

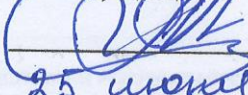
МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение
высшего образования
«ТЮМЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

ИНСТИТУТ МАТЕМАТИКИ И КОМПЬЮТЕРНЫХ НАУК
Кафедра программного обеспечения

РЕКОМЕНДОВАНО К ЗАЩИТЕ В ГЭК

Заведующий кафедрой

К.т.н., доцент



М. С. Воробьева

25 июня 2022 г.

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА
магистерская диссертация

**ПРОЕКТИРОВАНИЕ И РАЗРАБОТКА ПЛАГИНА ДЛЯ АВТОМАТИЗАЦИИ
РАСЧЕТА ХАРАКТЕРИСТИК ПОМЕЩЕНИЙ НА ОСНОВЕ МОДЕЛИ
ЗДАНИЯ В СИСТЕМЕ REVIT**

02.04.03 Математическое обеспечение и администрирование информационных систем

Магистерская программа «Разработка технологий Интернета вещей и больших данных»

Выполнили работу
(групповой проект)

студенты 2 курса
очной формы обучения



Бегун Илья Игоревич



Мухаметзянов Руслан
Аухатович

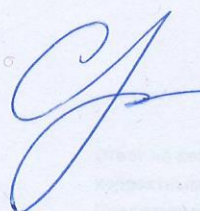
Научный руководитель
Профессор, д. п. н.



Захарова Ирина Гелиевна

Рецензент

Генеральный директор
АО «Институт
«Тюменькоммунстрой»



Никифоров Сергей
Владимирович

Тюмень
2022 год

ОГЛАВЛЕНИЕ

ГЛАВА 1. ИНФОРМАЦИОННОЕ МОДЕЛИРОВАНИЕ ЗДАНИЙ В REVIT	5
1.1 ОСОБЕННОСТИ ТЕХНОЛОГИИ BIM	5
1.2 ОСОБЕННОСТИ ПРОГРАММНОГО КОМПЛЕКСА REVIT	6
1.3. ИНФОРМАЦИОННОЕ МОДЕЛИРОВАНИЕ ТЕПЛОПОТЕРЬ	11
1.4. ИНФОРМАЦИОННОЕ МОДЕЛИРОВАНИЕ КВАРТИРОГРАФИИ	15
ГЛАВА 2. АЛГОРИТМЫ РАСЧЕТА ХАРАКТЕРИСТИК ПОМЕЩЕНИЙ	18
2.1. РАСЧЕТ ТЕПЛОВЫХ ПОТЕРЬ	18
2.1.1 ЧИСТАЯ ПЛОЩАДЬ СТЕН	19
2.1.2. КОЭФФИЦИЕНТЫ И НАДБАВКИ	22
2.1.3. ОПРЕДЕЛЕНИЕ СТОРОНЫ СВЕТА	23
2.1.4. ОПРЕДЕЛЕНИЕ ВНУТРЕННЕЙ/ВНЕШНЕЙ СТЕНЫ	23
2.2. РАСЧЕТ ПЛОЩАДЕЙ КВАРТИРОГРАФИИ	24
2.3. ВАРИАНТЫ ПРОЕКТИРОВАНИЯ ЭЛЕМЕНТОВ ЗДАНИЯ	26
ГЛАВА 3. ПРОЕКТИРОВАНИЕ И РАЗРАБОТКА ПЛАГИНА	39
3.1 АРХИТЕКТУРА ПО	39
3.2 ПОЛЬЗОВАТЕЛЬСКИЙ ИНТЕРФЕЙС	44
ЗАКЛЮЧЕНИЕ	50
БИБЛИОГРАФИЧЕСКИЙ СПИСОК	51
ПРИЛОЖЕНИЕ 1. КОД РАСЧЕТА ТЕПЛОПОТЕРЬ ПОМЕЩЕНИЯ	53

ВВЕДЕНИЕ

Современное общество тесно взаимосвязано с ускорением научно-технического прогресса – в бизнесе, на производстве, в учебных классах и современном офисе – везде применяются новые технологии. Интенсивное развитие автоматических и автоматизированных систем разного масштаба и назначения в первую очередь получило применение в области бизнес-приложений. Такие системы требуют оперативности в обработке, сокращение затрат на выполнение рутинных операций, повышение скорости выполняемых задач, снижение количества ошибок и повышение уровня качества выполняемого процесса.

Популярность автоматизации растет с каждым годом, но рынок услуг в этом направлении, не всегда может помочь компаниям и организациям решить определенные свои профессиональные цели и задачи. Поэтому есть необходимость в разработке таких технологий собственными силами предприятия. Для этого в крупных компаниях созданы собственные IT-отделы, которые занимаются такими информационными вопросами.

В настоящее время активно развиваются средства автоматизированного проектирования (САПР), что не может не сказываться на эффективности и разнообразии реализации строительных процессов. Среди различных вариаций САПР важное место на данный момент занимает информационное (интеллектуальное) проектирование. Проектирование ведётся с использованием технологии BIM (Building Information Modeling) [19].

BIM - совершенно другой, отличающийся от традиционного, чертежно-ориентированного, подход к созданию проекта, который позволяет представить проектный документ в виде единой 3D-модели (далее BIM-модель), наполненной огромным количеством информации, необходимой для эффективного строительства и эксплуатации объекта. При построении BIM-модели можно обеспечить любой уровень детализации, который позволит учесть все нюансы возведения и эксплуатации будущего здания, а также точно рассчитать объёмы и сметы на материалы, оборудование и строительно-

монтажные работы. Инструментами проектирования, поддерживающими идею BIM, являются продукты компании Autodesk [1], а в частности программа Revit [5], предназначенная для моделирования зданий и сооружений.

В числе многочисленных задач, встающих перед инженерами, можно выделить следующие расчетные задачи:

- расчет тепловых потерь;
- расчет и запись значений площадей помещений для квартирографии.

Сложность автоматизации решения таких задач обусловлена особенностями инженерных расчетов, характерных для каждой страны (нормы, правила, методики), индивидуальными геометрическими характеристиками объектов и составляющих их помещений, а также дифференциацией способов моделирования разными инженерами и погрешностями отрисовки элементов модели. Программный комплекс Revit, несмотря на его сложность, в отдельных направлениях не позволяет обеспечить инженера функционалом, позволяющим уйти от рутинных и длительных операций при решении данных задач, что определяет основную цель работы.

Цель: разработать инструмент, который позволит на основании параметров BIM-модели автоматизировать вычисление потерь тепла для выбора оптимальной системы теплоснабжения и расчет площадей помещений для составления декларации по квартирографии. Для этого необходимо выполнить следующие задачи:

1. Выполнить систематизацию подходов к моделированию для выбора алгоритма расчета технологических показателей.
2. Разработать алгоритмы подсчёта площадей и оценки теплопотерь.
3. Реализовать работу алгоритмов на базе программного комплекса Autodesk Revit.

ГЛАВА 1. ИНФОРМАЦИОННОЕ МОДЕЛИРОВАНИЕ ЗДАНИЙ В REVIT

1.1 ОСОБЕННОСТИ ТЕХНОЛОГИИ BIM

Появление технологии BIM существенно меняет процесс проектирования зданий и сооружений, обеспечивая возможность создания единой базы данных на основе информационной модели. Это способствовало повышению эффективности строительной отрасли за счет упрощения процедур взаимодействия между участниками проекта, снижения количества коллизий и работы над исправлениями. В числе преимуществ – ускорение работ, соответствие стандартам качества и безопасности, сокращение затрат на строительство и обслуживание. BIM можно рассматривать как результат эволюции технологии CAD (автоматизированное проектирование) [2]. BIM обладает множеством преимуществ по сравнению с устаревшей системой CAD. Также BIM-технологии обладают большой функциональностью, так как их применение целесообразно не только на стадии проектирования, но также и на стадиях строительства и эксплуатации зданий и сооружений. В настоящее время программы, реализующие технологию BIM, продолжают совершенствоваться. Разработчиками создаются все новые и новые инструменты, которые постоянно обновляются. В частности, ведется работа над дополнениями, которые экономят время, автоматизируя повторяющиеся задачи [13].

Однако использование инструментов для автоматизации работы программ — это процесс, требующий определенных навыков, в частности навыков программирования, который часто отсутствует у проектировщиков в области строительства, что создает трудности в освоении технологий и средств автоматизации. Понимая это, разработчики ПО осуществили интеграцию визуального языка программирования (программирование при помощи сценариев) в BIM программы. Это решение в результате позволило инженерам, не умеющим писать программный код, создавать алгоритмы — серии сценариев, интерпретируемых как связанные узлы, содержащих определенный набор

команд, которые автоматизируют работу программных инструментов или инструментов загрузки. Данное упрощение позволило вовлечь значительное количество пользователей в процесс создания новых инструментов и адаптации программы к современным задачам, что приводит к развитию навыков алгоритмизации и самостоятельного изучения основных принципов программирования.

На данный момент существует несколько способов автоматизации работы BIM программы, которые предназначены для разного уровня подготовки пользователя и сложности поставленной задачи и, следовательно, каждый из этих способов является востребованным. К таким способам можно отнести:

1. Создание алгоритма на языке визуального программирования внутри BIM программы с помощью узлов.
2. Разработка кода на языке программирования в отдельной программе с последующим созданием плагина и загрузкой его в BIM-программу.

1.2 ОСОБЕННОСТИ ПРОГРАММНОГО КОМПЛЕКСА REVIT

Инструментами проектирования, поддерживающими идею BIM, являются продукты компании Autodesk, а в частности программа Revit [5]. Программное обеспечение Revit – это полнофункциональная система автоматизированного проектирования, которая используется для информационного моделирования зданий и сооружений. База данных Revit может содержать информацию о проекте на различных этапах жизненного цикла здания, от разработки концепции до строительства и снятия с эксплуатации. Revit поддерживает межотраслевой процесс проектирования в среде для совместной работы, обеспечивает высокий уровень взаимодействия между экспертами различных дисциплин, независимо от их места нахождения. При этом значительно снижается количество ошибок, вызванных несогласованностью работы разных специалистов по проектированию. Revit позволяет создавать строительные конструкции и инженерные системы любой сложности. На основе проектируемых моделей, специалисты обладают способностью разрабатывать эффективные строительные

технологии и точно определять необходимое количество материалов. Мощные инструменты позволяют использовать процесс, основанный на использовании интеллектуальных моделей, для планирования, проектирования, строительства и эксплуатации зданий и объектов инфраструктуры.

