

Отчет о проверке на заимствования №1



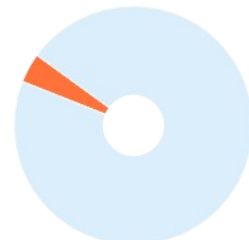
Автор: Андреева Анастасия nastyandreeva7@gmail.com / ID: 7763445
Проверяющий: Андреева Анастасия (nastyandreeva7@gmail.com / ID: 7763445)
 Отчет предоставлен сервисом «Антиплагиат» - <https://users.antiplagiat.ru>

ИНФОРМАЦИЯ О ДОКУМЕНТЕ

№ документа: 4
 Начало загрузки: 15.05.2020 16:45:21
 Длительность загрузки: 00:00:10
 Имя исходного файла: ВКР_Андреева.docx
 Название документа: ВКР_Андреева
 Размер текста: 1 кБ
 Символов в тексте: 69949
 Слов в тексте: 8492
 Число предложений: 642

ИНФОРМАЦИЯ ОБ ОТЧЕТЕ

Последний готовый отчет (ред.)
 Начало проверки: 15.05.2020 16:45:32
 Длительность проверки: 00:00:17
 Комментарии: не указано
 Поиск перефразирований: да
 Модули поиска: Цитирование, Модуль поиска Интернет



ЗАИМСТВОВАНИЯ

3,76%

САМОЦИТИРОВАНИЯ

0%

ЦИТИРОВАНИЯ

0%

ОРИГИНАЛЬНОСТЬ

96,24%

Заимствования — доля всех найденных текстовых пересечений, за исключением тех, которые система отнесла к цитированиям, по отношению к общему объему документа.
 Самоцитирования — доля фрагментов текста проверяемого документа, совпадающий или почти совпадающий с фрагментом текста источника, автором или соавтором которого является автор проверяемого документа, по отношению к общему объему документа.
 Цитирования — доля текстовых пересечений, которые не являются авторскими, но система посчитала их использование корректным, по отношению к общему объему документа. Сюда относятся оформленные по ГОСТу цитаты; общеупотребительные выражения; фрагменты текста, найденные в источниках из коллекций нормативно-правовой документации.
 Текстовое пересечение — фрагмент текста проверяемого документа, совпадающий или почти совпадающий с фрагментом текста источника.
 Источник — документ, проиндексированный в системе и содержащийся в модуле поиска, по которому проводится проверка.
 Оригинальность — доля фрагментов текста проверяемого документа, не обнаруженных ни в одном источнике, по которым шла проверка, по отношению к общему объему документа.
 Заимствования, самоцитирования, цитирования и оригинальность являются отдельными показателями и в сумме дают 100%, что соответствует всему тексту проверяемого документа.
 Обратите Ваше внимание, что система находит текстовые пересечения проверяемого документа с проиндексированными в системе текстовыми источниками. При этом система является вспомогательным инструментом, определение корректности и правомерности заимствований или цитирований, а также авторства текстовых фрагментов проверяемого документа остается в компетенции проверяющего.

№	Доля в отчете	Источник	Ссылка	Актуален на	Модуль поиска
[01]	0,26%	https://forum-nauka.ru/domains_data/files/sborniki/Zhurnal%20_2(30).pdf	https://forum-nauka.ru	10 Июн 2019	Модуль поиска Интернет
[02]	0,15%	ГОСТ Р ИСО/МЭК 12207-99 Информационная технология (ИТ). Процессы жизненного...	http://docs.cntd.ru	22 Мар 2020	Модуль поиска Интернет
[03]	0%	https://www.spbgasu.ru/upload-files/nauchinnovaz/sbornik_trudov/bimconf_2019.pdf	https://spbgasu.ru	06 Мар 2020	Модуль поиска Интернет
[04]	0,26%	Лекция 3. Документирование Файл	http://elab.pro	19 Июл 2018	Модуль поиска Интернет
[05]	0%	3355.Введение в программную инженерию Учебное пособие Ехлаков Ю. П. 2011. 14.	http://docme.ru	05 Мая 2017	Модуль поиска Интернет
[06]	0,4%	http://www.spbgasu.ru/upload-files/nauchinnovaz/konferenzii/bimconf_2018.pdf	http://spbgasu.ru	14 Сен 2018	Модуль поиска Интернет
[07]	0,63%	Скачать	http://worldreferat.ru	06 Апр 2020	Модуль поиска Интернет
[08]	0,48%	Скачать сборник (1/2)	https://ami.im	26 Янв 2020	Модуль поиска Интернет
[09]	0%	Учебно-методическое пособие на тему: КОНСПЕКТ ЛЕКЦИЙ ПО ДИСЦИПЛИНЕ «УС...	https://nsportal.ru	30 Мар 2020	Модуль поиска Интернет
[10]	0,4%	Материалы II международной научно-практической конференции «ВМ-моделиро...	https://spbgasu.ru	05 Мая 2020	Модуль поиска Интернет
[11]	0,26%	Скачать (4/4)	http://mgsu.ru	11 Мар 2020	Модуль поиска Интернет
[12]	0,22%	https://naukaip.ru/wp-content/uploads/2017/11/%D0%9C%D0%9A-240-%D0%A1%D0%...	https://naukaip.ru	06 Июн 2019	Модуль поиска Интернет
[13]	0%	ВМ-МОДЕЛИРОВАНИЕ В ЗАДАЧАХ СТРОИТЕЛЬСТВА И АРХИТЕКТУРЫ. Материалы В.	https://docplayer.ru	14 Июн 2019	Модуль поиска Интернет
[14]	0%	Материалы II международной научно-практической конференции «ВМ-моделиро...	https://spbgasu.ru	05 Мая 2020	Модуль поиска Интернет
[15]	0%	ГОСТ Р ИСО/МЭК ТО 15271-2002: Информационная технология. Руководство по при...	http://standartgost.ru	19 Мая 2016	Модуль поиска Интернет
[16]	0,19%	4.Наука и образование в современном обществе: вектор развития. Сборник научн...	http://co2b.ru	20 Авг 2017	Модуль поиска Интернет
[17]	0,15%	https://esu.citis.ru/dissertation/WEJDQRM5BL75PH3UAA4HWNBT	https://esu.citis.ru	10 Мая 2018	Модуль поиска Интернет
[18]	0,16%	https://www.irgups.ru/sites/default/files/irgups/doc/nauka/nauka_i_molodeg/sbornik_m...	https://irgups.ru	15 Сен 2018	Модуль поиска Интернет

[19]	0,16%	ВІМ-МОДЕЛІРОВАНИЕ КАК ЭЛЕМЕНТ СОВРЕМЕННОГО СТРОИТЕЛЬСТВА	https://yandex.ru	16 Ноя 2018	Модуль поиска Интернет
[20]	0,02%	Click here to get the file	http://usaaa.ru	02 Мая 2020	Модуль поиска Интернет

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«РОССИЙСКИЙ ГОСУДАРСТВЕННЫЙ
ПЕДАГОГИЧЕСКИЙ УНИВЕРСИТЕТ им. А. И. ГЕРЦЕНА»

12



Основная профессиональная образовательная программа
Направление подготовки 09.03.01 Информатика и вычислительная техника
Направленность (профиль) «Технологии разработки программного обеспечения»
форма обучения – очная

Выпускная квалификационная работа

Разработка решения для обработки данных программы автоматизированного
проектирования Revit

Обучающейся 4 курса
Андреевой Анастасии Александровны

Научный руководитель:
доктор педагогических наук, профессор
Абрамян Геннадий Владимирович

Санкт-Петербург

Оглавление

<u>ВВЕДЕНИЕ</u>	3
<u>ГЛАВА 1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ. ПОДГОТОВКА К СОЗДАНИЮ РЕШЕНИЯ ДЛЯ ОБРАБОТКИ ДАННЫХ</u>	5
<u>1.1. BIM-моделирование</u>	5
<u>1.1.1. BIM-моделирование и параметризация в Autodesk Revit</u>	7
<u>1.2. Обзор инструментальных средств и технологий обработки данных в Autodesk Revit</u>	13
<u>1.2.1. Анализ расширений и модулей разработчика в Revit</u>	13
<u>1.2.2. Анализ языковых средств и сред разработки</u>	20
<u>Выводы к главе 1</u>	26
<u>ГЛАВА 2. РЕАЛИЗАЦИЯ И РАЗРАБОТКА РЕШЕНИЯ ДЛЯ ОБРАБОТКИ ДАННЫХ ПРОГРАММЫ АВТОМАТИЗИРОВАННОГО ПРОЕКТИРОВАНИЯ REVIT</u>	28
<u>2.1. Описание заказчика</u>	28
<u>2.1.1. Данные о заказчике</u>	28
<u>2.1.2. Функциональные особенности существующей системы</u>	29
<u>2.1.3. Функциональные недостатки существующей системы</u>	30
<u>2.1.4. Требования заказчика</u>	31
<u>2.2. Разработка программного продукта</u>	31
<u>2.2.1. Выбор модели жизненного цикла разработки</u>	31
<u>2.2.2. Структура разработки решения для обработки данных</u>	34
<u>2.2.3. Получение доступа и экспорт данных с помощью Dynamo</u>	35
<u>2.2.4. Разработка пакета для обработки полученных данных</u>	41
<u>Выводы к главе 2</u>	43
<u>Заключение</u>	45
<u>СПИСОК ЛИТЕРАТУРЫ</u>	46

ВВЕДЕНИЕ

На сегодняшний день термин «большие данные» или "big data" используется всё чаще и чаще в большинстве областей науки и в целом сфере информационных технологий. Большинство объектов, процессов и явлений реального мира из разных отраслей моделируются с разной степенью абстракции и при использовании различных подходов к этому процессу и к выбору моделей. Одной из сфер, где постоянно необходимо получать, управлять и анализировать «большие данные» является строительство [22].

Строительная отрасль играет весомую роль в экономике всех стран мира и способствует темпам и тенденциям экономического развития. Учитывая рост информационных технологий в современном мире строительным компаниям всё тяжелее поддерживать конкурентоспособность на мировом и национальном уровне. Возникает потребность перехода строительных компаний к автоматизированному проектированию и внедрению электронных инструментов для выхода предприятий на более высокий уровень [8].

Для решения данной проблемы используются новые технологии, одной из которых является BIM (Building Information Modeling). Под термином BIM подразумевается процесс моделирования информационной модели здания. Данная технология помогает эффективнее планировать, проектировать, строить и эксплуатировать здания и объекты инфраструктуры [4].

В России внедрение BIM-технологии в строительные компании началось в 2014 году, после принятия на государственном уровне Плана поэтапного внедрения технологий информационного моделирования в области промышленного и гражданского строительства. Согласно исследованию [24] на 2019 год, активно применяют данную технологию лишь 22% строительных организаций. Такой низкий уровень внедрений технологий связан с

неготовностью компаний изучать и осваивать BIM-технологии. [19]

Одной из первых компаний, начавшей реализацию BIM-моделирования в 2012 году был Научно-технический центр «Эталон» [30]. BIM-моделирование реализуется с помощью ряда программного обеспечения, в том числе и программного комплекса Autodesk Revit. Помимо использования существующих сторонних решений в организации «Эталон» продолжает разрабатываться система управления данными о ходе реализации инвестиционно-строительных проектов, при этом используются данные из информационных моделей объектов строительства. Такие системы управления данными позволяют отслеживать все этапы разработки и реализации проектов, улучшают эффективность принятий решений, помогают контролировать и оптимизировать расходы на строительство объектов. Вышеупомянутой системе необходим прямой доступ к архитектурно-строительным данным моделей, но в программном обеспечении Autodesk Revit отсутствует такая возможность при работе с данными. В результате прохождения стажировки в ООО «НТЦ «Эталон» был получен заказ на разработку решения для обработки данных программы автоматизированного проектирования Revit согласно требованиям заказчика. Этим обуславливается **актуальность** выпускной квалифицированной работы.

