



Министерство науки и высшего образования Российской Федерации
федеральное государственное автономное образовательное учреждение высшего образования
«Национальный исследовательский Томский политехнический университет» (ТПУ)

Школа Инженерная школа информационных технологий и робототехники
Направление подготовки 09.03.04 Программная инженерия
ООП/ОПОП Разработка программно-информационных систем
Отделение школы (НОЦ) ОИТ

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА БАКАЛАВРА

Тема работы
Создание редактора стилей для ПО Geoserver

УДК 004.415.2:004.439

Обучающийся

Группа	ФИО	Подпись	Дата
8К93	Жабин Ефим Михайлович		

Руководитель ВКР

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОИТ ИШИТР	Друки Алексей Алексеевич	к.т.н.		

КОНСУЛЬТАНТЫ ПО РАЗДЕЛАМ:

По разделу «Финансовый менеджмент»

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Профессор ОСГН	Гасанов Магеррам Али оглы	д.э.н.		

По разделу «Социальная ответственность»

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Старший преподаватель ООД ШБИП	Мезенцева Ирина Леонидовна			

ДОПУСТИТЬ К ЗАЩИТЕ:

Руководитель ООП/ОПОП, должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОИТ ИШИТР	Чердынцев Евгений Сергеевич	к.т.н		

ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ ОБУЧЕНИЯ ООП

Код компетенции	Наименование компетенции
УК(У)-1	<i>Способен осуществлять поиск, критический анализ и синтез информации, применять системный подход для решения поставленных задач.</i>
УК(У)-2	<i>Способен определять круг задач в рамках поставленной цели и выбирать оптимальные способы их решения, исходя из действующих правовых норм, имеющихся ресурсов и ограничений.</i>
УК(У)-3	<i>Способен осуществлять социальное взаимодействие и реализовывать свою роль в команде.</i>
УК(У)-4	<i>Способен осуществлять деловую коммуникацию в устной и письменной формах на государственном языке Российской Федерации и иностранном(-ых) языке(-ах).</i>
УК(У)-5	<i>Способен воспринимать межкультурное разнообразие общества в социально-историческом, этическом и философском контекстах</i>
УК(У)-6	<i>Способен управлять своим временем, выстраивать и реализовывать траекторию саморазвития на основе принципов образования в течение всей жизни.</i>
ОПК(У)-1	<i>Способен применять естественнонаучные и общинженерные знания, методы математического анализа и моделирования, теоретического и экспериментального исследования в профессиональной деятельности.</i>
ОПК(У)-2	<i>Способен использовать современные информационные технологии и программные средства, в том числе отечественного производства, при решении задач профессиональной деятельности.</i>
ОПК(У)-3	<i>Способен решать стандартные задачи профессиональной деятельности на основе информационной и библиографической культуры с применением информационно-коммуникационных технологий и с учетом основных требований информационной безопасности.</i>
ОПК(У)-4	<i>Способен участвовать в разработке стандартов, норм и правил, а также технической документации, связанной с профессиональной деятельностью.</i>
ОПК(У)-5	<i>Способен устанавливать программное и аппаратное</i>

	<i>обеспечение для информационных и автоматизированных систем.</i>
<i>ПК(У)-1</i>	<i>Способен выполнять интеграцию программных модулей и компонент.</i>
<i>ПК(У)-2</i>	<i>Владение навыками моделирования, анализа и использования формальных методов конструирования программного обеспечения.</i>



Министерство науки и высшего образования Российской Федерации
федеральное государственное автономное образовательное учреждение высшего образования
«Национальный исследовательский Томский политехнический университет» (ТПУ)

Школа Инженерная школа информационных технологий и робототехники
Направление подготовки 09.03.04 Программная инженерия
Отделение информационных технологий

УТВЕРЖДАЮ:
Руководитель ООП/ОПОП
_____ Чердынцев Е.С.
(Подпись) (Дата) (ФИО)

**ЗАДАНИЕ
на выполнение выпускной квалификационной работы**

Обучающийся:

Группа	ФИО
8К93	Жабин Ефим Михайлович

Тема работы:

Создание редактора стилей для ПО Geoserver
Утверждена приказом директора (дата, номер)

Срок сдачи обучающимся выполненной работы:

--

ТЕХНИЧЕСКОЕ ЗАДАНИЕ:

Исходные данные к работе	Объектом исследования является настольное приложение с графическим пользовательским интерфейсом, позволяющее редактировать файлы с разметкой XML, содержащие внутри себя пространство имен SLD, описывающие визуальное оформление геометрии.
Перечень разделов пояснительной записки подлежащих исследованию, проектированию и разработке	<ol style="list-style-type: none"> 1. Проектирование настольного приложения 2. Программная реализация настольного приложения 3. Финансовый менеджмент, ресурсоэффективность и ресурсосбережение 4. Социальная ответственность
Перечень графического материала	<ol style="list-style-type: none"> 1. Рисунки, демонстрирующие результаты работы
Консультанты по разделам выпускной квалификационной работы	
Раздел	Консультант
Финансовый менеджмент,	Гасанов Магеррам Али оглы

ресурсоэффективность, ресурсосбережение	
Социальная ответственность	Мезенцева Ирина Леонидовна

Дата выдачи задания на выполнение выпускной квалификационной работы по линейному графику	
---	--

Задание выдал руководитель:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОИТ ИШИТР	Друки Алексей Алексеевич	к.т.н.		

Задание принял к исполнению обучающийся:

Группа	ФИО	Подпись	Дата
8К93	Жабин Ефим Михайлович		



Министерство науки и высшего образования Российской Федерации
федеральное государственное автономное образовательное учреждение высшего образования
«Национальный исследовательский Томский политехнический университет» (ТПУ)

Школа – Инженерная школа информационных технологий и робототехники
Направление подготовки – 09.03.04 «Программная инженерия»
Уровень образования – Бакалавриат

**КАЛЕНДАРНЫЙ РЕЙТИНГ-ПЛАН
выполнения выпускной квалификационной работы**

Обучающийся:

Группа	ФИО
8К93	Жабин Ефим Михайлович

Тема работы:

Создание редактора стилей для ПО Geoserver
--

Срок сдачи обучающимся выполненной работы:

Дата контроля	Название раздела (модуля) / вид работы (исследования)	Максимальный балл раздела (модуля)
05.03.2023	Постановка задачи	5
10.03.2023	Проектирование настольного приложения	25
15.04.2023	Программная реализация настольного приложения	40
10.05.2023	Финансовый менеджмент, ресурсоэффективность и ресурсосбережение	15
20.05.2023	Социальная ответственность	15

СОСТАВИЛ:

Руководитель ВКР

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОИТ ИШИТР	Друки Алексей Алексеевич	к.т.н.		

СОГЛАСОВАНО:

Руководитель ООП/ОПОП

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОИТ ИШИТР	Чердынцев Евгений Сергеевич	к.т.н.		

Обучающийся

Группа	ФИО	Подпись	Дата
8К93	Жабин Ефим Михайлович		

Оглавление

Реферат	9
Обозначения и сокращения	10
Введение.....	11
1. Постановка задачи	12
1.1. Описание предметной области	12
1.2. Анализ проблемы	12
1.3. Анализ существующих решений	12
1.4. Вывод.....	13
2. Проектирование настольного приложения.....	14
2.1. Функциональные требования.....	14
2.2. Нефункциональные требования.....	14
2.3. Техническая документация	15
3. Реализация программного обеспечения	17
3.1. Разработка настольного приложения	17
3.1.1. Написание библиотеки для конвертации файла в объект среды NET	17
3.1.2. Реализация основного функционала настольного приложения	22
3.1.3. Задание внешнего вида GUI приложения, локализация и добавление подсказок	29
3.1.4. Добавление предпросмотра	33
3.1.5. Поддержания работы со стилями ПО Geoserver.....	34
3.2. Исследование компиляции исходного кода настольного приложения, в Unity для получения WEB приложения.....	39
3.2.1. Подготовка решения с исходным кодом, для работы в Unity	41
3.2.2. Написание библиотеки для преобразования моделей с одинаковыми свойствами.....	47
3.2.3. Результаты исследования	48
4. Финансовый менеджмент, ресурсоэффективность и ресурсосбережение.....	52
4.1. Введение.....	54
4.2. Оценка коммерческого потенциала и перспективности проведения исследований с позиции ресурсоэффективности и ресурсосбережения.	54
4.2.1. Анализ конкурентных технических решений	54
4.2.2. SWOT-анализ	55
4.3. Планирование научно-исследовательских работ	58
4.3.1. Структура работ в рамках научного исследования.....	58

4.3.2.	Определение трудоемкости выполнения работ и разработка графика проведения.....	59
4.3.3.	Бюджет научно-технического исследовани.....	61
4.3.4.	Расчет материальных затрат научно-технического исследования	62
4.3.5.	Расчет амортизации специального оборудования.....	62
4.3.6.	Основная заработная плата исполнителей темы.....	63
4.3.7.	Дополнительная заработная плата исполнителей темы.....	64
4.3.8.	Отчисления во внебюджетные фонды (страховые отчисления)	65
4.3.9.	Накладные расходы.....	65
4.3.10.	Бюджетная стоимость НИР	66
4.4.	Определение ресурсной (ресурсосберегающей), финансовой, бюджетной, социальной и экономической эффективности исследования.....	66
4.5.	Вывод.....	69
5.	Социальная ответственность	70
5.1.	Введение.....	72
5.2.	Правовые и организационные вопросы обеспечения безопасности.....	72
5.3.	Производственная безопасность.....	74
5.3.1.	Длительное сосредоточенное наблюдение	75
5.3.2.	Отсутствие или недостаток необходимого искусственного освещения	75
5.3.3.	Повышенный уровень шума на рабочем месте	76
5.3.4.	Нагрузка на зрительный аппарат.....	78
5.3.5.	Монотонный режим работы.....	79
5.3.6.	Правила по охране труда при эксплуатации электроустановок.	79
5.4.	Экологическая безопасность	80
5.5.	Безопасность в чрезвычайных ситуациях.....	81
5.6.	Вывод.....	82
	Заключение.....	83
	Список используемых источников.....	84
	Приложение А.....	86

Реферат

Выпускная квалификационная работа содержит 86 страниц, 60 рисунков, 20 таблиц, 20 источников.

Ключевые слова: настольное приложение, Geoserver, карты, редактор стилей, SLD.

Объектом исследования является настольное приложение.

Предмет исследования – настольное приложение, являющиеся редактором стилей, с графическим пользовательским интерфейсом.

Цель работы – разработать настольное приложение для редактирования XML документов стилей, содержащих внутри себя структуру SLD, через дружелюбный графический пользовательский интерфейс, с последующим использованием стилей в ПО Geoserver.

В процессе работы было проведено исследование: пере использования кода, с целью добавления кроссплатформенность для компиляции приложения, как WEB приложения.

Степень внедрения: внедрено на предприятии.

Область применения: понижения уровня требуемых знаний для работы со стилями и повышение эффективности работы инженеров, работающих со стилями.

Обозначения и сокращения

ИС – информационная система

ПО – программное обеспечение.

API – Application Programming Interface – программный интерфейс приложения, интерфейс прикладного программирования.

HTTP – HyperText Transport Protocol – протокол передачи гипертекста.

URI – Universal Resource Identifier – универсальный идентификатор ресурса.

WEB – World Wide Web - всемирная система публичных веб-страниц в сети Интернет.

WEB приложение – программное обеспечение, запускаемое в веб-браузере.

XML – Xtensible Markup Language – расширяемый язык разметки.

Geoserver - программное обеспечение с открытым исходным кодом, написанное на Java, предоставляющее возможность администрирования и публикации геоданных на сервере.

C# - современный объектно-ориентированный и типобезопасный язык программирования.

NET - это модульная платформа для разработки программного обеспечения.

WPF – Windows Presentation Foundation – система для построения клиентских приложений Windows с визуально привлекательными возможностями взаимодействия с пользователем, графическая подсистема в составе .NET Framework, использующая язык XAML.

Unity – кроссплатформенная среда разработки компьютерных игр, разработанная американской компанией Unity Technologies.

SLD – Styled Layer Descriptor – спецификация пространств имен используемая, в файлах стилей для параметризации внешнего вида геометрии.

GUI – Graphical User Interface - графический интерфейс пользователя.

Noesis GUI – плагин для использования файлов XAML в ПО Unity.

Javascript - мультипарадигменный язык программирования.

Web Assembly – технология, предоставляющая новый тип кода, который можно запускать в современных веб-браузерах, что обеспечивает новые функции и значительное повышение производительности.

XAML - eXtensible Application Markup Language - основанный на XML язык разметки для декларативного программирования приложений, разработанный Microsoft, используемый в WPF для описания визуального оформления представлений.

Введение

В настоящее время, большинство промышленных предприятий, работающих в нефтяной, газовой и других подобных сырьевых индустриях, для работы с геоданными и графической отрисовки карт используют компьютер и ПО для создания информационных карт.

Для создания информационных карт и работы с геоданными используются общепринятые стандарты [18]. Стандарты используются для разработки ПО, используемого инженерами, в построении карт. Одним из популярных ПО с открытым исходным кодом и поддерживаемым сообществом разработчиков является – Geoserver.

Geoserver – позволяет в соответствии с указанными стандартами, работать с картографической информацией и использовать ее для отрисовки карт местностей. Данное ПО имеет ряд различного функционала, но его основным набор функций является: хранение геоданных, параметризация визуального стиля отрисовки геометрии, возможность указания порядка отображения слоев геометрии.

В данной работе, рассматривается функционал параметризации визуального стиля отрисовки геометрии. Имеющийся функционал используют XML файлы являющиеся стилями (параметрами) для отрисовки выбранной геометрии. Заполняемые XML файлы внутри себя содержат особую структуру, а именно пространство имен SLD. Заполнение стилей происходит вручную с помощью набора на клавиатуры необходимой текстовой разметки SLD.

С рассмотренным выше ПО Geoserver работаю инженеры, не являющиеся разработчиками ИС или программистами, по причине чего заполнение данных файлов является: затруднительным, неэффективным. Таким образом, в настоящее время, необходимо решения для повышения эффективности работы и понижения требуемого уровня знаний пространств имен инженерами.

1. Постановка задачи

1.1. Описание предметной области

Документ стиля представляет собой XML документ, с пространством имен SLD [18]. Стилль задает визуальный способ отображения геометрии, с помощью: правил, фильтра, символа:

- Правило – содержит внутри себя фильтр и символы, для отображения геометрии; стилль в свою очередь содержит список правил.
- Фильтр – хранит условия для применения параметров визуального оформления геометрии.
- Символ – является способом описания определенного вида геометрии; в правиле может содержаться все виды символов.

В стиллях имеются виды символов для описания разной геометрии: точка, линия, полигон, текст.

Стили применяются определенном слое геометрии, для которого будут использоваться указанные в стиллях параметры отображения геоданных.

Геоданные хранятся в отдельных базах данных, которые подключаются к ПО Geoserver.

1.2. Анализ проблемы

Трудность заполнения данного файла, отображается в следующих пунктах:

- 1) Текстовая разметка XML;
- 2) Знание наименований элементов и атрибутов в пространстве имен SLD;
- 3) Ручное создание, редактирование и удаление стилей в ПО Geoserver;

Из указанных выше пунктов, во время работы инженера проявляется неэффективное использование временных ресурсов сотрудника, или необходимость содержания в штате предприятия разработчиков ИС, или взаимодействия с компаниями, предоставляющими услуги по разработке ИС.

1.3. Анализ существующих решений

На момент написания работы существует, два программных продукта чем-то решающих рассмотренные ранее пункты.

QGIS - геоинформационная система, состоящая из настольной и серверной части: QGIS Desktop – настольное приложение ГИС для создания, редактирования,

визуализации, анализа и публикации геопространственной информации. В данной системе имеется инструментарий, который позволяет создать стили по готовым шаблонам, параметризуя их несколькими параметрами.

Atlas Styler - удобное для пользователя настольное приложение для стилизации геоданных. Результирующие стили могут быть сохранены в виде файлов SLD / SE, соответствующих стандартам OGC. Является прямым зарубежным конкурентом, со схожими необходимыми пользовательскими возможностями. В данном ПО нет возможности: работы напрямую со стилями в ПО Geoserver и полноценной работы с фильтрами, для корректной отрисовки слоев.

1.4. Вывод

Исходя из полученной информации можно сделать вывод, что на рынке решения, но данные решения не закрывают большую часть пунктов для эффективной работы инженера при построении карт по геоданным.

2. Проектирование настольного приложения

2.1. Функциональные требования

Основные функциональные требования необходимые инженеры представлены, на диаграмме вариантов.



Рисунок 1. Диаграмма вариантов использования приложения.

Если обобщить и упростить функциональные требования к приложению, ПО должно иметь возможность:

- работать с файлами, хранящими внутри себя разметку XML со структурой SLD: открытие, сохранение;
- необходим функционал графического предпросмотра визуального оформления геометрии с использованием редактируемого стиля;
- пользователь может сохранять, загружать, удалять стили в ПО Geoserver;

2.2. Нефункциональные требования

Требования, которые должны реализоваться в работе приложения:

- в редакторе должна быть видна и разделена рабочая область для редактирования: правил, фильтра, символов;
- предпросмотр работает при изменении и выборе редактируемого правила;
- GUI приложения имеет локализацию на русском языке;
- GUI приложения содержит подсказки на элементах отображаемых, в редакторе;
- визуальный стиль приложения является темным;

Необходимо провести исследования для возможности пере использования кода. Исследование должно заключаться, в проверке возможности компиляции и трансляции кода для получения WEB приложения, посредством Unity.

Стек технологий, используемый для разработки приложения: C#, WPF, Geoserver, Unity, NoesisGUI.

2.3. Техническая документация

Разработка приложения велась с использованием языка программирования C#. В данном языке программирование имеются понятия сборки, иначе модули приложения. Разработанное приложение имеет следующую структуру модулей, отображенную на диаграмме ниже.

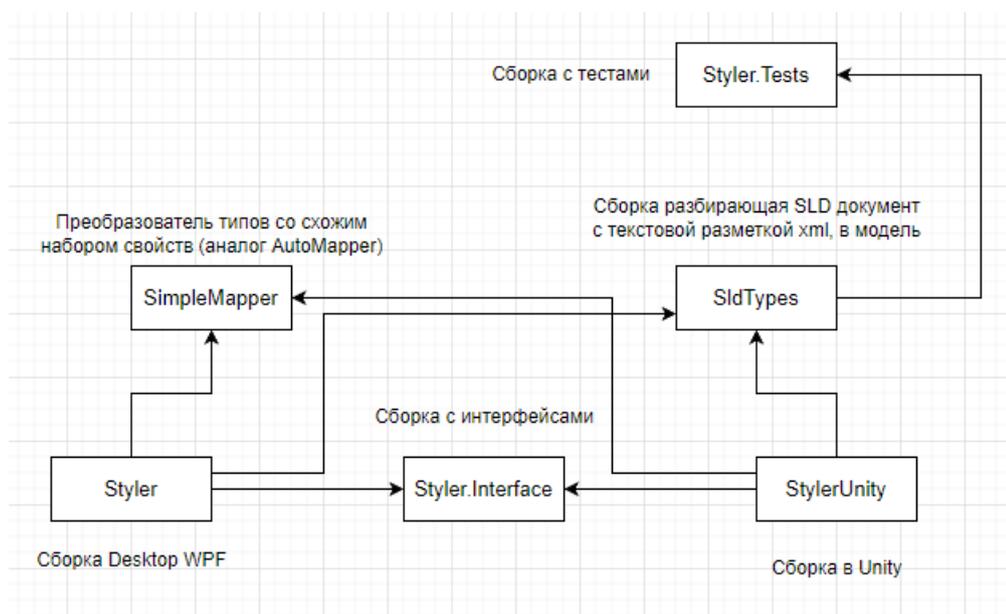


Рисунок 2. Диаграмма модулей приложения.

- SldTypes – является библиотекой для сериализации и десериализации текстовой разметки XML в объект используемой средой NET.
- SimpleMapper – необходим для минимизации кода по преобразованию объектов в NET со схожими свойствами.
- Styler – является проектом для компиляции запускаемого настольного приложения.
- StylerUnity – является проектом для разработки на платформе Unity.

Идея использования Unity платформы и ее компиляторов, для получения WEB приложения, исходит из следующих причин:

- 1) В Unity платформе используется разработчиком язык программирования C#;
- 2) Для поддержания XAML разметки описывающих представления редактора, для Unity имеется плагин Noesis GUI;

3) Unity может компилировать приложения под несколько платформ, в том числе под WEB платформу.

Если использовать Unity для компиляции под WEB платформу, можно получить следующую последовательность работы компиляторов.

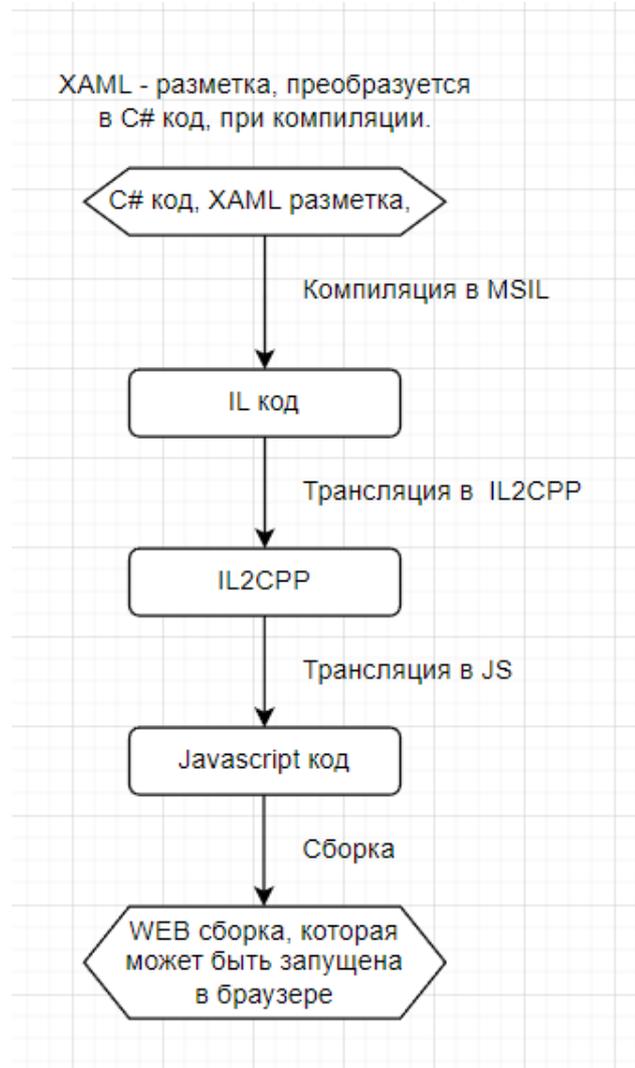


Рисунок 3. Последовательность операций выполняемых для компиляции WEB приложения.

Исследование заключается в проверке возможности получения, работающего WEB приложения посредством пере использования кода, ранее задействованного при разработке настольного приложения.

3. Реализация программного обеспечения

3.1. Разработка настольного приложения

3.1.1. Написание библиотеки для конвертации файла в объект среды NET

Документ стиля представляет собой XML документ, для того чтобы взаимодействовать с его данными необходимо десериализовать документ (привести данные документа в необходимый вид). Для операции десериализации XML документа в модель данных приложения, на языке C# уже имеются готовые библиотеки, но все имеющиеся библиотеки, не могут удовлетворить все возможности заполнения XML документа, по спецификации SLD, следствием из чего разработан сериализатор для преобразования данных, в обе стороны.

Сериализатор, представлен отдельной сборкой, которая используется для отображения документа в модель.

При разработке стало известно, что SLD структура может поддерживать несколько пространств имен. Для сериализатора, было использовано популярное и более задокументированное пространство; по итогу их различие в большинстве своем заключались в префиксах к тегам.

```
public const string SE = "http://www.opengis.net/se";
```

Рисунок 4. Применяемое при разработке пространство имен.

Для большинства тегов используемых в документе были отображены сущности-классы в сборке, такая иерархия необходима для того чтобы воспользоваться стандартным XML сериализатором имеющимся в C#.

С помощью атрибутов, используемых в языке C# можно указать, является вложенным элементом в XML документе, а что является атрибутом.

Важно, не путать смысл слова атрибут в разных контекстах, атрибут в XML документе, это простое свойство сущности, которое имеет простую структура данных, к примеру, текстовую; а атрибут в языке C# подразумевает, настройки компиляции и дополнительные метаданные поля, объекта.