Основные возможности программного обеспечения Revit:

- параметрическое моделирование. Все связи между объектами и элементами задаются параметрами, которые можно динамически изменять;
- двунаправленная ассоциативность. Этот механизм обеспечивает централизованное хранение всей информации о проекте. Возможно изменение любого вида. Внесенные изменения отображаются на всех видах;
- инструменты концептуального проектирования. Создание и свободное управление эскизами и моделями произвольной формы. Возможность создавать формы и геометрию в качестве реальных компонентов здания, что облегчает переход к разработке проекта и выдаче документации;
- цветовые схемы на планах систем. Цветовые схемы позволяют визуализировать дизайнерские идеи. Любые изменения в планах с цветовыми метками автоматически переносятся на модель;
- браузер систем. Позволяет работать с системами по отдельности. Можно выбирать только ту систему, с которой в данный момент нужно вести работу, что позволяет руководить связью элементов любой системы;
- материалы при моделировании. Возможность использования свойств различных строительных материалов при проектировании строительных объектов;
- наличие API. У пользователей есть возможность создавать свои надстройки;
- двусторонняя связь с различными расчетными программами. Такая связь дает возможность модели обновляться, учитывая полученные результаты;
- визуализация проектов. Высокое качество визуализации обеспечивается системой рендеринга, поддерживающей высокую скорость работы.

Интерфейс Autodesk Revit представлен на рисунке 1.1.

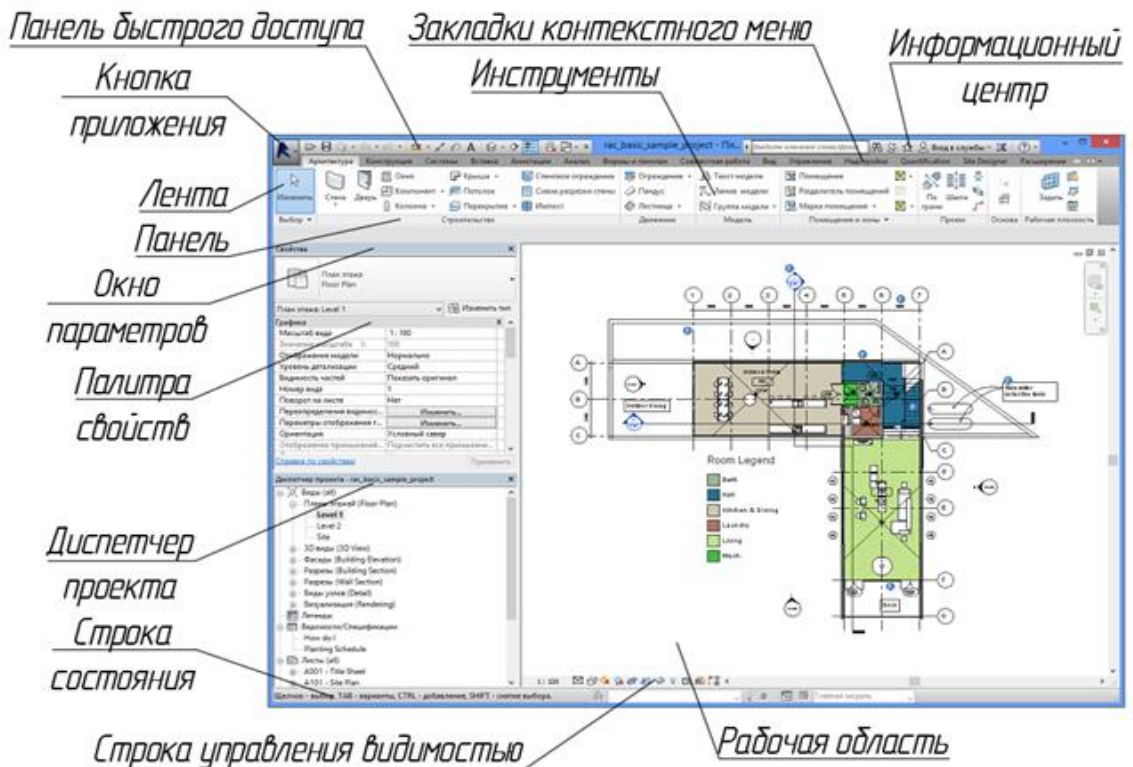


Рис. 1.1. Интерфейс Autodesk Revit

Программный комплекс Autodesk Revit имеет интерфейс прикладного программирования Revit API, который позволяет пользователям добавлять функции программного обеспечения и создавать собственные инструменты и плагины. Плагин - это подключаемый модуль, предназначенный для установки на конкретное ПО и позволяющий расширять его функционал или исправлять изначальные недоработки [20]. Плагин отличается от полноценного софта тем, что он не может использоваться отдельно, как самостоятельный элемент, а обязательно устанавливается и работает внутри программного продукта, на основе его интерфейса.

Также Revit содержит модуль визуального программирования Dynamo, который разрабатывался как плагин для Revit с использованием Revit API и построен с использованием Windows Presentation Foundation [2]. Интерфейс Dynamo изображен на рисунке 1.2.

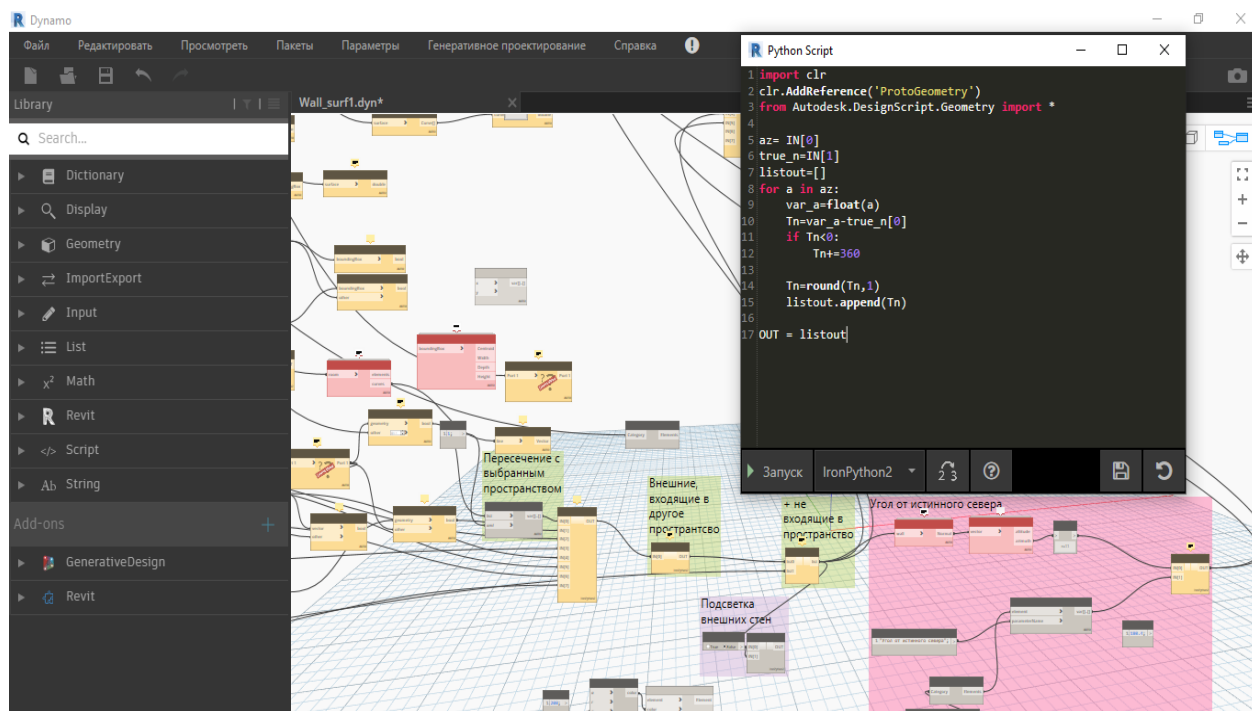


Рис. 1.2. Интерфейс Dynamo

Как механизм параметрического моделирования, Dynamo предназначен для расширения возможностей параметрического моделирования Revit за счет добавления уровня ассоциативности, которого нет в готовом приложении, включая параметры управления, основанные на внешних входных данных, таких как датчики или данные, полученные в результате анализа. Можно сопоставить соответствующие параметры и динамически изменять каждое значение со значением, полученным из входного источника.

Элементы, с которыми пользователи взаимодействуют в Dynamo, называются «узлами». Каждый узел может иметь несколько «портов», которые обеспечивают связь между узлами по «соединителям». Порты могут быть подключены только к другим портам, тип вывода которых соответствует типу ввода порта, или к любому порту, тип вывода которого находится дальше по иерархии наследования типа ввода порта. Вместе эти связанные элементы создают «рабочий процесс».

Базовым классом для Dynamo, от которого наследуются все элементы, является `dynElement`. Как абстрактный класс, `dynElement` предоставляет структуру, на которой строятся все элементы. В этот класс включены три метода,

которые составляют «цикл сборки» Dynamo: destroy, draw и update. Метод destroy используется для очистки элементов или объектов, созданных и принадлежащих узлу. Метод рисования используется для создания элементов или объектов Revit на основе входных соединений. И метод обновления используется для указания «нижестоящим» узлам выполнять свои собственные циклы сборки. Последующие нижестоящие узлы оцениваются до тех пор, пока на узле не возникнет исключение или больше не останется узлов для обработки. Пользователь может реализовать пользовательские функции в цикле сборки, переопределив методы dynElement destroy, build или update.