Цель работы — разработка решения, которое удовлетворяет требованиям заказчика, а **предмет** — автоматизированная обработка архитектурно-строительных данных. Для достижения поставленной цели необходимо решить следующие **задачи**:

1. Проанализировать справочную литературу, документацию и интернет-источники по BIM-моделированию в Autodesk Revit и расширению функциональных возможностей.
2. Осуществить обоснованный отбор технологий для экспорта и обработки архитектурно-строительных данных из Autodesk Revit.
3. Реализовать решение для получения доступа и экспорта архитектурно-строительных данных с помощью отобранной технологии.
4. Разработать пакет чтения и обработки экспортированных данных для

преобразования их в единый текстовый файл (в формат JSON).

В рамках решения последней задачи разработанное решение для обработки данных в программе автоматизированного проектирования Revit должны внедрить в систему управления данными Научно-технического центра «Эталон».

ГЛАВА 1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ. ПОДГОТОВКА К СОЗДАНИЮ РЕШЕНИЯ ДЛЯ ОБРАБОТКИ ДАННЫХ

1.1. BIM-моделирование

В наше время цифровые технологии играют всё более значительную роль в различных отраслях жизни. Растут объемы информации и вместе с ними появляются всё больше новых способов и технологий хранения информации. Строительство является одной из основных отраслей экономики, которая также не отстаёт и стремится использовать и применять последние технологии и разработки в своей сфере. Одним из таких прорывов является активно развивающаяся технология BIM.

Информационное моделирование зданий выводит строительную индустрию на новый «цифровой» уровень «безбумажного» проектирования и управления эксплуатацией зданий. Поток информации при возведении и строительстве того или иного сооружения никогда не заканчивается на этапе построения, ведь после ввода в эксплуатацию объект начинает взаимодействовать с окружающей средой и его жизненный цикл не заканчивается. Хранить, управлять и передавать огромный объем информации намного проще и практичнее с помощью технических средств и технологий. Информационное моделирование — это продолжительный процесс с разными стадиями формирования и развития самой информационной модели. Поэтому под термином BIM подразумевают не только информационное здания, но и непосредственно информационную модель.

BIM представляет собой трёхмерную информационную, цифровую модель здания с атрибутивной составляющей, которая содержит в себе не только все элементы сооружения, но и подробную информацию о них. Основа концепции BIM – единая информационная модель. Единая модель сооружения – это главное условие реализации BIM-технологии. Такая модель должна содержать полную и согласованную информацию, необходимую для решения конкретной задачи информационного моделирования.

Данные в модель добавляются и обновляются на протяжении всех этапов жизненного цикла здания: от планирования и составления технического задания до эксплуатации и ремонта. Поэтому использование BIM-технологии актуально на каждой ступени жизненного цикла объекта строительства.

Такие данные можно извлекать, сортировать, обрабатывать и менять,

поэтому, можно сказать, что модель представляет собой своеобразную базу данных, которая, в свою очередь, имитирует реальный объект. Одним из важных свойств BIM-модели является то, что при внесении графических изменений, автоматически будет изменяться и база данных. Поэтому при изменении какого-либо из параметров модели происходит автоматическое изменение связанных с ним параметров и объектов. На этом основании формируется рабочая документация. Таким образом, модель состоит из двух частей: геометрической и информационной, которые взаимосвязаны друг с другом.

Одним из важнейших принципов BIM является открытый, общий доступ к модели для всех специалистов, причастных к моделированию данной модели. То есть BIM помогает объединять команды архитекторов, инженеров, строителей для совместной работы над одним проектом. Данный принцип позволяет ввести безбумажный документооборот между отделами, помогает в своевременном выявлении допущений и коллизий. BIM-технология существенно влияет и улучшает планирование бизнеса, увеличивает рост контроля над возможными расходами, сокращает ошибки и стоимость строительства, а также продолжительность проекта. Еще одним преимуществом BIM - это использование модели непосредственно на самой строительной площадке, на монтаже. Всё это становится возможным, если на стройплощадках у специалистов есть планшеты с предварительно установленным ПО и доступом к Wi-Fi. Доступ и работа с моделью в реальном времени позволит сократить ошибки и ускорить процесс возведения здания.

Для создания информационной модели требуется определенное программное обеспечение, которое смогло бы в полной мере или частично реализовать данную технологию. На российском рынке сейчас конкурируют четыре BIM-решения: ArchiCAD, Revit, Tekla и Renga. Все программные продукты отличаются между собой не только инструментарием, но и способом взаимодействия с компонентами, а именно методами их создания. Еще одним важным отличием программ является то, какой функционал предоставляют BIM ПО. Есть программы, которые позволяют решать только узкоспециализированные

задачи из определенных областей строительства или архитектуры, например, программный пакет ArchiCAD решает лишь задачи, связанные с архитектурой.

Наиболее приближенным к концепции BIM является программный комплекс Autodesk Revit, который охватывает несколько разделов (архитектурный, конструктивный, инженерный и др.) и дает возможность проектировщикам разных отделов работать в едином формате файла, тем самым обеспечивает возможность обмена данными между существующими системами предприятия. Все это помогает данной программе отвечать главному принципу BIM-моделированию — принципу единой информационной модели.

Рассмотрим программный комплекс Autodesk Revit подробнее.

1.1.1. BIM-моделирование и параметризация в Autodesk Revit

Изначально Revit представлял собой набор из трёх отдельных утилит для различных стадий проектирования зданий:

- Revit Architecture – САПР для архитекторов и дизайнеров (создание информационных моделей сооружений);
- Revit Structure – САПР для проектировщиков строительных конструкций;
- Revit MEP – САПР для инженеров систем.

В 2013 году все три версии были объединены в единый программный комплекс – Revit, который положил начало BIM-моделированию для данного продукта.

Можно сказать, что программный комплекс является некой виртуальной стройкой, на которой есть и моделируемое здание, и окружающая обстановка, и, конечно, все строительные и инженерные системы. Revit, как и другие программные продукты, отвечающие принципам BIM, поддерживает объектно-ориентированное проектирование. В таком подходе моделирование происходит на основе библиотечных элементов.

В Revit информационная модель здания или сооружения состоит из категории семейств. Категория семейств – это параметрические компоненты

программы, которые представляют из себя графические объекты. Такие объекты являются типами возможных элементов, из которых может состоять здание, например, колонны, стены, двери и т. д. В свою очередь категория семейств подразделяется на отдельные семейства, например, для категории колонн семействами будут: прямоугольная колонна, круглая колонна и другие виды колонн. Все объекты семейств имеют свою геометрию и параметры – данные, полностью описывающие объект. Геометрия объекта отрисовывается один раз и может изменяться только за счет изменения пользователем параметров. Семейства дают возможность наполнять модель детальной информацией о структуре моделируемого здания.

Каждый объект семейства представляет собой аналог реального изделия с полностью наследуемыми параметрами и свойствами. Так, например, у стен есть свой тип, размеры (длина, толщина, высота, глубина и т.д.), на которые впоследствии будут ориентироваться и опираться рабочие на стройке и возводить их согласно смоделированному проекту.

Семейства в Revit классифицируются в три типа:

- системные семейства (System families);
- разовое семейство (In-place families);
- обычные семейства (Standard component families).

16

Системные семейства являются встроенными (базовыми) в Revit. К ним относятся основные строительные конструкции: стены, перекрытия, двери, крыши и др. Данные семейства создают каркас виртуального здания и являются основой для присоединения других элементов. Создавать новые системные семейства нельзя, можно лишь изменять их параметрические свойства. Все системные семейства возводятся, в основном, на строительной площадке.

Разовые и обычные семейства могут самостоятельно моделироваться проектировщиками в редакторе семейств. Такие семейства являются пользовательскими и создаются в отдельных файлах, параметрические свойства таких семейств определяются пользователем. В отличие от системных они изготавливаются на заводах и затем уже монтируются на строительных

площадках,

Параметризацией называется процесс присвоения параметров объектам. Параметры хранят в себе и отображают информацию о семействе. С их помощью можно легко редактировать семейства без обращения к редактору семейств. Они используются для формирования спецификаций, фильтрации и сортировки в проекте. Параметры могут быть созданы пользователем (тогда они называются пользовательскими) или они могут существовать внутри семейств по умолчанию (системные параметры).

Параметры имеют следующую классификацию:

- общие параметры;
- параметры проекта;
- глобальные параметры;
- встроенные параметры;
- параметры семейства.

Общие параметры используются, в основном, тогда, когда нужно вынести параметры для марок и спецификации. Это позволяет дать большую информацию об элементе на чертеже. Общие параметры записываются в отдельный файл «Файл Общих Параметров» формата txt, в котором хранятся наименования, типы и уникальные идентификационные номера параметров (GUID). С помощью этого файла можно открывать или закрывать возможность редактирования и видимости параметров непосредственно в самом Revit. Так же такой файл позволяет использовать ранее созданные общие параметры в различных проектах, лишь подгружая этот файл к проекту.

Проектные параметры создаются для категории семейств, тем самым добавляя дополнительные свойства. Они используются для спецификации и фильтраций. Проектные параметры могут дополнить список системных параметров. В отличие от общих параметров, проектные параметры не могут быть перемещены или скопированы в другой проект, а также не могут быть сносками в аннотациях.

Глобальные параметры используются для параметризации внутри проектов,

а не семейств, их значения приравниваются параметрам других объектов модели.

Встроенные параметры – это параметры, которые изначально присутствуют в любом проекте и семействе. Их можно вносить в спецификации и аннотации, использовать для фильтрации, перемещать из проекта в проект. Примерами таких параметров является «Комментарий», «Код по классификатору». Встроенные параметры недоступны для редактирования.

Параметры семейства используются только внутри семейств, такие параметры нельзя никуда вынести и обычно они являются вспомогательными, могут участвовать в формулах. Параметрами семейства можно описывать размеры и параметрические свойства, которые не содержатся в спецификациях или марках.

У параметров Revit имеется ряд таких свойств как:

- вид параметра;
- имя параметра;
- подсказка;
- группирование;
- параметр типа или экземпляра;
- параметр для отчетов;
- категория параметра;
- тип данных и единица измерения.

При создании параметров первым делом указывается его вид — параметр общий или параметр семейства. Это зависит от назначения параметра — использование в марках и спецификация или просто вспомогательный параметр. Далее параметру присваивается имя и иногда указывается подсказка. Существуют правила наименования параметров, которые регламентированы в установленных стандартах самой компании. Обязательно нужно указать группировку параметра для дальнейшего быстрого поиска и управления параметром. Также необходимо выбрать тип параметра:

- параметр экземпляра;
- параметр типа.

Значение параметра экземпляра (или параметра объекта) является индивидуальным для каждого отдельного элемента модели. Поэтому при изменении такого параметра меняются свойства только у конкретного, выделенного объекта модели. Параметры экземпляра находятся в свойствах объекта, те, которые доступны при выделенные объекта. Например, колонны могут иметь один тип «Колонна бетонная_Квадратная_ADSK_2", но длина их может отличаться, у одной колонны длина - 550 на 650 миллиметров, у другой - 500 на 600 миллиметров. Параметр «длина» является параметром экземпляра, т.к у каждой колонны может быть своя длина. Через этот параметр экземпляра легко можно корректировать длину колонн.

Параметр типа управляет всеми экземплярами данного типа семейства. Что касается значений параметр типа, то изменение такого параметра приводит к изменению свойств у всех семейств этого типа. В отличие от параметров экземпляров они управляют семействами на более высоком уроком, позволяя за один раз изменить все объекты определенного семейства. Например, в проекте есть пустые дверные проемы с размерностью 1150 х 2100, но возникла необходимость сузить проемы до размера 1100 х 2000, тогда изменение параметра типа будет применено ко всему семейству.