Для того, чтобы понять, что является атрибутом в XML документе, а что является вложенным элементом рассмотрим пример XML документа.

Пример XML-документа

- `<?xml version="1.0" encoding="UTF-8"?>`
`<bookstore>`
 `<book>`
 `<title lang="en">Harry Potter</title>`
 `<author>J K. Rowling</author>`
 `<year>2005</year>`
 `<price>29.99</price>`
 `</book>`
 `...`
`</bookstore>`

Рисунок 5. Пример XML документа.

В данном примере «lang» является атрибутом, а «price» вложенным элементом.

В C# по умолчанию присутствует класс который позволяет сериализовать объекты по их типу.

```
public string Serialize(StyledLayerDescriptor sld, string filePath = "Template.sld")
{
    if (sld == null)
        throw new ArgumentNullException(nameof(sld));

    // Определяем пространства для Xml документа.
    var xmlSerializerNamespaces = new XmlSerializerNamespaces();
    xmlSerializerNamespaces.Add("ogc", Prefix.OGC);
    xmlSerializerNamespaces.Add("xsi", "http://www.w3.org/2001/XMLSchema-instance");
    xmlSerializerNamespaces.Add("xlink", "http://www.w3.org/1999/xlink");
    xmlSerializerNamespaces.Add(
        "schemaLocation",
        "http://www.opengis.net/sld http://schemas.opengis.net/sld/1.1.0/StyledLayerDescriptor.xsd");
    xmlSerializerNamespaces.Add("se", Prefix.SE);
    xmlSerializerNamespaces.Add("xsd", "http://www.w3.org/2001/XMLSchema");

    var xmlWriterSettings = new XmlWriterSettings
    {
        Encoding = Encoding.UTF8,
        Indent = true
    };

    using (var fileStream = File.Create(filePath))
    using (var writer = XmlWriter.Create(fileStream, xmlWriterSettings))
    {
        // Десериализуем структуру.
        var serializer = new XmlSerializer(typeof(StyledLayerDescriptor), "http://www.opengis.net/sld");
        serializer.Serialize(writer, sld, xmlSerializerNamespaces);
    }

    return filePath;
}
```

Рисунок 6. Пример использования «XmlSerializer» для сериализации структуры SLD.

```

/// <summary>
/// Возвращает новый SLD объект.
/// </summary>
/// <param name="pathToFile">Путь до документа со стилями. (only sld)</param>
/// <returns>Вернется новый объект SLD.</returns>
Ссылка: 2
public StyledLayerDescriptor Deserialize(string pathToFile)
{
    using (var fileStream = File.Open(pathToFile, FileMode.Open))
    {
        // Сериализуем структуру.
        var serializer = new XmlSerializer(typeof(StyledLayerDescriptor), "http://www.opengis.net/sld");
        var styledLayerDescriptor = (StyledLayerDescriptor)serializer.Deserialize(fileStream);
        styledLayerDescriptor.FilePath = pathToFile;
        return styledLayerDescriptor;
    }
}

```

Рисунок 7. Пример использования «XmlSerializer» для десериализации структуры SLD.

Настроить сериализацию, в сериализаторе можно несколькими способами:

- использовать C# атрибуты,
- наследовать интерфейс IXmlSerializable,

Примеры:

Рассмотрим 1 случай, с применением, атрибута «XmlElement» можно указать что будет являться, вложенным элементом.

```

/// <summary>
/// FeatureTypeStyle содержит в себе правила стиля, направленные на один единственный слой.
/// </summary>
/// <remarks>Нельзя выбрать тип объекта, к которому должен применяться стиль.</remarks>
[Serializable]
Ссылка: 4
public class FeatureTypeStyle
{
    /// <summary>
    /// FeatureTypeStyle содержит в себе правила стиля, направленные на один единственный слой.
    /// </summary>
    /// <remarks>Нельзя выбрать тип объекта, к которому должен применяться стиль.</remarks>
    Ссылка: 2
    public FeatureTypeStyle()
    {
        Rules = new List<Rule>();
    }

    /// <summary>
    /// Список, а не один элемент, т.к. UserStyle может содержать в себе несколько FeatureTypeStyle.
    /// </summary>
    /// <remarks>При всем при этом, в наших файлах стилей, больше одного FeatureTypeStyle объекта, не используется.</remarks>
    [XmlElement("Rule")]
    Ссылка: 5
    public List<Rule> Rules { get; }
}

```

Рисунок 8. Пример использования «XmlElement» атрибута.

Пример, с помощью атрибута «XmlAttribute», можно указать является ли свойство XML атрибутом, а не вложенным элементом/ Как можно заметить также можно указать, название атрибута и пространство имен; тип значения берется исходя из типа поля или свойства.

```

/// <summary>
/// Онлайн ресурс, позволяющий указать ссылку.
/// </summary>
[Serializable]

public class OnlineResources
{
    /// <inheritdoc cref="OnlineResources" />

    public OnlineResources()
    {
        XLinkType = "simple";
    }

    /// <summary>
    /// Ссылка на файл изображения.
    /// </summary>
    [XmlAttribute("href", Namespace = "http://www.w3.org/1999/xlink")]

    public string XLinkHref { get; set; }

    /// <summary>
    /// Тип соединения.
    /// </summary>
    [XmlAttribute("type", Namespace = "http://www.w3.org/1999/xlink")]

    public string XLinkType { get; set; }
}

```

Рисунок 9. Пример использования «XmlAttribute» атрибута.

Атрибутов, для указания настройки сериализатора множество, все их рассматривать не будем.

Рассмотрим 2 случая, с наследованием интерфейса «IXmlSerializable».

```

namespace System.Xml.Serialization
{
    using ...

    public interface IXmlSerializable
    {
        XmlSchema GetSchema();
        void ReadXml(XmlReader reader);
        void WriteXml(XmlWriter writer);
    }
}

```

Рисунок 10. Интерфейс «IXmlSerializable».

Интерфейс «IXmlSerializable» является базовым из пространства имен System.Xml.Serialization; если наследовать этот интерфейс для класса, то при сериализации класса будут использоваться не атрибуты компиляции, а будут вызваны его методы «ReadXml» и «WriteXml». Такой подход несомненно удобен, когда мы хотим полностью контролировать процесс сериализации объекта.

```

/// <summary>
/// Цветовой фон для надписи.
/// </summary>
Ссылка: 6
public class Halo : IXmlSerializable
{
    // ...

/// <summary>
/// Считываем все теги из текущего элемента, для Halo.
/// </summary>
Ссылка: 0
public void ReadXml(XmlReader reader)
{
    Injector.Get<ILogger>().Info($"Halo начало-считывания - {reader.LocalName} - {reader.NodeType}");
    reader.ReadStartElement();

    // С помощью цикла считываем все теги, которые нам нужны.
    while (reader.NodeType != XmlNodeType.EndElement)
    {
        switch (reader.LocalName)
        {
            case Tags.FILL:
                // Помещаем параметры в тег Fill.
                reader.ReadStartElement();
                SvgParameterUtils.ReadListXml(reader, Fill);
                reader.ReadEndElement();
                break;
            case Tags.RADIUS:
                if (float.TryParse(
                    reader.ReadElementString(),
                    NumberStyles.Float,
                    CultureInfo.InvariantCulture,
                    out var radius))
                    Radius = radius;
                break;
            default:
                reader.Skip();
                break;
        }
    }

    reader.ReadEndElement();
}

/// <summary>
/// Записываем с помощью writer, всю информацию о Halo.
/// </summary>
Ссылка: 0
public void WriteXml(XmlWriter writer)
{
    writer.WriteElementString(Tags.RADIUS, Prefix.SE, Radius.ToString(CultureInfo.InvariantCulture));

    writer.WriteStartElement(Tags.FILL, Prefix.SE);
    SvgParameterUtils.WriteListXml(writer, Fill);
    writer.WriteEndElement();
}
}

```

Рисунок 11. Пример реализации интерфейса «IXmlSerializable».

По итогу получилась следующая иерархия классов:

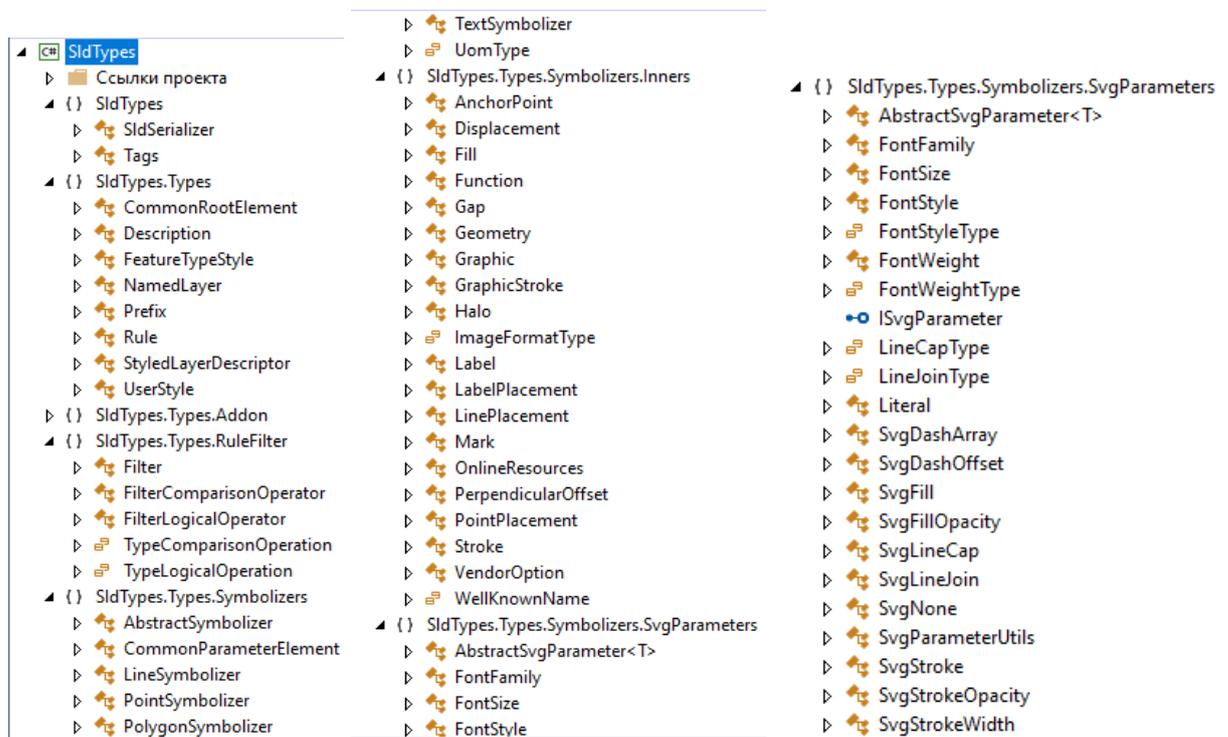


Рисунок 12. Классы сборки сериализатора.

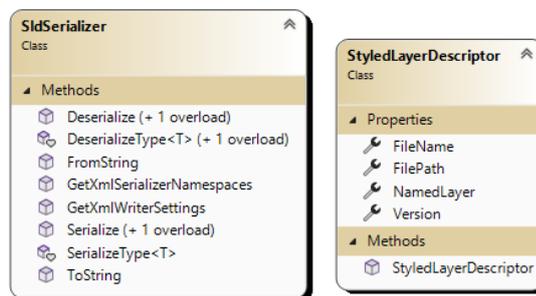


Рисунок 13. Главные классы в реализованной сборке для конвертации стилей, в модель.

В иерархии классов доступен класс сериализатор для SLD типов, реализации которого мы рассмотрели ранее, с помощью него можно преобразовывать текстовые файлы с разметкой XML в объект, и обратно.

3.1.2. Реализация основного функционала настольного приложения

В технологии WPF, приложения разрабатываются с уже сложившимися правилами и паттернами разработки. Сама технология работает следующим образом:

В C# был введен функционал «частичных классов», когда можно объявить один тот же класс в двух файлах, ключевым словом «partial», и при сборке данный класс соберется

воедино и полученный программа будет работать с единым классом. Благодаря такой языковой возможности и имеется технология WPF, в которой мы можем создать окно от наследовавшись от базового класса «Window», и реализовать его в классе «SomeWindow.cs» логику путем написания кода, и в XAML документе «SomeWindow.xaml» отредактировать представление.

```

namespace Styler
{
    using Styler.ViewModels.Displays;

    /// <summary>
    /// Логика взаимодействия для MainUserControl.xaml
    /// </summary>

    public partial class MainUserControl
    {
        //Check

        public MainUserControl()
        {
            InitializeComponent();
            DataContext = new EditorVM();
        }
    }
}

```

Рисунок 14. Пример реализации «Window» класса.

```

1 <Page x:Class="Styler.MainUserControl"
2   xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
3   xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
4   xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
5   xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
6   mc:Ignorable="d"
7   d:DesignHeight="720" d:DesignWidth="1280"
8   FontSize="13">
9
10  <Grid Background="{DynamicResource Background}">
11    <Grid.RowDefinitions>
12      <RowDefinition Height="Auto" />
13      <RowDefinition Height="*" />
14    </Grid.RowDefinitions>
15    <!--Buttons-->
16    <StackPanel Orientation="Horizontal" Grid.Column="0" Margin="0 2 0 0"...>
17      <!--Workspace-->
18    <Grid Grid.Row="1" Background="{DynamicResource WorkspaceBackground}">
19      <Grid.RowDefinitions>
20        <RowDefinition Height="Auto" />
21        <RowDefinition Height="*" />
22      </Grid.RowDefinitions>
23      <ListBox ItemsSource="{Binding SldFileList}" Background="Transparent"
24        ItemContainerStyle="{DynamicResource SldFileItemContainer}"
25        Style="{DynamicResource SldFileList}"
26        Padding="20 7 0 0"
27        SelectedItem="{Binding CurrentEditSldFile}" />
28      <!--Editor-->
29      <Grid Grid.Row="1" DataContext="{Binding CurrentEditSldFile}"...>
30      </Grid>
31    </Grid>
32    <!--PreviewDisplay-->
33    <ContentControl Grid.Row="1" Content="{Binding}" Margin="0 5 0 0"
34      ContentTemplate="{DynamicResource PreviewDisplay}" />
35    <!--todo Убрать сверху строку файл-->
36    <ContentControl Grid.Row="1" ContentTemplate="{StaticResource DisplayCreating}"
37      Content="{Binding FileCreatorVM}" />
38  </Grid>
39 </Page>

```

Рисунок 15. Пример реализации «Window» класса, в документе Xaml.

Для данной технологии, архитектурный паттерн приложения был сформирован сразу при ее появлении, паттерн MVVM (Model, View, ViewModel; модель, представление, представление-модель или вью-модель).

Паттерн предполагает следующее применение, имеется модель бизнес логики, для модели создается «вью-модель», в котором указываются необходимые свойства для связывания с представлением, пример которого рассмотрен выше, назначается в качестве контекста данных «DataContext» - ViewModel, привязываем свойство View к ViewModel свойствам.

DataContext – подразумевает собой, контекст данных окна который будет использоваться, во время выполнения приложения.

Привязка свойств – является отличительной особенностью WPF технологии и паттерна MVVM. С помощью данной особенности обновляются свойства как на представлении из вью-модели, так и пользователь, обновляя свойства на представлении может изменить свойства во вью-модели, таким образом получается некая предметная область во главе с вью-моделью, которая изменяется через привязанные свойства.

```
<StackPanel Orientation="Horizontal" Grid.Column="0" Margin="0 2 0 0">
  <Menu>
    <MenuItem Header="Файл">
      <MenuItem Header="Создать" Command="{Binding CreateFileCommand}" CommandParameter="{Binding}" />
      <MenuItem Header="Открыть" Command="{Binding OpenFileCommand}" CommandParameter="{Binding}" />
      <MenuItem Header="Сохранить" Command="{Binding SaveFileCommand}" CommandParameter="{Binding}" />
      <MenuItem Header="Сохранить как" Command="{Binding SaveAsFileCommand}" CommandParameter="{Binding}" />
      <MenuItem Header="Выйти" Command="{Binding ExitProgramCommand}" CommandParameter="{Binding}" />
    </MenuItem>
  </Menu>
</StackPanel>
```

Рисунок 16. Пример привязки свойств в xaml (Binding-и).

```

/// <summary>
/// VM для работы в редакторе.
/// </summary>
Ссылка: 19
public class EditorVM : ViewModelBase
{
    /// <summary>
    /// Содержит в себе, выбранный, в редакторе SldFile;
    /// переменная изменяется благодаря <see cref="CurrentEditSldFile" />
    /// </summary>
    private SldVM _currentEditSldFile;

    /// <summary>
    /// VM для представления, Sld структуры, и взаимодействию с ней.
    /// </summary>
}
Ссылка: 2
public EditorVM()...

    /// <summary>
    /// Открыть файл.
    /// </summary>
    Ссылка: 1
    public CreateFileCommand CreateFileCommand { get; }

    /// <summary>
    /// Текущий редактируемый файл, выбранный в редакторе.
    /// </summary>
    Ссылка: 9
    public SldVM CurrentEditSldFile...

    /// <summary>
    /// Выйти из программы.
    /// </summary>
    Ссылка: 1
    public ExitProgramCommand ExitProgramCommand { get; }

    /// <summary>
    /// Окно создания файла.
    /// </summary>
    Ссылка: 2
    public FileCreatorVM FileCreatorVM { get; }
}

```

Рисунок 17. Вью-модель, к которой привязываются свойства представления.

Таким способом реализуется пользовательский интерфейс приложения, с использованием WPF технологии.

Было разработано и создано приложение для редактирования SLD структур, для которой ранее был написан сериализатор получения объекта NET, из файла стиля

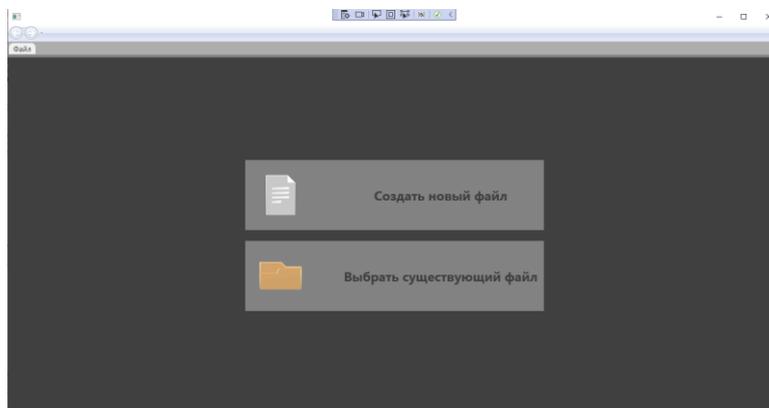


Рисунок 18. Preview окно приложения.

В данном окне мы можем выбрать существующий файл и открыть его, или же создать новый документ.

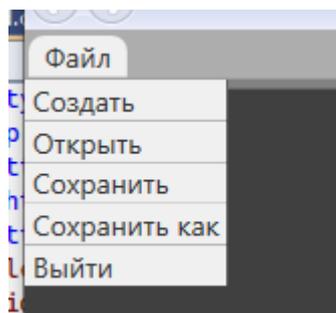


Рисунок 19. Базовые операции приложения.

Также сверху у нас имеется кнопка, по нажатию которой мы можем выполнить базовый операции программы.



Рисунок 20. Рабочая область приложения.

В верхней части приложения, можно увидеть вкладки, внутри которых указаны названия открытых в данный момент документов.

В самой рабочей области имеются три колонки в соответствии заполнению SLD документа: правила, фильтр, символы.

- Правило – это отдельная сущность применимая к конкретному случаю, для правила указываются в колонке фильтр условия, в колонке символы настраивается графическое отображение.
- Фильтр – обычные логические условия, с некоторой спецификой SLD документа, и его области применения.
- Символ – это указание и настройка стиля, графического отображения: точки, линии, полигона, текста.

Работа в приложении происходит для настройки выше описанных сущностей.

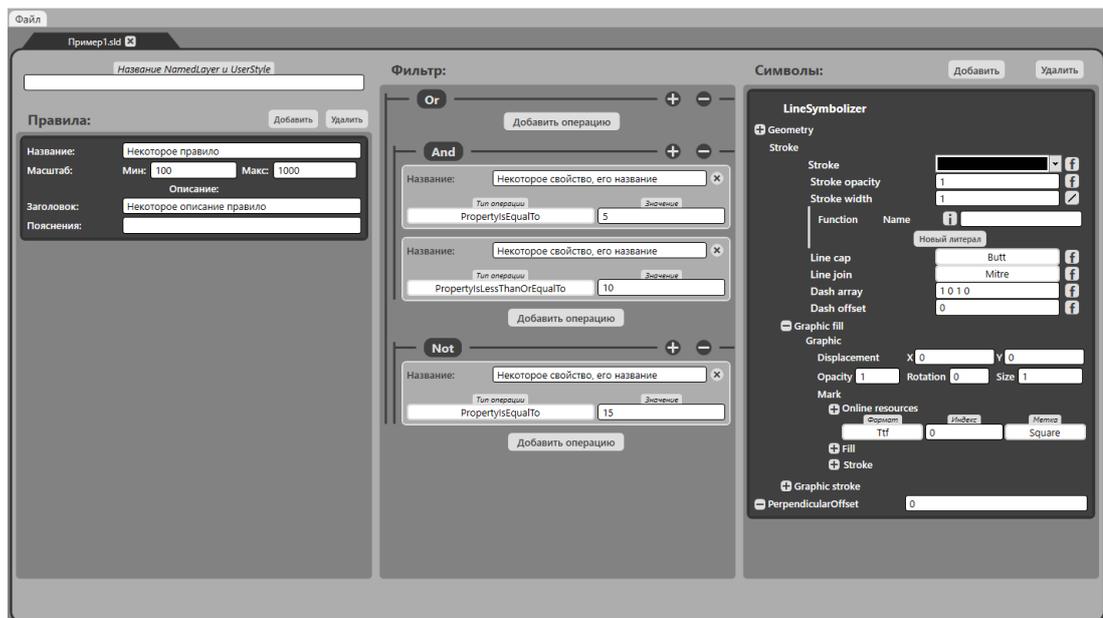


Рисунок 21. Заполненная сущность SldVM (Вью-модель SLD).

Если сохранить текущую открытый файл стиля, то получим, в необходимом xml формате стиль, который может быть использован для описания геометрии.