Узлы классифицируются как «транзакционные» или «нетранзакционные». Транзакционные узлы открывают транзакцию базы данных на этапе сборки, и все элементы, созданные на этом этапе, фиксируются в базе данных; по сути, это означает, что транзакционные действия позволяют обновлять базу данных сборки. Нетранзакционные узлы не открывают транзакции, поскольку им не нужно изменять базу данных Revit. Примером такого транзакционный узел – это узел, который создает или редактирует группу экземпляров семейства. Для сравнения, нетранзакционным узлом может быть узел, который взаимодействует с последовательными данными. Если узел выдает исключение или ошибку во время обработки, транзакция базы данных отменяется, и база данных возвращается к своему предыдущему состоянию.

Визуальное программирование обеспечивает широкие возможности, которых может хватить для многих задач. Этот способ доступен большинству пользователей, чем привлекает их. Но также следует отметить, что работа сложного алгоритма насыщена узлами в Dynamo, особенно при доступе к файлам вне программы, является более продолжительной и тяжелой для компьютера и для самого Revit.

Основным способом создания новых инструментов или автоматизации существующих, предоставляющим наиболее широкий выбор команд и функций, является написание кода. Revit имеет многофункциональный API и встроенные средства для разработки плагинов [2]. Revit API меняется между версиями Revit

и даже обновлениями. Важно знать какие изменения произошли между версиями API, поскольку написанный код может быть устаревшим или быть изменен в более поздних версиях. В программе пользователь может применять любой язык программирования, использующий платформу .NET (C#, VB.NET, F# и т.д.), для создания плагинов.

Revit API содержит множество команд, методов, которые выполняют требуемую пользователем операцию. Однако для того, чтобы вписать эти команды в программный код и обеспечить их необходимыми корректными данными, извлеченными из модели, необходимо иметь навыки кодирования, что не входит в обязанности строителя-проектировщика, поэтому вызывает затруднения. Обучение пользователей прикладному использованию языков программирования в таких случаях сталкивается с рядом трудностей, которые делают невозможным повсеместное применение этого метода. Однако при наличии навыков программирования этот метод является наиболее эффективным, как в плане скорости алгоритма, так и в плане нагрузки на компьютер и сам программный комплекс.

1.3. ИНФОРМАЦИОННОЕ МОДЕЛИРОВАНИЕ ТЕПЛОПТЕРЬ

При расчете систем отопления для любых помещений основной целью является определение теплопотерь. Теплопотери — это тепло, бесцельно уходящее за пределы здания [11]. Суммарные теплопотери складываются из основных и добавочных. Основные тепловые потери определяют путем суммирования утечек теплоты через ограждающие конструкции помещения. Добавочные же зависят от ориентации ограждающих конструкций по сторонам света, а также от расположения здания на открытой местности, скорости ветра в данном географическом районе.

Расчет теплопотерь является важнейшим этапом проектирования систем отопления. Для определения тепловой мощности, покрывающей максимальную нагрузку на систему отопления, необходимо знать теплопотери здания в самую суровую расчетную часть холодного периода года. Для решения вопроса о

соответствии уровня теплотребления системой отопления здания современным требованиям, особенно учитывая проблему энергосбережения, необходимо определить теплотери здания за весь отопительный период.

Теплотери нельзя рассчитать, не зная теплозащитных качеств ограждений, коэффициентов теплообмена на поверхностях, расчетных наружных и внутренних условий. Поэтому в работе достаточно большое место уделено этим характеристикам. Кроме того, по многим вопросам приведены обоснования общеизвестных рекомендаций и указаны их авторы. Вместе с тем представленный материал не претендует на всеохватывающее изложение сопутствующих вопросов.

Существуют различные подходы к выбору расчетных значений коэффициентов теплопроводности строительных материалов. При этом тщательность в выборе значения данного коэффициента крайне важна, принимая во внимание тот факт, что производители теплоизоляции зачастую приводят в рекламных материалах теплопроводность не при эксплуатационных условиях, а в сухом состоянии. Необходимо также правильно оценивать значения коэффициентов теплообмена на поверхностях ограждений, особенно коэффициента теплоотдачи на внутренней поверхности, т.к. при завышенном его значении будет завышена и расчетная температура на внутренней поверхности, например, окна.

При определении теплотерь здания важна правильная оценка коэффициентов теплопередачи ограждающих конструкций.

В настоящий момент, подбирая мощность отопительной системы помещения, часто руководствуются средним значением в 100 Вт на 1 кв. м площади при стандартной высоте потолков до трех метров [15]. Однако не всегда эта мощность достаточна для полного восполнения теплотерь. Здания различаются по составу строительных материалов, их объему, нахождению в разных климатических зонах и т.д. Для грамотного расчета теплоизоляции и подбора мощности отопительных систем необходимо знать о реальных теплотерях дома.

Основные потери тепловой энергии зданий приходится на стены, крышу, окна и полы. Значительная часть тепла покидает помещения через системы вентиляции.

В основном на теплопотери влияют следующие два фактора:

- разница температур в помещении и на улице, т.е. чем она выше, тем больше теплопотери;
- теплоизоляционные свойства ограждающих конструкций (стены, перекрытия, окна).

Ограждающие конструкции препятствуют проникновению тепловой энергии наружу, потому что обладают определенными теплоизоляционными свойствами, которые измеряют величиной, называемой сопротивлением теплопередаче.

Эта величина показывает, каков будет перепад температур при прохождении определенного количества тепла через 1 м^2 ограждающей конструкции или сколько тепла уйдет через 1 м^2 при определенном перепаде температур.

Для расчета теплопотерь в оконные заполнения необходимо знать следующие параметры: количество камер и переплетов, наличие покрытия и заполнение газом.

При расчете заполнения дверных проемов необходимо учитывать добавку на врывание холодного воздуха через наружные двери, не оборудованные воздушными или воздушно-тепловыми завесами, при открывании их на короткие периоды времени. Эта добавка относится к теплопотерям дверей и учитывает потребность в расходе тепла на подогрев врывающегося через открытые двери наружного воздуха. Перечисленные добавки не учитываются, если двери являются летними или запасными, т. е. не открываются постоянно. В промышленных зданиях врывание холодного воздуха через ворота при открывании их в общей сложности не более чем на 15 минут в смену учитывается тем, что теплопотери через ворота утраиваются. При большом времени открытия

ворот врывание холодного воздуха должно локализоваться путем устройства специальных воздушных завес или тамбуров.

Инфильтрация — это перемещение воздуха через ограждающие конструкции из окружающей среды в помещения за счет ветрового и теплового напоров, формируемых разностью температур и перепадом давления воздуха снаружи и внутри помещений. Она происходит через небольшие щели в дверных и оконных рамах. Воздух поступает в помещение также из неотапливаемых частей здания — чердаков, подвалов и так далее. Он проникает через отверстия в стенах, полах и потолках, таких как трещины в местах сопряжения двух стен или стены и потолка. Для определения количества фильтрующегося воздуха через окна и стены необходимо найти разность давлений воздуха на наружной и внутренней стороне ограждающей конструкции.

Теплопотери зданий неоднородны и зависят от множества факторов. Подход к расчету может занимать огромное количество времени и усилий специалистов, так как необходимо обратить на все особенности помещений.

На рисунке 1.3 представлена неоднородность теплопотерь объекта. Наиболее сильные теплопотери помечены красным цветом.

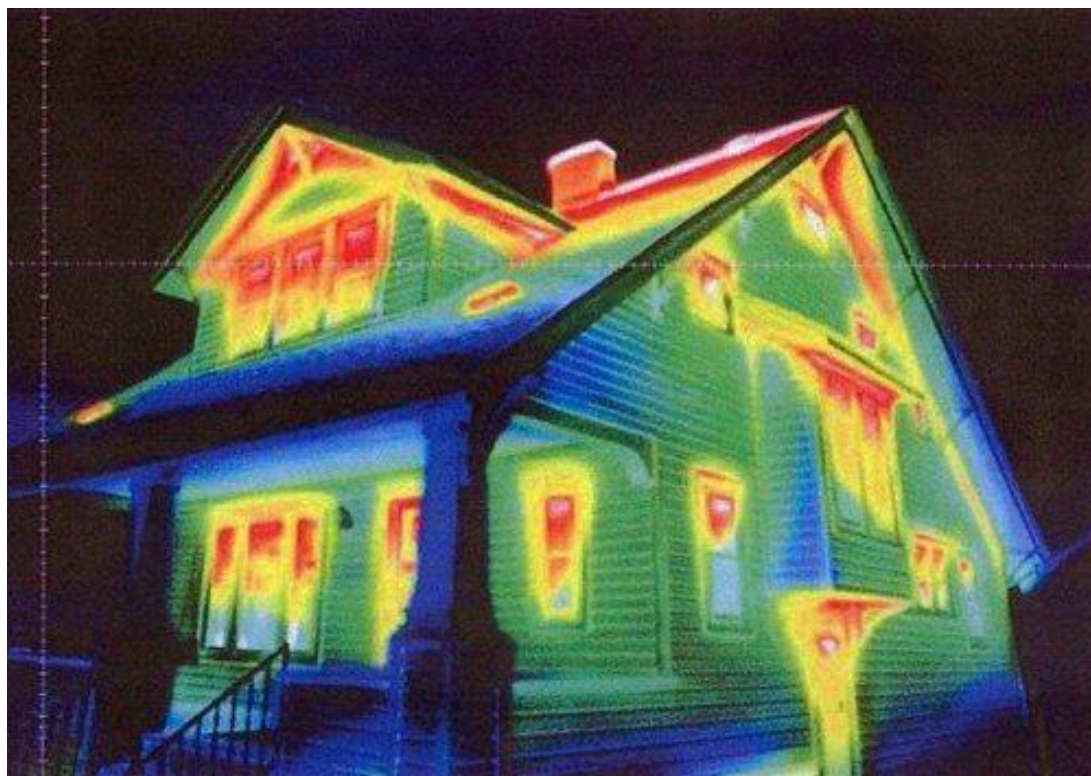


Рис. 1.3. Неоднородность теплопотерь

1.4. ИНФОРМАЦИОННОЕ МОДЕЛИРОВАНИЕ КВАРТИРОГРАФИИ

Квартирография - одно из важнейших решений любого объекта жилой недвижимости. От того, какие площади, конфигурации предусмотрит застройщик, зависит дальнейший успех и продажи новостройки. Ошибка в расчетах соотношения разного типа квартир может свести на нет остальные усилия. Подобные нюансы утверждает заказчик объекта, когда пишет техническое задание и отмечает, по какой цене планирует продавать каждый квадратный метр. Застройщик выполняет анализ спроса покупателей и разрабатывает квартирографию на основе полученных данных. В зависимости от этого закладывается бюджет на строительные материалы, а также определяются планировки квартир. Возможны ситуации, когда заказчик вложил в проект меньше средств, а продажи при этом растут. В данном случае он прогадал, так как не сделал класс жилья выше и не продал квадратные метры дороже. Застройщикам нужно более гибко подходить к изменениям, которые происходят

на рынке, быстро на них реагировать и при необходимости вовремя корректировать проект. Пример квартирографии помещений на секции с указанными площадями представлен на рисунке 1.4.