Существуют ещё один тип параметра- параметр для отчёта. Такой параметр может быть только параметром экземпляра и служит для вывода размерных параметров семейства в спецификацию или в формулу. В проекте нельзя изменить значение такого параметра, его можно только считать.

При определении размеров параметров всегда требуется определение их типа данных. Набор типов данных параметра изменяется в зависимости от его категории: общие, несущие конструкции, ОВК, трубопроводы, электросети, энергия. Наиболее часто используемыми типами данных являются:

- длина;
- площадь;
- объём;
- текст;

- многострочный текст;
- число;
- целое;
- угол;
- уклон;
- материал.

Параметры с типом данных «длина» позволяют управлять размерами элемента. Все размерные параметры имеют единицы измерения, заданные в проекте. Например, длина в миллиметрах, площадь в метрах квадратных, объём в метрах кубических. Типы данных «объём» и «площадь» чаще всего задаются с помощью формул. Тип данных «целое» используется для сложных формул с условными выражениями. Тип «материал» позволяет управлять отображением элемента, его представлением в различных графических стилях, управлять штриховками и выносить дополнительные описания параметра.

Таким образом, ПО Revit позволяет моделировать достаточно сложные объекты, редактировать параметры компонентов модели, однако в программе отсутствует возможность организовывать прямой экспорт параметров и их значений.

1.2. Обзор инструментальных средств и технологий обработки данных в Autodesk Revit

Для обработки данных в программном комплексе Autodesk Revit необходимо было обратиться к возможным расширениям и модулям разработчика, так как сам процесс обработки данных, а именно экспорт данных невозможен непосредственно в Revit и требует обращения к дополнительным утилитам.

1.2.1. Анализ расширений и модулей разработчика в Revit

Внутри Autodesk Revit действительно существуют различные возможности расширения функциональности программного комплекса. Такие расширения позволяют выйти за пределы обычного проектирования и дают возможность автоматизировать рутинные задачи разного характера, добавить и усовершенствовать существующие функции, повысить эффективность работы прямо внутри программного комплекса, не прибегая к стороннему программному обеспечению. Благодаря данной возможности строительные компании могут не только добиваться большей результативности, но и составлять достойную конкуренцию другим компаниям на рынке. Среди таких расширений выделяют следующие:

- .NET API и SDK;
- RevitPythonShell и RevitRubyShell;
- PyRevit;
- Dynamo.

Опишем кратко приведенные выше технологии.

.NET API и SDK

Autodesk Revit предоставляет сильный .NET API и SDK. .NET API — это кроссплатформенная среда исполнения плагинов и макросов, написанных на языках программирования, использующие платформу .NET. А Revit SDK содержит в себе справочную литературу, для понимания концепции API, документацию и примеры кодов макросов.

Revit API содержит в себе огромную библиотеку классов, методов, свойств всех семейств и объектов. Именно благодаря доступу Revit API к объектам исходного кода можно создавать собственные инструменты и функции внутри программного комплекса.

Внутри Revit API лежит концепция пространства имён (namespace), которая помогает объединять классы в логические группы для избежания конфликтов имён. Revit API представляет своеобразную иерархию из этих пространств имён и

для того, чтобы достать информацию о каком-либо объекте внутри API необходимо «спуститься» по пространству имён. Например, если надо получить доступ к объекту Column (колонна), то последовательность будет следующей: Autodesk.Revit.DB Namespace> BuiltInCategory Enumeration >OST_Columnn.

Как уже отмечалось, доступ к Revit API имеет любой язык программирования, поддерживающий .NET. Но на данный момент Revit поддерживает четыре языка программирования:

- C#;
- VB.NET;
- Ruby;
- Python.

Рассмотрим и выделим преимущества использования того или иного языка в Revit API.

C# – это объектно-ориентированный язык программирования, который был разработан для платформы разработки приложений ASP.NET, которая входит в состав платформы .NET Framework. Язык относится к семейству языков C. Программы, написанные на данном языке, являются компактными, но не совсем читабельными из-за сложного синтаксиса. Несмотря на это C# был и остается популярным языком при написании плагинов для Autodesk Revit. Доступны различные курсы обучения по данному языку, а также примеры кода на C# для Revit.

Visual Basic .NET (VB.NET) это ещё одним объектно-ориентированным языком программирования, который является ответвлением языка Visual Basic реализованным на платформе Microsoft .NET. В отличие от языка C#, VB.NET менее строгий и более понимаемый синтаксис. Для VB.NET не так много готовых примеров кодов, но из-за его схожести с C# примеры можно проинтерпретировать.

Кроссплатформенные языки Ruby и Python были добавлены в API в 2014 году для упрощения вхождения и погружения в программирования в Autodesk Revit. Данные языки легки для обучения и не требуют огромного количества

времени для освоения, как C# и VB.NET. Благодаря их внедрению, у проектировщиков и других строительных инженеров появилась возможность самим в короткий срок овладеть программированием в Revit и начать упрощать и совершенствовать свои ежедневные задачи. Единственным минусом является то, примеров использования кода на Python в макросах сравнительно меньше, чем для других языков (C#, Visual Basic, Ruby).

Создавать, редактировать и запускать макросы позволяет встроенный редактор макросов SharpDevelop. Создаваемые макросы могут быть двух уровней: уровня конкретного документа или уровня всего приложения. В редакторе отсутствует возможность создавать внешние команды или внешние приложения. Вне зависимости от используемого языка программирования, код макроса всегда можно сконвертировать с одного языка на другой. Встроенный редактор позволяет сэкономить большое количество времени на инициализации и отладки макроса, нежели если бы программа подгружалась из сторонней среды разработки, такой как Microsoft Visual Studio.

На Рисунке 1.1 представлено главное окно редактора SharpDevelop. Прежде чем создать модуль необходимо выбрать назначение модуля: модуль уровня приложения или уровня проекта. Затем необходимо нажать на кнопку «Модуль» — «Создать», задать имя будущему модулю и выбрать один из четырех языков программирования, после этого можно приступить к непосредственному написанию макроса.

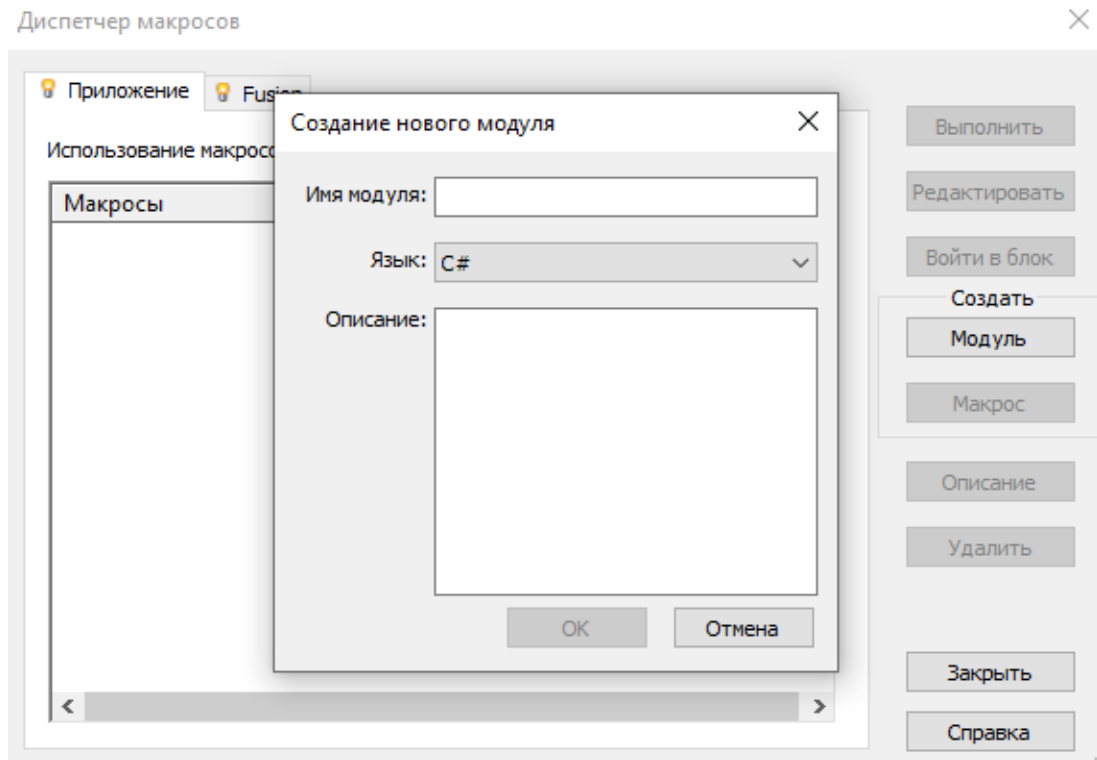


Рисунок 1.1. Диспетчер макросов в Revit

RevitPythonShell и RevitRubyShell

После внедрения Python и Ruby в Revit начала появляться различные дополнительные плагины, в том числе такие как RevitPythonShell и RevitRubyShell. На Рисунке 1.2 представлен один из данных плагинов - RevitPythonShell.

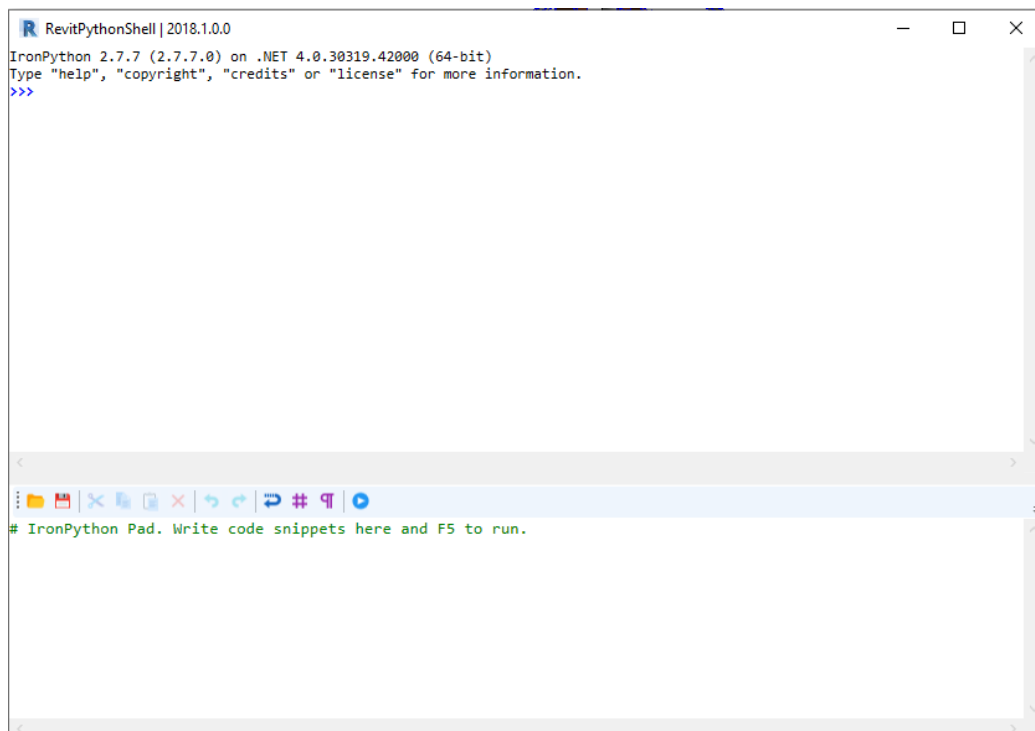


Рисунок 1.2. RevitPythonShell

Данные консоли (RevitPythonShell и RevitRubyShell) являются командными строками IronPython и IronRuby соответственно. IronPython и IronRuby являются интерпретаторами языков программирования Python и Ruby для языка C# на платформе Microsoft .NET. Рассматриваемые интерпретаторы позволяют использовать динамические языки в среде .NET, которые способны выполнять задачи непосильные языкам со статической типизацией.