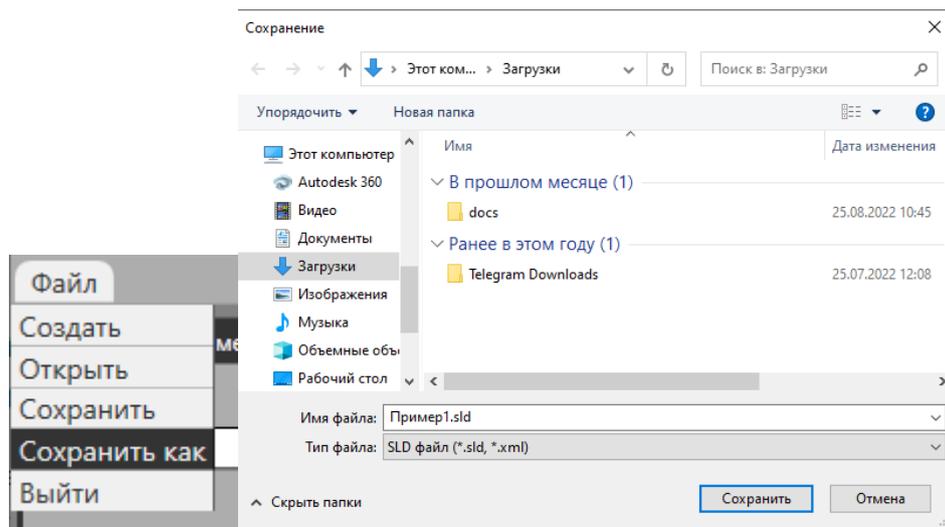


Рисунок 22. Итерации сохранения документа.

```

1 <?xml version="1.0" encoding="utf-8"?>
2 <StyledLayerDescriptor xmlns:se="http://www.opengis.net/se" xmlns:xsd="http://www.w3.org/2001/XMLSchema-
http://schemas.opengis.net/sld/1.1.0/StyledLayerDescriptor.xsd" xmlns:xlink="http://
3 <NamedLayer>
4   <se:Description />
5   <UserStyle>
6     <se:Description />
7     <se:FeatureTypeStyle>
8       <se:Rule>
9         <se:Name>Некоторое правило</se:Name>
10        <se:Description>
11          <se:Title>Некоторое правило заголовок</se:Title>
12        </se:Description>
13        <ogc:Filter xmlns:ogc="http://www.opengis.net/ogc">
14          <ogc:And>
15            <ogc:And>
16              <ogc:PropertyIsLessThan>
17                <ogc:PropertyName>Некоторое свойство</ogc:PropertyName>
18                <ogc:Literal>10</ogc:Literal>
19              </ogc:PropertyIsLessThan>
20              <ogc:PropertyIsEqualTo>
21                <ogc:PropertyName>Некоторое свойство</ogc:PropertyName>
22                <ogc:Literal>15</ogc:Literal>
23              </ogc:PropertyIsEqualTo>
24            </ogc:And>
25          </ogc:And>
26        </ogc:Filter>
27        <se:MinScaleDenominator>0</se:MinScaleDenominator>
28        <se:MaxScaleDenominator>0</se:MaxScaleDenominator>
29        <se:LineSymbolizer>
30          <se:PerpendicularOffset>0</se:PerpendicularOffset>
31          <se:Stroke>
32            <se:SvgParameter name="stroke">#000000</se:SvgParameter>
33            <se:SvgParameter name="stroke-opacity">1</se:SvgParameter>
34            <se:SvgParameter name="stroke-width">1</se:SvgParameter>
35            <se:SvgParameter name="stroke-linecap">butt</se:SvgParameter>
36            <se:SvgParameter name="stroke-linejoin">mitre</se:SvgParameter>
37            <se:SvgParameter name="stroke-dasharray">1 0 1 0</se:SvgParameter>
38            <se:SvgParameter name="stroke-dashoffset">0</se:SvgParameter>
39          </se:Stroke>
40        </se:LineSymbolizer>
41      </se:Rule>
42    </se:FeatureTypeStyle>
43  </UserStyle>
44 </NamedLayer>
45 </StyledLayerDescriptor>

```

Рисунок 23. Xml файл с SLD структурой.

Также если откроем сохраненный xml файл, то отобразятся те же сущности, что были созданы при редактировании.

Иерархия классов в сборке Desktop выглядит следующим образом:

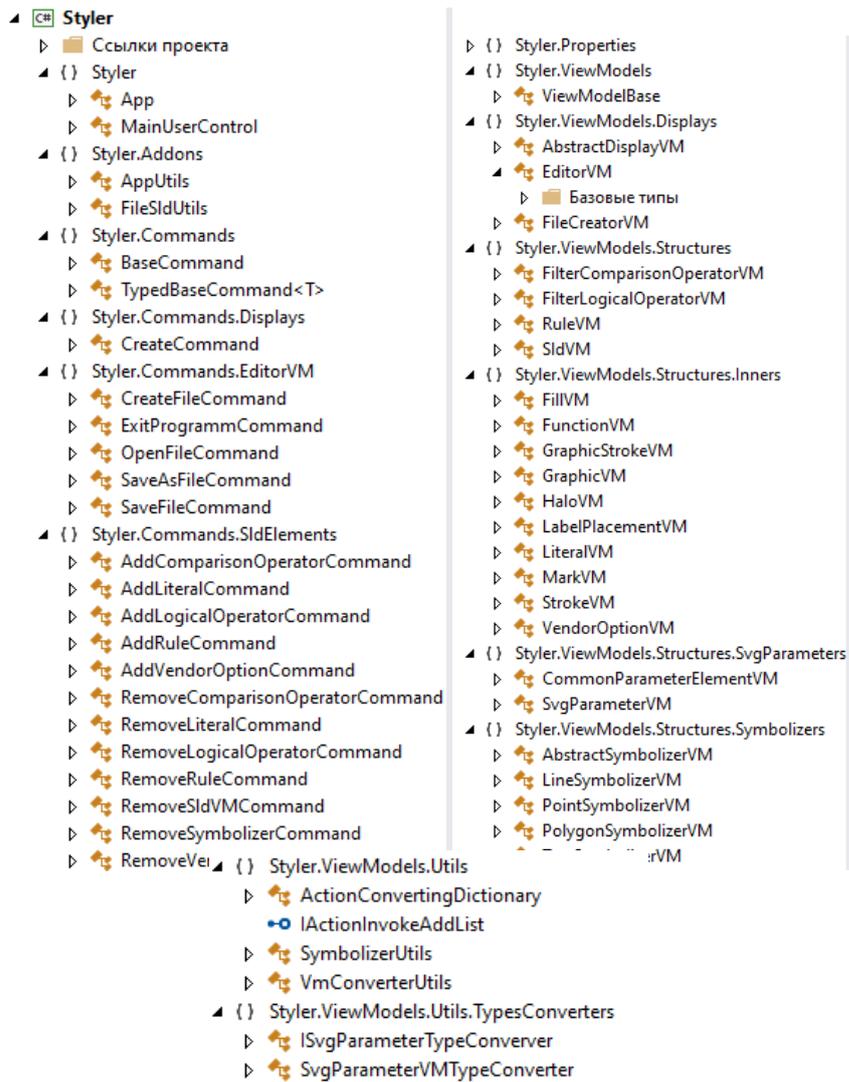


Рисунок 24. Иерархия классов Desktop сборки.

3.1.3. Задание внешнего вида GUI приложения, локализация и добавление подсказок

Были проведены работы над XAML стилями приложения, описывающие его внешний вид, была выбрана цветовая гамма черно-серый; переработаны шаблоны кнопок меню, и способы отображения графических пользовательских элементов.

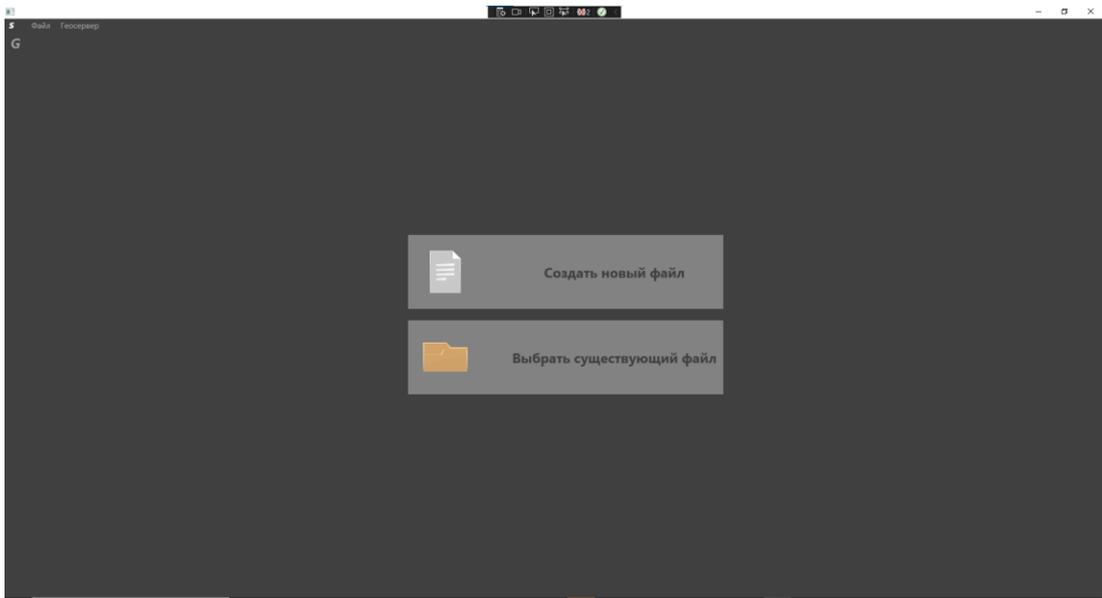


Рисунок 25. Начальное окно при открытии приложения.

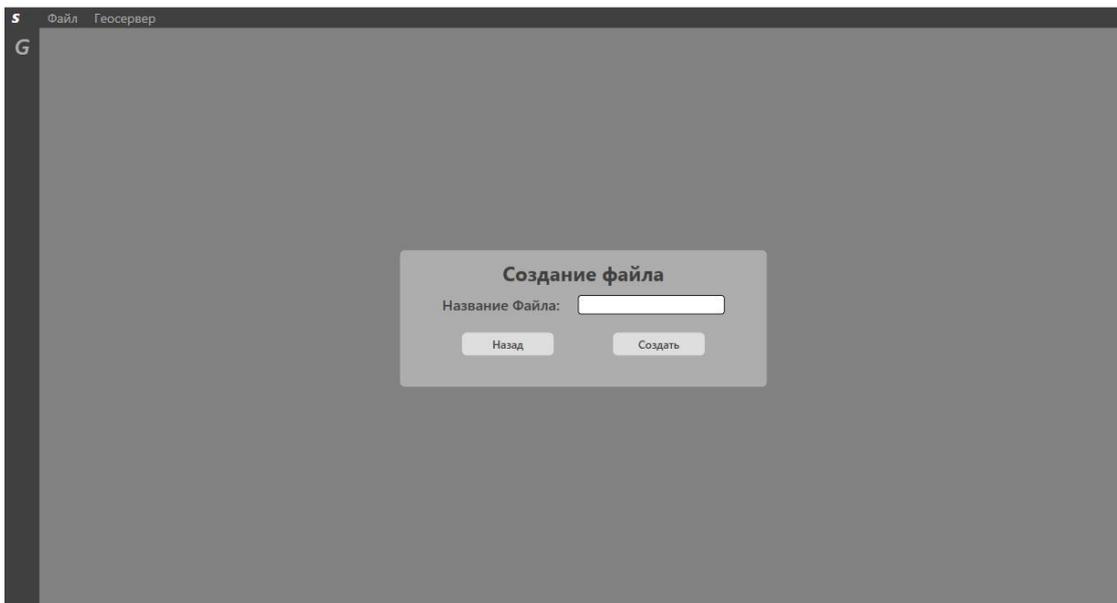


Рисунок 26. Окно при создании стиля.

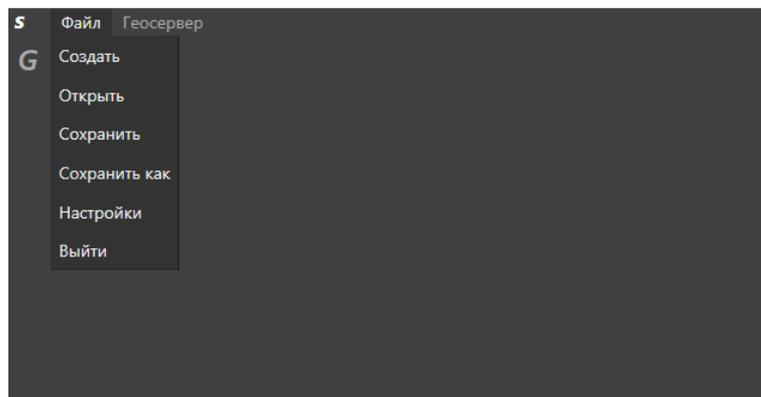


Рисунок 27. Панель меню кнопок приложения.

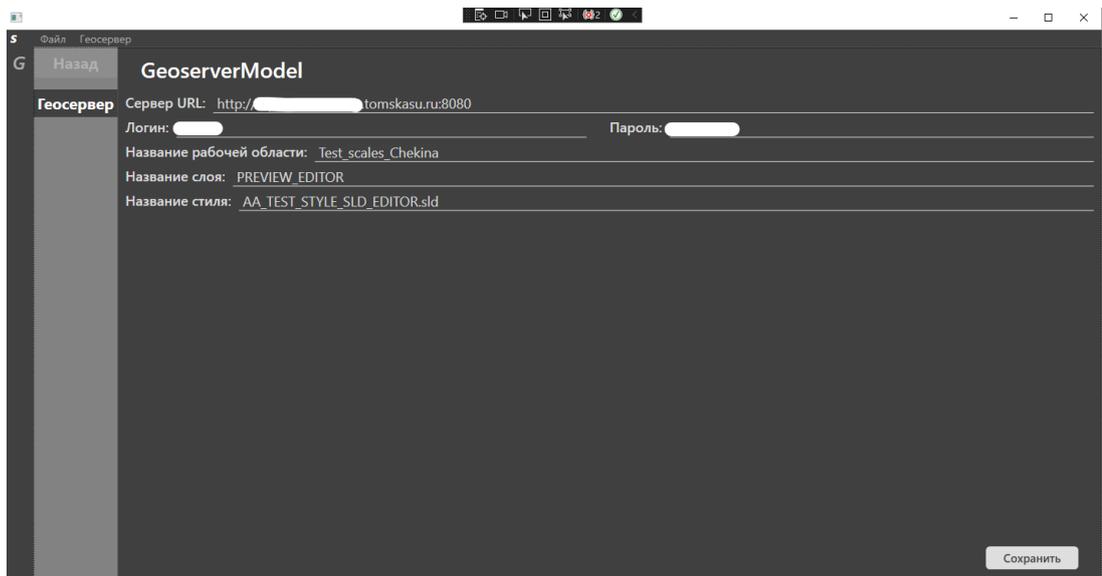


Рисунок 28. Окно настроек приложения, с выбранной вкладкой настроек для ПО Геосервер.

Предполагалось использовать ПО, в первую очередь внутреннем рынке страны, поэтому для легко-понимаемости GUI, была произведена локализация приложения на русский язык.

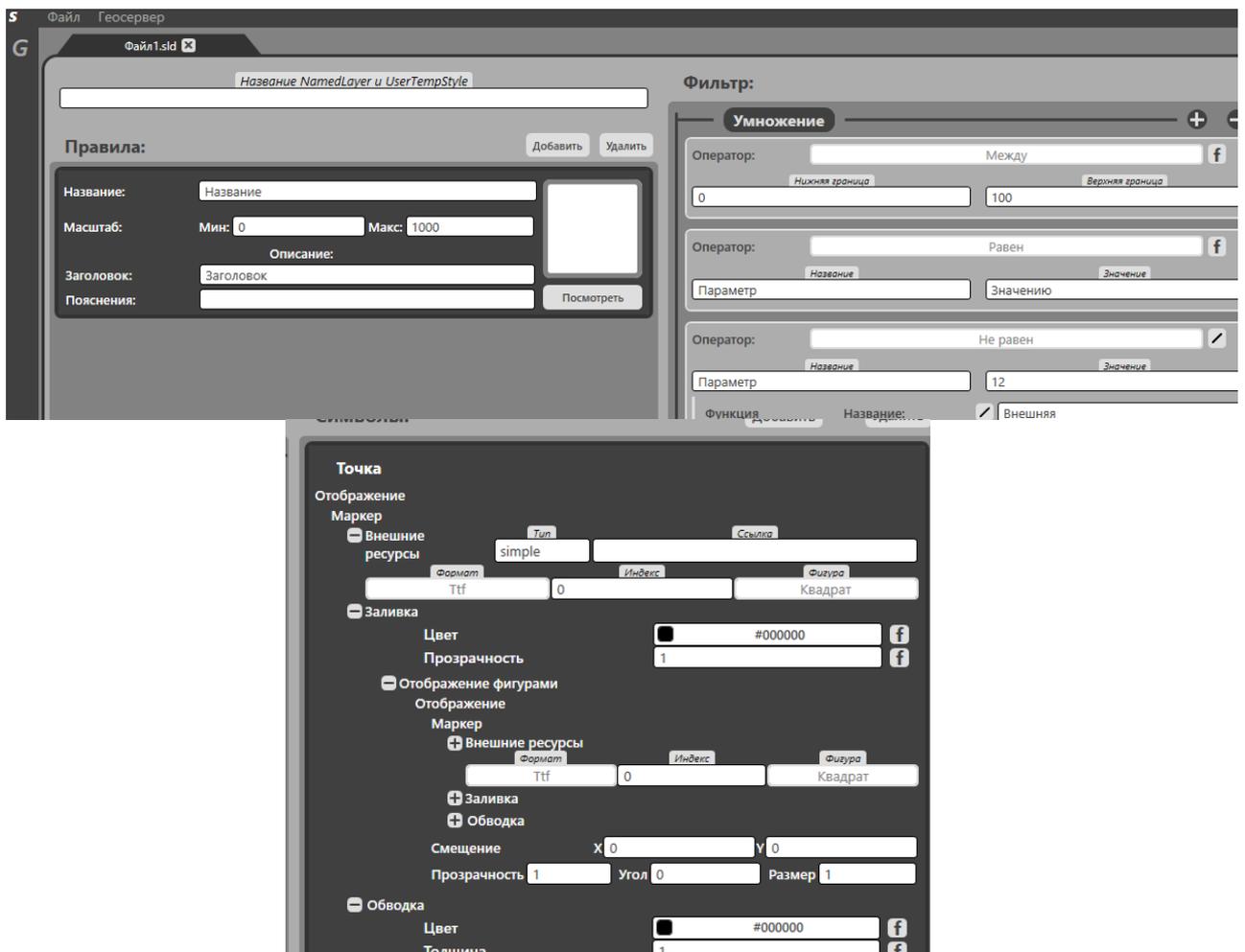


Рисунок 29. Редактор с открытым стилем.

Если присмотреться к GUI, то можно заметить, что все отображаемые надписи локализованы.

Для облегчения понимания, как работать с программой, какой параметр является какой сущностью в SLD структуре, были добавлены подсказки к каждому визуальному элементу представления.

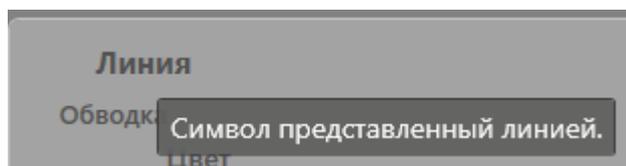


Рисунок 30

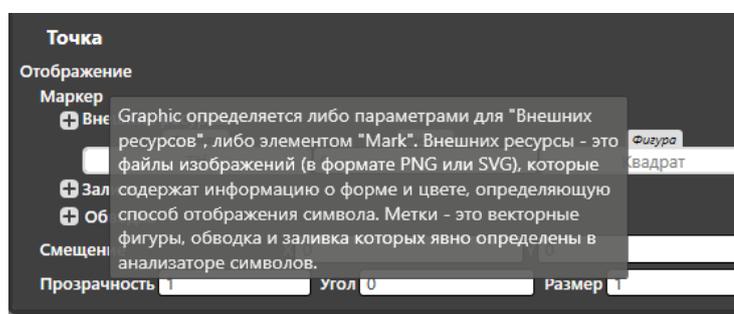


Рисунок 31

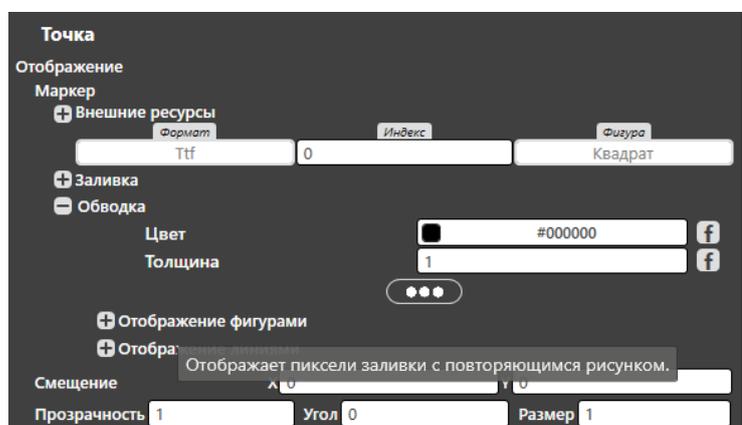


Рисунок 32

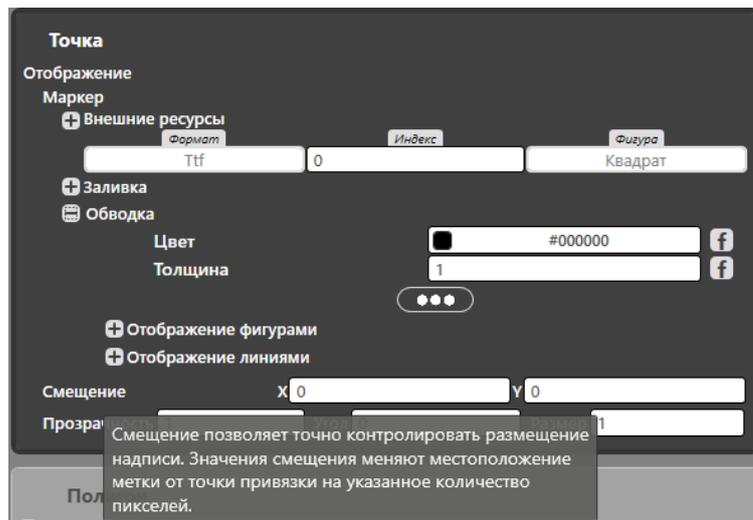


Рисунок 33.

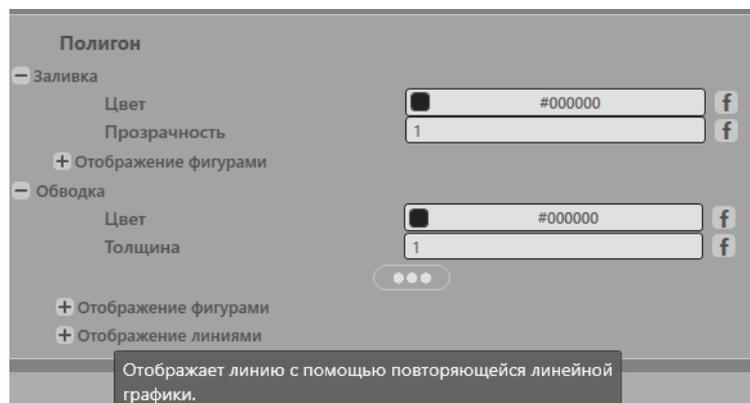


Рисунок 34

3.1.4. Добавление предпросмотра

Была добавлена возможность предпросмотра, редактируемого стиля, для указанного правила и его конфигурации.

Для конфигурации подключения и работы с ПО Geoserver, был добавлен раздел настроек «Геосервер», в котором мы указываем:

- адрес подключения,
- аутентификационные данные,
- уточняем стиль, который будет использоваться для прорисовки изображений предпросмотра

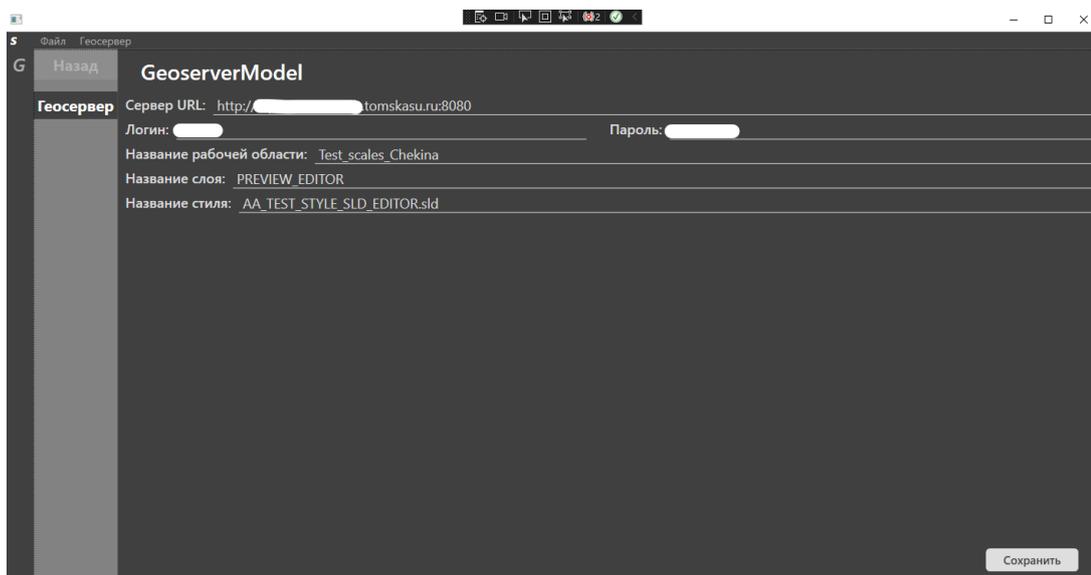


Рисунок 35. Раздел настроек «Геосервер».

Во время редактирования стиля, для каждого правила отображается панель, в которой можно увидеть редактируемый стиль. Изображение изменяется при изменении выбранного правила.