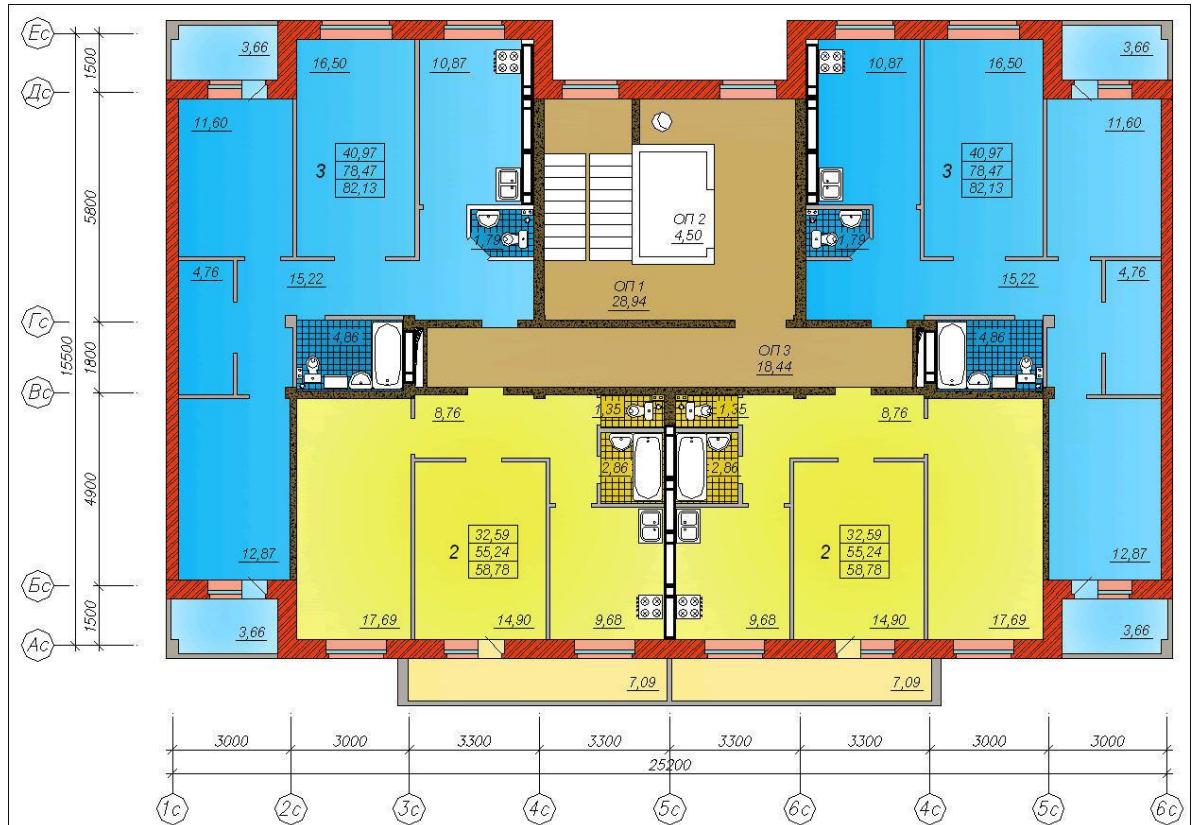


Рис. 1.4. Квартирография

В процессе определения квартирографии рассчитывается точная площадь всех помещений с учетом всех сложностей их плановой конфигурации для оценки стоимости в договорах с покупателями. Данные площади используются для контроля экономической эффективности проекта и составления декларации, которую предъявляют в банк для оценки проектного финансирования и открытия эскроу-счетов.

В программном комплексе Autodesk Revit нет возможности стандартными средствами получить квартирографию - помимо того, что в программе нет понятия «квартира», в марку помещения невозможно вывести

суммарные значения площадей, тем более, с учетом коэффициентов.

Безусловно, есть обходные пути, но они или требуют много ручной работы, или решают задачу лишь частично. Однако данные по квартирографии являются важной частью документации и отсутствие удобных функций зачастую снижает потенциальный круг пользователей.

ГЛАВА 2. АЛГОРИТМЫ РАСЧЕТА ХАРАКТЕРИСТИК ПОМЕЩЕНИЙ

2.1. РАСЧЕТ ТЕПЛОВЫХ ПОТЕРЬ

На рисунке 2.1 представлена схема IDEF0 расчета тепловых потерь помещений.



Рис. 2.1. Процесс расчета тепловых потерь помещения

Входные параметры: ограждающие конструкции, грани помещений.

В ограждающих конструкциях встречались разные семейства. В основном это были элементы из категории стен. Также в некоторых моделях попадали колонны.

В роли механизмов выступают: алгоритм по расчету чистой площади стены, алгоритм определения пирога стены, алгоритм по определению

расположения стены относительно внешнего пространства, алгоритм нахождения ориентации стены по сторонам света.

В роли управления: нормативная документация, документация по Revit API, методики расчета.

Нормативная документация - ГОСТы, нормы и правила проектирования зданий [16, 17].

Документация по Revit API - правила взаимодействия программным кодом с системой Revit и ее элементами. Функциональный блок отображает процесс расчета теплотерь помещений. На выходе результатом выполнения процесса будет значение, характеризующее потери тепла.

2.1.1 ЧИСТАЯ ПЛОЩАДЬ СТЕН

Для расчета теплотерь необходимо определить чистую площадь всех ограждающих конструкций. Для этого необходимо получить помещение, по которому будут рассчитываться теплотери. Получаем помещение при помощи Revit API [10], далее с этого помещения получаем его геометрию (метод `GetGeometry()`), полученная геометрия - `solid` (твердое тело - это объект, граница которого определяется гранью и краем). Необходимо получить составляющие `solid` при помощи свойства `Faces` объекта. Данные составляющие - грани фигуры (`face`).

При работе с гранями и их свойствами возникало множество вопросов по ним. Например, стоит ли грани, с маленькой площадью учитывать при подсчетах. Revit не позволяет посмотреть отдельную грань и понять где она находится и что из себя представляет. Возникла необходимость в наглядном представлении этой геометрии. При помощи класса `Shape`, были созданы экземпляры объекта, размеры которой соответствуют граням помещения. После того, как появилась возможность посмотреть на каждую грань и ее расположение в помещении, стало понятно с чем мы имеем дело.

Далее необходимо было отобрать только вертикальные грани, так как нужна площадь только со стен помещения. Обратимся к ориентации этих граней.

Ориентация представлена в виде координаты, показывающей направление единичного вектора, падающего на плоскость грани. Эта координата имеет три параметра - X , Y , Z . Напрямую у `face` взять это направление не получится, так как этого параметра нет. Необходимо преобразовать `face` в `PlanarFace`. `PlanarFace` - это грань тела или оболочки, ограниченная контуром.

После того, как получили ориентацию, требуется отбросить все грани, у которых значение Z равняется 1, или -1, так как они являются верхними и нижними в помещении и соответствуют перекрытиям. После того, как отобрана вертикальная грань, определим у нее пять значений - максимальное значение координаты X , максимальное значение Y , минимальное значение X , минимальное значение Y , центральную точку грани. Минимальные и максимальные значения формируются с помощью метода `Triangulate()`, обращенный к `face` и свойства `Vertices`.

Таким образом, получаем набор вершин, из которых состоит грань. Перебирая эти вершины, собираем максимальные и минимальные значения X , Y у грани. Центральную точку получаем с помощью метода `GetBoundingBox()` и `Evaluate()`. Эти методы дают понять максимальное и минимальное значение координат у грани, после чего необходимо сложить максимальное и минимальное значения и разделить на два. Собранные максимальные и минимальные значения X , Y , а также центральная точка у грани необходимы для соотнесения к ограждающей конструкции текущего помещения.

Обратимся к нашему помещению и выявим у него список ограждающих конструкций. Необходима проверка на наличие связанного документа, так как при работе в запущенном документе, не получится обработать конструкцию, если она была создана в другом файле, нужно будет обращаться к родному файлу для ограждающей конструкции для получения всего набора информации по ней. Перебирая ограждающие конструкции и их грани, необходимо предварительно проверить на идентичность ориентации с ориентацией грани помещения. Они должны совпадать.

Также собираем максимальные и минимальные значения X , Y , центральную точку. И проверяем входит ли центральная точка грани конструкции в диапазон значений наших координат у грани помещения. Встречались различные случаи моделирования объекта в Revit, поэтому нужно проверить входит ли центральная точка грани помещения в диапазон максимальных минимальных точек ограждающей конструкции. При успешном исходе присваиваем ограждение к нашей грани и записываем чистую площадь (не исключая проемы). Если другая грань помещения также определит ограждение, у которой уже есть соответствие, площади суммируются. Из этой полученной площади нужно вычесть площадь проемов, которые относятся к этой стене и входят в рамки помещения. В ограждающие конструкции попадают только ближайšie элементы.