Командные строки позволяют вводить и компилировать код плагинов в консоли, находясь в Revit. Более того они имеют интерактивную оболочку, что позволяет видеть результат кода при запуске программы. В основном, данные командные строки используются лишь для изучения Revit API и написания небольших плагинов.

pyRevit

pyRevit — это библиотека IronPython скриптов, написанных для Revit. Данная библиотека представлена в виде рабочего набора инструментов, полностью написанного на Python. Данные инструменты добавляют и дополняют функции и «сценарии» в Revit, как и все расширения, но главным отличием является то, что они имеют готовый пользовательский интерфейс и достаточно лишь нажать на кнопку, чтобы сценарий был вызван.

В pyRevit есть панель с кнопками, которые предоставляются при загрузке данной библиотеки. Эти кнопки помогают быстро создавать моделирующие и аннотативные штриховки, маркировать помещения, работать с 3D-видом и многое другое. При этом есть возможность написания и добавления собственных сценариев в pyRevit.

Для данной библиотеки не так много ресурсов для изучения и написания собственных сценариев, так как изначально она создавалась как готовый инструмент автоматизации, который требует только установки в Revit.

Dynamo

Dynamo — это среда визуального программирования для автоматизации

работы Revit. Это бесплатное дополнение помогает расширять функционал Revit с помощью графических алгоритмов, а именно:

- управлять информацией об объектах;
- создавать элементы по заданным параметрам;
- формировать и модифицировать объекты;
- автоматизировать рутинные процессы.

Инструментом данной среды является язык визуального программирования (Visual Programming Language или VPL). С помощью данного языка можно создавать геометрию, выполнять различные команды, манипулировать существующими объектами или системами графически, без написания кода в привычном нам понимании. Данный подход в программировании даёт возможность обычным инженерам и проектировщикам без особых знаний в программировании овладеть средой разработки.

Такие VPL-языки основываются на принципе «блоки и связи», где блоки представляют сущность программы, а связи - отношения между сущностями. Узлы и блоки, или в Dynamo в их ещё называют ноды (узлы) — это объекты, которые вы размещаете в рабочем пространстве и соединяете вместе с проводами для формирования визуальной программы. Ноды могут представлять собой как элементы Revit, так и математические операции. Все ноды представлены в библиотеке и сгруппированы по категориям и подкатегориям, например, есть категория для обработки данных, категория для создания и редактирования геометрии. У каждого нода есть входной и выходной порт, цветной индикатор состояния соединения нодов. Ноды соединяются между собой связями (проводами) через порты, тем самым создавая отношения между блоками. На Рисунке 1.3 представлен пример программы, выполненной в Dynamo. На данном рисунке видны ноды (серые блоки) связанные друг с другом «проводами», обозначенные линиями. В данной среде разработки простой пользовательский интерфейс, который интуитивно понятен любому пользователю Revit.

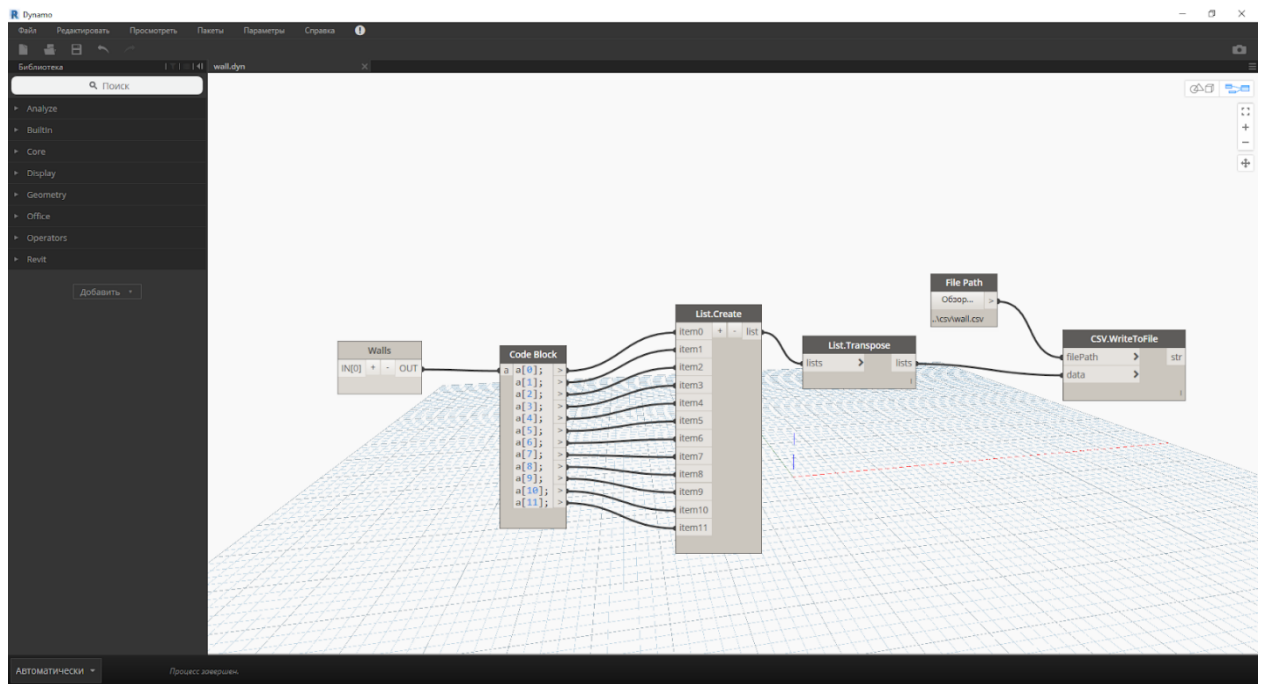


Рисунок 1.3. Пример программы в Dynamo

Помимо существующих нодов, можно загружать новые категории из сторонних пакетов или самостоятельно создавать собственные ноды, расширяющие функционал, и делиться ими с другими разработчиками по всему миру. Среди огромного количества нодов есть один, который существенно увеличивает возможности Dynamo и позволяет совмещать визуального программирования с обычным, это нод с названием «Python Script». Внутри данного нода можно писать полноценную программу на языке Python, при этом есть возможность добавлять входные данные из Revit напрямую, так как это обычный нод, который имеет входные и выходные слоты. Кроме того Dynamo поддерживает ещё один собственный язык программирования Design Script, с помощью которого можно также сделать программу более эффективной. Таким образом, в Dynamo благодаря таким скриптам есть поддержка Revit API.

Также в Dynamo есть дополнительные возможности, такие как:

- импорт или экспорт данных в Excel;
- сравнение моделей;
- автоматизированная проверка моделей.

Программы, созданные в данной среде разработки, могут быть любой сложности и объема. Dynamo поддерживает BIM-концепцию и позволяет его

разработчикам работать в одном файле, то есть поддерживает общий доступ к файлам.

По сравнению с рассмотренными выше расширениями, главным преимуществом Dynato является огромное количество видеоуроков, книг, статей и форумов для быстрого изучения и начала работы со средой.

1.2.2. Анализ языковых средств и сред разработки

После осуществления процесса экспорта данных из Revit, обработка данных не закончилась и включила в себя такие операции как формализацию, фильтрацию, сортировку и преобразования данных для дальнейшего их использования. Необходимо было создать пакет обработки данных, который бы включал в себе все упомянутые раньше операции.

На сегодняшний день существует огромный выбор как языковых средств для обработки данных, так и инструментальных. Наиболее подходящими для выполнения манипуляций с данными являются следующие инструменты:

- R;
- Python;
- Scala;
- MATLAB.

В качестве основного языкового средства для реализации выпускной квалификационной работы (программного продукта) был выбран язык Python. Опишем более подробно и детально выбранное языковое средство.

В настоящее время высокоуровневый, интерпретируемый язык программирования Python является одним из самых популярных языков программирования, применяемым в различных сферах IT: от разработки веб-приложений различного вида и масштаба до приложений, связанных с машинным обучением и искусственным интеллектом. Благодаря тому что Python поддерживает структурированное, объектно-ориентированное и функциональное программирование с довольно простым и понятным синтаксисом, разработчики

начинают активно использовать данный язык в новых областях ИТ. В последние годы Python приобрел популярность и в области анализа и обработки данных, то есть в области Data Science.

Обработка и анализ данных — это сложный и долгий процесс, включающий в себя целый ряд этапов, на каждом из которых возможно реализовать различные по типу, способу решения задачи. Таким образом, используя исключительно Python возможно решение самого широкого спектра задач на всех этапах обработки данных и не задействовать другие языки программирования и инструменты. Кроме того, язык имеет обширный набор модулей и библиотек, с помощью которых можно ускорить и упростить процесс разработки, в том числе и обработку данных. Данные библиотеки способны обрабатывать и визуализировать данные для аналитических проектов любой сложности. К наиболее популярным инструментам для работы с данными в Python относятся:

- NumPy;
- Pandas;
- SciPy;
- Matplotlib.

Опишем кратко приведенные выше технологии.

NumPy

NumPy — одна из популярных библиотек Python с открытым исходным кодом используемая для анализа данных и машинного обучения, входит в состав пакета Python SciPy Stack. Python SciPy Stack включает в себя основные библиотеки для обработки данных: Matplotlib, Pandas и SymPy и дополнительные инструменты.

NumPy даёт возможность быстро выполнять математические и числовые операции, такие как:

- сложение вычитание, деление, умножение, транспонирование и индексирование многомерных массивов и матриц;
- вычисление высокоуровневых математических функций;
- решение задач линейной алгебры.

NumPy используется, в основном, для облегчения работы с массивами, в которых хранятся значения одного типа данных. В данной библиотеке заложен механизм векторизации, который увеличивает производительность уменьшает время выполнение операций.

Pandas

Pandas — одна из фундаментальных и быстроразвивающихся библиотек Python, которая является дополнением и надстройкой NumPy, а также частью Python SciPy Stack. Очень часто эти две библиотеки работают в связке, что позволяет увеличить функциональность и производительность программы. Дополняя NumPy, Pandas поддерживает и позволяет выполнять следующие операции:

- поддержка и работа с данными в разных форматах (.xlsx, .xls, .csv, .sql, .html и другие);
- различное манипулирование, сортировка, переименование DataFrame и Series;
- построение прямоугольной диаграмма и гистограммы.

Pandas является незаменимым инструментом для быстрой и высокоэффективной работы с данными.

SciPy

SciPy — одна из ключевых библиотек в составе пакета Python SciPy Stack. Библиотека построена на объекте массива NumPy и также, как и Pandas расширяет функционал. SciPy содержит в себе множество хорошо документированных модулей, позволяющих решать разнообразные задачи из таких областей как:

- интерполяция;
- интеграция;
- линейная алгебра;
- процедуры численной интеграции и оптимизации;
- дифференциальные уравнения;
- статистика.

Данная библиотека характеризуется сравнительно большой

производительностью за счет использования программной оптимизации алгоритмов вычислений.

Matplotlib

Matplotlib — основная библиотека для визуализации данных на Python, которая также входит в пакет Python SciPy Stack. Библиотека предоставляет широкие возможности для различных видов визуализаций, таких как:

- линейные диаграммы и графики;
- круговые диаграммы;
- контурные графики;
- спектрограммы;
- гистограммы;
- столбцовые диаграммы;
- точечные диаграммы;
- графики рассеяния.

