.Navigation;

// Документацию по шаблону элемента "Пустая страница" см. по адресу
// https://go.microsoft.com/fwlink/?LinkId=402352&clcid=0x419

namespace AppTaxationCard
{
    /// <summary>
    /// Пустая страница, которую можно использовать саму по себе или для перехода внутри фрейма.
    /// </summary>
    public sealed partial class MainPage : Page
    {
        public MainPage()
        {
            this.InitializeComponent();
            this.Loaded += Page_Loaded;
        }

        private void Page_Loaded(object sender, RoutedEventArgs e)
        {
            contentFrame.Navigate(typeof(DashboardPage));
        }

        private void GoToMainPage()
        {
            if (Frame.CanGoBack)
                Frame.GoBack();
            else
                Frame.Navigate(typeof(DashboardPage));
        }

        private void NavigationViewControl_ItemInvoked(NavigationView sender, NavigationViewItemInvokedEventArgs args)
        {
            if (args.IsSettingsInvoked)
            {
                contentFrame.Navigate(typeof(SettingsPage));
            }
            else
            {
                TextBlock ItemContent = args.InvokedItem as TextBlock;
                if (ItemContent != null)
                {
                    switch (ItemContent.Tag)
                    {
                        case "Nav_Home":
                            contentFrame.Navigate(typeof(DashboardPage));
                            break;

                        case "Nav_ControlDatabase":
                            contentFrame.Navigate(typeof(ControlDataBasePage));
                            break;
                    }
                }
            }
        }
    }
}

```



```

using AppTaxationCard.Models;
using Microsoft.EntityFrameworkCore;
using System;
using System.Collections.Generic;
using System.IO;
using System.Linq;
using System.Runtime.InteropServices.WindowsRuntime;
using Windows.Foundation;
using Windows.Foundation.Collections;
using Windows.UI.Xaml;
using Windows.UI.Xaml.Controls;
using Windows.UI.Xaml.Controls.Primitives;
using Windows.UI.Xaml.Data;
using Windows.UI.Xaml.Input;
using Windows.UI.Xaml.Media;
using Windows.UI.Xaml.Navigation;

// Документацию по шаблону элемента "Пустая страница" см. по адресу https://go.microsoft.com/fwlink/?LinkId=234238

namespace AppTaxationCard.Views
{
    /// <summary>
    /// Пустая страница, которую можно использовать саму по себе или для перехода внутри фрейма.
    /// </summary>
    public sealed partial class DashboardPage : Page
    {

        public DashboardPage()
        {
            this.InitializeComponent();
            this.Loaded += Page_Loaded;
        }

        private void Page_Loading(FrameworkElement sender, object args)
        {

        }

        // загрузка страницы и загрузка данных из бд
        private void Page_Loaded(object sender, RoutedEventArgs e)
        {

            using (ModelContext db = new ModelContext())
            {
                newItemGridView.ItemsSource = db.Videls.Include(x => x.Kvartal).Where(c => c.UserId == ).ToList();
            }
        }

        //переход на страницу добавления
        private void Add_Click(object sender, RoutedEventArgs e)
        {
            Frame.Navigate(typeof(CardVidel));
        }

        // получение выделенного элемента
        private void newItemGridView_SelectionChanged(object sender, SelectionChangedEventArgs e)
        {
            if (newItemGridView.SelectedItem != null)
            {
                Videl company = newItemGridView.SelectedItem as Videl;

                if (company != null)
                    Frame.Navigate(typeof(VidelListPage), company.KvartalId);
            }
        }
    }
}

```

```

// кнопка удаления объекта
private void Delete_Click(object sender, RoutedEventArgs e)
{
    if (newItemGridView.SelectedItem != null)
    {
        // Kvartal company = newItemGridView.SelectedItem as Kvartal;
        if (company != null)
        {
            using (ModelContext db = new ModelContext())
            {
                // db.Kvartals.Remove(company);
                db.SaveChanges();
                // newItemGridView.ItemsSource = db.Kvartals.ToList();
            }
        }
    }
}

}

using System;
using System.Collections.Generic;
using System.IO;
using System.Linq;
using System.Runtime.InteropServices.WindowsRuntime;
using Windows.Foundation;
using Windows.Foundation.Collections;
using Windows.UI.Xaml;
using Windows.UI.Xaml.Controls;
using Windows.UI.Xaml.Controls.Primitives;
using Windows.UI.Xaml.Data;
using Windows.UI.Xaml.Input;
using Windows.UI.Xaml.Media;
using Windows.UI.Xaml.Navigation;
using AppTaxationCard.Models;
using Microsoft.EntityFrameworkCore;
using Windows.UI.Xaml.Media.Animation;
// Документацию по шаблону элемента "Пустая страница" см. по адресу https://go.microsoft.com/fwlink/?LinkId=234238

namespace AppTaxationCard.Views
{
    /// <summary>
    /// Пустая страница, которую можно использовать саму по себе или для перехода внутри фрейма.
    /// </summary>
    public sealed partial class LoginPage : Page
    {
        public LoginPage()
        {
            this.InitializeComponent();
        }

        private async void Login_Click(object sender, RoutedEventArgs e)
        {
            using (ModelContext db = new ModelContext())
            {
                var account = db.Accounts.FirstOrDefault(u => u.Username == usernameBox.Text);
                if (account != null)
                {
                    if (account.Password == passwordBox.Password)
                    {
                        db.SessionUsers.Add(new SessionUser { SId = account.Id, Username = usernameBox.Text, SessionTime =
DateTime.Now, });
                        db.SaveChanges();
                    }
                }
            }
        }
    }
}

```