Для того, чтобы получить все проемы, относящиеся к конкретному ограждению, необходимо получить так называемый «пирог» стены. То есть пробуем получить всю многослойную конструкцию стены. Это нам нужно для того, чтобы дойти от отделки, до самой стены, в которую и встроены проемы. Этот модуль реализован с помощью луча, который ведется от центра помещения в сторону центральной точки проекции грани помещения на линию стены каждый раз сдвигая конец этого луча на небольшой промежуток. По этому лучу строится геометрический объект (solid) и происходит проверка на пересечения с объектами. Луч двигается до тех пор, пока не попадет в другое помещение, или вовсе текущая точка не будет относиться ни к какому помещению. Используем метод FindInserts к стене, получая список проемов.

Теперь каждый проем необходимо проверить на вхождение в диапазон максимальных и минимальных точек грани помещения, которая уже имеет связь к ограждающей конструкции, с которой был получен пирог. С помощью метода `get_BoundingBox`, примененного к проему, получаем его центральную точку. После этого необходимо провести перпендикуляр к грани помещения с помощью метода `Project`, если получилось, то считаем, что проем относится к

помещению и его нужно учитывать. Считываем площадь проема и вычитаем из площади текущей грани помещения. Чистая площадь для стены получена.

2.1.2. КОЭФФИЦИЕНТЫ И НАДБАВКИ

Основные потери (Вт) рассчитываются по формуле

$$Q_{осн} = S_{ст} * \lambda * \Delta t \cdot n * \sum \beta + 1, \quad (2.1)$$

где $S_{ст}$ (m^2) - чистая площадь стен,

λ (Вт/м \cdot °C) - коэффициент теплопроводности стены,

$\Delta t \cdot n$ – коэффициент,

$\sum \beta + 1$ - сумма всех надбавок (надбавка на ориентацию, надбавка на две стены, прочие надбавки) плюс единица.

Коэффициент $\Delta t \cdot n$ получаем по формуле

$$n * (t_{вн} - t_{нар}), \quad (2.2)$$

где n - коэффициент, зависящий от положения строительной конструкции в пространстве,

$t_{вн}$ (°C) - внутренняя температура равная 22 (температура по ГОСТу от 16 до 24, оптимальные от 21 до 22),

$t_{нар}$ (°C) - расчетная температура наружного воздуха наиболее холодной пятидневки (минус 35 для города Тюмень).

Коэффициент n для стены рассчитывается, исходя из вектора, определяющего направление стены (ориентация). Если координата Z равна нулю, то есть стена строго вертикальная, то этот коэффициент n равен единице.

Надбавка на ориентацию (для внешних стен): присваивается исходя из ориентации стены (север - 0.1, юг - 0, запад - 0.05, восток - 0.05).

Надбавка на две стены: присваивается коэффициент 0.1, если образуется угол из двух внешних стен.

Прочие надбавки. Если у ограждающей конструкции найден проем (дверь), коэффициент равняется 1.98.

Потери на инфильтрацию (Вт) рассчитываются по формуле

$$Q_{инф} (Вт) = Q_{осн} * K_{инф}, \quad (2.3)$$

где $Q_{осн}$ (Вт) - основные потери,

$K_{инф}$ - коэффициент инфильтрации.

Инфильтрация - это процесс проникновения воздуха в помещение, через неплотности окон и дверей. Коэффициент инфильтрации зависит от типа зданий, величины и количества окон, силы и направления ветра, количества и типа дверей и ворот [12]. Его значение при расчетах равняется 0.09.

Общие тепловые потери находятся по формуле

$$Q_{общ} \text{ (Вт)} = Q_{осн} + Q_{инф}, \quad (2.4)$$

где $Q_{осн}$ (Вт) - сумма основных потерь,

$Q_{инф}$ (Вт) - потери на инфильтрацию.

Внутренняя температура, коэффициент теплопроводности (сопротивление конструкции, оказываемое наружному воздуху) для стены.

2.1.3. ОПРЕДЕЛЕНИЕ СТОРОНЫ СВЕТА

Для определения стороны света для стен, относящихся к помещению, необходимо определить нормаль стены и развернуть ее направление от помещения. Это нужно для того, чтобы две противоположные стены в помещении не имели одинаковую сторону света. Это может происходить, если сами конструкции стен развернуты в одну сторону.

После того, как получена нормаль стены, необходимо найти угол между севером (направление севера представлена в виде координаты (0,1,0)) и нормалью. Полученный угол необходимо идентифицировать как направление. Есть диапазон градусов к которым относится каждое направление: север - от 315 градусов (включительно) до 45 градусов (включительно), запад - от 225 (включительно) градусов до 315 градусов, восток - от 45 градусов до 135 градусов (включительно), юг - от 135 градусов до 225 градусов.

2.1.4. ОПРЕДЕЛЕНИЕ ВНУТРЕННЕЙ/ВНЕШНЕЙ СТЕНЫ

Стены могут быть внутренними или внешними, в зависимости от того имеет ли стена выход на улицу. Если стена имеет выход на улицу, то она

считается внешней, иначе внутренней. Для того, чтобы определить направление стены для начала требуется определить корректную нормаль стены, которая должна быть направлена от помещения. Если нормаль стены направлена в помещение, то ее нужно развернуть.

Далее необходимо координату центра грани стены передвигать в направлении нормали стены до тех пор, пока не будет найдено другое помещение. Если помещение было найдено, то стена является внутренней, если нет, то внешней. Для определения принадлежности координаты какому-либо помещению используется метод `GetRoomAtPoint` из Revit API.

2.2. РАСЧЕТ ПЛОЩАДЕЙ КВАРТИРОГРАФИИ

При проектировании выделяют несколько различных типов помещений, такие как жилое, нежилое отапливаемое, нежилое неотапливаемое (например, лоджия), балкон. Для расчета площадей, необходимых для квартирографии, сначала необходимо в зависимости от типа помещения определить коэффициент площади. В таблице 2.1 приведено соответствие коэффициента к типу помещения.

Таблица 2.1

Соответствие коэффициента площади типу помещения

Тип помещения	Коэффициент	Описание
1	1	Жилое
2	1	Нежилое отапливаемое
3	0.5	Нежилое неотапливаемое (лоджия)
4	0.3	Балкон

В таблице 2.2 представлен список параметров, рассчитываемых при составлении квартирографии. В таблице указывается название параметра, как выполняется расчет данного параметра, и тип данных.

Таблица 2.2

Параметры для квартирографии

Название параметра	Формула расчета	Тип
ADSK_Площадь квартиры общая	Сумма площадей всех помещений с одинаковыми значениями в параметре «ADSK_Номер квартиры»	Площадь
ADSK_Площадь квартиры жилая	Сумма площадей всех помещений с одинаковыми значениями в параметре «ADSK_Номер квартиры» и со значением параметра «ADSK_Тип помещения» = 1	Площадь
ADSK_Количество комнат	Количество всех помещений с одинаковыми значениями в параметре «ADSK_Номер квартиры» и со значением параметра «ADSK_Тип помещения» = 1	Число
ADSK_Индекс помещения	«ADSK_Номер квартиры»_ «ADSK_Тип помещения»	Текст
ADSK_Коэффициент площади	Заполняется в зависимости от значения параметра «ADSK_Тип помещения»	Числовой
ADSK_Площадь с коэффициентом	«Площадь»*«ADSK_Коэффициент площади»	Площадь

ADSK_Площадь квартиры	Сумма площадей всех помещений с одинаковыми значениями в параметре «ADSK_Номер квартиры» и со значениями параметра «ADSK_Тип помещения» = (1 или 2)	Площадь
ADSK_Площадь квартиры общая с коэфф	Сумма значений параметра «ADSK_Площадь с коэффициентом» всех помещений с одинаковыми значениями в параметре «ADSK_Номер квартиры»	Площадь
ITKS_Площадь комбинированная	Для типов помещения 1 и 2 - «Площадь» Для типов помещения 3 и 4 - «Площадь»/«ADSK_Площадь с коэффициентом»	Текст
Площадь комбинированная	Аналогично «ITKS_Площадь комбинированная»	Текст

2.3. ВАРИАНТЫ ПРОЕКТИРОВАНИЯ ЭЛЕМЕНТОВ ЗДАНИЯ

Процесс проектирования в Revit состоит из следующих этапов:

1. Подготовка моделей компонентов зданий, конструкций или систем.
2. Моделирование зданий, конструкций или систем.
3. Оформление спецификаций.
4. Оформление чертежей.

При моделировании зданий в программном комплексе Autodesk Revit, архитекторы применяют различные способы и подходы. Также проекты бывают разной степени сложности в плане проектирования. От одноэтажных зданий с несколькими помещениями до больших многоэтажных жилых комплексов. Кроме того, сама по себе модель постепенно усложняется в процессе

прохождения этапов информационного проектирования. От всего этого зависит и работа алгоритмов, реализуемая в плагине. Необходимо предусмотреть и учесть все варианты элементов и их проектирование, которые могут встретиться при работе с помещениями. А также правильно их обработать и считать информацию с элементов.

Архитекторы и инженеры-механики, совместно работающие над моделью, должны понимать некоторые особенности помещений и пространств.

В Revit архитекторы используют помещения и области для разделения модели здания в соответствии с использованием, вместимостью и другими критериями. В то же время инженеры, проектирующие системы зданий, используют пространства и зоны для анализа отопительных и холодильных нагрузок.

При формировании связи архитектурной модели с моделью MEP необходимо учитывать следующее:

Пространства (созданные в модели MEP) могут быть ограничены элементами в связанных моделях, в главной модели, или возможны оба варианта.

Пространства зависят от положения линий-разделителей помещений. На помещения не влияет положение линий-разделителей пространств.

Пространства измеряются от чистой поверхности стены. В них используется расчетная высота, определенная в архитектурной модели.

Несколько пространств могут обращаться к образу одного помещения в связанной модели. Помещения могут существовать в параметрах конструкции. Пространства не могут существовать в параметрах конструкции.

При изменении архитектурной модели в основной модели MEP пространства не удаляются. Пространства могут стать незамкнутыми, избыточными или неоднозначными при внесении изменений в главную модель.

Изменение одной модели не распространяется на связанные модели. Если архитектурная модель и модель MEP связаны друг с другом, то изменения в архитектурной модели могут не совпадать с изменениями в модели MEP до тех пор, пока не будут выполнены операции открытия, повторного сохранения и перезагрузки модели MEP [18].