Matplotlib поддерживает различные форматы изображений (.jpg, .png, .pdf, .svg, .eps и другие), а также позволяет создавать анимированные изображения. Из-за большого перечня возможностей представленной библиотеки Matplotlib полноценно и равноправно может конкурировать с большими средами разработок, например, с MATLAB. Именно поэтому данная библиотека поддерживается различными платформами и средами разработки.

Анализ сред разработок

После выбора языкового средства необходимо было проанализировать и выбрать среду разработки, с помощью которой можно было бы создать модуль на языке Python. Были рассмотрены следующие инструментальные средства:

- Jupyter Notebook;
- Visual Studio Code;
- PyCharm;
- Atom.

Кратко опишем приведенные выше технологии.

Jupyter Notebook

Jupyter Notebook — среда с открытым исходным кодом, которая позволяет интерактивно разрабатывать программы на языках Python, Ruby, Haskell, R. Jupyter является графической оболочкой для интерактивной оболочки IPython. Интерфейс данной среды разработки довольно простой и удобный, он состоит из двух частей: веб-приложение и «ноутбуки». Веб-приложение позволяет запускать, редактировать код прямо в браузере. А «ноутбуки» представляют собой файлы, в которых хранится данный код. Jupyter Notebook доступен бесплатно и может быть запущен как локально, так и на удаленном сервере.

Главной особенностью инструмента является то, что он позволяет в одном документе совместить не только программы и скрипты на указанных выше языках программирования, но и содержать текстовые блоки, отображать графические объекты, мультимедиа-контент. «Ноутбуки», созданные в Jupyter Notebook могут содержать различные графики и диаграммы, формулы, уравнения и другие разнообразные заметки. Среда также поддерживает работу с языками разметки такими как LaTeX и Markdown. Такие возможности позволяют дополнить программу и увеличить понимание кода, особенно когда речь идёт о совместной разработке.

Jupyter является отличной средой для разработки программ в области анализа данных. Он содержит в себе все библиотеки для обработки данных, которые входят в состав пакета Python SciPy Stack: Pandas, NumPy, Scipy, Matplotlib. Встроенные библиотеки намного ускоряют и упрощают процесс обработки данных, а возможности, связанные с отображением различного вида графики и интеграцией мультимедиа-контента, позволяют создать документацию непосредственно в "ноутбуке", сделать его максимально дружелюбным для программиста.

Visual Studio Code

Visual Studio Code или VS — редактор и отладчик исходного кода, поддерживающий большое количество языков программирования, таких как C, C#, C++, CSS, HTML, JavaScript, Java, PHP и Python. Несмотря на то, что данный продукт был создан компанией Windows Microsoft, редактор запускается не только

на Windows, но и на Linux, и Mac OS. Редактор позволяет разрабатывать веб-приложения, а благодаря поддержки всеми операционными системами продукты являются кроссплатформенными.

У Visual Studio Code достаточно простой и дружелюбный интерфейс со всеми требуемыми для создания приложения функциями. Позволяет создавать индивидуальные темы, собственные сочетания клавиш для более быстрой и комфортной разработки. Также поддерживает возможность работы с системой контроля версий Git, подсветку синтаксиса, навигацию по коду и возможности загрузки плагинов и расширений для дополнительного расширения функционала. У редактора открытый исходный код и он бесплатен для загрузки как частным лицам, так и компаниям.

В 2018 году было выпущено расширение для языка Python. Плагин распространяется под свободной лицензией и также имеет открытый программный код. Имеет интерфейс для установки модулей и пакетов, поддерживает модульное тестирование, использование виртуальных сред, редактирование кода с подсказками. Всё это в совокупности увеличивает производительность и делает Visual Studio Code одним из наиболее оптимальных и часто используемых инструментов для разработки на Python.

PyCharm

PyCharm — это интегрированная среда разработки, предназначенная непосредственно для работы с языком программирования Python, хотя и имеется поддержка других языков. Среда является кроссплатформенной и может быть установлена как на Windows, так и на Linux и Mac OS.

PyCharm имеет полный набор средств для решения задач любого уровня и разработки проектов любой сложности. Так, для веб-разработки в среде есть поддержка популярных веб-фреймворков таких как Django и Flask. В PyCharm есть также поддержка интеграции с IPython Notebook для комбинирования возможностей и увеличения списка решаемых задач. Что касается анализа данных, среда имеет поддержку научного стека библиотек, включая библиотеки пакета Python SciPy Stack.

PyCharm имеет интуитивно понятный и приятный интерфейс с дополнительными и настраиваемыми функциями: автоматическое форматирование, подсветки синтаксиса, умный редактор кода, быстрая навигация по проекту и исходному коду. Среди преимуществ среды выделяют следующие инструменты:

- встроенный графический отладчик и быстрый рефакторинг;
- систему управления версиями, в том числе поддержка Git;
- юнит-тестирование;
- интерактивная консоль;
- возможность создавать свои плагины или загружать из сети Интернет дополнительные расширения.

Среда имеет две версии бесплатную PyCharm Community Edition и платную PyCharm Professional Edition, отличающихся набором доступных функций.

Выводы к главе 1

1. Проведенный анализ справочной литературы, документации и интернет-источников, а также изучение возможностей программного продукта Autodesk Revit позволяет сделать вывод о том, что данный продукт является оптимальным для решения задач в области BIM-моделирования.
2. В результате анализа расширений и модулей разработчика в Autodesk Revit были выявлены слабые и сильные стороны плагинов. В качестве среды для экспорта данных выбрана среда визуального программирования Dynamo, как наиболее функциональная и соответствующая поставленной заказчиком задачи, связанной с обработкой данных.
3. Для обработки данных были проанализированы и рассмотрены языковые и инструментальные средства. В результате выявления преимуществ и недостатков языковых средств был выбран язык программирования Python, как наиболее подходящий под задачи обработки данных. В качестве

инструментального средства для работы с Python была выбрана интегрированная среда разработки PyCharm.

Во второй главе дипломной работы описан процесс реализации и разработки решения для экспорта и обработки архитектурно-строительных данных с помощью отобранных средств разработки.

ГЛАВА 2. РЕАЛИЗАЦИЯ И РАЗРАБОТКА РЕШЕНИЯ ДЛЯ ОБРАБОТКИ ДАННЫХ ПРОГРАММЫ АВТОМАТИЗИРОВАННОГО ПРОЕКТИРОВАНИЯ AUTODESK REVIT

2.1. Описание заказчика

2.1.1. Данные о заказчике

Заказчиком решения для обработки данных является общество с ограниченной ответственностью «Научно-технический центр «Эталон» (ООО

«НТЦ «Эталон»).

Центр существует и решает различного рода научно-технические задачи с 1997 года, входит в Группу компаний «Эталон», являющегося одним из лидеров жилищного строительства в России.

У центра обширные направления своей деятельности, связанные с различными отраслями промышленности, основные из них это:

- проектирование строительства;
- разработка и внедрение технологий контроля над объектами;
- исследования в области конструкций материалов;
- разработка информационно-измерительных комплексов.

В НТЦ «Эталон» используют передовые технологические разработки в строительстве для реализации инвестиционно-строительных объектов. На всех этапах жизненного цикла строительства используется технология информационного моделирования BIM, открывая на каждом этапе новые возможности решения задач строительства. Компания не один раз получала премию «BIM-Лидера» России, подтверждая свой статус и профессионализм.

В Центре есть целый IT-отдел, который занимается разработками в данной области. В качестве инструмента контроля за безопасностью НТЦ разработали и начали использовать собственную систему управления данными о ходе реализации инвестиционно-строительных проектов Группы «Эталон», которая интегрирована с технологией BIM и методикой оценки рисков. Тесная взаимосвязь между строительной лабораторией и инженерами на строительной площадке позволяют предусмотреть все возможные риски, связанные с безопасностью и охраной труда для контроля качества работ на самом высоком уровне.

Целями создания системы управления данными являются:

- проведение мониторинга охраны труда и техники безопасности на строительных площадках;
- получение информации с подконтрольных площадок в режиме реального времени;

- просмотр в 3D-представлении места нарушений техники безопасности;
- определение мест возникновения критических ситуаций.

2.1.2. Функциональные особенности существующей системы

Система управления данными или официально она называется «Система комплексного контроля строительства (СККС)» представляет собой веб-приложение. Данная платформа является облачным сервисом с помощью которого можно решать разные задачи и управлять проектами на базе BIM-моделей. В результате применения данной разработки создается полноценная структура хранения и обработки данных о конкретном строительном проекте.

В системе есть несколько разделов, разделенных по направлениям решаемых задач. Объединяет их одно – доступ к разделам с цифровых устройств (планшетов, смартфонов, компьютеров) в реальном времени. Такой универсальный доступ позволяет:

- своевременно получать информацию в требуемой детализации;
- проводить анализ тенденций хода строительных работ (опережение или отставание по срокам работ);
- получать текущее 3D-представление о строящемся объекте с отражением фактов изменения по календарному графику строительства.

После внесения или изменений данных, в системе происходит обработка данных, а затем формирование отчета о проделанных результатах, который доступен для импорта.

Система основывается на базе облачной платформы Autodesk Forge, которая позволяет создавать собственные инженерные веб-сервисы и приложения для объединения и комбинации различных разделов строительства, строить совместную работу над проектом и визуализировать данные проекта. Для решения задач с помощью Autodesk Forge на стороне сервера применяется REST API, а на стороне клиента используется популярный фронтенд стек технологий: HTML,

CSS и JavaScript. Платформа позволяет работать с разными типами данных, так как поддерживает более 60 форматов данных.

Продукт является уникальным на рынке охраны труда и был впервые представлен в 2018 году как программное обеспечение строительного контроля. Разработка запатентована и применяется только внутри Группы «Эталон», позволяя экономить время и денежные средства.

2.1.3. Функциональные недостатки существующей системы

Летом 2019 года была пройдена стажировка в НТЦ «Эталон». В этот период НТЦ находился в процессе внедрения комплексной системы в составе которой Autodesk Forge. Одним из недостатков на тот момент была невозможность прямого доступа к параметрам проекта для учета и обработки их данных при формировании конечного отчета.

Часть данных возможно было получить через API Autodesk Forge, но полученные данные были неполными и неупорядоченные, что сказывалось на достоверности формируемых отчетов. Именно поэтому возникла потребность в заказе на решение данной задачи.

2.1.4. Требования заказчика

В результате рассмотрения системы и переговоров с заказчиком были выделены и поставлены следующие задачи:

- получить доступ к определенному типу параметров Autodesk Revit, используя одно из доступных расширений для работы с программным комплексов;
- экспортировать полученные данные в один из форматов данных;
- создать модуль для чтения данных;
- разработать модуль для фильтрации, типизации и сортировки считываемых данных;
- выгрузить обработанные данные в формат данных JSON.

2.2. Разработка программного продукта

2.2.1. Выбор модели жизненного цикла разработки

В области разработки программного продукта одним из основных стандартов, регламентирующий процесс разработки является международный стандарт ISO/IEC 12207:1995 "Information Technology - Software Life Cycle Processes" (ГОСТ Р ИСО/МЭК 12207-99 "Информационная технология. Процессы жизненного цикла программных средств"). Данный стандарт устанавливает общую структуру и порядок процессов жизненного цикла программных продуктов от первоначальной идеи до введения в эксплуатацию продукта.

Согласно стандарту ГОСТ Р ИСО/МЭК 12207-99 в жизненном цикле выделяется три группы процессов, которые включают в себя перечень относящихся к той или иной группе процессов. Первая группа – это основные процессы жизненного цикла, к ним относятся:

- процесс приобретения;
- процесс поставки;
- процесс разработки;
- процесс функционирования;
- процесс сопровождения.