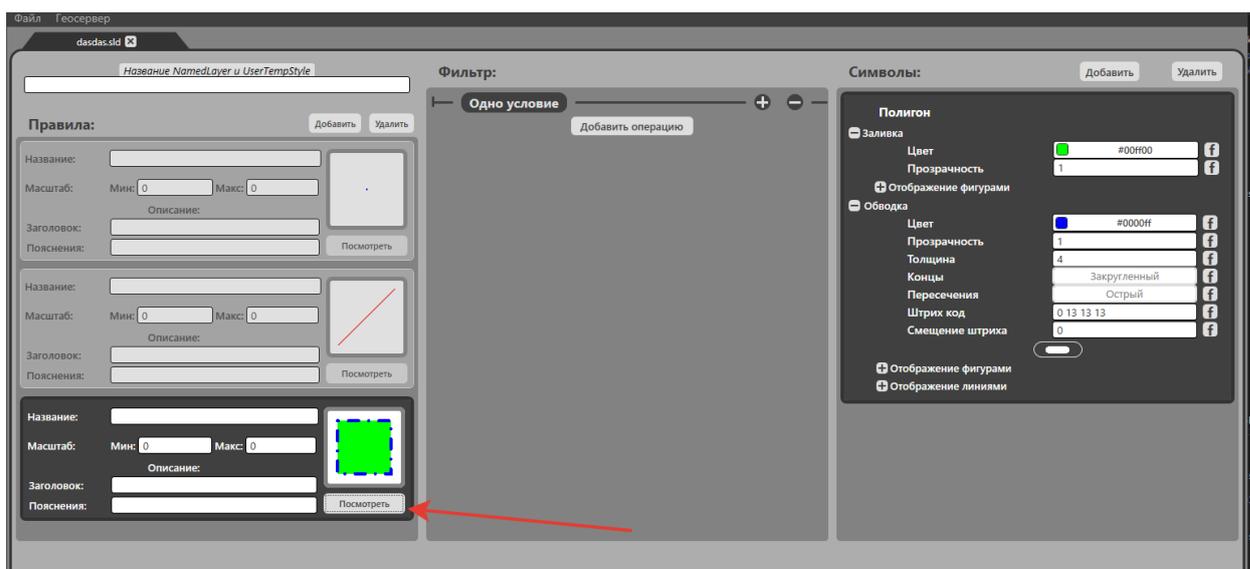


Рисунок 36. Отображение правил с панелью предпросмотра.

Для обновления панели предпросмотра, для выбранного правила используется кнопка «Посмотреть».

3.1.5. Поддержания работы со стилями ПО Geoserver

В начале запуска приложения и при изменении параметров настроек для геосервера, была добавлена проверка соединения. Если соединение отсутствует, отображается представление «Уведомление».

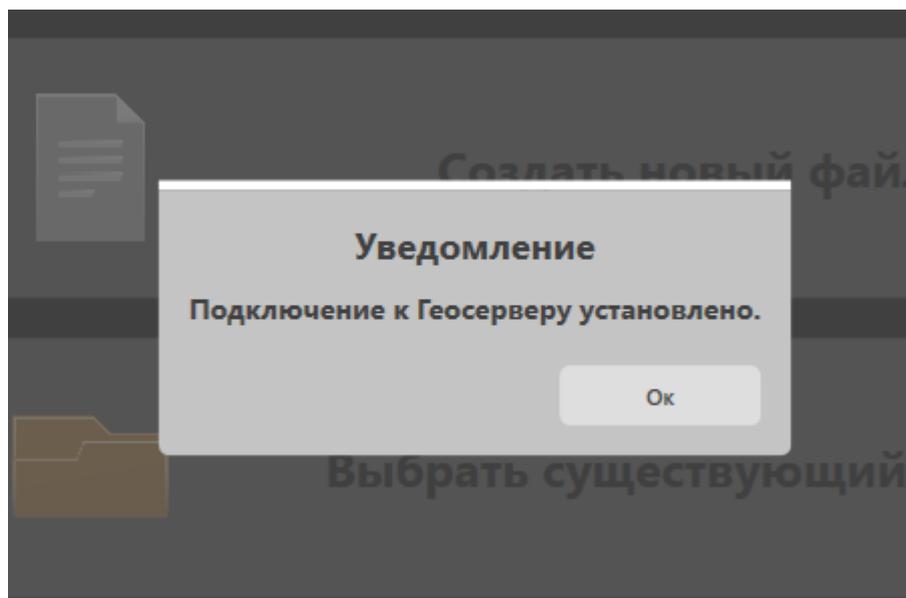


Рисунок 37. Представление «Уведомление».

Была добавлена панель, с имеющимися стилями на развернутом ПО Geoserver.

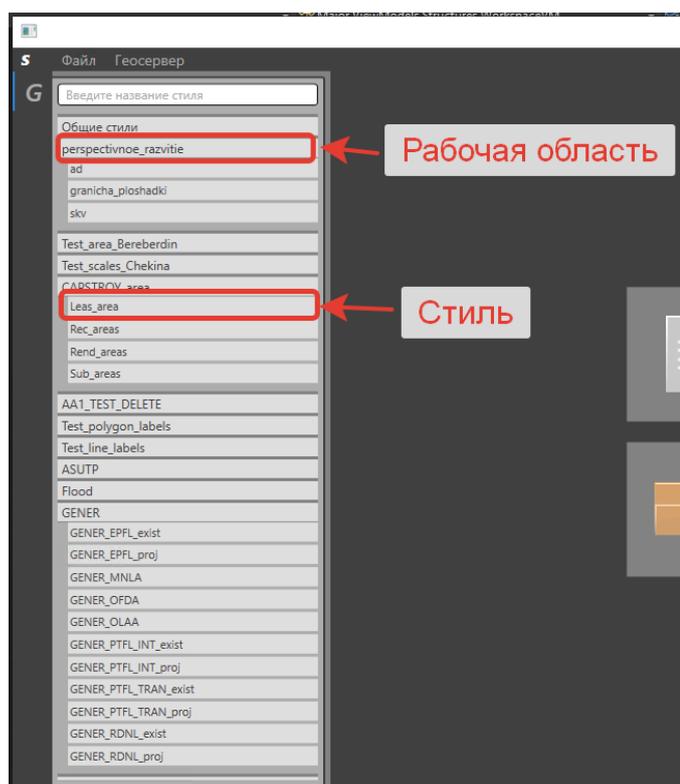


Рисунок 38. Окно с открытой панелью, в которой отображены рабочие области, и стили расположенные, в данных рабочих областях.

В панели отображается иерархия рабочих областей, и вложенных в них стилей. Рабочие области можно свернуть/развернуть для просмотра имеющихся стилей.

По двойному нажатию на стиль открывается, специальный формат документа, который связан со стилем, выгруженным из ПО Geoserver. Особенность открытого с сервера стиля отображена в иконке у названия документа-стиля.

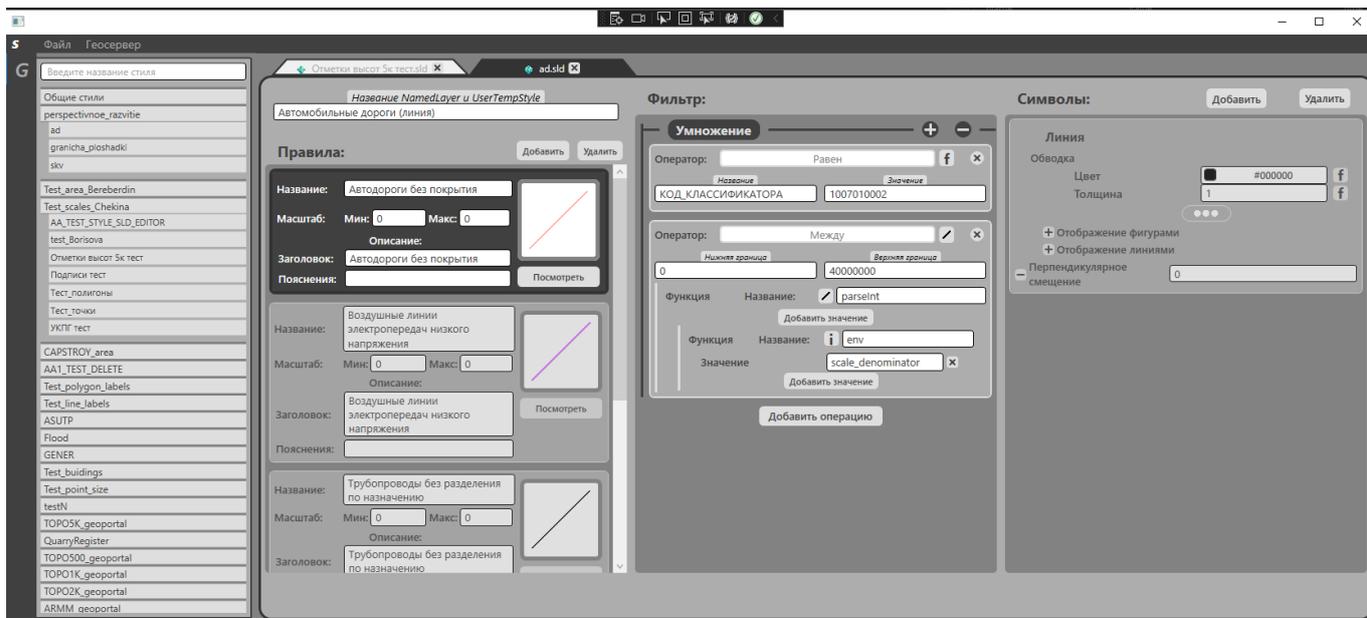


Рисунок 39. Выгруженный стиль.

При создании нового стиля имеется возможность загрузить данный стиль в Geoserver, и в дальнейшем взаимодействовать с ним через сервер.

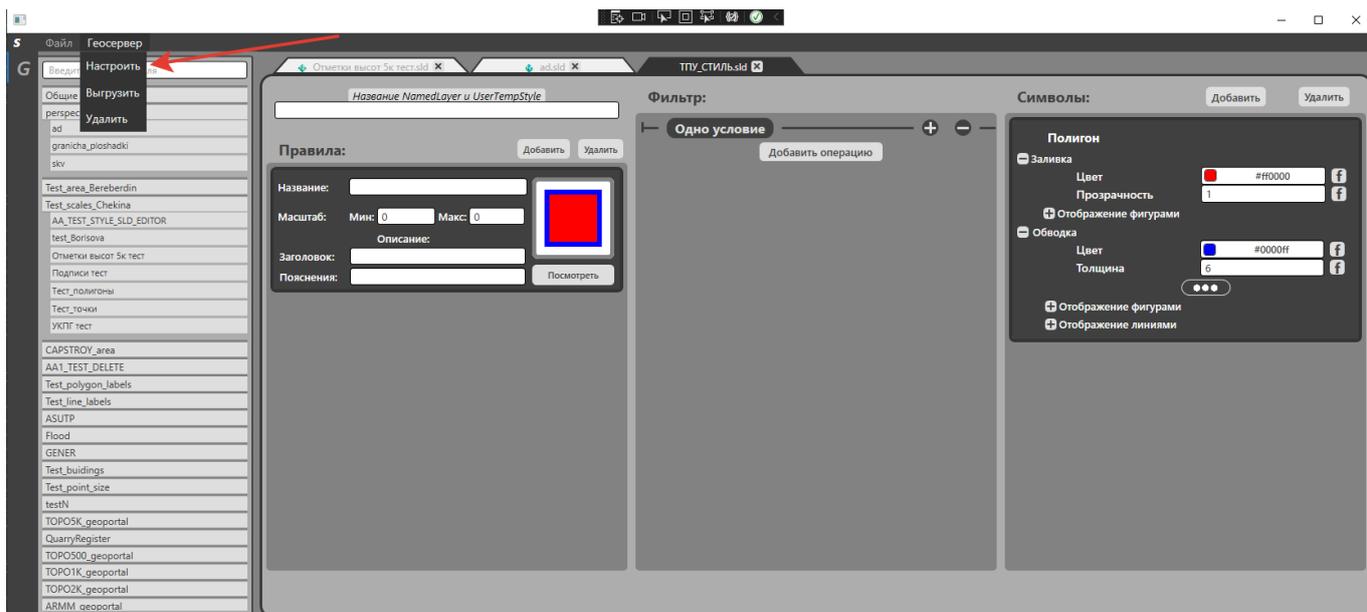


Рисунок 40. Уточнена кнопка «Настроить» для нового стиля.

С помощью кнопки настроить открывается представление, для связывания стиля с геосервером.

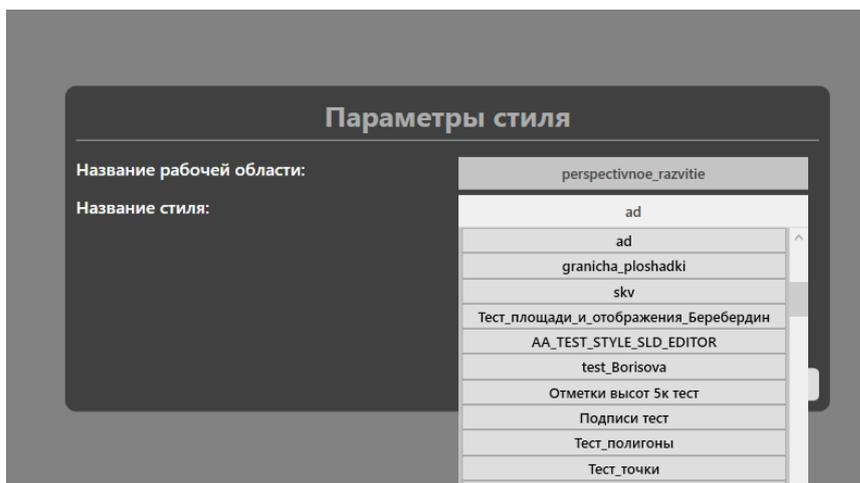


Рисунок 41. Представление для связывания стиля с геосервером.

Для связывания необходимо указать рабочую область и стиль, из имеющихся.

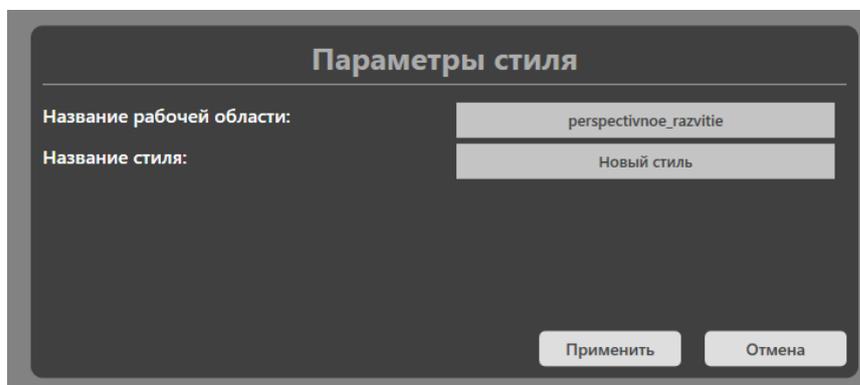


Рисунок 42. Представление для связывания стиля с геосервером, при создании нового стиля.

При создании нового стиля на геосервере, нужно выбрать вариант «Новый стиль», тогда будет добавлен новый стиль.

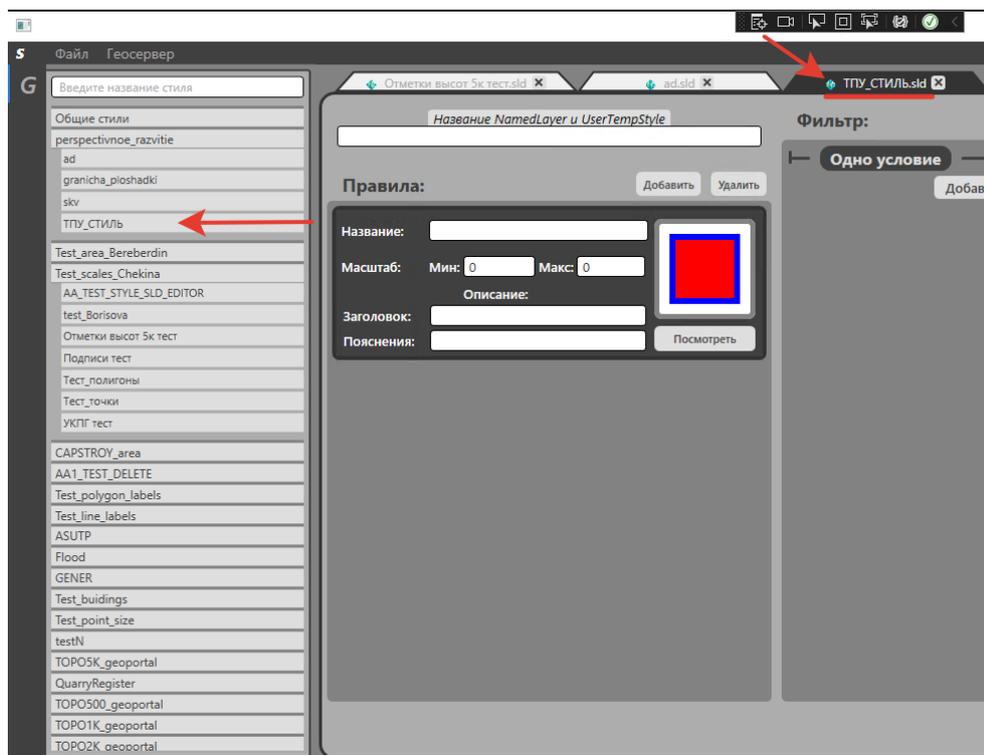


Рисунок 43. Добавленный стиль в Geoserver.

Для выгрузки-обновление стиля на сервере необходимо нажать кнопку выгрузить. После перезагрузки приложения, стиль также отобразится в нашей панели.

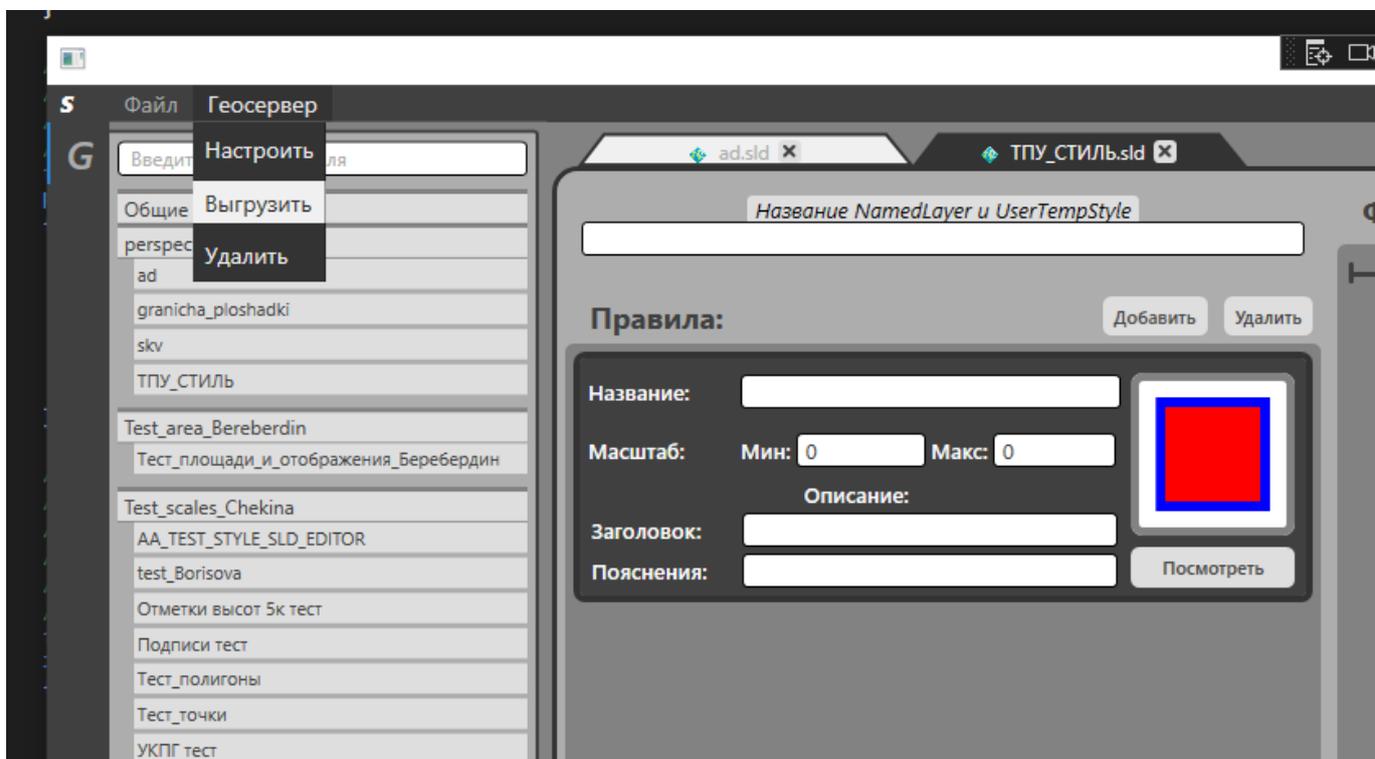


Рисунок 44. Кнопка «Выгрузить».

При необходимости стиль может быть удален из геосервера.

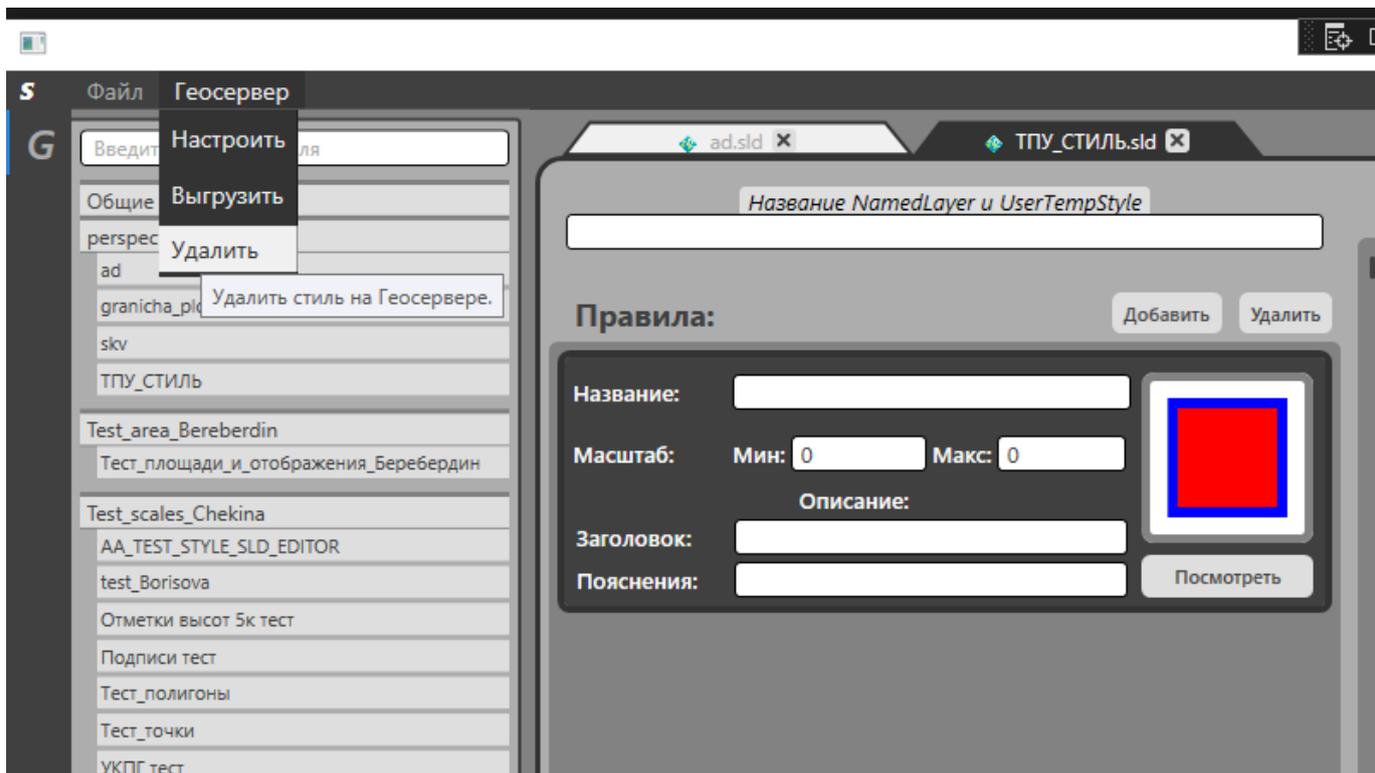


Рисунок 45. Кнопка «Удаление».