В Revit пространства в основном используются для решения задач инженеров, а помещения для архитекторов. Также отличие состоит в том, что в Revit у них разный набор параметров.

Модель здания может быть построена с помощью пространств или помещений. По своей сути именно с поиска помещений/пространств плагин начинает взаимодействие с моделью. В моделях в основном используются помещения. Помещения были взяты за основу, так как не было необходимости в использовании сразу двух из этих сущностей.

На рисунке 2.2 изображена часть плана с использованием пространства при проектировании.

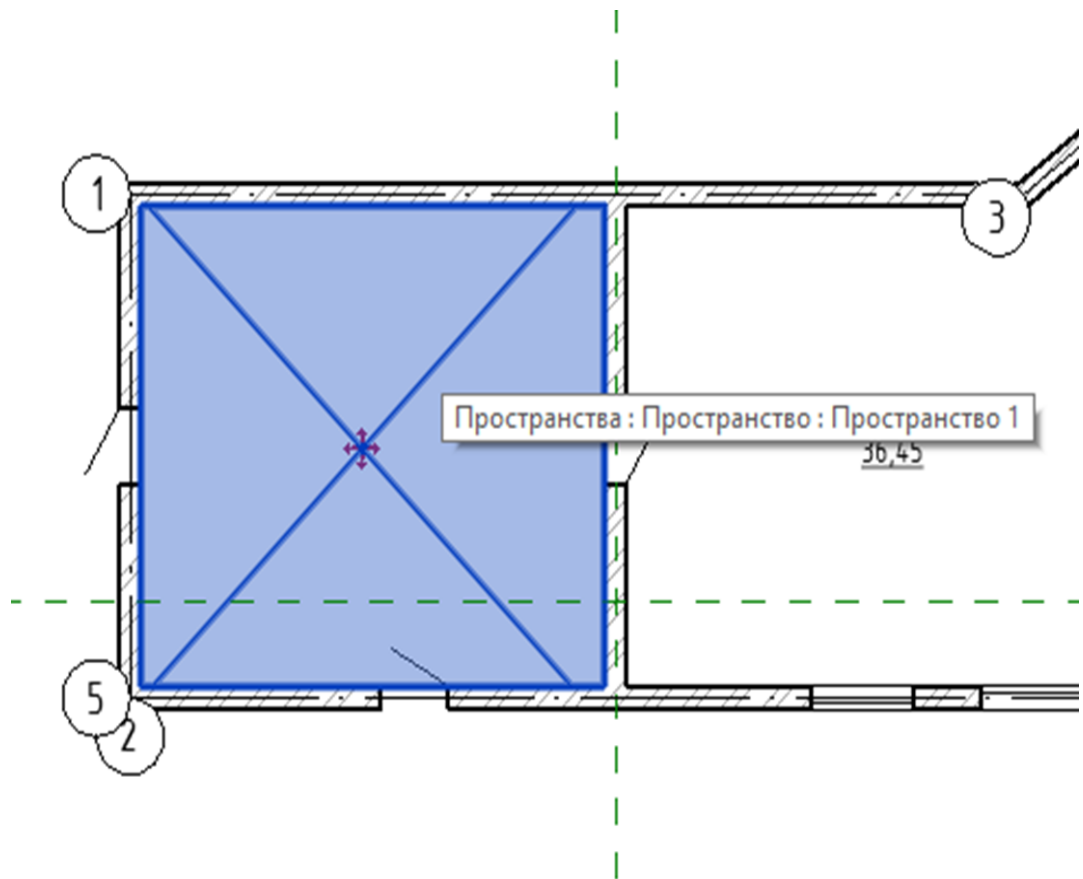


Рис. 2.2. Пространство при проектировании

На рисунке 2.3 изображена часть плана с использованием помещения при проектировании.



Рис. 2.3. Помещение при проектировании

После того, как помещения были найдены, требуется понять, что является ограждающими конструкциями.

По умолчанию для ограничения помещений используются следующие элементы:

1. Стены: витражи, стандартные, контекстные и стены по граням.
2. Крыши: стандартные, контекстные и крыши по граням.
3. Перекрытия: стандартные, контекстные и перекрытия по граням.
4. Потолки: стандартные, контекстные и потолки по граням.
5. Колонны: архитектурные и несущие железобетонные.
6. Витражные системы.
7. Линии, разделяющие помещения.
8. Основание здания.

Изменяя свойства элементов, можно задать большее или меньшее количество элементов, представляющих собой границы комнат. Например, туалетные перегородки можно отнести к неограничивающим элементам, поскольку они обычно не учитываются при расчетах помещений. Если элемент определен как не являющийся ограничивающим, он не используется в Revit при

расчете площади и объема помещения, а также всех смежных с ним помещений, для которых данный элемент является общим.

Также в качестве границ помещения можно использовать различные элементы в связанных файлах IFC.

«Линия-разделитель помещений» служит для добавления и корректировки границ помещений.

Разделители служат границами помещений. При помощи разделителей удобно обозначать функциональное назначение помещений, не разделяя их стенами, например обеденную зону в гостиной. Линии-разделители помещений видны на видах в плане и 3D-видах.

При создании помещения, ограниченного стенами, его площадь по умолчанию рассчитывается по внутренним поверхностям стен. Если в эти стены нужно добавить проемы, не изменяя при этом рассчитанные значения площади помещений, то надо провести линии-разделители помещений через эти проемы; при этом значения площади помещений останутся такими же, как и полученные при первом расчете.

Если в ограждающие конструкции попал разделитель, то площадь грани помещения, относящаяся к ней по расположению, не должна учитываться. Что отличается от всех других элементов ограждающих конструкций.

На рисунке 2.4 изображена часть проекта, где красными пунктирными линиями обозначены Линии-разделители помещений.

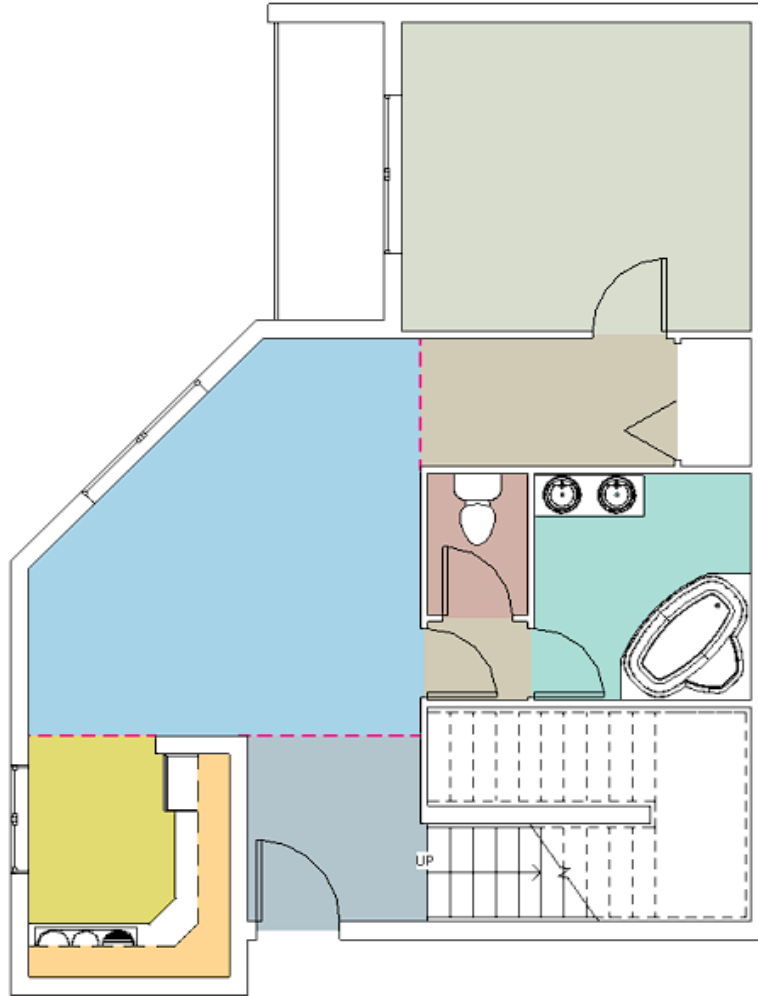


Рис. 2.4. Разделители помещений

Наиболее встречающееся семейство в ограждающих конструкциях помещений - стены. В это семейство могут входить, как и привычные газобетонные стены, так и отделка. В зависимости от того, что мы получили, применяется и разный подход к подсчету площадей. Все из-за того, что отделка под собой и подразумевает чистую площадь. Отделка идет вдоль основной стены, в рамках комнаты и не проходит там, где есть проемы. С самой отделки можно просто считать площадь, что и будет являться чистой площадью.

В том случае, если в список ограждающих конструкций попала стена в обычном понимании, то метод подсчета другой и сложнее. Нам нужно определить часть стены, которая входит в выбранное помещение и определить площадь этой части, а также исключить проемы.

На рисунке 2.5 изображено помещение с отделкой.

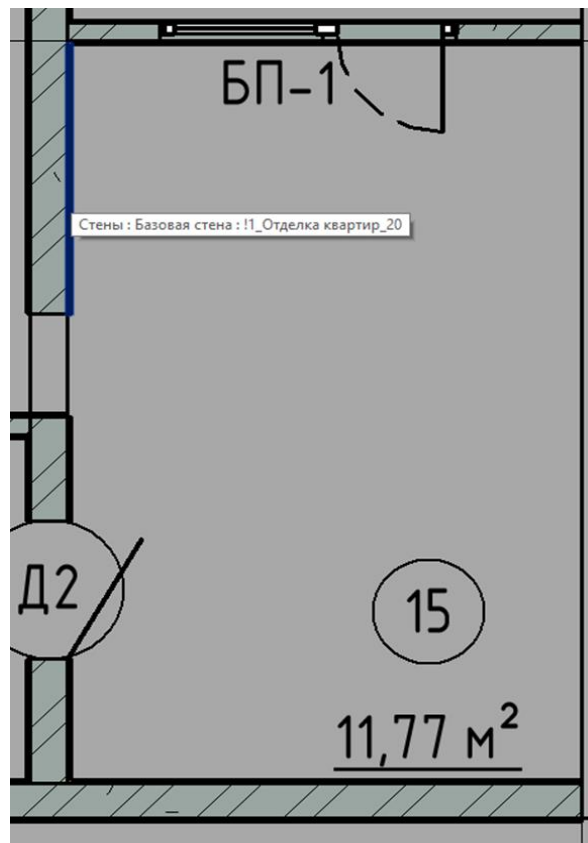


Рис. 2.5. Помещение с отделкой

На рисунке 2.6 изображены помещения со стеной в виде ограждающих конструкций.