Вторая группа – это вспомогательные процессы, к которым относятся следующие:

- процесс решения проблем;
- процесс документирования;
- процесс управления конфигурацией;
- процесс обеспечения качества;
- процесс верификации;
- процесс аттестации;
- процесс совместной оценки;
- процесс аудита.

И последняя третья группа – это организационные процессы, которые включают в себя:

- процесс управления;
- процесс создания инфраструктуры;
- процесс усовершенствования;
- процесс обучения.

Разработка решения для обработки данных программы автоматизированного проектирования Revit осуществлялась в соответствии с данным стандартом. Для разработки была выбрана одна из моделей жизненного цикла – инкрементная модель.

Инкрементная модель является усовершенствованием классической каскадной модели и представляет процесс разработки как последовательная, частичная реализация функциональных возможностей с их дальнейшим наращиванием. Такой подход позволяет сократить затраты и ускоряет процесс разработки будущего продукта, так как в конце выполнения каждого инкремента получается функциональный продукт. Данная модель может применяться в проектах разного масштаба и сложности, для её применения обязательно необходимо следующее:

- основные требования всегда понятны и определены, но могут в процессе быть доработаны;
- время выполнения и разработки продукта не ограничено в короткий срок;
- разработка программ имеет среднюю или низкую степень риска;
- для будущего продукта оптимален принцип применения пофазовой разработки.

Рассмотрим подробнее фазы разработки продукта при применении инкрементной модели жизненного цикла. На первых этапах жизненного цикла, когда идут планирование, анализ и разработка, происходит конструирование будущего продукта. На этих стадиях выделяются инкременты и функции, а затем уже каждый инкремент по очереди проходит через остальные этапы жизненного цикла: анализ, проектирование, кодирование, тестирование и поставка (см.

Рисунок 2.1). Сначала выполняются и реализуются инкременты, играющие важную и основную роль для формирования основы продукта для выпуска в базовой комплектации. А затем уже выполняются последующие инкременты для улучшения и наращивания функций продукта. Все инкременты реализуются строго в соответствии с поставленными требованиями.



Рисунок 2.1. Инкрементная модель процесса разработки

Среди преимуществ применения данной модели выделяют следующее:

- экономия вкладываемых средств на начальных этапах;
- получение функционального продукта после выполнения каждого инкремента;
- снижение возможных рисков и стабилизирование требований;
- быстрый вывод продукта на рынок;
- с каждым инкрементом улучшается понимание требований для следующих инкрементов.

В результате применения инкрементной модели к разработке, решение разрабатывается поэтапно с выделением инкрементов. В применении к разработке решения для обработки данных инкрементная модель гарантирует работоспособность и качественную функциональность решения на самых ранних этапах разработки. Так, обработка данных сначала происходила с экспортом, чтением, фильтрацией одним типом параметров Revit, а затем уже постепенно прибавлялись остальные типы, приближая решение к конечному результату. Благодаря применению данной модели разработка решения становится оптимальной и помогает спрогнозировать изменения в требованиях дальнейших

инкрементов, тем самым упростив работу и уменьшить срок разработки.

2.2.2. Структура разработки решения для обработки данных

Структура разработки решения для обработки данных является последовательной, и каждая её часть взаимосвязана с другой и их порядок не может быть изменен (см. Рисунок 2.2). Решение для обработки данных начинается с получения самих данных, то есть с получения доступа к конкретным параметрам Revit, требуемых и определенных заказчиком. В качестве инструмента для получения данных была выбрана среда визуального программирования Dynamo (см. Главу 1). После получения параметров данные необходимо экспортировать в удобный для дальнейшей обработки формат данных. Следующим шагом является создание модуля для чтения полученных данных, который находит файл с данными по относительному или абсолютному пути и производит чтение данных программой. Далее идет разработка модуля для фильтрации, сортировки, типизирования считываемых данных. И последним шагом является выгрузка данных в формат JSON для последующей интеграции и использования данных в систему управления данными НТЦ «Эталон».

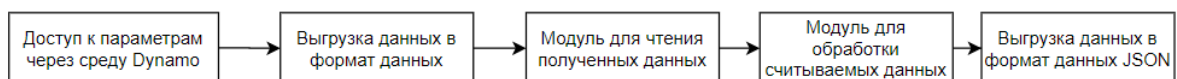


Рисунок 2.2. Структура разработки решения для обработки данных

2.2.3. Получение доступа и экспорт данных с помощью Dynamo

В процессе разработки решения для доступа к архитектурно-строительным данным программы Revit была использована среда визуального программирования Dynamo. Заказчиком был представлен список требуемых параметров для извлечения из BIM-модели здания:

- id;
- unique_id;

- navisworks;
- уровень;
- номер секции №;
- категория;
- тип;
- имя;
- длина;
- площадь;
- объем;
- смещение снизу;
- смещение сверху;
- ширина;
- высота;
- отметка уровня над уровнем земли.

Все вышеперечисленные параметры относятся к разным группам параметров и имеют разный тип данных. Категории, из которых необходимо извлечь данные параметры, следующие:

- стены;
- колонны;
- двери;
- окна;
- пандусы;
- специальное оборудование;
- лестницы;
- перекрытия;
- этажи;
- вентиляционные блоки.

В результате применения инкрементной модели жизненного цикла получение параметров из категорий происходило последовательно. Сначала была создана программа в Dynamo, которая получала данные из параметров только из

одной категории – из стен. Далее в программу постепенно добавлялись ноды, связанные с другими категориями, и на выходе получилась программа, содержащая подпрограммы для категорий (см. Рисунок 2.3).

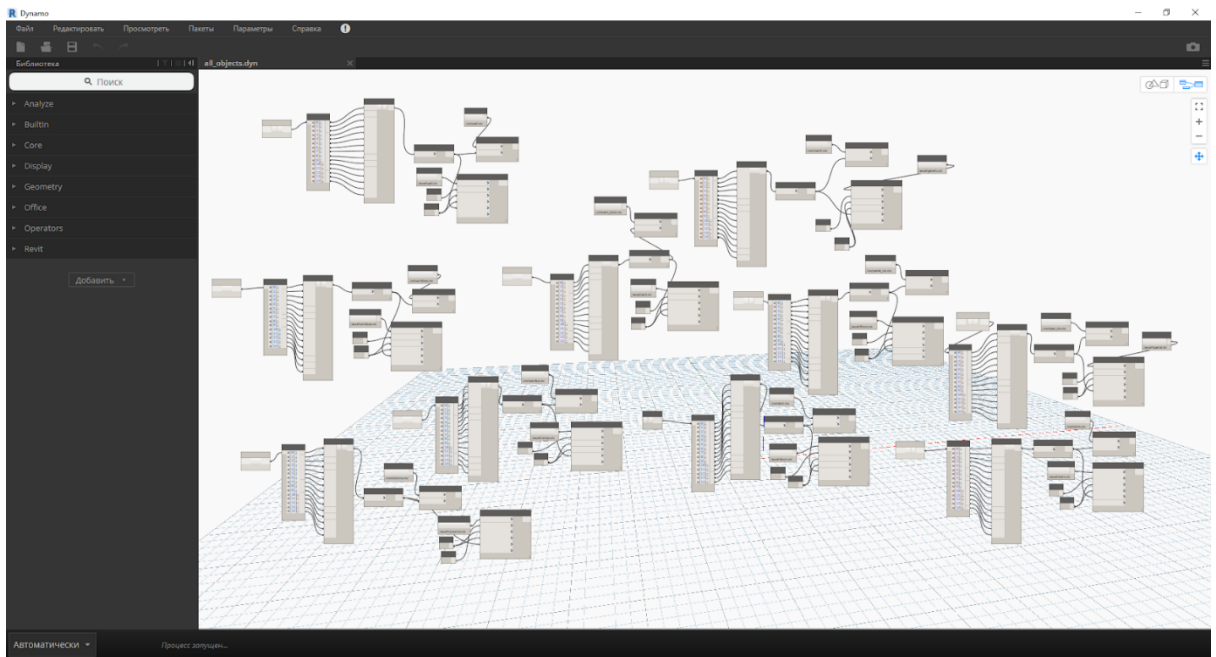


Рисунок 2.3. Итоговая программа в Dynamo

Каждая из подпрограмм содержит в себе следующие ноды:

- Python Script;
- Code Block;
- Level.Elevation;
- List.Create;
- List.Transpose;
- String, Number;
- File.Path;
- CSV.WriteToFile;
- Excel.WriteToFile.

Рассмотрим и опишем задачи представленных нодов.

Python Script - нод, относящийся к группе нодов «сценарии», который позволяет писать внутри код на языке программирования Python, при этом имеет, как и любой другой нод - входные и выходные порты. Данный нод можно использовать как для ускорения выполнения небольших вычислений для Revit,

так и для написания более сложных программ с использованием Revit API.

Для извлечения параметров и получения категорий внутри ноды Python Script написана программа с обращением к классам, методам и свойствам Revit API. Классы используются согласно заданным категориям, из которых необходимо достать данные параметров.

В процессе разработки были использованы следующие методы Revit API:

- `.GetTypeId()` — получения id типа объекта;
- `.GetElement()` — получения типа объекта;
- `.LookupParameter()` — получения параметра по заданному имени параметра;
- `.AsString()` — доступ к строковому содержанию параметра;
- `.AsStringValue()` — получение значения параметра в виде строки с единицами измерения;
- `.AsDouble()` — возвращает значение параметра с типом плавающей точки (double/float);
- `.OfCategory()` — метод получения категории;
- `.WhereElementIsNotElementType()`. — позволяет связать несколько вызовов категории в одной строке;
- `.ToElements()` — возвращает полный набор элементов, которые проходят фильтр(ы).

Тип данных у получаемых параметров из категорий требует типизации с помощью методов. `AsString()`, `AsStringValue()`, `AsDouble()` для дальнейшей обработки данных.

Также в процессе разработки в ноды Python Script были использованы такие свойства Revit API, как:

- `.Symbol` — свойства символов заголовка;
- `.Document` — возвращает документ, в котором находится элемент;
- `.Id` — уникальный идентификатор элемента в проекте Autodesk Revit;
- `.UniqueId` — стабильный уникальный идентификатор для элемента внутри документ;
- `.Name` — читаемое человеком имя для элемента;

- `.Category` — извлекает объект категории, представляющий категорию или подкатеорию, в которой элемент находится.

Для получения отметки уровня над уровнем длины был использован нод `Level.Elevation`. На вход подается нод `Python Script`, с помощью которого получается тип объекта категории. На выходе - список из отметок уровня.

В результате выполнения нодов `Python Script` и `Level.Elevation` на их выходных портах получается список с всеми параметрами, который необходимо упорядочить. Для приведения списка в сгруппированный вид по каждому отдельному объекту были использованы ноды `Code Block`, `List.Create` и `List.Transpose`.

Внутри нода `Code Block` можно писать программы на языке программирования `DesignScript`, разработанного специально для `Dynamo`. `Code Block` – это интерфейс текстовых сценариев в среде визуальных сценариев. Они могут использоваться в качестве чисел, строк, формул и других типов данных. В `Code Block` можно задавать и выбирать способ как задать произвольные переменные, и эти переменные автоматически добавляются на входы узла.

`List.Create` и `List.Transpose` – одни из главных нодов по работе со списками в `Dynamo`. `List.Create` позволяет создавать список, на вход подаются объекты разных типов данных, а на выходе получается массив из этих объектов с нумерацией от 0 (нумерация — это индексы объектов в списке). Нод `List.Transpose` транспонирует (переворачивает) список, то есть заменяет строки столбцами, а столбцы — строками.

После получения и приведения списка в сгруппированный вид, данные необходимо записать в файл формата данных (`.csv` или `.xls`). Для этого были использованы ноды `CSV.WriteToFile` и `Excel.WriteToFile`. Ноды относятся к стандартным узлам для записи данных в файлы `Excel` и `CSV`. У нода `Excel.WriteToFile` имеется 6 входных слотов (переменных):

- `filePath`;
- `sheetName`;
- `startRow`;

- startCol;
- data;
- overWrite.