3.2. Исследование компиляции исходного кода настольного приложения, в Unity для получения WEB приложения

Для разработки настольного программного обеспечения для Windows, использовалась технология WPF компании Microsoft. Во время разработки GUI использовалась технология привязок и написания UI через файлы XAML – которые по своей сути являются XML документами с текстовой разметкой пространств имен и их классов из WPF.

В Unity по умолчанию во время разработки ПО используется программный язык C#. Unity не поддерживает файлы задающие способ отображения GUI, представленные расширением XAML. Для того чтобы построить пользовательский интерфейс в Unity используются свои инструменты (разработчик может дополнить этот инструментарий, разработав их на C#).

Кроссплатформенность – данный термин под собой подразумевает пере использование одного и того же кода, для компиляции или использования приложения на разных платформах: WEB, Windows, Mac, IOS, Android и так далее.

Ранее, было невозможным использовать файлы XAML в Unity, но с 2017 года появился поддерживаемый и постоянно обновляющийся плагин NoesisGUI для Unity, который позволял использовать те же файлы XAML для разработки GUI программы.

NoesisGUI – это много платформенное промежуточное программное обеспечение

графического пользовательского интерфейса для приложений реального времени, которое понимает XAML.

NoesisGUI содержит пространство имен и описание классов, подобно библиотекам используемых в платформе WPF. Благодаря чему нам достаточно указывать эти различия в используемых пространствах имен через «директивы».

```
#if UNITY_2019_4_OR_NEWER
using Noesis;

#else
using System.Windows.Controls;
using System.Windows;

#endif
namespace Major
{
    using System;

    using Major.Utils;
    using Major.ViewModels.Displays;

    using SldTypes.Types.Addon;

    using Styler.Interface;
#if UNITY_2019_4_OR_NEWER
    using GUI = GUI;

#endif

    /// <summary>
    /// Логика взаимодействия для MainUserControl.xaml
    /// </summary>
#if UNITY_2019_4_OR_NEWER
    public class MainUserControl : UserControl
#else
    public partial class MainUserControl
#endif
    {
        .....
```

Рисунок 46. Определение используемых пространств имен, в файле.

Как можно заметить директивы также используются в редких случаях для разделения проектных отличий или кодовых отличий.

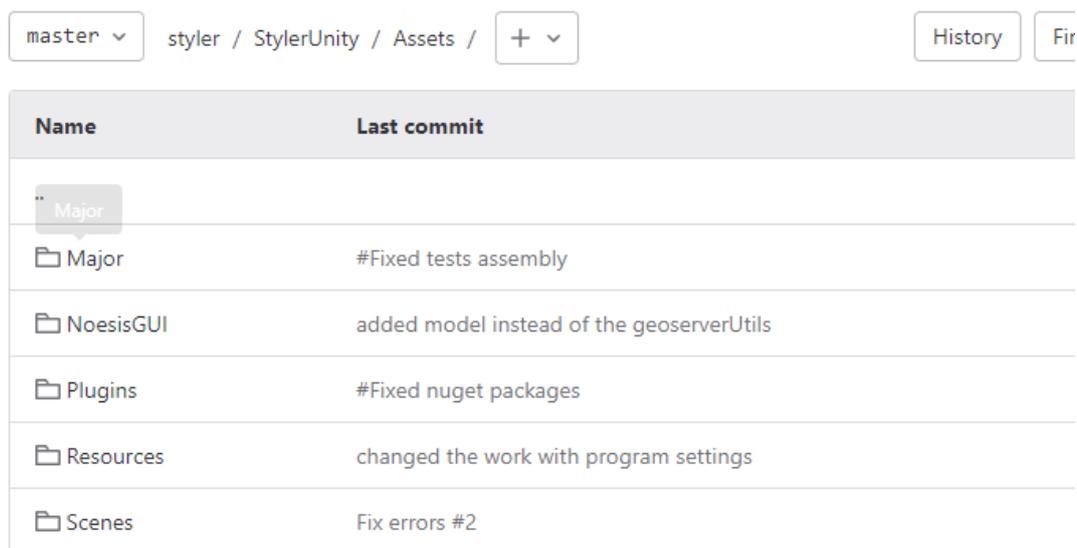
Директивы – это константы компиляции, которые используют C# компиляторы. В данном случае компилятор Unity, будет использовать код определенный без директив и определенный в директиве UNITY_2019_4_OR_NEWER, а WPF сделает иначе,

скомпилировав код определенный после #else.

3.2.1. Подготовка решения с исходным кодом, для работы в Unity

При разработке приложения на Unity все файлы ресурсов, в том числе и кодовые файлы, разработчики располагают по определенной системе иерархии, которая задумана в Unity.

В частности, для данного проекта были перенесены все файлы и исправлены файлы сборки .csproj для Unity и для WPF, с целью их пере использования.



Name	Last commit
Major	#Fixed tests assembly
NoesisGUI	added model instead of the geoserverUtils
Plugins	#Fixed nuget packages
Resources	changed the work with program settings
Scenes	Fix errors #2

Рисунок 47. Иерархия папок и файлов.

Все используемые при компиляции файлы перенесены в Major.

Name	Last commit	Last update
..		
└─ Addons	fixed error with point	10 months ago
└─ Commands	fixed error with point	10 months ago
└─ Constants	fixed remarks	10 months ago
└─ Converters	fixed comments	10 months ago
└─ Elements	fixed remarks	10 months ago
└─ Fonts	Fix errors #1	1 year ago
└─ Images	added of a hiding	10 months ago
└─ Models	fixed a bug with dragdrop	10 months ago
└─ Selectors	fixed comments	10 months ago
└─ StylesDictionary	added function in comparisonVM	10 months ago
└─ TestUI	Fix errors #2	1 year ago
└─ Utils	#Fixed tests assembly	9 months ago
└─ ViewModels	added function in comparisonVM	10 months ago
└─ Workers	fixed errors related with INotifier	11 months ago
└─ MainUserControl.asset	Fix errors #2	1 year ago
└─ MainUserControl.xaml	fixed remarks	11 months ago

Рисунок 48. Содержание папки Major.

Чтобы использовать NoesisGUI, необходимо зарегистрироваться на официальном информационном портале Noesis и получить лицензию. В частности, во время разработки ПО, достаточно бесплатной лицензии.

Для установки плагина, достаточно зайти в Package Manager – Unity, найти этот плагин и установить или же вручную скачать файл установки и добавить его, в Unity Assets и также установить.

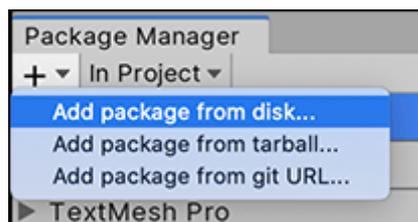


Рисунок 49. PackageManager.

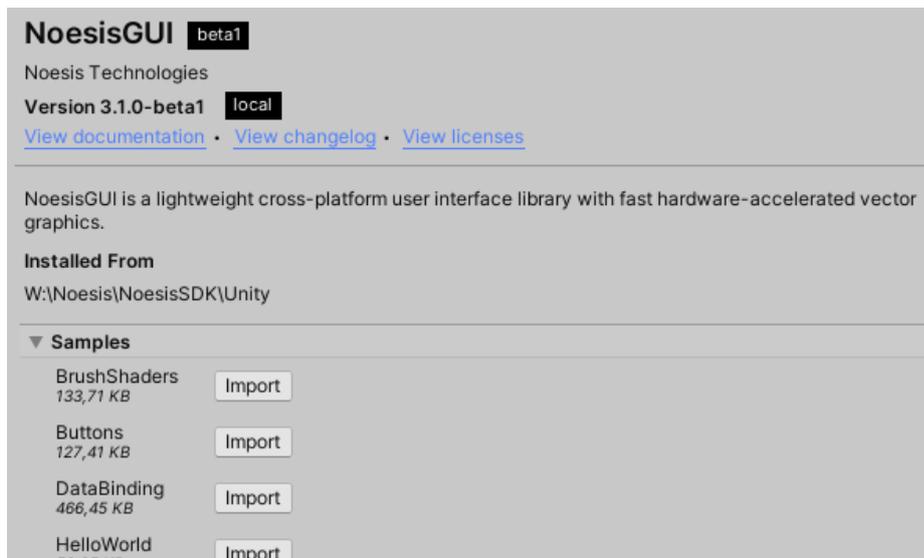


Рисунок 50. Плагин NoesisGUI, в Unity.

Когда разработчик получает лицензию, ему достаточно ввести авторизационные данные, в Unity в данном плагине; помимо указания лицензии, разработчик может указать дополнительные конфигурации для работы плагина. В нашем проекте были переписаны на XAML стили всех используемых UI элементов, поэтому необходимо указать словарь ресурсов XAML. Для использования символов кириллицы, был добавлен свой шрифт.

Словарь ресурсов XAML – представляет собой словарь, где ключом является название ресурса-стиля с описанием UI элемента, к примеру кнопки.

```
<ResourceDictionary xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation">
    <ResourceDictionary.MergedDictionaries>
        <ResourceDictionary Source="StylesDictionary/MenuStyleDictionary.xaml" />
        <ResourceDictionary Source="StylesDictionary/RuleStyleDictionary.xaml" />
        <ResourceDictionary Source="StylesDictionary/FilterStyleDictionary.xaml" />
        <ResourceDictionary Source="StylesDictionary/SymbolStyleDictionary.xaml" />
        <ResourceDictionary Source="StylesDictionary/DisplayStyleDictionary.xaml" />
    </ResourceDictionary.MergedDictionaries>
</ResourceDictionary>
```

Рисунок 51. Словарь, объединяющий в себе, все словари ресурсов.

```
ElementsStyleDictionary.xaml 11.95 KiB
1 <ResourceDictionary xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
2   xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
3   xmlns:elements="clr-namespace:Major.Elements">
4
5   <ResourceDictionary.MergedDictionaries>
6     <ResourceDictionary Source="BasicStyles.xaml" />
7     <ResourceDictionary Source="EnumsDictionary.xaml" />
8   </ResourceDictionary.MergedDictionaries>
9
10  <LinearGradientBrush x:Key="ThumbBackground">
11    <GradientStop Color="#828282" Offset="0"/>
12    <GradientStop Color="ffffff" Offset="0.5"/>
13    <GradientStop Color="faf3a2" Offset="1"/>
14  </LinearGradientBrush>
15
16  <!--ColorPicker-->
17  <Style x:Key="ThumbSlider" TargetType="Thumb">
18    <Setter Property="Background" Value="{StaticResource BlockBorder_Select}"/>
19    <Setter Property="Template">
20      <Setter.Value>
21        <ControlTemplate TargetType="Thumb">
22          <Grid Background="Transparent" Width="10" Height="15" Cursor="Hand">
23            <Border x:Name="bg" Width="5" Background="{StaticResource ThumbBackground}"
24              BorderBrush="{TemplateBinding Background}" BorderThickness="1"
25              CornerRadius="1"/>
26          </Grid>
27        </ControlTemplate>
28      </Setter.Value>
29    </Setter>
30    <Style.Triggers>
31      <Trigger Property="IsMouseOver" Value="true">
32        <Setter Property="Background" Value="DodgerBlue"></Setter>
33      </Trigger>
34    </Style.Triggers>
35  </Style>
```

Рисунок 52. Словарь, с описанием дополнительных UI элементов, разработанных во время разработки редактора стилей.

Как можно видеть по картинке выше, словарем является сам файл ElementsStyleDictionary, а ключем – ThumbSlider, который указывает на стиль-описание GUI элемента Thumb.

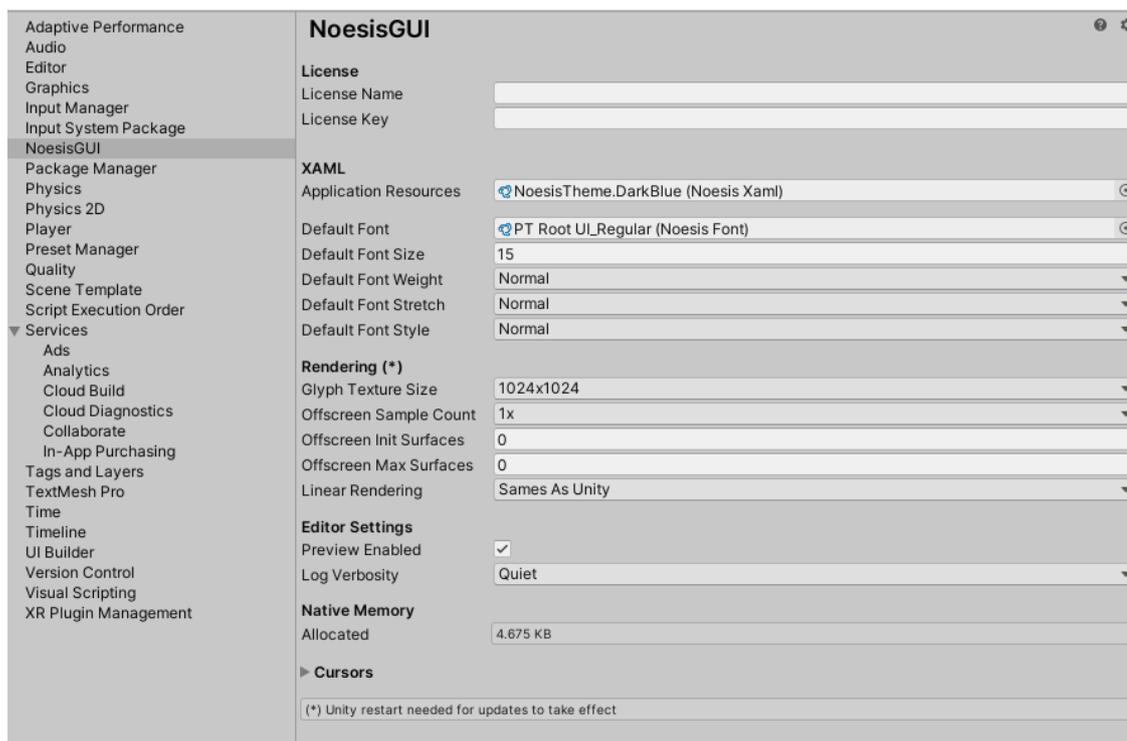


Рисунок 53. Пример конфигурирования работы плагина, в Unity проекте.

В Unity все представлено игровыми объектами, которые характеризуются с помощью компонентов, прикрепляемых к объектам. Объект кнопка является GUI кнопкой, потому что у нее есть компонент: UI трансформации и компонент Button.

Как говорилось ранее, в Unity для разработки GUI приложения используется свой инструментарий, который можно расширить. NoesisGUI – расширяет данный инструментарий, в частности мы можем прикрепить к игровому объекту в Unity, компонент который отобразит нам представление, которое ранее было написано на XAML.

Компонент, который позволяет отобразить XAML файл в GUI приложения – NoesisView. Для его настройки, достаточно указать главный XAML файл, с представлением.

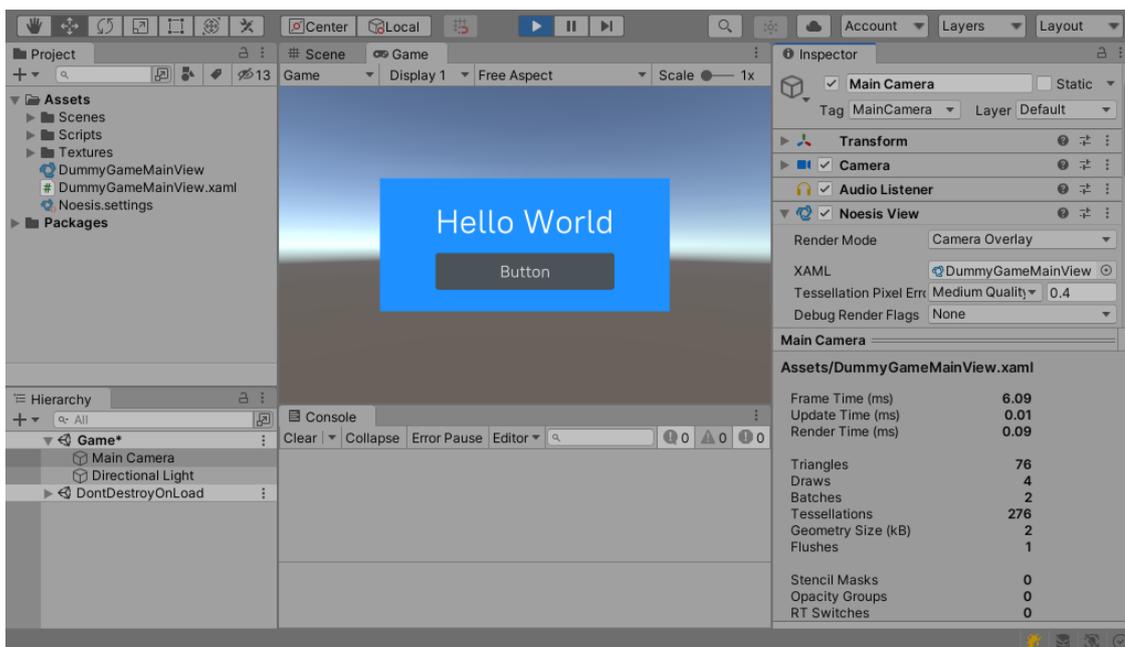


Рисунок 54. Пример использования компонента NoesisView.

Данный плагин позволяет читать разметку XAML, и отображать ее в UNITY GUI, но данном плагине по умолчанию не встроено XAML описание для базовых элементов; все XAML GUI элементы отображаются розовым цветом, указывая на то, что для них необходимо определить стили.

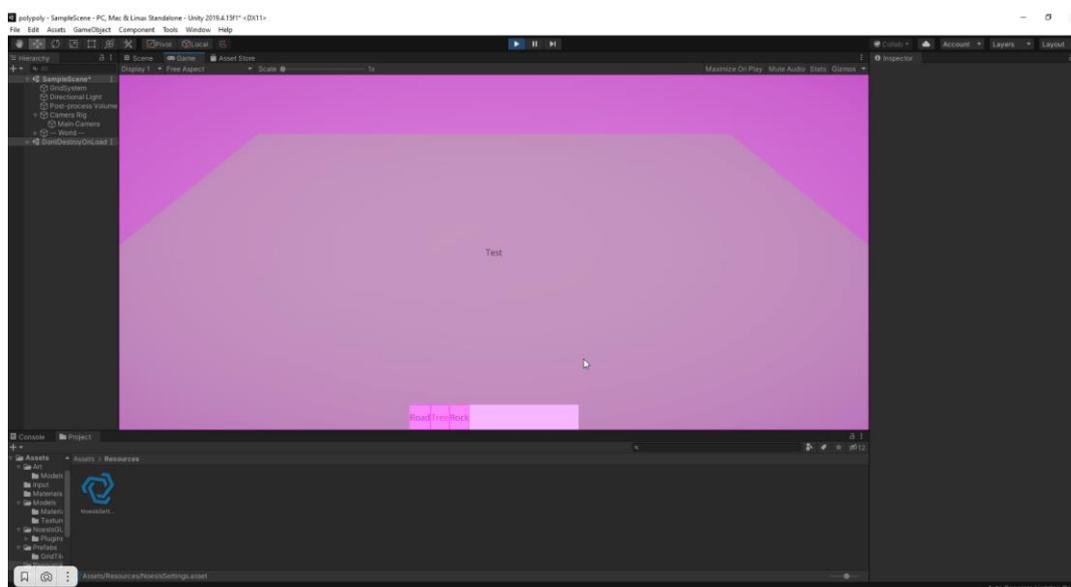


Рисунок 55. GUI отображается розовым потому что, нет стилей для GUI элементов по умолчанию, в NoesisGUI.

Написаны стили для всех стандартных NoesisGUI элементов.

BasicStyles.asset	Fix errors #4. All visual and assemblies erros fixed	1 year ago
BasicStyles.xaml	translated all names in editor	10 months ago
ColorBrushDictionary.asset	Fix errors #2	1 year ago
ColorBrushDictionary.xaml	added functional -CU- from CRUD	11 months ago
DisplayStyleDictionary.asset	Fix errors #4. All visual and assemblies erros fixed	1 year ago
DisplayStyleDictionary.xaml	added dragdrop for rules	10 months ago
ElementsStyleDictionary.asset	ColorHex fixed	1 year ago
ElementsStyleDictionary.xaml	ColorHex fixed	1 year ago
EnumsDictionary.asset	Fix errors #4. All visual and assemblies erros fixed	1 year ago
EnumsDictionary.xaml	added dragdrop for rules	10 months ago
FilterStyleDictionary.asset	#fix remarks and configured tests	1 year ago
FilterStyleDictionary.xaml	added function in comparisonVM	10 months ago
InnersStyleDictionary.asset	Fix errors #4. All visual and assemblies erros fixed	1 year ago
InnersStyleDictionary.xaml	added dragdrop for rules	10 months ago
MenuItemsStyleDict.xaml	fix remarks	11 months ago
MenuStyleDictionary.asset	Add new plugin - SimpleMapper	1 year ago
MenuStyleDictionary.xaml	added of a hiding	10 months ago
RuleStyleDictionary.asset	Fix errors #2	1 year ago
RuleStyleDictionary.xaml	Change names of ui elements	10 months ago

Рисунок 56. Множество словарей ресурсов, с описанием XAML GUI элементов.

3.2.2. Написание библиотеки для преобразования моделей с одинаковыми свойствами

Основная логика работы WPF приложения отображается следующей схемой.



Рисунок 57. Схема работы редактора, WPF приложения.

Как можно увидеть на схеме для работы с со стилем, в кодовой базе необходимо иметь вью-модель, к свойствам, которой привязывается представление, также необходимо иметь модель стиля для конвертации его в текстовую разметку.

Модель и модель-представления в себе повторяют сложную структуру, которая содержится в файлах стилей. Написание кода по преобразованию таких моделей, где вручную указывается преобразование каждого свойства, является рутинной работой,

которую можно автоматизировать.

Для данного случая имеется популярная библиотека AutoMapper, которая решает данную проблему, но т.к. платформа NET Framework используемая в WPF отличается от NET Mono используемой в Unity, была написана библиотека SimpleMapper.

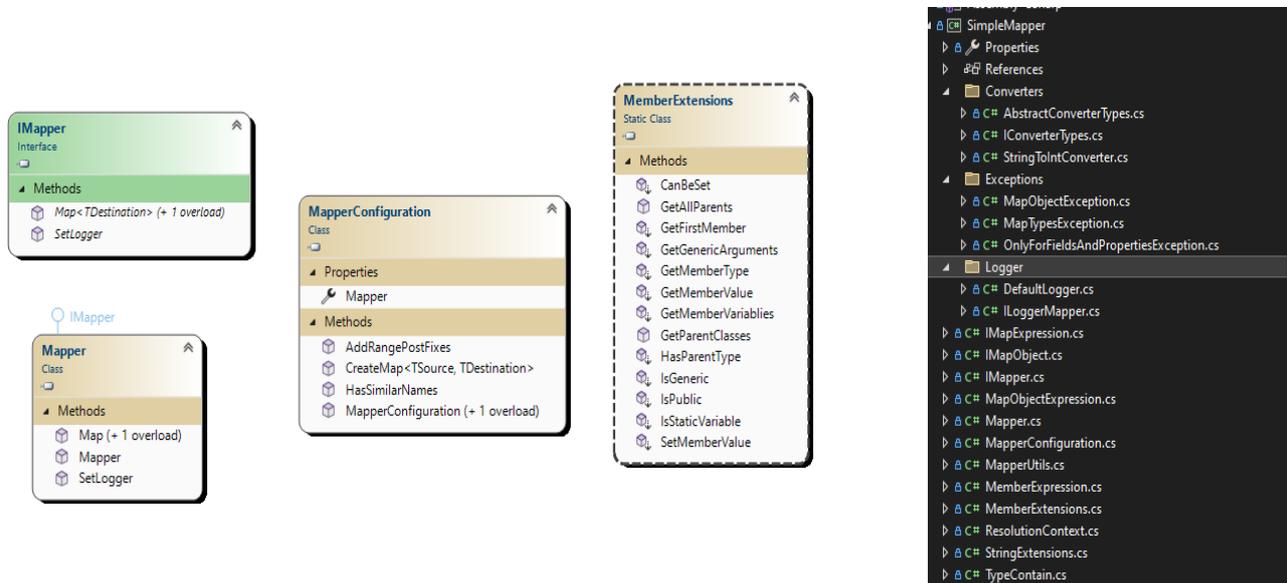


Рисунок 58. Классы библиотеки SimpleMapper.