```

// если логин и пароль верен переход на главный фрейм
Frame.Navigate(typeof(MainPage), new DrillInNavigationTransitionInfo());
    }
    else
    {
        ContentDialog deleteFileDialog = new ContentDialog()
        {
            Title = "Сообщение об ошибке",
            Content = "Не верный пароль, желаете продолжить? ",
            PrimaryButtonText = "ОК",
            SecondaryButtonText = "Отмена"
        };
        ContentDialogResult result = await deleteFileDialog.ShowAsync();
    }
}
else
{
    ContentDialog deleteFileDialog = new ContentDialog()
    {
        Title = "Сообщение об ошибке",
        Content = "Пользователь не найден ",
        PrimaryButtonText = "ОК"
    };
    ContentDialogResult result = await deleteFileDialog.ShowAsync();
}
}
}
private void PhonesList_SelectionChanged(object sender, SelectionChangedEventArgs e)
{
    //Account selectedPhone = (Account)phonesList.SelectedItem;
    //await new Windows.UI.Popups.MessageDialog($"Выбран {selectedPhone.username}").ShowAsync();
}
private void Page_Loaded(object sender, RoutedEventArgs e)
{
    //// добавление нового объекта
    //using (ModelContext db = new ModelContext())
    //{

        // db.Accounts.Add(new Account { Username = "Ilya", Password = "121790", CreateTime = DateTime.Now, });
        // db.Accounts.Add(new Account { Username = "Vova", Password = "121790", CreateTime = DateTime.Now, });
        // db.Accounts.Add(new Account { Username = "Den", Password = "121790", CreateTime = DateTime.Now, });
        // db.Accounts.Add(new Account { Username = "Evgen", Password = "121790", CreateTime = DateTime.Now, });
        // db.SaveChanges();


        // db.Videls.Add(new Videl { NumKvartal = 1, Area = 200, AccountId = 1, Bonitet = "C", });
        // db.Videls.Add(new Videl { NumKvartal = 2, Area = 300, AccountId = 1, Bonitet = "C", });
        // db.Videls.Add(new Videl { NumKvartal = 1, Area = 100, AccountId = 2, Bonitet = "C", });
        // db.Videls.Add(new Videl { NumKvartal = 1, Area = 150, AccountId = 3, Bonitet = "C", });
        // db.SaveChanges();

        // phonesList.ItemsSource = db.Accounts.ToList();
    //} } }
}

```

Приложение Б (справочное) Формы выходных документов

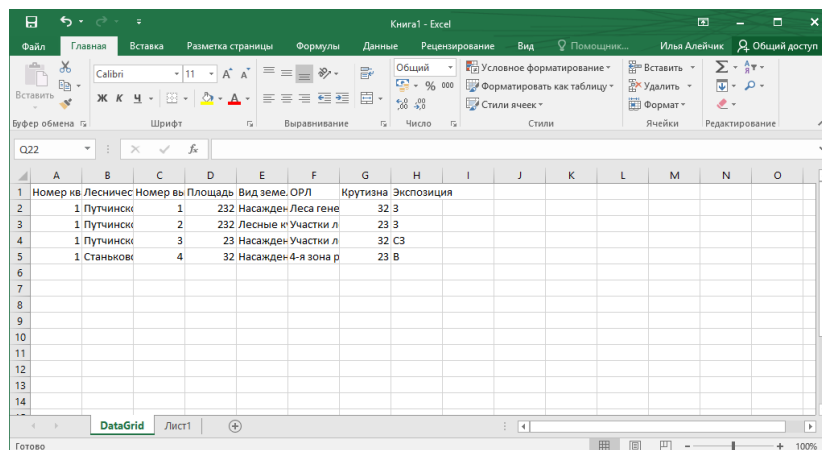
Отчет о выделе



Макет 1

Дата и время создания документа:	24.05.2019 18:42:00
Номер квартала:	2
Лесничество:	Держинское
Номер выдела:	2
Площадь:	21
Вид земли:	Квартальные просеки
ОРЛ:	Участки леса в поймах рек
Крутизна:	23
Экспозиция:	В

Рисунок Б.1 –Карточка таксации



Номер кв	Лесничес	Номер вы	Площадь	Вид земе.	ОРЛ	Крутизна	Экспозиция
1	Путчинск	1	232	Насажден	Леса гене	32	3
2	Путчинск	2	232	Лесные к	Участки л	23	3
3	Путчинск	3	23	Насажден	Участки л	32	С3
4	Станьков	4	32	Насажден	4-я зона р	23	В

Рисунок Б.2 – Вывод списка выделов

```
{
  "type": "AppTaxationCard.Models.Videl, AppTaxationCard",
  "id": 5,
  "NumVidel": 1,
  "Area": 1,
  "Krut": 32,
  "TypeErosionId": 1,
  "TypeErosion": null,
  "DegreeErosionId": 2,
  "DegreeErosion": null,
  "ExpositionSlopeId": 2,
  "ExpositionSlope": null,
  "TypeOrlId": 3,
  "TypeOrl": null,
  "TypeEarthId": 3,
  "TypeEarth": null,
  "ForestryId": 3,
  "Forestry": null,
  "CreateDateVidel": "2019-05-29T17:57:21.7257028",
  "KvartalId": 4,
  "Kvartal": null,
  "AccountId": 1,
  "Account": null
}
```

Рисунок Б.3 – Вывод карточки таксации в JSON