Вход `filePath` позволяет задать путь к файлу Excel, куда будет идти запись данных. Для этого необходимо воспользоваться нодом `filePath` и выбрать уже существующий, нужный файл для записи. Или же создать новый с помощью нода `Code Block`, в котором будет указано имя нового файла, и файл Excel создается по умолчанию в папке «Мои документы». Во входе `sheetName` необходимо выбрать лист в файле Excel, а затем с помощью слотов `startRow` и `startCol` задаются индексы для начала операции записи, то есть индексы первой строки и столбца записи. В слот `data` подается список и вложенный список с данными. Вход `overWrite` позволяет контролировать будет ли Excel перезаписывать существующую информацию (`true`) или сохранять ее (`false`).

Нод `CSV.WriteToFile` имеет меньше входных слотов по сравнению с `Excel.WriteToFile`: `filePath` и `data`. Слоты имеют такую же функциональность, как и в нод `Excel.WriteToFile`.

В результате переговоров с заказчиком был выбран формат экспорта данных в Excel. На выходе после выполнения программы в Dupaмо получается 10 файлов Excel, разделенных по категориям (см. Рисунок 2.4).

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	AA	AB	AC
13	7120248	e565a576	2.4.3.3.1	(1) Первый этаж	Стены	Базовая с АР, склини АР, склини	674,7293	2 м²	0.29 м³	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
14	7120341	e565a576	2.4.3.3.1	(1) Первый этаж	Стены	Базовая с АР, склини АР, склини	465,5887	1 м²	0.18 м³	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
15	7120427	e565a576	2.4.3.3.1	(1) Первый этаж	Стены	Базовая с АР, склини АР, склини	445,9985	1 м²	0.18 м³	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
16	7120468	e565a576	2.4.3.3.1	(1) Первый этаж	Стены	Базовая с АР, склини АР, склини	343,5088	1 м²	0.10 м³	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
17	7120541	e565a576	2.4.3.3.1	(1) Первый этаж	Стены	Базовая с АР, склини АР, склини	351,4132	1 м²	0.13 м³	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
18	7126929	e565a576	2.4.3.3.1	(1) Первый этаж	Стены	Базовая с АР, склини АР, склини	930,3792	4 м²	0.50 м³	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
19	7127207	e565a576	2.4.3.3.1	(1) Первый этаж	Стены	Базовая с АР, склини АР, склини	930,3792	4 м²	0.50 м³	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
20	7127257	e565a576	2.4.3.3.1	(1) Первый этаж	Стены	Базовая с АР, склини АР, склини	380	1 м²	0.12 м³	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
21	7127334	e565a576	2.4.3.3.1	(1) Первый этаж	Стены	Базовая с АР, склини АР, склини	380	1 м²	0.12 м³	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
22	7127371	e565a576	2.4.3.3.1	(1) Первый этаж	Стены	Базовая с АР, склини АР, склини	930	3 м²	0.37 м³	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
23	7127485	e565a576	2.4.3.3.1	(1) Первый этаж	Стены	Базовая с АР, склини АР, склини	1950	7 м²	0.97 м³	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
24	7127691	e565a576	2.4.3.3.1	(1) Первый этаж	Стены	Базовая с АР, склини АР, склини	2010	7 м²	0.91 м³	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
25	7127776	e565a576	2.4.3.3.1	(1) Первый этаж	Стены	Базовая с АР, склини АР, склини	2010	7 м²	0.91 м³	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
26	7128197	e565a576	2.4.3.3.1	(1) Первый этаж	Стены	Базовая с АР, склини АР, склини	445,2692	2 м²	0.24 м³	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
27	7128239	e565a576	2.4.3.3.1	(1) Первый этаж	Стены	Базовая с АР, склини АР, склини	680,5047	2 м²	0.32 м³	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
28	7128337	e565a576	2.4.3.3.1	(1) Первый этаж	Стены	Базовая с АР, склини АР, склини	405,0048	1 м²	0.18 м³	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
29	7128419	e565a576	2.4.3.3.1	(1) Первый этаж	Стены	Базовая с АР, склини АР, склини	680,5047	2 м²	0.26 м³	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
30	7128499	e565a576	2.4.3.3.1	(1) Первый этаж	Стены	Базовая с АР, склини АР, склини	2895	11 м²	1.29 м³	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
31	7128577	e565a576	2.4.3.3.1	(1) Первый этаж	Стены	Базовая с АР, склини АР, склини	405	1 м²	0.16 м³	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
32	7128886	e565a576	2.4.3.3.1	(1) Первый этаж	Стены	Базовая с АР, склини АР, склини	1590,415	5 м²	0.68 м³	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
33	7129259	2e641d1e	2.4.3.3.1	(1) Первый этаж	Стены	Базовая с АР, склини АР, склини	458,8609	2 м²	0.29 м³	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
34	7129453	2e641d1e	2.4.3.3.1	(1) Первый этаж	Стены	Базовая с АР, склини АР, склини	6105,6005	2 м²	0.29 м³	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
35	7129555	2e641d1e	2.4.3.3.1	(1) Первый этаж	Стены	Базовая с АР, склини АР, склини	664,4322	2 м²	0.25 м³	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
36	7129642	2e641d1e	2.4.3.3.1	(1) Первый этаж	Стены	Базовая с АР, склини АР, склини	2050	7 м²	0.87 м³	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
37	7129712	2e641d1e	2.4.3.3.1	(1) Первый этаж	Стены	Базовая с АР, склини АР, склини	2200	7 м²	0.94 м³	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
38	7129784	2e641d1e	2.4.3.3.1	(1) Первый этаж	Стены	Базовая с АР, склини АР, склини	205,0123	2 м²	0.33 м³	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
39	7129882	2e641d1e	2.4.3.3.1	(1) Первый этаж	Стены	Базовая с АР, склини АР, склини	2165	7 м²	0.94 м³	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
40	7129929	2e641d1e	2.4.3.3.1	(1) Первый этаж	Стены	Базовая с АР, склини АР, склини	400	1 м²	0.13 м³	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
41	7144637	2e641d1e	2.4.3.3.1	(1) Первый этаж	Стены	Базовая с АР, склини АР, склини	415	1 м²	0.12 м³	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
42	7144894	2e641d1e	2.4.3.3.1	(1) Первый этаж	Стены	Базовая с АР, склини АР, склини	2218,892	11 м²	1.49 м³	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
43	7144777	2e641d1e	2.4.3.3.1	(1) Первый этаж	Стены	Базовая с АР, склини АР, склини	1526,562	5 м²	0.64 м³	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
44	7144900	2e641d1e	2.4.3.3.1	(1) Первый этаж	Стены	Базовая с АР, склини АР, склини	199,8557	1 м²	0.09 м³	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
45	7144962	2e641d1e	2.4.3.3.1	(1) Первый этаж	Стены	Базовая с АР, склини АР, склини	438,4388	1 м²	0.14 м³	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
46	7145075	2e641d1e	2.4.3.3.1	(1) Первый этаж	Стены	Базовая с АР, склини АР, склини	1960,001	6 м²	0.84 м³	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
47	7145151	2e641d1e	2.4.3.3.1	(1) Первый этаж	Стены	Базовая с АР, склини АР, склини	678,7046	3 м²	0.38 м³	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
48	7145220	2e641d1e	2.4.3.3.1	(1) Первый этаж	Стены	Базовая с АР, склини АР, склини	380	1 м²	0.12 м³	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
49	7145332	2e641d1e	2.4.3.3.1	(1) Первый этаж	Стены	Базовая с АР, склини АР, склини	380	1 м²	0.12 м³	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
50	7145392	2e641d1e	2.4.3.3.1	(1) Первый этаж	Стены	Базовая с АР, склини АР, склини	1370	5 м²	0.64 м³	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Рисунок 2.4. Пример файла Excel с выгруженными данными

Следующим шагом после получения файлов с данными было создание пакета для чтения и обработки этих данных вне Revit.

2.2.4. Разработка пакета для обработки полученных данных

Разработка решения для обработки данных велась в стиле модульного программирования. Модуль – это функционально законченная часть программы, которая объединяет и содержит в себе различные функции. В результате модули могут объединяться и составлять пакет для решения определенного класса задач.

Пакет для обработки данных состоит из следующих взаимосвязанных друг с другом модулей:

- ModelParamEntity - модуль объявления объектов;
- ModelParamCollection - модуль создания коллекции;
- ModelParamParser - модуль чтения и записи файлов.

Рассмотрим и опишем вышеперечисленные модули.

ModelParamEntity

Модуль ModelParamEntity содержит в себе класс, который создает и объявляет объекты, которые будут заполнены полученными параметрами из Revit. Данные объекты должны пройти проверку на типизацию и заполненность, так как некоторые параметры (Id и UniqueId) не могут быть пустыми. При непрохождении проверки поднимается ошибка и дальнейшее выполнение программы приостанавливается. Также в этом модуле объявляются функции: функция создания списка и функция создания словаря объектов.

ModelParamCollection

Модуль ModelParamCollection создает коллекцию объектов. Коллекция – это структура данных, содержащей в себе набор объектов того или иного типа. Коллекция позволяет записывать, хранить, обрабатывать и извлекать хранящиеся внутри объекты. Коллекция в ModelParamCollection позволяет выполнять следующей действия над объектами:

- добавлять объект в коллекцию;
- подсчитывать длину полученной коллекции;
- добавлять объект в список коллекции;
- добавлять объект в словарь коллекции;
- преобразовывать словарь коллекции в формат JSON.

Для работы с модулем и создании коллекции необходимо импортировать модуль объектов ModelParamEntity.

ModelParamParser

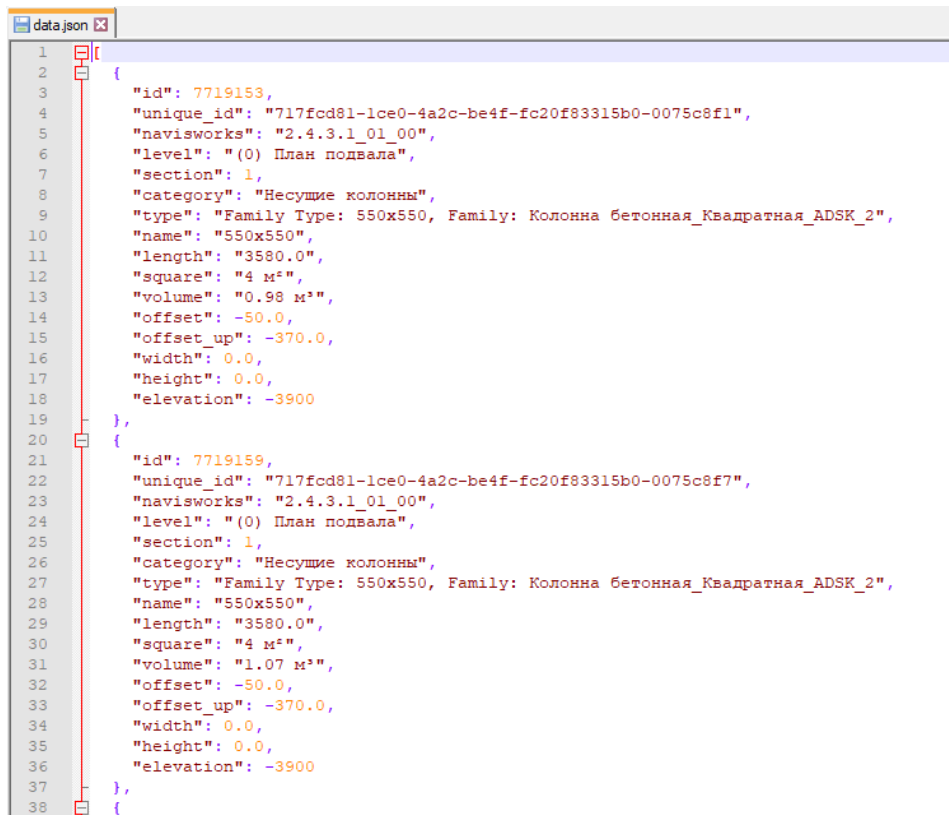
ModelParamParser – это модуль чтения и записи данных в файлы, для функционирования которого необходимо импортировать модули ModelParamCollection и ModelParamEntity. Модуль имеет следующие функции:

- чтение файлов через путь;
- проверка пути;
- поиск и проверка файлов xls;
- создание объектов коллекции;
- заполнение значениями объектов.