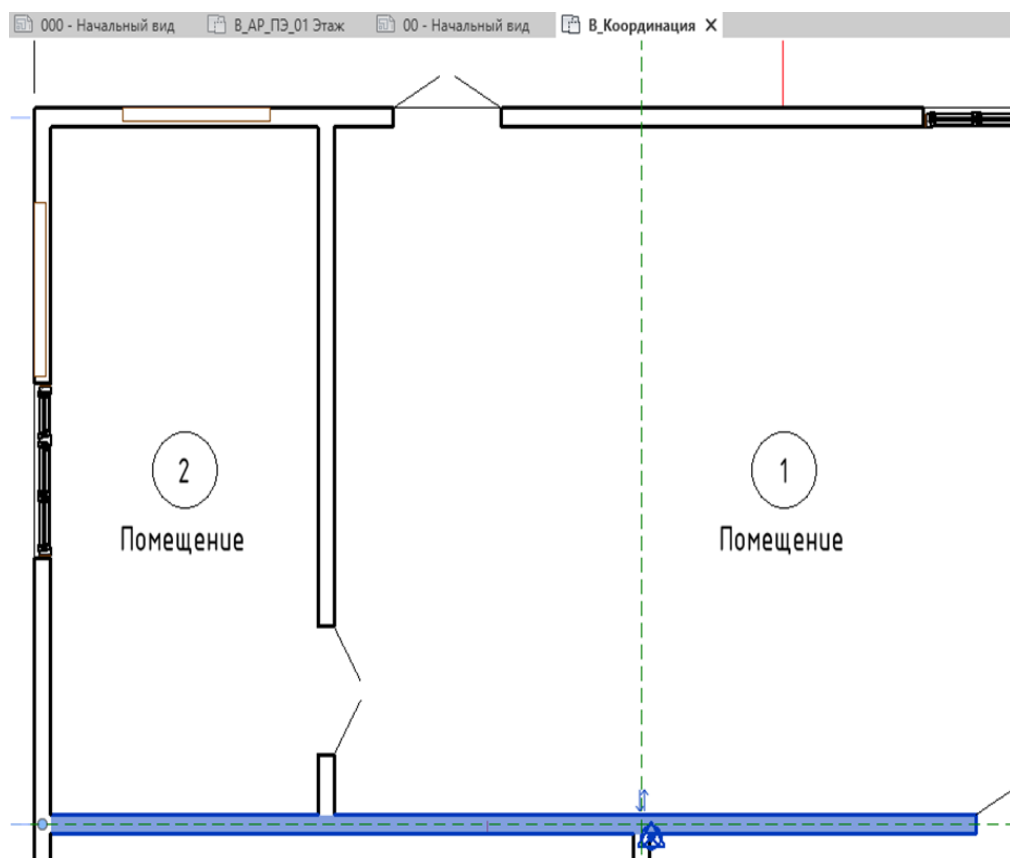


Рис. 2.6. Стена в виде ограждающей конструкции

Довольно часто в рабочих Revit-проектах существует наличие связанных моделей. Связанная модель представляет собой отдельный файл, в котором смоделирована часть проекта. Модели можно связывать для упрощения работы с отдельными деталями или для повышения производительности при работе с большим проектом.

При связывании модели, содержащей другие связанные модели, образуются вложенные связи. В главной модели можно отобразить или скрыть вложенные связанные модели. Отображение вложенных связей зависит от значения параметра «Тип связи», заданного для родительской модели.

Наложение: вложенные модели не загружаются в главную модель и поэтому не отображаются в проекте.

Прикрепление: вложенные модели загружаются в главную модель и отображаются в проекте.

На следующем рисунке 2.7 проект А представляет собой связанный проект в составе проекте В (т. е. проект В является родительской моделью для проекта А). Для проекта А параметру «Тип связи» в родительском проекте (проект В) присвоено значение «Наложение». При импорте проекта В в проект С проект А не отображается.

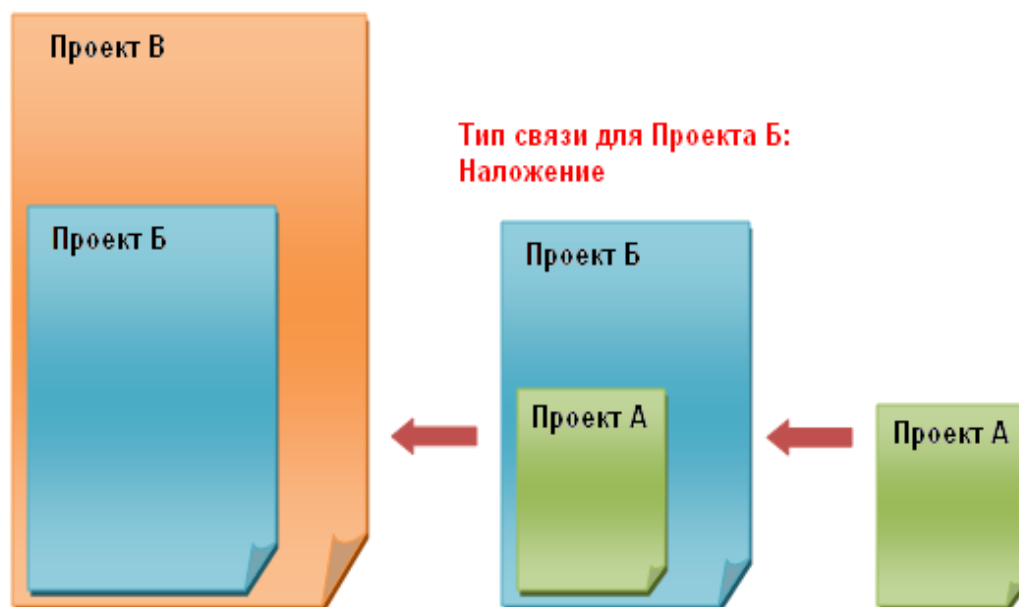


Рис. 2.7. Тип связи проекта «Наложение»

При изменении типа связи для проекта А на «Прикрепление» (в его родительской модели, проекте В) при импорте проекта В в проект С вложенная связь (проект А) будет отображаться (рисунок 2.8).

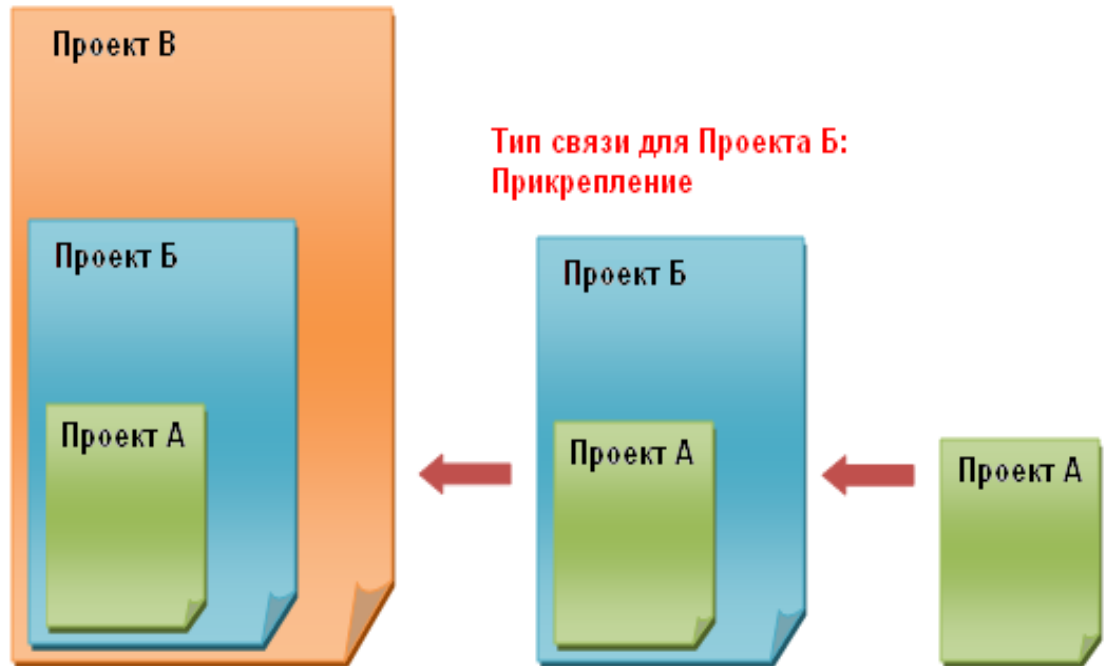


Рис. 2.8. Тип связи проекта «Прикрепление»

Если в главной модели может отображаться вложенная модель, можно задать режим отображения вложенной модели на виде: с фильтром из главной модели, с фильтром, примененным к виду в связанной или вложенной модели, или без фильтров. См. раздел Фильтры просмотра для связанных моделей.

Работая с таким типом проекта при применении связи «прикрепление» подразумевается умение обрабатывать и считывать информацию с элементов, к которым нет напрямую доступа из основного Revit-документа. Если помещение содержит в ограждающих конструкциях связанный элемент, то категория этого элемента будет `RevitLinkInstance`.

В этом случае нам нужно первоочередно получить родительский документ для этого элемент. После чего получить сам элемент с категорией, присущей в родительском файле. Зная, что это за элемент, извлекаем нужную информацию.

На рисунке 2.9 изображен пример стены, которая является связанным элементом.