Библиотека позволяет произвести преобразование между разными типами. Преобразователь рефлексивно анализирует наименование свойств и их типы при совпадении переназначает данные между этими свойствами. В библиотеки имеется дополнительный функционал для параметризации преобразователя.

3.2.3. Результаты исследования

После устранения всех различий между платформами Unity и WPF, и подготовки решения. Удалось пере использовать код и скомпилировать WEB приложение.

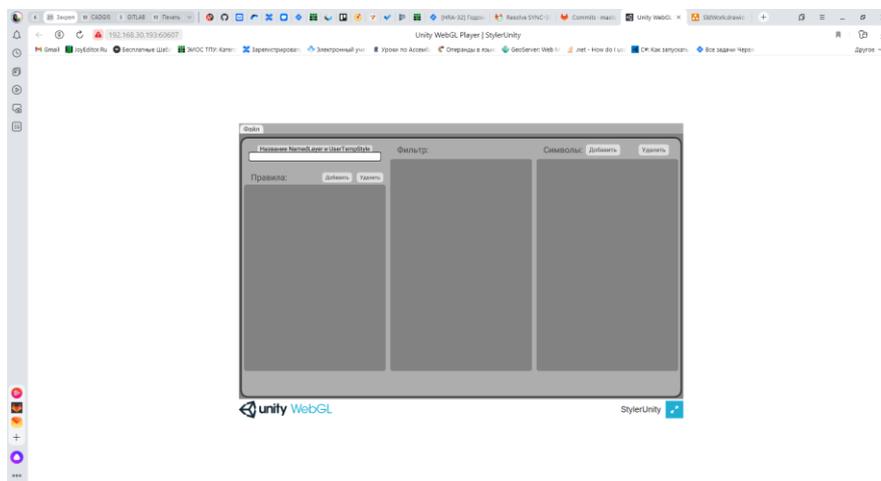


Рисунок 59. Работа приложения в браузере.

Во время исследования и использования скомпилированного приложения появлялись ошибки, без возможности определения причины.

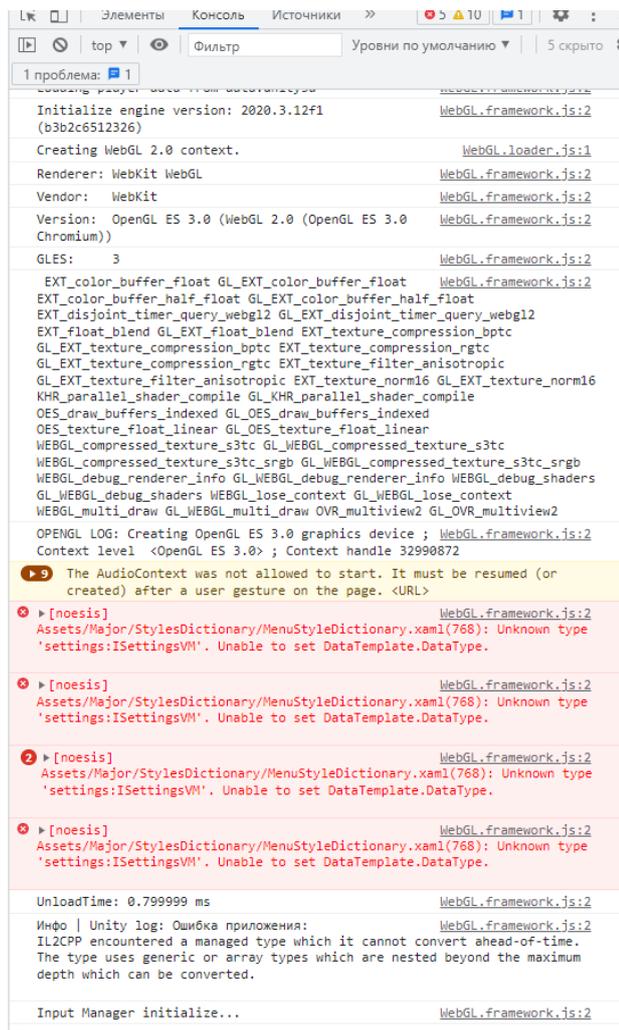


Рисунок 60. Примеры ошибок из отладки браузера.

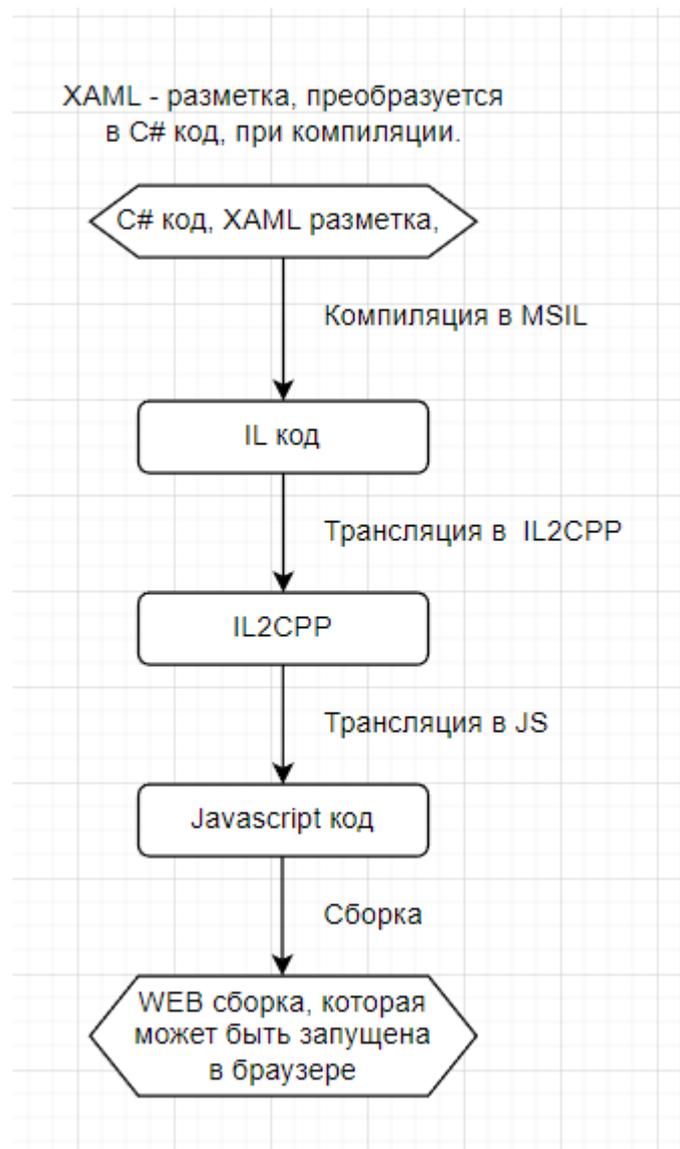


Рисунок 61. Схема преобразования исходного при компиляции WEB приложения.

Из представленной схемы выше, можно заметить множество изменений, которых происходит с кодом, во время получения итоговой WEB сборки. У этого сложного-комплексного действия были выявлены следующие последствия:

- 1) Долгая сборка на высокопроизводительном компьютере
 - Причиной долгой сборки является большая обработка кода при получении конечного WEB приложения.
 - Во время компиляции каждый раз, компилируются сам плагин NoesisGUI, со своей большой библиотекой классов.
- 2) Различие поведения WEB приложения из тестового в Unity.
 - Unity во время воспроизведения (запуск в редакторе), не выполняет такую большую работу по компиляции и трансляции кода. (Unity компилирует все в IL и запускает IL код в среде Mono)
 - Процесс отладки итогового приложения в WEB среде не возможен, по

причине сложности чтения ретраслированного кода

- Различие поведения представляется кодовыми ошибками, которые тяжело выявляются и тяжело исправляются.

3) NoesisGUI имеет свои недоработки и особенности.

- Недоработки NoesisGUI – тяжело выявляются на стадии уже готового WPF продукта и адаптации его под Unity с последующей адаптацией под WEB.

Из всего выше сказанного, становится ясно, для получения кроссплатформенного кода, компиляция кода и тестирования приложения должна происходить на всех необходимых платформах, с самого начала разработки.

4. Финансовый менеджмент, ресурсоэффективность и ресурсосбережение

ЗАДАНИЕ ДЛЯ РАЗДЕЛА «ФИНАНСОВЫЙ МЕНЕДЖМЕНТ, РЕСУРСОЭФФЕКТИВНОСТЬ И РЕСУРСОСБЕРЕЖЕНИЕ»

Студенту:

Группа	ФИО
8К93	Жабину Ефиму Михайловичу

Школа	Инженерная школа информационных технологий и робототехники	Отделение школы (НОЦ)	ОИТ
Уровень образования	Бакалавриат	Направление/специальность	09.03.04 «Программная инженерия»

Исходные данные к разделу «Финансовый менеджмент, ресурсоэффективность и ресурсосбережение»:

1. Стоимость ресурсов научного исследования (НИ): материально-технических, энергетических, финансовых, информационных и человеческих	Бюджет проекта – не более 300 тыс. руб., в т.ч. затраты на оплату труда – 200 тыс. руб. Оклад разработчик – 40 тыс. руб.
2. Нормы и нормативы расходования ресурсов	Премимальный коэффициент 30% Доплаты и надбавки разработчика 20% Дополнительной заработной платы 15% Накладные расходы 15% Районный коэффициент 1,3.
3. Используемая система налогообложения, ставки налогов, отчислений, дисконтирования и кредитования	Тариф отчислений во внебюджетные фонды 7,6%

Перечень вопросов, подлежащих исследованию, проектированию и разработке:

1. Оценка коммерческого потенциала, перспективности и альтернатив проведения НИ с позиции ресурсоэффективности и ресурсосбережения	Определение потенциального потребителя результатов исследования. Анализ конкурентных технических решений. SWOT-анализ разработанной стратегии.
2. Планирование и формирование бюджета научных исследований	Структура работ в рамках научного исследования; Определение трудоемкости выполнения работ и разработка графика проведения научного исследования; Бюджет проекта.
3. Определение ресурсной (ресурсосберегающей), финансовой, бюджетной, социальной и экономической эффективности исследования	Рассчитать показатели финансовой эффективности, ресурсоэффективности и эффективности исполнения

Перечень графического материала*(с точным указанием обязательных чертежей):*

1. Оценка конкурентоспособности технических решений
2. Матрица SWOT
3. График проведения и бюджет НИ
4. Оценка ресурсной, финансовой и экономической эффективности НИ

Дата выдачи задания для раздела по линейному графику

04.03.2023

Задание выдал консультант:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Профессор ОСГН	Гасанов Магеррам Али ОГЛЫ	д.э.н.		

Задание принял к исполнению студент:

Группа	ФИО	Подпись	Дата
8К93	Жабин Ефим Михайлович		04.03.2023

4.1. Введение

Основная цель данного раздела – оценить перспективность развития и планировать финансовую и коммерческую ценность конечного продукта, представленного в рамках исследовательской работы. Коммерческая ценность определяется не только наличием более высоких технических характеристик над конкурентными разработками, но и тем, насколько быстро разработчик сможет ответить на следующие вопросы – будет ли продукт востребован на рынке, какова будет его цена, каков бюджет научного исследования, какое время будет необходимо для продвижения разработанного продукта на рынок.

Данный раздел, предусматривает рассмотрение следующих задач:

- Оценка коммерческого потенциала разработки.
- Планирование научно-исследовательской работы;
- Расчет бюджета научно-исследовательской работы;
- Определение ресурсной, финансовой, бюджетной эффективности исследования.

Целью НИР является исследование и разработка настольного приложения для редактирования XML документов, содержащих внутри себя структуру SLD, через дружелюбный графический пользовательский интерфейс.

4.2. Оценка коммерческого потенциала и перспективности проведения исследований с позиции ресурсоэффективности и ресурсосбережения.

4.2.1. Анализ конкурентных технических решений

В ходе исследования были рассмотрены несколько схожих реализаций для редактирования файлов стилей:

- 1) QGIS;
- 2) Atlas Styler;

Таблица 1 - Сравнение конкурентных технических решений (исследовательских работ).

Критерии оценки	Вес критерия	Баллы			Конкурентоспособность		
		Б _ф	Б _{к1}	Б _{к2}	К _ф	К _{к1}	К _{к2}
1	2	3	4	5	6	7	8
Технические критерии оценки ресурсоэффективности							
1. Актуальность исследования	0,05	5	5	5	0,25	0,25	0,25
2. Энергэффективность	0,2	3	2	2	0,6	0,4	0,4
3. Качество дизайна	0,05	5	3	3	0,25	0,15	0,15

Критерии оценки	Вес критерия	Баллы			Конкурентоспособность		
		Б _ф	Б _{к1}	Б _{к2}	К _ф	К _{к1}	К _{к2}
1	2	3	4	5	6	7	8
4. Скорость работы	0,05	4	3	5	0,2	0,15	0,25
5. Надёжность	0,25	4	3	3	1	0,75	0,75
6. Компактность	0,2	5	2	2	1	0,4	0,4
7. Устойчивость к внешним воздействиям	0,1	4	3	2	0,4	0,3	0,2
Экономические критерии оценки эффективности							
1. Цена на расходы	0,025	5	5	5	0,125	0,125	0,125
2. Конкурентоспособность исследовательской работы	0,025	3	3	3	0,075	0,075	0,085
3. Финансирование научного исследования	0,05	4	4	4	0,2	0,2	0,1
Итого	1	42	33	34	4,15	2,8	2,8

Анализ конкурентных технических решений определяется по формуле:

$$K = \sum B_i \cdot B_j$$

где K – конкурентоспособность научной разработки или конкурента;

B_i – вес показателя (в долях единицы);

B_j – балл i -го показателя.

По результатам, которые предоставляет оценочная карта можно сделать вывод, что наиболее конкурентоспособным решением будет разработка собственного настольного приложения. Основными преимущественными факторами являются: совмещение данного модуля с отечественным ПО и подключение, к продуктовым технологиям, а также возможность добавления нового функционала, отсутствующего у конкурентов.

4.2.2. SWOT-анализ

Для исследования внешней и внутренней среды проекта, в этой работе проведен SWOT-анализ с детальной оценкой сильных и слабых сторон исследовательского проекта, а также его возможностей и угроз.

Первый этап, составляется матрица SWOT, в которую описаны слабые и сильные стороны проекта и выявленные возможности и угрозы для реализации проекта, которые проявились или могут появиться в его внешней среде, приведены в таблице:

Таблица 2 – Матрица SWOT-анализа

Сильные стороны	Слабые стороны
С1. Заявленная экономичность и энергоэффективность технологии.	Сл1. Отсутствие пользовательского тестирования продукта
С2. Более точные результаты исследования по сравнению с другими технологиями.	Сл2. Не достаточное количество исполнителей
С3. Своя технология с возможностью переиспользования на других платформах.	Сл3. Маленькая оплата труда
С4. Дешевизна разработки	Сл4. Не большой опыт в данной сфере
С5. Квалифицированный персонал.	Сл5. Отсутствие аналитики во время реализации функционала
Возможности	Угрозы
В1. Появление дополнительного спроса на полученные результаты исследования в учебной, научной и коммерческой сфере	У1. Срыв проекта из-за отсутствия исполнителей
В2. Появление потенциального спроса на новые разработки в сфере геоинженерии.	У2. Появление или существование зарубежных аналогов и более ранний их выход на рынок.
В3. Легкое внедрение технологии в коммерческой области.	У3. Уход исполнителей
В4. Продажа технологии более крупному производителю	У4. Отсутствие спроса на новые результаты исследования в связи с не большой решаемой задачей.
В5. Привлечение новых исполнителей для выполнения всего комплекса работ	У5. Потеря актуальности в связи с моно-платформенностью приложения.

На втором этапе на основании матрицы SWOT строятся интерактивные матрицы возможностей и угроз, позволяющие оценить эффективность проекта, а также надёжность его реализации. Соотношения параметров представлены в таблицах 3–6:

Таблица 3 – Интерактивная матрица проекта «Возможности проекта и сильные стороны».

Сильные стороны проекта						
Возможности проекта		С1	С2	С3	С4	С5
	В1	-	+	+	-	-
	В2	-	-	+	-	-
	В3	-	-	+	+	+
	В4	+	-	-	-	-
	В5	-	-	-	-	+

Таблица 4 – Интерактивная матрица проекта «Возможности проекта и слабые стороны».

Слабые стороны проекта						
Возможности проекта		Сл1	Сл2	Сл3	Сл4	Сл5
	В1	+	+	-	-	+
	В2	+	+	-	-	+
	В3	+	-	+	-	+
	В4	+	-	+	+	+
	В5	-	+	-	+	-

Таблица 5 – Интерактивная матрица проекта «Угрозы проекта и сильные стороны».

Сильные стороны проекта						
Угрозы проекта		C1	C2	C3	C4	C5
	У1	-	-	-	-	+
	У2	-	-	-	-	-
	У3	-	-	-	-	-
	У4	-	-	-	-	-
	У5	-	-	+	-	-

Таблица 6 – Интерактивная матрица проекта «Угрозы проекта и слабые стороны».

Слабые стороны проекта						
Угрозы проекта		Сл1	Сл2	Сл3	Сл4	Сл5
	У1	+	+	+	+	+
	У2	-	-	-	-	-
	У3	-	+	+	+	-
	У4	-	-	-	-	-
	У5	-	-	-	-	+

В рамках третьего этапа должна быть составлена итоговая матрица SWOT-анализа, которая приводится в работе.

Таблица 7 - SWOT-анализ

	<p>Сильные стороны научно-исследовательского проекта: С1. Заявленная экономичность и энергоэффективность технологии. С2. Более точные результаты исследования по сравнению с другими технологиями. С3. Своя технология с возможностью пере использования на других платформах. С4. Дешевизна разработки С5. Квалифицированный персонал.</p>	<p>Слабые стороны научно-исследовательского проекта: Сл1. Отсутствие пользовательского тестирования продукта Сл2. Не достаточное количество исполнителей Сл3. Маленькая оплата труда Сл4. Не большой опыт в данной сфере Сл5. Отсутствие аналитики во время реализации функционала</p>
<p>Возможности: В1. Появление дополнительного спроса на полученные результаты исследования в учебной, научной и коммерческой сфере В2. Появление потенциального спроса на новые разработки в сфере геоинженерии. В3. Легкое внедрение технологии в коммерческой области. В4. Продажа технологии более крупному производителю В5. Привлечение новых исполнителей для выполнения всего комплекса работ всего комплекса работ</p>	<p>Наилучший исход событий это получение дополнительных ресурсов в случае если технология окажется интересна каким либо инвесторам/покупателям</p>	<p>Главная проблема - это отсутствие достаточного финансирования для дальнейшей разработки, причина которого является отсутствие заказчика с большим списком требований к ПО.</p>
<p>Угрозы: У1. Срыв проекта из-за отсутствия исполнителей</p>	<p>Все сильные стороны проекта, а также специфика работы компенсируют угрозы, которые</p>	<p>Для устранения всех угроз и слабых сторон, необходимо сформулировать сегмент рынка</p>

<p>У2. Появление или существование зарубежных аналогов и более ранний их выход на рынок. У3. Уход исполнителей У4. Отсутствие спроса на новые результаты исследования в связи с не большой решаемой задачей. У5. Потеря актуальности в связи с моно-платформенностью приложения.</p>	<p>существуют из вне. Так как работа является студенческим проектом и не носит большого коммерческого характера.</p>	<p>гео-инженеров и гео-компаний, для которых будет разрабатываться ПО, а также организовать постоянную группу обратной связи для понимания БП инженеров в ходе работы.</p>
---	--	--

Вывод SWOT-анализу: в ходе анализа были рассмотрены все сильные и слабые стороны научного проекта, а также разьяснены все его последствия. Как позитивные, так и негативные – возможности и угрозы. SWOT–анализ показал что, данная разработка может иметь два координально отличающихся исхода событий. Первый это очень хороший вариант, при котором к данной разработке появится интерес у сторонних производителей, исследователей. Что приведет к увеличению финансирования и появлению новых кадров, которые ускорят разработку либо при помощи своей высокой квалификации, либо простым увеличением количеством исполнителей, что позволит более быстро проводить некоторые этапы разработки. Второй вариант это тот при котором исследование не найдет должного финансирования, и будет либо остановлено, либо будет преобразовано в базу знаний для другого проекта.

4.3. Планирование научно-исследовательских работ

4.3.1. Структура работ в рамках научного исследования

Планирование комплекса научно-исследовательских работ осуществляется в порядке:

- определение структуры работ в рамках научного исследования;
- определение количества исполнителей для каждой из работ;
- установление продолжительности работ;
- построение графика проведения научных исследований;

Для оптимизации работ удобно использовать классический метод линейного планирования и управления.

Результатом такого планирования является составление линейного графика выполнения всех работ. Порядок этапов работ и распределение исполнителей для данной научно-исследовательской работы, приведен в таблице:

Таблица 8 – Перечень этапов, работ и распределение исполнителей

Основные этапы	№ раб	Содержание работ	Должность исполнителя
Разработка темы диссертации	1	Составление и утверждение темя диссертации, утверждение плана-графика.	Научный руководитель
	2	Календарное планирование выполнения диссертации.	Инженер, Научный руководитель
Теоретические исследования	3	Изучение академической литературы и конкурентных решений в данной сфере.	Инженер
	4	Изучение программного обеспечения для расчетов.	Инженер
Экспериментальные исследования	5	Технический расчет оценки времени разработки.	Инженер
	6	Разработка библиотеки для работы с SLD структурой.	Инженер
	7	Написание Desktop приложения для редактирования стилей.	Инженер
	8	Адаптация GUI под обычного пользователя.	Инженер
	9	Тестирование.	Инженер
Оформление отчета по НИР	10	Составление пояснительной записки.	Инженер

4.3.2. Определение трудоемкости выполнения работ и разработка графика проведения

При проведении научных исследований основную часть стоимости разработки составляют трудовые затраты, поэтому определение трудоемкости проводимых работ является важным этапом составления бюджета.

Для определения ожидаемого (среднего) значения трудоемкости использована следующая формула:

$$t_{ожи} = \frac{3t_{мини} + 2t_{макси}}{5} \quad (4,2)$$

Где:

$t_{ожи}$ – ожидаемая трудоемкость выполнения i -ой работы, человеко-дни;

$t_{мини}$ – минимально возможная трудоемкость выполнения заданной i -ой работы, человеко-дни;

$t_{макси}$ – максимально возможная трудоемкость выполнения заданной i -ой работы, человеко-дни;

Зная величину ожидаемой трудоемкости, можно определить продолжительность каждой i -ой работы в рабочих днях T_{pi} , при этом учитывается параллельность выполнения работ разными исполнителями. Данный расчёт позволяет определить

величину заработной платы.

$$T_{Pi} = \frac{t_{ожі}}{Ч_i} \quad (4,3)$$

Где:

T_{Pi} – продолжительность одной работы, рабочие дни;

$t_{ожі}$ – ожидаемая трудоемкость выполнения одной работы, человеко-дни;

$Ч_i$ – численность исполнителей, выполняющих одновременно одну и ту же работу на данном этапе, чел;

Для перевода длительности каждого этапа из рабочих в календарные дни, необходимо воспользоваться формулой:

$$T_{ki} = T_{Pi} \times k \quad (4,4)$$

Где:

T_{ki} – продолжительность выполнения i -й работы в календарных днях;

T_{Pi} – продолжительность выполнения i -й работы в рабочих днях;

k – календарный коэффициент.

Календарный коэффициент определяется по формуле:

$$k = \frac{T_{\text{кал}}}{T_{\text{кал}} - T_{\text{вых}} - T_{\text{праз}}} = \frac{365}{365 - 89 - 29} = 1,48 \quad (4,5)$$

Где:

$T_{\text{кал}}$ – общее количество календарных дней в году;

$T_{\text{вых}}$ – общее количество выходных дней в году;

$T_{\text{праз}}$ – общее количество праздничных дней в году;

Расчеты временных показателей проведения научного исследования представляются в таблице:

Таблица 9 – Временные показатели проведения научного исследования.