Чтение и получение файлов с данными происходит через путь, который задает пользователь. Заданный путь может быть как относительным (путь относительно местонахождение файла), так и абсолютным (полным) путем. Путь проходит на проверку существования, а далее в нем осуществляется поиск файлов с расширением .xls и вывод путей полученных файлов в список. После этого происходит открытие и считывание данных с файлов, которые создают объекты коллекции. Последним шагом является заполнение объектов коллекциями значениями из файлов.

Полученная коллекция объектов, согласно требованиям заказчиков, должна быть выгружена в формат данных JSON. JSON (JavaScript Object Notation) – это простой текстовый формат обмена данными, удобный для хранения и чтения структурированных данных. Выбор данного формата обусловлен дальнейшим применением и внедрением файла в систему на базе облачной платформы Autodesk Forge.

Созданный файл JSON представляет собой словарь, состоящий из набора пар ключ:значение, где ключ – это название параметра, а значение – это данные параметра (см. Рисунок 2.5).



```

1  {
2
3    "id": 7719153,
4    "unique_id": "717fcd81-lce0-4a2c-be4f-fc20f83315b0-0075c8f1",
5    "navisworks": "2.4.3.1_01_00",
6    "level": "(0) План подвала",
7    "section": 1,
8    "category": "Несущие колонны",
9    "type": "Family Type: 550x550, Family: Колонна бетонная_Квадратная_ADSK_2",
10   "name": "550x550",
11   "length": "3580.0",
12   "square": "4 м²",
13   "volume": "0.98 м³",
14   "offset": -50.0,
15   "offset_up": -370.0,
16   "width": 0.0,
17   "height": 0.0,
18   "elevation": -3900
19 },
20 {
21   "id": 7719159,
22   "unique_id": "717fcd81-lce0-4a2c-be4f-fc20f83315b0-0075c8f7",
23   "navisworks": "2.4.3.1_01_00",
24   "level": "(0) План подвала",
25   "section": 1,
26   "category": "Несущие колонны",
27   "type": "Family Type: 550x550, Family: Колонна бетонная_Квадратная_ADSK_2",
28   "name": "550x550",
29   "length": "3580.0",
30   "square": "4 м²",
31   "volume": "1.07 м³",
32   "offset": -50.0,
33   "offset_up": -370.0,
34   "width": 0.0,
35   "height": 0.0,
36   "elevation": -3900
37 },
38 {

```

Рисунок 2.5. Структура файла JSON

Выводы к главе 2

При разработке решения для обработки архитектурно-строительных данных было выполнено следующее:

1. На основе анализа потребностей заказчика и функциональных недостатков в существующей системе управления данными были сформулированы требования к решению для обработки данных.
2. Реализовано и разработано решение для получения доступа, экспорта и обработки архитектурно-строительных данных на основе инкрементной модели жизненного цикла.

В результате разработано полностью готовое к внедрению в систему

программное решение.

Заключение

При выполнении данной выпускной квалифицированной работы был разработан программный продукт «Решение для обработки данных программы автоматизированного проектирования Revit», который позволяет экспортировать и обрабатывать архитектурно-строительные данные для системы управления данными НТЦ «Эталон».

В процессе выполнения работы были решены следующие задачи:

1. Проанализирована справочная литература, документация и интернет-источники по BIM-моделированию в Autodesk Revit и расширению функциональных возможностей.
2. Проанализированы и отобраны инструментальные средства реализации разработки решения для обработки данных.
3. Реализовано решение для получения доступа и экспорта архитектурно-строительных данных с помощью среды визуального программирования Dynamo.
4. Разработан пакет чтения и обработки экспортированных данных.

Таким образом, поставленные задачи полностью реализованы, а цель выпускной квалифицированной работы достигнута.

Решение для обработки данных было внедрено ТА \I " было внедрено " \s "it" \с 1 в «Систему комплексного контроля строительства (СККС)» Группы «Эталон». Данная система была представлена на конференции Autodesk University Russia 2019, которая проходила в Московской школе управления «Сколково».

СПИСОК ЛИТЕРАТУРЫ

1. Культин Н.Б. Инструментальные средства управления проектами строительства: – СПб.: Политехника, 2012. – 234 с.
2. Маккинли У. Python и анализ данных / Уэс Маккинли ; пер. с англ. А.А. Слинкина. – Москва : ДМК Пресс, 2015. – 482 с.
3. Попов В. Л. Управление проектами строительства/ – СПб: Изд-во ПГТУ, 2012. – 176 с.
4. Талапов В.В. Технология BIM: суть и основы внедрения информационного моделирования зданий – М.: ДМК-пресс, 2015. – 410 с.
5. Андреев И.И., Мальцев В.Л. Dynamo. Визуальное программирование в строительстве // Сборник статей II Международной научно-практической конференции; ТИУ. – Тюмень, 2018. – С.169-171.
6. Гогин А.Г. BIM технологии и их будущее // Молодежный научный форум. – 2016. – № 8. – С. 9-14.
7. Казнади С.К. Python и R в анализе данных // Сборник материалов V Международной научно-практической конференции; ООО "ЦНС "Интерактив Плюс". – Чебоксары, 2018. – С.259-261.
8. Кан Д. Некоторые тенденции и перспективы развития строительной отрасли в России // Международный студенческий научный вестник. – 2017. – № 2. – С. 19.
9. Каренгин Г.В., Есипов А.В. Dynamo как способ расширения возможностей Autodesk Revit // Сборник статей II Международной научно-практической конференции; ТИУ. – Тюмень, 2018. – С.216-218.
10. Каширцев М.С., Насырова А.Н. BIM технологии в строительстве // Научное сообщество студентов XXI столетия. ТЕХНИЧЕСКИЕ НАУКИ: сб. ст. по мат. LIV междунар. студ. науч.-практ. конф. – 2017. – № 6(53) – С. 35-39.
11. Кизянов А.О., Лучанинов Д.В. Анализ программных средств, реализующих

- язык программирования Python // ПОСТУЛАТ. – 2016. – № 8. – С. 8.
12. Костюченко Ю.А. Анализ подходов к моделированию данных с помощью библиотек языка Python // XLVII науч. и учебно-метод. конф. Университета ИТМО по тематикам: экономика; менеджмент [30 января – 2 февраля 2018 г.]; Университет ИТМО. – СПб, 2018. – С.175-178.
 13. Кушнарев Ю.А., Калачев О.Н. Язык Python и внедрение в учебный процесс Autodesk Fusion 360 // 69-ая всероссийская науч.-тех. конф. студентов, магистрантов и аспирантов высших учеб. завед. с междунар. участием; Изд. дом ЯГТУ. – Ярославль, 2016. – С. 562-569.
 14. Младзиевский Е.П. Расширение возможностей BIM-проектирования // ACADEMY. – 2020. – № 1. – С. 19-20.
 15. Могилина В.С. Программирование оболочек в Dynamo с использованием Python / В.С. Могилина, А.Н. Сазанова, К.А. Шумилова // материалы II Междунар. науч.-практич. конф. [29–30 марта 2018 г.]; СПбГАСУ. – СПб, 2018. – С.173-177.
 16. Особенности семейств ПК Autodesk Revit Structure / Д.С. Назарова, Н.А. Переверзев, А.В. Васильев, Н.А. Васильева // Современные научные исследования и инновации. – 2019. – № 8.
 17. Русских А. Г. Применение программного комплекса Autodesk Revit в обследовании зданий // Международный научно-исследовательский журнал. – 2016. – № 11 (53) Часть 2. – С. 75-77. 10
 18. Черных М.А. BIM-технология и программные продукты на его основе в России // Вестник ИЖГТУ им. М. Т. Калашникова. — 2014. — № 1 (61). — С. 119-121. 19
 19. Чиковская И., Новоженкина И. Тенденции развития BIM в России // САПР и Графика. – 2014. – № 8. – С. 8-10.
 20. Шишина Д.Л., Сергеев Ф.В. Revit | Dynamo: проектирование объектов сложных форм // материалы II Междунар. науч.-практич. конф. [15–17 мая 2019 г.]; СПбГАСУ. – СПб, 2019. – С.194-199.
 21. BIM-технологии / Е.Н. Рыбин, С.К. Амбарян, В.В. Аносов [и др] // Известия

- вузов. Инвестиции. Строительство. Недвижимость. – 2019. – № 9(1). – С.98-105.
22. Интеллектуальный анализ данных завоевывает все больше популярности в строительной отрасли // DMSTR. – URL: <https://dmstr.ru/articles/intellektualnyy-analiz-dannykh-zavoevyvaet-vse-bolshe-populyarnosti-v-stroitelnoy-otrasli> (дата обращения: 30.04.2020)
23. Назарова Д.С., Переверзев Н.А., Васильев А.В., Васильева Н.А. Особенности семейств ПК Autodesk Revit Structure [Электронный ресурс] // Современные научные исследования и инновации. 2019. № 8 — URL: <http://web.snauka.ru/issues/2019/08/90049> (дата обращения: 12.02.2020).
24. Отчет по исследованию «Уровень применения BIM в России 2019» // Concurator. — URL: http://concurator.ru/information/bim_report_2019/ (дата обращения: 25.03.2020)
25. Сидоров А.Г. BIM – это инструмент прозрачного контроля со стороны инвестора [Электронный ресурс] // Строительный Еженедельник. 2015. №31 — URL: <https://asninfo.ru/magazines/html-version/668-spb/14996-arsentiy-sidorov-bim-eto-instrument-prozrachnogo-kontrolya-so-storony-investora> (дата обращения: 12.02.2020).
26. Сидоров А.Г. BIM. Лучшая практика внедрения ИТ-технологий в градостроительной сфере [Электронный ресурс] // Строительный эксперт — URL: http://ardexpert.ru/article/4239?_utl_t=tw (дата обращения: 03.05.2020).
27. Строительство.ru – Всероссийский отраслевой интернет-журнал. Техника и технологии. Почему Минстрой предпочел BIM-технологии: [Электронный ресурс] – URL: <http://rcmm.ru/tehnika-i-tehnologii/22401-pochemu-minstroy-predpochel-bim-tehnologii.html> (дата обращения: 19.04.2020)
28. Чегодаева, М. А. Трудности внедрения и развития BIM-технологий в России [Электронный ресурс] / М. А. Чегодаева // Молодой ученый. — 2017. — № 29 (163). — С. 29-32 — URL: <https://moluch.ru/archive/163/45194/> (дата обращения: 03.05.2020).
29. Чегодаева, М. А. Этапы формирования и перспективы развития BIM-

технологий [Электронный ресурс] / М. А. Чегодаева // Молодой ученый. — 2017. — № 10 (144). — С. 105-108 — URL: <https://moluch.ru/archive/144/40481/> (дата обращения: 03.05.2020).

30. BIM на стройке. Опыт Группы «Эталон» // Строительный эксперт— URL: <https://ardexpert.ru/article/17691> (дата обращения: 10.04.2020)