Название работы	Трудоёмкость работ						Длительность работ в рабочих днях T_{Pi}	Длительность работ в календарных днях T_{ki}
	t_{mini} , чел-дни		t_{maxi} , чел-дни		$t_{ожі}$, чел-дни			
	Исп.1	Исп.2	Исп.1	Исп.2	Исп.1	Исп.2		
1	2	3	4	5	6	7	8	9
1. Составление и утверждение темя диссертации, утверждение плана-графика.	1	-	2	-	1,4	-	1,4	2

2. Календарное планирование выполнения диссертации.	1	2	2	3	1,4	2	2	3
3. Изучение академической литературы и конкурентных решений в данной сфере	-	40	-	60	-	48	48	71
4. Изучение программного обеспечения для расчетов	-	5	-	8	-	6,2	6,2	9
5. Технический расчет оценки времени разработки.	-	5	-	8	-	6,2	6,2	9
6. Разработка библиотеки для работы с SLD структурой.	-	60	-	80	-	68	68	75
7. Написание Desktop приложения для редактирования стилей.	-	50	-	70	-	58	58	86
8. Адаптация GUI под обычного пользователя.	-	40	-	50	-	44	44	65
9. Тестирование.	3	8	5	10	3,8	8,8	8,8	9
10. Составление пояснительной записки.	-	20	-	40	-	28	28	41
Итого:	5	230	9	319	5,6	269,2	250,8	370

Примечание:

Исп.1 – научный руководитель; Исп.2 – инженер;

На основе таблицы составлен календарный план-график выполнения проекта с использованием диаграммы Ганта (см. Приложение А).

4.3.3. Бюджет научно-технического исследования

При планировании бюджета научно-технического исследования учитывались все

виды расходов, связанных с его выполнением. В этой работе использовать следующую группировку затрат по следующим статьям:

- материальные затраты научно-исследовательской работы (НИР);
- затраты на специальное оборудование для экспериментальных работ;
- основная заработная плата исполнителей темы;
- дополнительная заработная плата исполнителей темы;
- отчисления во внебюджетные фонды (страховые отчисления);
- накладные расходы НИР.

4.3.4. Расчет материальных затрат научно-технического исследования

Данная часть включает затрат всех материалов, используемых при исследовании. Результаты расчета затрат представлены в таблице.

Таблица 10 – Материальные затраты.

Наименование материалов	Цена за ед., руб.	Кол-во, ед.	Сумма, руб.
Блокнот	400	1	400
Ручка	70	2	140
Итого:			540

4.3.5. Расчет амортизации специального оборудования

Расчет сводится к определению амортизационных отчислений, так как оборудование было приобретено до начала выполнения данной работы и эксплуатировалось ранее, поэтому при расчете затрат на оборудовании учитываем только рабочие дни по данной теме.

Расчет амортизации проводится следующим образом:

Норма амортизации: рассчитывается по формуле:

$$H_A = \frac{1}{n} \quad (4,6)$$

Где: n – срок полезного использования в количестве лет.

Амортизация оборудования рассчитывается по формуле:

$$A = \frac{H_A I}{12} \times t \quad (4,7)$$

Где:

I – итоговая сумма, тыс. руб.; t – время использования, мес.

Таблица 11 – Затраты на оборудование.

№	Наименование	Кол-во,	Срок полезного	Цены единицы	Общая
---	--------------	---------	----------------	--------------	-------

	оборудования	шт.	использования, лет	оборудования, тыс. руб.	стоимость оборудования, тыс. руб.
1	Персональный компьютер (ноутбук)	1	4	50	50
Итого		50 тыс. руб.			

Рассчитывается норма амортизации для ноутбука, с учётом того, что срок полезного использования составляет 4 года (по формуле 4.6):

$$H_A = \frac{1}{n} = \frac{1}{4} = 0,25$$

Общая сумма амортизационных отчислений (по формуле 4.7):

$$A = \frac{H_A I}{12} \times m = \frac{0,25 \times 50000}{12} \times 13 = 13500 \text{ руб}$$

4.3.6. Основная заработная плата исполнителей темы

В данном разделе рассчитывается заработная плата инженера и руководителя, помимо этого необходимо рассчитать расходы по заработной плате, определяемые трудоемкостью проекта и действующей системой оклада.

Основная заработная плата одного работника рассчитывается по следующей формуле:

$$Z_{\text{осн}} = Z_{\text{дн}} \times T_p \quad (4,8)$$

Где:

$Z_{\text{дн}}$ – среднедневная заработная плата, руб.;

T_p – продолжительность работ, выполняемых работником, раб. дней. (по таблице 4.9 для инженера: $T_{p2} = 270$ дней, для руководителя: $T_{p1} = 6$ дней).

Среднедневная заработная плата рассчитывается по формуле 4.9:

$$Z_{\text{дн}} = \frac{Z_m \times M}{F_d} \quad (4,9)$$

Где:

Z_m – месячный должностной оклад работника, руб.;

F_d – действительный годовой фонд рабочего времени научно-технического персонала, раб. Дней (в данном случае $F_d = 247$ дней);

M – количество месяцев работы без отпуска в течение года (при отпуске в 118 раб. дней, $M = 8,1$ месяц, 6-дневная рабочая неделя);

Должностной оклад работника за месяц определяется по формуле 4.10:

$$Z_m = Z_{mc} \times (1 + k_{\text{пр}} + k_d) \times k_p \quad (4,10)$$

Где:

Z_{mc} – заработная плата, согласно тарифной ставке, руб (для руководителя $Z_{mc1} = 28600$ руб, а для инженера $Z_{mc2} = 6500$ руб);

$k_{пр}$ – премиальный коэффициент, равен 0,3;

k_d – коэффициент доплат и надбавок, равен 0,2;

k_p – районный коэффициент, равен 1,3 (для г. Томск);

По формуле 4.10 определяется должностной оклад руководителя за месяц:

$$Z_{m1} = Z_{mc1} \times (1 + k_{пр} + k_d) \times k_p = 28600 \times (1 + 0,3 + 0,2) \times 1,3 = 55770 \text{ руб}$$

По формуле 4,10 определяется должностной оклад инженера за месяц:

$$Z_{m2} = Z_{mc2} \times (1 + k_{пр} + k_d) \times k_p = 6500 \times (1 + 0,3 + 0,2) \times 1,3 = 12675 \text{ руб}$$

Среднедневная заработная плата у руководителя рассчитывается по формуле 4.9:

$$Z_{дн1} = \frac{Z_m \times M}{F_d} = \frac{55770 \times 8,1}{247} = 1828,89 \text{ руб}$$

Среднедневная заработная плата у инженера рассчитывается по формуле 4.9:

$$Z_{дн2} = \frac{Z_m \times M}{F_d} = \frac{12675 \times 8,1}{247} = 415,65 \text{ руб}$$

Основная заработная плата руководителя рассчитывается по формуле 4,8:

$$Z_{осн1} = Z_{дн1} \times T_{p1} = 1828,89 \times 6 = 10973,34 \text{ руб}$$

Основная заработная плата инженера рассчитывается по формуле 4,8:

$$Z_{осн2} = Z_{дн2} \times T_{p2} = 415,65 \times 270 = 112225,5 \text{ руб}$$

Таким образом, затраты на общую основную заработную плату составляют:

$$Z_{осн\text{ обще}} = Z_{осн1} + Z_{осн2} = 10973,34 + 112225,5 = 123198,39 \text{ руб}$$

Перечисленные информации представляются в таблице:

Таблица 12 – Расчеты основной заработной платы исполнителей.

Исполнители НИ	Z_{mc} , руб	$k_{пр}$	k_d	k_p	Z_m , руб	$Z_{дн}$, руб	T_p , дн	$Z_{осн}$, руб
Руководитель	28600	0,3	0,2	1,3	55770	1828,89	6	10973,34
Инженер	6500	0,3	0,2	1,3	12675	415,65	270	112225,5
Итого:								123198,39

4.3.7. Дополнительная заработная плата исполнителей темы

Дополнительная заработная плата определяется по формуле 4.11:

$$Z_{доп} = k_{доп} \times Z_{осн} \quad (4,11)$$

Где:

$Z_{доп}$ – дополнительная заработная плата;

$Z_{осн}$ – основная заработная плата;

$k_{\text{доп}}$ – коэффициент дополнительной заработной платы (на стадии проектирования принимаем равным 0,15);

По формуле 4.11 определяется дополнительная заработная плата для руководителя:

$$З_{\text{доп1}} = k_{\text{доп}} \times З_{\text{осн}} = 0,15 \times 10973,34 = 1\,646 \text{ руб}$$

По формуле 4.11 определяется дополнительная заработная плата для инженера:

$$З_{\text{доп2}} = k_{\text{доп}} \times З_{\text{осн}} = 0,15 \times 112227,6 = 16834,14 \text{ руб}$$

Таким образом, общая дополнительная заработная плата составляет:

$$З_{\text{доп общ}} = З_{\text{доп1}} + З_{\text{доп2}} = 1\,646 + 16834,14 = 18480,14 \text{ руб}$$

4.3.8. Отчисления во внебюджетные фонды (страховые отчисления)

Отчисления во внебюджетные фонды определяются по формуле 4.12:

$$З_{\text{внеб1}} = k_{\text{внеб}} \times (З_{\text{осн1}} + З_{\text{доп1}}) \quad (4,12)$$

Где:

$k_{\text{внеб}}$ – коэффициент отчислений на уплату во внебюджетные фонды (пенсионный фонд, фонд ОМС и социальное страхование). Общая ставка взносов составляет в 2021 году –30% (ст. 425 НК РФ).

Отчисления во внебюджетные фонды для руководителя определяются по формуле 4.12:

$$З_{\text{внеб1}} = k_{\text{внеб}} \times (З_{\text{осн1}} + З_{\text{доп1}}) = 0,3 \times (10973,34 + 1646) = 3785,8 \text{ руб}$$

Отчисления во внебюджетные фонды для инженера определяются по формуле 4.12:

$$З_{\text{внеб2}} = k_{\text{внеб}} \times (З_{\text{осн2}} + З_{\text{доп2}}) = 0,3 \times (112225,5 + 16834,14) = 42\,503,694 \text{ руб}$$

Таким образом, общие затраты на составляется отчисления во внебюджетные фонды:

$$З_{\text{внеб общ}} = З_{\text{внеб1}} + З_{\text{внеб2}} = 3785,802 + 42503,69 = 46289,49 \text{ руб}$$

4.3.9. Накладные расходы

Накладные расходы учитывают прочие затраты организации, не попавшие в предыдущие статьи расходов: печать и ксерокопирование материалов исследования, оплата услуг связи, электроэнергии, почтовые и т.д.

Величина накладных расходов определяется по формуле 4.13:

$$З_{\text{накл}} = (\text{сумма статей } 1 \div 5) k_{\text{нр}} \quad (4,13)$$

Где:

$k_{\text{нр}}$ – коэффициент, учитывающий накладные расходы. Величина коэффициента

принимается равной 0,16.

$$Z_{\text{накл}} = (540 + 13500 + 123198,39 + 18480,14 + 46289,49) \cdot 0,16 = 149\,959,4 \text{ руб.}$$

4.3.10. Бюджетная стоимость НИР

Группировка затрат по статьям представляется в таблице:

Таблица 13 – Группировка затрат по статьям.

Статьи							
1	2	3	4	5	6	7	8
Материалы, руб	Амортизация, руб	Основная заработная плата, руб	Дополнительная заработная плата, руб	Отчисления на социальные нужды, руб	Итого без накладных расходов, руб	Накладные расходы, руб	Стоимость бюджета, руб
540	13500	123198,39	18480,14	46289,49	202007,39	32321,12	234328,51

4.4. Определение ресурсной (ресурсосберегающей), финансовой, бюджетной, социальной и экономической эффективности исследования

Для определения эффективности исследования рассчитан интегральный показатель эффективности научного исследования путем определения интегральных показателей финансовой эффективности и ресурсоэффективности.

Интегральный показатель финансовой эффективности научного исследования получен в процессе оценки бюджета затрат трех вариантов исполнения научного исследования. Для этого наибольший интегральный показатель реализации технической задачи принят за базу расчета (как знаменатель), с которым соотносятся финансовые значения по всем вариантам исполнения.

Интегральный финансовый показатель разработки рассчитывается по формуле 4.14:

$$I_{\text{финр}}^{\text{исп.}i} = \frac{\Phi_{ri}}{\Phi_{\text{max}}} \quad (4,14)$$

Где:

$I_{\text{финр}}^{\text{исп.}i}$ – интегральный финансовый показатель разработки;

Φ_{ri} – стоимость i -го варианта исполнения;

Φ_{max} – максимальная стоимость исполнения из всех вариантов;

По перечисленным вычислениям определяются общие затраты для всех вариантов:

$$\Phi_{\text{текущ.проект}} = 234328,51 \text{ руб.},$$

$$\Phi_{\text{исп.1}} = 234328,51 \text{ руб}$$

$$\Phi_{\text{max}} = \Phi_{\text{исп.1}} = 234328,51 \text{ руб}$$

По формуле 4,14 определяется интегральный финансовый показатель для текущего

проекта:

$$I_{\text{финр}}^{\text{текущ.проект}} = \frac{\Phi_{\text{текущ.проект}}}{\Phi_{\text{max}}} = \frac{234328,51}{234328,51} = 1$$

В результате расчетов интегральных финансовых показателей по трем вариантам разработки текущий проект с меньшим перевесом признан считается более приемлемым с точки зрения финансовой эффективности.

Интегральные показатели ресурсоэффективности всех вариантов определяются путем сравнительной оценки их характеристик, распределенных с учетом весового коэффициента каждого параметра.

Таблица 14 – Сравнительная оценка характеристик всех вариантов

Объекты исследования Критерии	Весовой коэффициент параметра	Текущий проект	Исп.2	Исп.3
1. Безопасность при использовании ПО	0,3	5	5	4
2. Стабильность работы	0,15	4	4	5
3. Функциональные возможности	0,2	5	4	4
4. Масштабируемость	0,2	5	4	3
5. Удобство использования	0,15	5	5	4
ИТОГО	1	4,8	4,4	4

По данным из таблицы 4.17 определяется интегральный показатель ресурсоэффективности для текущего проекта:

$$I_p^{\text{текущ.проект}} = 0,3 \times 5 + 0,15 \times 4 + 0,2 \times 5 + 0,2 \times 5 + 0,15 \times 5 = 4,85$$

По данным из таблицы 4.17 определяется интегральный показатель ресурсоэффективности для первого конкурентного проекта:

$$I_p^{\text{исп.1}} = 0,3 \times 5 + 0,15 \times 4 + 0,2 \times 4 + 0,2 \times 4 + 0,15 \times 5 = 4,45$$

По данным из таблицы 4.17 определяется интегральный показатель ресурсоэффективности для второго конкурентного проекта:

$$I_p^{\text{исп.2}} = 0,3 \times 4 + 0,15 \times 5 + 0,2 \times 4 + 0,2 \times 3 + 0,15 \times 4 = 3,95$$

В результате расчетов интегральных показателей ресурсоэффективности по трем вариантам разработки текущий проект с большим перевесом признан считается более приемлемым с точки зрения ресурсной эффективности.

Интегральные показатели эффективности всех вариантов вычисляются на основании показателей ресурсоэффективности и интегральных финансовых показателей по формуле 4.15:

$$I_{\text{эф.}i} = \frac{I_p^{\text{исп.}i}}{I_{\text{финр}}^{\text{исп.}i}} \quad (4,15)$$

Где:

$I_{\text{эф.}i}$ – интегральный показатель эффективности i -ого варианта разработки;

$I_p^{\text{исп.}i}$ – интегральный показатель ресурсной эффективности i -ого варианта разработки;

$I_{\text{финр}}^{\text{исп.}i}$ – интегральный финансовый показатель i -ого варианта разработки;

По формуле 4.15 определяется интегральный показатель эффективности для текущего проекта:

$$I_{\text{эф.текущ.проект}} = \frac{I_p^{\text{текущ.проект}}}{I_{\text{финр}}^{\text{текущ.проект}}} = \frac{4,85}{0,043} = 112,79$$

По формуле 4.15 определяется интегральный показатель эффективности для первого конкурентного проекта:

$$I_{\text{эф.исп.1}} = \frac{I_p^{\text{исп.1}}}{I_{\text{финр}}^{\text{исп.1}}} = \frac{4,45}{1} = 4,45$$

По формуле 4.15 определяется интегральный показатель эффективности для второго конкурентного проекта:

$$I_{\text{эф.исп.2}} = \frac{I_p^{\text{исп.2}}}{I_{\text{финр}}^{\text{исп.2}}} = \frac{3,95}{0,047} = 84,04$$

Далее среднее значение интегрального показателя эффективности каждого варианта НИР сравнивалось с средним значением интегрального показателя эффективности текущего проекта с целью определения сравнительной эффективности проектов.

Таблица 15 – Сравнительные эффективности разработок.

№ п/п	Показатели	Текущий проект	Исп.2	Исп.3
1	Интегральный финансовый показатель разработки	0,043	1	0,047
2	Интегральный показатель ресурсоэффективности разработки	4,85	4,45	3,95
3	Интегральный показатель эффективности	112,79	4,45	84,04
4	Сравнительная эффективность вариантов исполнения	1	0,084	0,748

Сравнение среднего интегрального показателя сопоставляемых вариантов позволило сделать вывод о том, что наиболее финансовым и ресурсным эффективным вариантом является текущий проект. Наш проект является более эффективным по сравнению с конкурентами.

4.5. Вывод

В ходе работы был проведен экономический анализ проекта, посчитаны разного рода затраты, зарплатные отчисления, составлен SWOT анализ, выявлены сильные и слабые стороны проекта, выявлены конкуренты и произведено сравнение с похожими конкурирующими разработками, составлен календарный рейтинг план с расчетом трудозатрат, а так же построена диаграмма Ганта на основе этих данных.

5. Социальная ответственность

ЗАДАНИЕ ДЛЯ РАЗДЕЛА «СОЦИАЛЬНАЯ ОТВЕТСТВЕННОСТЬ»

Студенту:

Группа		ФИО	
8К93		Жабин Ефим Михайлович	
Школа	ИШИТР	Отделение (НОЦ)	ОИТ
Уровень образования	Бакалавриат	Направление/специальность	09.03.04 Программная инженерия

Тема ВКР:

Создание редактора стилей для ПО Geoserver.

Исходные данные к разделу «Социальная ответственность»:

<p>Введение</p> <ul style="list-style-type: none"> – Характеристика объекта исследования (вещество, материал, прибор, алгоритм, методика) и области его применения. – Описание рабочей зоны (рабочего места) при разработке проектного решения/при эксплуатации 	<p><i>Объект исследования:</i> настольное приложение, для редактирование стилей ПО Geoserver, в дружелюбном графическом пользовательском интерфейсе.</p> <p><i>Область применения:</i> проектировщики, системные администраторы, дизайнеры</p> <p><i>Рабочая зона:</i> офис</p> <p><i>Размеры помещения:</i> 10*15 м².</p> <p><i>Количество и наименование оборудования рабочей зоны:</i> компьютерный стол, компьютер, компьютер с сервисом, компьютерная периферия.</p> <p><i>Рабочие процессы, связанные с объектом исследования, осуществляющиеся в рабочей зоне:</i> обсуждение внутри группы, разработка с использованием компьютера.</p>
--	--

Перечень вопросов, подлежащих исследованию, проектированию и разработке:

<p>1. Правовые и организационные вопросы обеспечения безопасности при разработке проектного решения:</p> <ul style="list-style-type: none"> – специальные (характерные при эксплуатации объекта исследования, проектируемой рабочей зоны) правовые нормы трудового законодательства; – организационные мероприятия при компоновке рабочей зоны. 	<p>Трудовой кодекс РФ; СанПиН 1.2.3685-21. Гигиенические нормативы и требования к обеспечению безопасности и (или) безвредности для человека факторов среды обитания ТК РФ Статья 91. Понятие рабочего времени. Нормальная продолжительность рабочего времени; ГОСТ 12.2.032-78. Система стандартов безопасности труда. Рабочее место при выполнении работ сидя.</p>
<p>2. Производственная безопасность при разработке проектного решения:</p> <ul style="list-style-type: none"> – Анализ выявленных вредных и опасных производственных 	<p>Вредные:</p> <ol style="list-style-type: none"> 1. Длительное сосредоточенное наблюдение 2. Отсутствие или недостаток необходимого искусственного освещения; 3. Повышенный уровень шума на рабочем месте; 4. Нагрузка на зрительный аппарат;

факторов	<p>5. Монотонный режим работы.</p> <p>Опасные:</p> <p>1. Факторы, связанные с электрическим током, вызываемым разницей электрических потенциалов, под действие которого попадает рабочий.</p> <p>Требуемые средства коллективной защиты от выявленных факторов: системы естественного освещения, приборы искусственного освещения, изоляционные средства, предохранительные устройства.</p>
<p>3. Экологическая безопасность при эксплуатации</p>	<p>Воздействие на селитебную зону: не выявлено</p> <p>Воздействие на литосферу: загрязнение почв веществами от компьютерной техники, батареек и других элементов питания</p> <p>Воздействие на гидросферу: из-за неверного способа утилизации рабочей техники</p> <p>Воздействие на атмосферу: не выявлено</p>
<p>4. Безопасность в чрезвычайных ситуациях при разработке проектного решения</p>	<p>Возможные ЧС:</p> <p>Природные (землетрясения, лесные пожары и т.д.);</p> <p>Геофизические (землетрясения и т.д.);</p> <p>Техногенные (транспортные аварии, пожары, аварии с выбросом химически/радиоактивно опасных веществ и т.д.);</p> <p>Биолого-социального (пандемия)</p> <p>Наиболее типичная ЧС: пожар</p>
<p>Дата выдачи задания для раздела по линейному графику</p>	

Задание выдал консультант:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Старший преподаватель	Мезенцева Ирина Леонидовна			

Задание принял к исполнению студент:

Группа	ФИО	Подпись	Дата
8К93	Жабин Ефим Михайлович		

5.1. Введение

В рамках выпускной квалификационной работы было разработано настольное приложение, позволяющие пользователям с помощью дружелюбного графического пользовательского интерфейса, редактировать файлы стилей формата XML, содержащие внутри себя пространство имен SLD. Пользователи – люди и организации, которые занимаются инженерной проектировочной деятельностью, по разработке карт местностей, для которых необходимо отредактировать файлы стилей, являющиеся конфигурационными файлами с параметрами отображения геометрии.

Данное настольное приложение разрабатывалось во время прохождения производственной практики в ООО «ТОМСКАСУПРОЕКТ», место проведения работ – офис компании. Проектируемое рабочее место представляет собой офис, в котором будет работать разработчик.

Характеристика помещения: площадь – 72 м²; объем – 288 м³; в помещении установлен кондиционер, имеется естественная вентиляция – вытяжное вентиляционное отверстие, дверь, окно, щели; в помещении установлено искусственное освещение, имеется естественное освещение.

В данном помещении максимальное количество сотрудников в одну смену – 12. В среднем на одного сотрудника приходится 6 м² площади и около 24 м³ объема помещения. Данное размещение сотрудников удовлетворяет санитарным нормам, согласно которым на одного работника должно приходиться не менее 6 м² площади и 24 м³.

Оборудование рабочей зоны: компьютерный стол, компьютер с основной периферией, блокнот, ручка.

Рабочие процессы: обсуждение внутри группы, разработка с использованием компьютера.

Целью данной главы является определение и оценка параметров рабочей среды, а также исследование социальной ответственности при работе в офисе и написания приложения.

5.2. Правовые и организационные вопросы обеспечения безопасности

Любая трудовая деятельность в нашем государстве регулируется Трудовым кодексом Российской Федерации [1], в том числе, и деятельность по разработке программного обеспечения. Специфика данной области такова, что для полноценной работы достаточно иметь лишь персональный компьютер и стабильный доступ в сеть

Интернет. Но также немаловажными факторами производительности сотрудника являются удобство его рабочего места, режим рабочего дня, атмосфера в коллективе и т.д.

Можно выделить основные положения для выполнения:

- Нормальная продолжительность рабочего времени не может превышать 40 часов в неделю.
- Всем работникам предоставляются выходные дни (еженедельный непрерывный отдых).
- Организация-работодатель выплачивает заработную плату работникам. Возможно удержание заработной платы только в случаях, установленных ТК РФ ст. 137. В случае задержки заработной платы более чем на 15 дней, работник имеет право приостановить работу, письменно уведомив работодателя.
- Согласно статье 108 Трудового Кодекса РФ, в течение рабочего дня работнику должен быть предоставлен перерыв для отдыха и питания продолжительностью не более двух часов и не менее 30 минут. Правилами внутреннего трудового распорядка или трудовым договором может быть предусмотрено, что указанный перерыв может не предоставляться работнику, если установленная для него продолжительность ежедневной работы (смены) не превышает четырех часов (в ред. Федерального закона от 18.06.2017 N 125ФЗ).

Рабочее место должно быть организовано с учетом требований ГОСТ 12.2.032-78 «Система стандартов безопасности труда (ССБТ). Рабочее место при выполнении работ сидя. Общие эргономические требования» [2]. При организации работы с ЭВМ, согласно указанному выше требованию, должны быть соблюдены следующие условия:

- 1) Конструкция рабочей мебели должна обеспечивать возможность индивидуальной регулировки соответственно росту пользователя и создавать удобную позу для работы;
- 2) Рабочее место должно быть установлено так, чтобы, подняв глаза от экрана, можно было увидеть какой-нибудь удаленный предмет в помещении или на улице, таким образом, предоставляя эффективный способ разгрузки зрительного аппарата;
- 3) Окна в помещениях с ПК должны быть оборудованы регулируемыми устройствами – жалюзи, занавески, внешние козырьки;
- 4) Помещения, где размещаются рабочие места с ПЭВМ, должны быть оборудованы защитным заземлением (занулением) в соответствии с техническими требованиями по эксплуатации;

- 5) В помещениях, оборудованных ПЭВМ, проводится ежедневная влажная уборка и систематическое проветривание после каждого часа работы на ЭВМ;
- 6) Высота рабочего стола с клавиатурой должна составлять 680 - 800 мм над уровнем стола;
- 7) Высота экрана над полом – 900-1280 мм, монитор должен находиться на расстоянии 600-700 мм от работника на 20 градусов ниже уровня глаз;
- 8) Положение тела пользователя относительно монитора должно соответствовать направлению просмотра под прямым углом 90 градусов или под углом 75 градусов.

При выполнении выпускной квалификационной работы правовых и организационных нарушений по указанным требованиям не было выявлено, рабочее место было оборудовано согласно всем нормам и правилам.

5.3. Производственная безопасность

Согласно производственным факторам ГОСТ 12.0.003-2015 ССБТ [3], на оператора ПЭВМ в течение рабочего дня воздействует множество различных производственных факторов, каждый из которых влияет на производительность, работоспособность и физическое состояние.

В процессе работы химические и биологические факторы не оказывают влияния на состояние здоровья, поэтому рассмотрим только физические и психофизиологические факторы.

К вредным и опасным производственным факторам можно отнести:

- 1) Производственные факторы, связанные с аномальными микроклиматическими параметрами воздушной среды на местонахождении работающего.
- 2) Отсутствие или недостаток необходимого искусственного освещения.
- 3) Монотонность труда, вызывающая монотонию.
- 4) Производственные факторы, связанные с электрическим током, вызываемым разницей электрических потенциалов, под действие которого попадает работающий.
- 5) Длительное сосредоточенное наблюдение.

Все выявленные факторы приведены в таблице.

Таблица 16 – Возможные опасные и вредные факторы производственные факторы на рабочем месте за персональным компьютером

Факторы (ГОСТ 12.0.003-2015)	Нормативные документы
1. Длительное сосредоточенное наблюдение	СанПиН 1.2.3685-21 [4]
2. Отсутствие или недостаток необходимого освещения	СП 52.13330.2016 [5]
3. Повышенный уровень шума на рабочем месте	ГОСТ 26329-84 [9]
4. Нагрузка на зрительный аппарат	ГОСТ 12.2.032-80 [10]
5. Монотонный режим работы	Трудовой кодекс Российской Федерации от 30.12.2001 N 197-ФЗ (ред. от 25.02.2022) (с изм. и доп., вступ. в силу с 01.03.2022)
6. Факторы, связанные с электрическим током, вызываемым разницей электрических потенциал	Правила по охране труда при эксплуатации электроустановок [8]

5.3.1. Длительное сосредоточенное наблюдение

Работа на ПК сопровождается постоянным и значительным напряжением функций зрительного анализатора. Одной из основных особенностей является иной принцип чтения информации, чем при обычном чтении. Чтобы снизить зрительное напряжение нужно соблюдать допустимые уровни ультрафиолетового излучения для мониторов, которые регламентируются в соответствии с пунктом 136 СанПиН 1.2.3685-21 «Гигиенические нормативы и требования к обеспечению безопасности и (или) безвредности для человека факторов среды обитания».

Таблица 17 – Допустимые уровни ультрафиолетового излучения

Вид изделий	Спектральный диапазон длин волн, нм	Допустимая интенсивность облучения, Вт/м ²
Экраны телевизоров, видеомониторов, осциллографов измерительных и других приборов, средств отображения информации с визуальным контролем	Свыше 315 до 400	Не более 0,1
	Свыше 280 до 315	Не более 0,0001
	От 200 до 280	Не допускается

5.3.2. Отсутствие или недостаток необходимого искусственного освещения

При выполнении работы по разработке настольного приложения было

использовано в основном искусственное освещение.

Недостаточная освещенность приводит к понижению работоспособности, а также может вызвать проблемы со здоровьем, а именно может повлиять на качество зрения работников.

Согласно СП 52.13330.2016 зрительную работу разработчика программного обеспечения можно характеризовать как работу разряда Б – высокой точности (наименьший эквивалентный размер объекта различения составляет 0,3-0,5 мм), подразряда 1 (относительная продолжительность зрительной работы при направлении зрения на рабочую поверхность не менее 70%). В таблице представлены требования к освещению рабочего помещения для указанного разряда.

Таблица 18 - Требования к освещению рабочего помещения для разряда Б1

Искусственное освещение			
Освещенность на рабочей поверхности от системы общего освещения, лк	Цилиндрическая освещенность, лк	Объединенный показатель дискомфорта, не более	Коэффициент пульсации освещенности, Кп, %, не более
300	100	21	15

Для снижения влияния фактора недостаточной освещенности на рабочем месте необходимо, чтобы уровень естественного освещения и яркость экрана персонального компьютера были приблизительно одинаковыми, так как яркий свет в зоне периферийного зрения заметно увеличивает глазное напряжение и приводит к быстрой утомляемости. Путем решения проблемы недостаточной освещенности помещения может стать расширение оконного проема или установка качественных источников искусственного освещения.

5.3.3. Повышенный уровень шума на рабочем месте

Основными источниками шумов в офисных помещениях являются электроприборы, компьютеры, ноутбуки, транспорт, городской шум на улицах, разговоры людей и другие источники. Допустимый уровень шума – это уровень, который не вызывает у человека значительного беспокойства и существенных изменений показателей функционального состояния систем и анализаторов, чувствительных к шуму.

Беспорядочные звуковые колебания оказывают вредное влияние на организм человека. Реакция на них со стороны нервной системы начинается при уровне 40 дБ. Уже при 35 дБ может наблюдаться нарушение сна. При 70 дБ происходят глубокие изменения

в нервной системе, вплоть до психического заболевания, а также заболевания органов зрения, слуха, изменение состава крови и т.д. Шум снижает производительность труда, особенно при выполнении точных работ, затрудняет восприятие опасности от движущихся машин и механизмов, снижает разборчивость речи. Шум оказывают негативное влияние на организм человека и даже может вызвать шумовую болезнь, которая характеризуется тугоухостью, гипертонией (гипотонией), головными болями.

Предельно допустимые уровни звукового давления в октавных полосах частот, уровни звука и эквивалентные уровни звука для трудовой деятельности программиста, разработанные с учётом категории тяжести и напряжённости труда, представлены в таблице.

Таблица 19 - Предельно допустимые уровни звукового давления для программиста

Наименование устройства	Принцип действия устройства, обуславливающий излучение шума	Условия применения устройств	Допустимый скорректированный уровень звуковой мощности, дБ, не более
Устройства ввода-вывода на перфокарты и перфоленту	Преимущественно электромеханический	Работа в специальных помещениях, рассчитанных на повышенное воздействие шума на рабочих местах (например вычислительные центры)	85
Запоминающие устройства на магнитных дисках, магнитной ленте, устройства вывода на микрофильм	Электронный с небольшим числом механических блоков		75
Процессоры, оперативные запоминающие устройства и устройства управления для ЭВМ	Электронный с принудительным охлаждением		70
Устройства последовательной печати	Преимущественно электромеханический	Работа в нормальных рабочих помещениях	70
Запоминающее устройство на гибких магнитных дисках	Электронный с небольшим числом механических блоков		60
Процессоры, оперативные запоминающие устройства	Электронный с принудительным охлаждением		55

управления для мини-ЭВМ			
Устройства отображения	Электронный с конвекционным охлаждением		50
Клавиатура	Механический		50

Борьба с шумом осуществляется при помощи технических и организационных мероприятий. Они проводятся в соответствии с комплексными планами охраны труда и развития предприятия. Среди мероприятий по борьбе с шумом можно отметить такие, как:

- выявление источников шума;
- проверка эффективности звукоизоляции помещений;
- разработка системы мер снижения уровней шума до регламентированных действующими нормативами;
- организация постоянного контроля за уровнем шума на рабочих местах и в рабочих помещениях, замена или модернизация оборудования и технологий для исключения шумоопасных источников или снижения интенсивности шума от них.

5.3.4. Нагрузка на зрительный аппарат

Работа на ПК сопровождается постоянным и значительным напряжением функций зрительного анализатора.

Чтобы снизить зрительное напряжение, необходимо выбрать такой монитор, параметры которого удовлетворяли бы параметрам таблицы 2. Соблюдение этих норм позволит снизить зрительное напряжение.

Характеристики освещения в зрительно воспринимаемом пространстве воздействуют одновременно на физиологические зрительные функции (зрительную работоспособность) и на психологические зрительные функции (условия комфорта). Таким образом, они могут способствовать производительности труда, безопасности, зрительному комфорту и удовлетворению работающего в зрительно воспринимаемом пространстве.

Освещение рабочего помещения естественным и электрическим светом должно обеспечивать оптимальные условия для выполнения требуемых заданий и для правильного восприятия окружающего пространства, когда рабочий переводит взгляд от выполняемого задания для отдыха или для того, чтобы перейти к другой деятельности. Особые требования могут предъявляться к освещению торговых предприятий, промышленных зданий, бюро, больниц и т.д.

5.3.5. Монотонный режим работы

Эмоциональное состояние сотрудника зависит от центральной нервной системы, нагрузку на которую отражает фактор «напряженность труда». К факторам, характеризующим напряженность труда, относятся: интеллектуальные, сенсорные, эмоциональные нагрузки, степень монотонности нагрузок, режим работы.

Эмоциональные перегрузки и стресс снижают работоспособность сотрудника, и наоборот, повышают конфликтность и раздражительность. Также может развиваться апатия к тому, чем человек занимается, что в затяжной форме перерастает в болезнь, называемую профессиональным выгоранием. Кроме психологического здоровья, в большинстве случаев, страдает и физическое – повышение или понижение давления, расстройства желудочнокишечного тракта, нарушения деятельности сердца.

При первых симптомах психического перенапряжения необходимо:

- дать нервной системе расслабиться;
- рационально чередовать периоды отдыха и работы;
- начать заниматься спортом;
- ложиться спать в одно и то же время;
- в тяжелых случаях обратиться к врачу.

Естественно, что полностью исключить провоцирующие факторы из жизни вряд ли удастся, но можно уменьшить их негативное воздействие, давая нервной системе необходимый отдых.

5.3.6. Правила по охране труда при эксплуатации электроустановок.

Оперативное обслуживание электроустановок должны выполнять работники субъекта электроэнергетики (потребителя электрической энергии), из числа оперативного и оперативно-ремонтного персонала, а также работники из числа административно-технического персонала в случаях предоставления соответствующих прав оперативного (оперативно-ремонтного) персонала, имеющие V группу по электробезопасности при эксплуатации электроустановок напряжением выше 1000 В, IV группу по электробезопасности при эксплуатации электроустановок напряжением до 1000 В.

В электроустановках напряжением выше 1000 В работники из числа оперативного персонала, единолично обслуживающие электроустановки, и старшие по смене должны иметь группу по электробезопасности не ниже IV, остальные работники в смене - группу не ниже III.

Работник, выдающий разрешение на подготовку рабочих мест и на допуск, является ответственным за:

за выдачу команд по отключению и заземлению линий электропередачи (далее - ЛЭП) и оборудования, находящихся в его технологическом управлении, и получение подтверждения их выполнения, а также за самостоятельные действия по отключению и заземлению ЛЭП и оборудования, находящихся в его технологическом управлении.

Таблица 20 - Допустимые расстояния до токоведущих частей электроустановок, находящихся под напряжением

Напряжение электроустановок, кВ	Расстояние от работников и применяемых ими инструментов и приспособлений, от временных ограждений, м	Расстояния от механизмов и подъемных сооружений в рабочем и транспортном положении от стропов, грузозахватных приспособлений и грузов, м
ВЛ до 1	0,6	1,0
Остальные электроустановки:		
до 1	не нормируется (без прикосновения)	1,0
1-35	0,6	1,0
60 (постоянный ток) - 110	1,0	1,5
150	1,5	2,0
220	2,0	2,5
330	2,5	3,5
400 (постоянный ток) - 500	3,5	4,5
750	5,0	6,0
1150	8,0	10,0

5.4. Экологическая безопасность

Загрязнение почвы веществами от компьютерной техники, батареек и других элементов питания может иметь серьезное воздействие на литосферу. Эти вещества, такие как свинец, кадмий, ртуть и другие токсичные вещества, могут проникать в почву и затем попадать в грунтовые воды, загрязняя их.

Это может привести к серьезным последствиям для окружающей среды и здоровья людей, таких как отравления, рак и другие заболевания. Помимо этого, загрязнение почвы может привести к снижению плодородия почвы и ухудшению качества растительности.

Неверный способ утилизации рабочей техники может привести к серьезному загрязнению гидросферы. Когда рабочая техника утилизируется неправильно, отходы могут попадать в реки, озера и другие водные источники, загрязняя их.

Отходы, содержащие токсичные вещества, такие как ртуть, свинец и другие тяжелые металлы, могут проникать в воду, заражая ее и оказывая негативное воздействие на живые организмы, в том числе рыб и других морских животных. Это может привести к сокращению численности популяций рыб и других водных животных, а также снижению качества воды для использования человеком.

Помимо этого, неправильная утилизация техники может привести к образованию мусорных свалок на берегах рек и озер, что также может привести к загрязнению водных ресурсов.

Поэтому очень важно правильно утилизировать рабочую технику, используя методы, которые не оказывают негативного воздействия на окружающую среду. Это может включать в себя переработку и утилизацию техники с использованием безопасных методов и в соответствии с законодательством и нормативами, а также соблюдение мер предосторожности при работе с техникой.

На атмосферу влияние не оказывается.

На селитебную зону влияние не оказывается.

5.5. Безопасность в чрезвычайных ситуациях

Выпускная квалификационная работа по разработке веб-приложения проходила в офисе. Ниже перечислены возможные ЧС:

- 1) Техногенные (взрывы, пожары, обрушение помещений);
- 2) Природные (наводнения, ураганы, бури, природные пожары);
- 3) Биологические (эпидемии, пандемии);
- 4) Антропогенные (война, терроризм).

Наиболее типичной ЧС для помещения, в котором проводилась работа по разработке мобильного приложения, является пожар. Рабочее помещение, представленное для выполнения ВКР, согласно пункту 5 СП 12.13130.2009, можно отнести к категории В (пожароопасное) [16]. Он может возникнуть вследствие причин электрического и неэлектрического характеров. К причинам электрического характера можно отнести короткое замыкание, искрение, статическое электричество. К причинам неэлектрического характера относится неосторожное обращение с огнём, курение, оставление без присмотра нагревательных приборов.

Для предотвращения возникновения пожара необходимо:

- Регулярно проводить инструктажи сотрудников предприятия по пожарной безопасности;
- Разместить в помещении план эвакуации и плакаты с краткой информацией с действиями при возникновении пожара;
- Соблюдать правила и нормы при монтаже электронных приборов и проведении электрической проводки;
- Оборудовать помещение пожарной сигнализацией и красными кнопками, а также средствами тушения пожара.

- Если все же не удалось предотвратить пожар, то каждый сотрудник должен:
- Незамедлительно сообщить об это в пожарную охрану;
- Принять меры по эвакуации людей, каких-либо материальных ценностей согласно плану эвакуации;
- Отключить электроэнергию, приступить к тушению пожара первичными средствами пожаротушения при отсутствии прямых угроз здоровью и жизни.

5.6. Вывод

В результате создания социальной части ВКР были изучены правовые и организационные вопросы обеспечения безопасности, изучены правовые нормы трудового законодательства и требования к компоновке рабочей зоны, в которой проводилась разработка настольного приложения.

Была рассмотрена производственная безопасность, в т.ч. проанализированы вредные и опасные факторы, которые могут возникнуть при разработке и эксплуатации настольного приложения. Предложены мероприятия по защите разработчика от действия опасных и вредных факторов.

Был рассмотрен характер воздействия настольного приложения на окружающую среду и проведен анализ возможных чрезвычайных ситуаций, которые могут возникнуть при разработке и эксплуатации проектируемого решения.

Согласно правилам установок электроустановок, рабочее помещение инженера-программиста относится к помещениям без повышенной опасности поражения электрическим током.

Рабочее помещение оборудовано в соответствии с требованиями электро и пожарной безопасности. Работа программиста относится к категории тяжести труда Ia и требует 1 группы по электробезопасности.

Рабочее помещение является пожароопасным и относится к категории В, а возможный пожар может относиться к классу А, так и классу Е.

По категории тяжести труда в соответствие с СанПин 1.2.3685-21 работа разработчика-программиста относится к категории Ia, так как работы производятся сидя и сопровождаются незначительным физическим напряжением.

Подводя итоги, можно отметить, что нарушений по организации рабочего процесса при выполнении ВКР выявлено не было, а все необходимые требования и нормы безопасности соблюдены.

Заключение

Была разработана программа для создания файлов стилей, описывающих графическое отображение геометрии; пример практического применения, редактирование файлов стилей для отображения в ПО «Geoserver». Стандарт SLD и работа с ним является общепринятым стандартом, что способствуют множеству мест применения разработки.

Программа запускается на платформе Windows, так как написана на C# с использованием WPF технологии с использованием рабочей области NET Framework 4.7.2, что способствует ее дальнейшей, легкой адаптации для обновленной кроссплатформенной рабочей области NET 5.0, 6.0, 7.0, которая в свою очередь позволит запустить программу на других операционных системах: Mac, Linux.

Пользовательский интерфейс программы понятен и адаптивен, время на создание файлов стилей сократилось в разы. Проведены дополнительные работы по упрощению и облегчению понимания графического пользовательского интерфейса, пользователем. Добавлены подсказки для отображаемых в окне элементов, обновлены используемые графические стили в приложении.

Добавлена панель предпросмотра, с помощью которой можно отсматривать для каждого правила, формируемое им изображение, которое в дальнейшем применится на геометрию.

Реализованы операции по взаимодействию со стилями в ПО Geoserver, через клиент настольного приложения.

Приложение внедрено на предприятии.

Было проведено исследования по возможности добавления кроссплатформенности коду, реализованного настольного приложения на платформе WPF, путем использования платформы Unity, для дальнейшей компиляции как WEB приложение. Исследование показало, что данный вариант разработки мало возможен на текущий момент; эффективней разрабатывать WEB приложение, отдельным продуктом, с помощью инструментов применяемыми на платформе WEB разработки.

Список используемых источников

- 1) Трудовой кодекс Российской Федерации от 30.12.2001 N 197-ФЗ (ред. от 25.02.2022) (с изм. и доп., вступ. в силу с 01.03.2022) [Электронный ресурс]. URL: http://www.consultant.ru/document/cons_doc_LAW_34683/ (Дата обращения: 10.05.23).
- 2) ГОСТ 12.2.032-78 «ССБТ. Рабочее место при выполнении работ сидя. Общие эргономические требования» [Электронный ресурс]. URL: <http://docs.cntd.ru/document/1200003913> (дата обращения: 13.05.2023).
- 3) ГОСТ 12.0.003-2015 Система стандартов безопасности труда (ССБТ). Опасные и вредные производственные факторы. Классификация (с Поправками) [Электронный ресурс]. URL: <https://docs.cntd.ru/document/1200136071> (дата обращения: 10.05.2023).
- 4) СанПиН 1.2.3685-21 "Гигиенические нормативы и требования к обеспечению безопасности и (или) безвредности для человека факторов среды обитания" [Электронный ресурс]. URL: <https://docs.cntd.ru/document/573500115> (дата обращения: 17.05.2023).
- 5) СП 52.13330.2016 «Естественное и искусственное освещение». Актуализированная редакция СНиП 23-05-95 [Электронный ресурс]. URL: <https://docs.cntd.ru/document/456054197> (дата обращения: 12.05.2023).
- 6) ГОСТ 12.1.019-2017 Система стандартов безопасности труда (ССБТ). Электробезопасность. Общие требования и номенклатура видов защиты [Электронный ресурс]. URL: <https://docs.cntd.ru/document/1200161238> (дата обращения: 10.05.2023).
- 7) СП 12.13130.2009 Определение категорий помещений, зданий и наружных установок по взрывопожарной и пожарной опасности [Электронный ресурс]. URL: <https://docs.cntd.ru/document/1200071156> (дата обращения: 12.05.2023).
- 8) Правила по охране труда при эксплуатации электроустановок [Электронный ресурс]. URL: <https://docs.cntd.ru/document/573264184> (дата обращения: 14.05.2023).
- 9) Допустимые уровни шума технических средств и методы их определения. [Электронный ресурс]. URL: <https://docs.cntd.ru/document/1200016755> (дата обращения: 16.05.2023).
- 10) ГОСТ 12.4.082-80 «ССБТ. Метод определения остроты зрения человека в средствах индивидуальной защиты» [Электронный ресурс]. URL: <https://docs.cntd.ru/document/1200012646> (дата обращения: 23.05.2023).
- 11) Windows Presentation Foundation: [Электронный ресурс]., 2022. URL: <https://docs.microsoft.com/ru-ru/dotnet/desktop/wpf/?view=netframeworkdesktop-4.8>. (Дата

- обращения: 05.06.2022).
- 13) Руководство по WPF: [Электронный ресурс], 2019. URL: <https://metanit.com/sharp/wpf>.
(Дата обращения: 12.06.2022).
 - 14) User Manual: [Электронный ресурс], 2016. URL: <https://docs.geoserver.org> (Дата обращения: 01.06.2022).
 - 15) Документация Unity: [Электронный ресурс], 2020. URL: <https://docs.unity3d.com/Manual/IL2CPP.html> (Дата обращения: 22.09.2022).
 - 16) Документация Noesis: [Электронный ресурс], 2017. URL: <https://www.noesisengine.com/docs/Gui.Core.Index.html> (Дата обращения: 09.10.2022).
 - 17) Запуск WEB Application: [Электронный ресурс], 2021. URL: <https://docs.unity.cn/2020.2/Documentation/Manual/webgl-building.html> (Дата обращения: 03.11.2022).
 - 18) OGC Standards: [Электронный ресурс], 2018. URL: <https://www.ogc.org/standard> (Дата обращения: 27.05.2023).

Приложение А

Таблица 21 – Диаграмма Ганта

Название работы	Исполнитель	Май (2022)	Июнь	Июль	Август	Сентябрь	Октябрь	Ноябрь	Декабрь	Январь	Февраль	Март	Апрель	Май
Составление и утверждение тема диссертации, утверждение плана-графика.	Исп.1	■												
Календарное планирование выполнения диссертации.	Исп.1, Исп.2	■												
Изучение академической литературы и конкурентных решений в данной сфере.	Исп.2		■	■										
Изучение программного обеспечения для расчетов.	Исп.2				■									
Технический расчет оценки времени разработки.	Исп.2					■	■	■						
Разработка библиотеки для работы с SLD структурой.	Исп.2							■	■	■	■			
Написание Desktop приложения для редактирования стилей.	Исп.2										■	■		
Адаптация GUI под обычного пользователя.	Исп.2											■		
Тестирование.	Исп.2											■	■	
Составление пояснительной записки.	Исп.2												■	■

Примечание:



– Исп. 1 (научный руководитель);



– Исп. 2 (инженер)