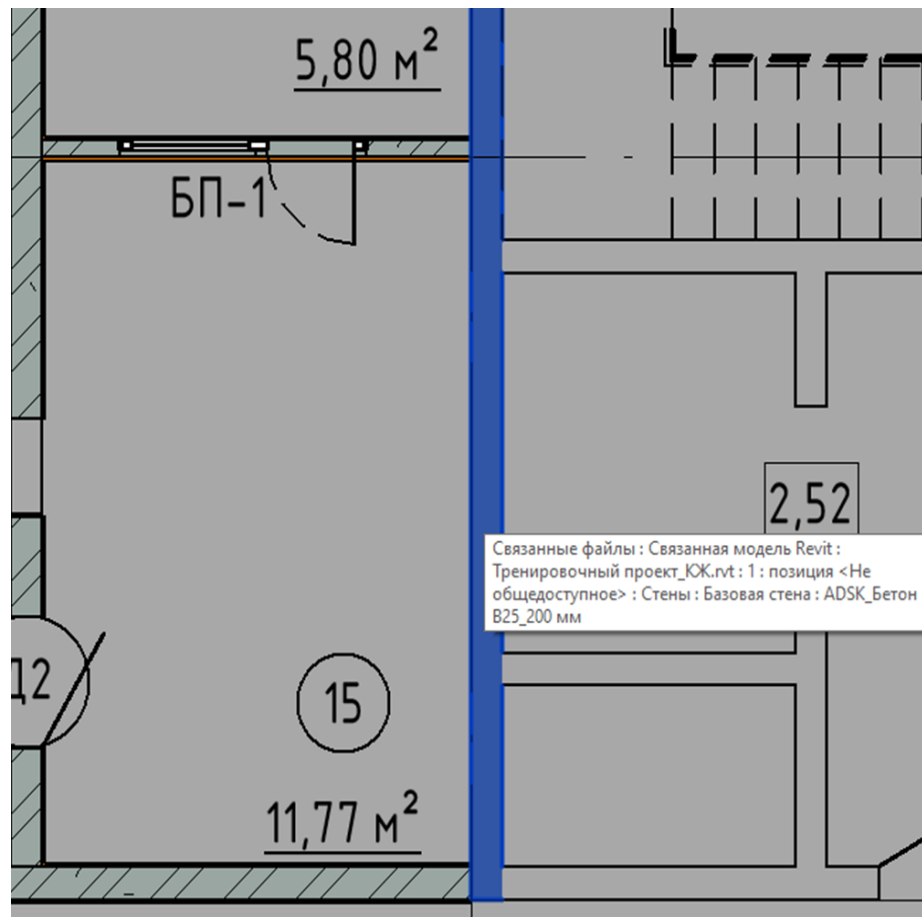


Рис. 2.9. Пример связанного элемента

Модель в Revit состоит из частей. Можно провести аналогию с реальным зданием. Здание состоит из фундаментов, стен, балок, перекрытий. Стены в здании в свою очередь могут отличаться по толщине, составу, типу и в конце концов в здании вполне может быть множество одинаковых стен. Из этих небольших частей, как одно перекрытие, один кирпич, один лестничный марш и состоит здание в целом. Точно такая же иерархия в модели Revit:

Модель – это здание в целом.

Категории семейств – это типы элементов, из которых может состоять здание: колонны, перекрытия, балки и т.д.

Семейства – отдельные типы колонн: например, прямоугольная колонна, круглая колонна, колонна из прокатного двутавра.

Типоразмеры в семействе – далее прямоугольная колонна может быть сечением 400x400, 500x500 и т.д.

Вхождение – в здание может быть несколько колонн 400х400. Одна колонна называется вхождением.

Проектируя модель используются разные семейства стен.

Могут быть использованы однослойные стены - из одного материала, также есть вариант использования стены, которая представляет собой набор слоев из разных материалов.

На рисунке 2.10 представлен пример «пирога» стены.



Рис. 2.10. Пример «пирога» стены

Для определения теплопроводности и получения проемов помещения требуется знать, что следует за элементом ограждающей конструкции. Так как с помещения можно определить лишь элемент, который ограничивает размеры помещения. Требуется проверить всю конструкцию стены, путем сканирующего

отрезка, а также проверить каждый полученный элемент на многослойность и разбить его на составляющие. В приведенном примере присутствуют 8 слоев.

ГЛАВА 3. ПРОЕКТИРОВАНИЕ И РАЗРАБОТКА ПЛАГИНА

3.1 АРХИТЕКТУРА ПО

В программном комплексе Autodesk Revit пользователь может применять любой язык программирования, использующий платформу .NET (C#, VB.NET, F# и т.д.), для создания плагинов. У каждого из этих языков есть свои достоинства, но в рамках данной работы плагин разрабатывался с использованием языка программирования C#. Был выбран данный язык так как наиболее популярный и более мощный среди использующих платформу .NET.

Использование технологии MVVM [14] позволяет организовать и структурировать код для написания поддерживаемых, тестируемых и расширяемых программ. Model представляет сущности системы, такие как «помещение», «ограждающая конструкция», «грань стены», «проем» и т. д. Во ViewModel происходит связывание наших моделей с представлением (View), отслеживаются изменения в данных произведенные пользователем, обрабатывается логика работы представления с помощью механизма команд. View содержит только пользовательский интерфейс, который отображает различные элементы, такие как кнопки, текстовые поля, вкладки, таблицы и т. д.

View

View отвечает за определение структуры, макета и внешнего вида того, что пользователь видит на экране. В идеале каждое представление определяется в XAML с ограниченным кодом программной части, который не содержит бизнес-логику. Однако в некоторых случаях код программной части может содержать логику пользовательского интерфейса, которая реализует визуальное поведение, которое трудно выразить в XAML, например анимации.

В приложении представление обычно является производным или ContentView производным классом Page. Однако представления также могут быть представлены шаблоном данных, который указывает элементы пользовательского интерфейса, которые будут использоваться для визуального представления объекта при его отображении. Шаблон данных в виде

представления не имеет кода программной части и предназначен для привязки к определенному типу модели представления.

ViewModel

ViewModel реализует свойства и команды, к которым представление может привязать данные, и уведомляет представление о любых изменениях состояния с помощью событий уведомления об изменениях изменений. Свойства и команды, предоставляемые моделью представления, определяют функциональные возможности, предоставляемые пользовательским интерфейсом, но представление определяет способ отображения этой функции.

Model

Классы моделей — это не визуальные классы, которые инкапсулируют данные приложения. Таким образом, модель можно рассматривать как представляющую модель предметной области приложения, которая обычно включает модель данных вместе с бизнес-логикой и логикой проверки. Примерами объектов модели являются объекты передачи данных (DTO), обычные старые объекты CLR (POC), а также созданные сущности и прокси-объекты.

Классы моделей обычно используются в сочетании со службами или репозиториями, которые инкапсулируют доступ к данным и кэширование [21].

Пример применения паттерна MVVM при поиске помещений приведен на рисунке 3.1. При нажатии на кнопку «Найти помещения» в пользовательском интерфейсе (View) вызывается команда «FindRoomsCmd» из класса «MainPageViewModel». При выполнении команды происходит заполнение модели «RoomElement» данными. Модель уведомляет о своем изменении свойство «RoomElements» (список помещений) во ViewModel, которая в свою очередь уведомляет о своем изменении View. Таким образом список помещений обновляется.

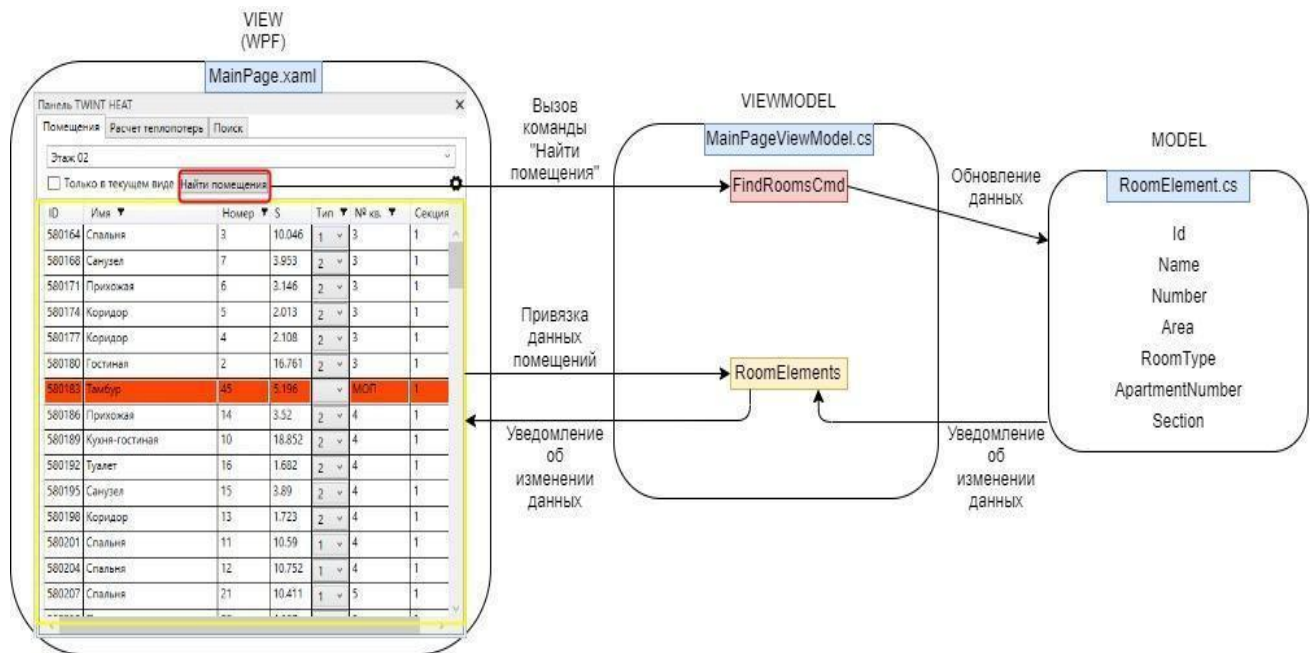


Рис. 3.1. Пример применения MVVM

Для программной реализации было разработано 29 классов. Общая схема классов проекта представлена на рисунке 3.2.

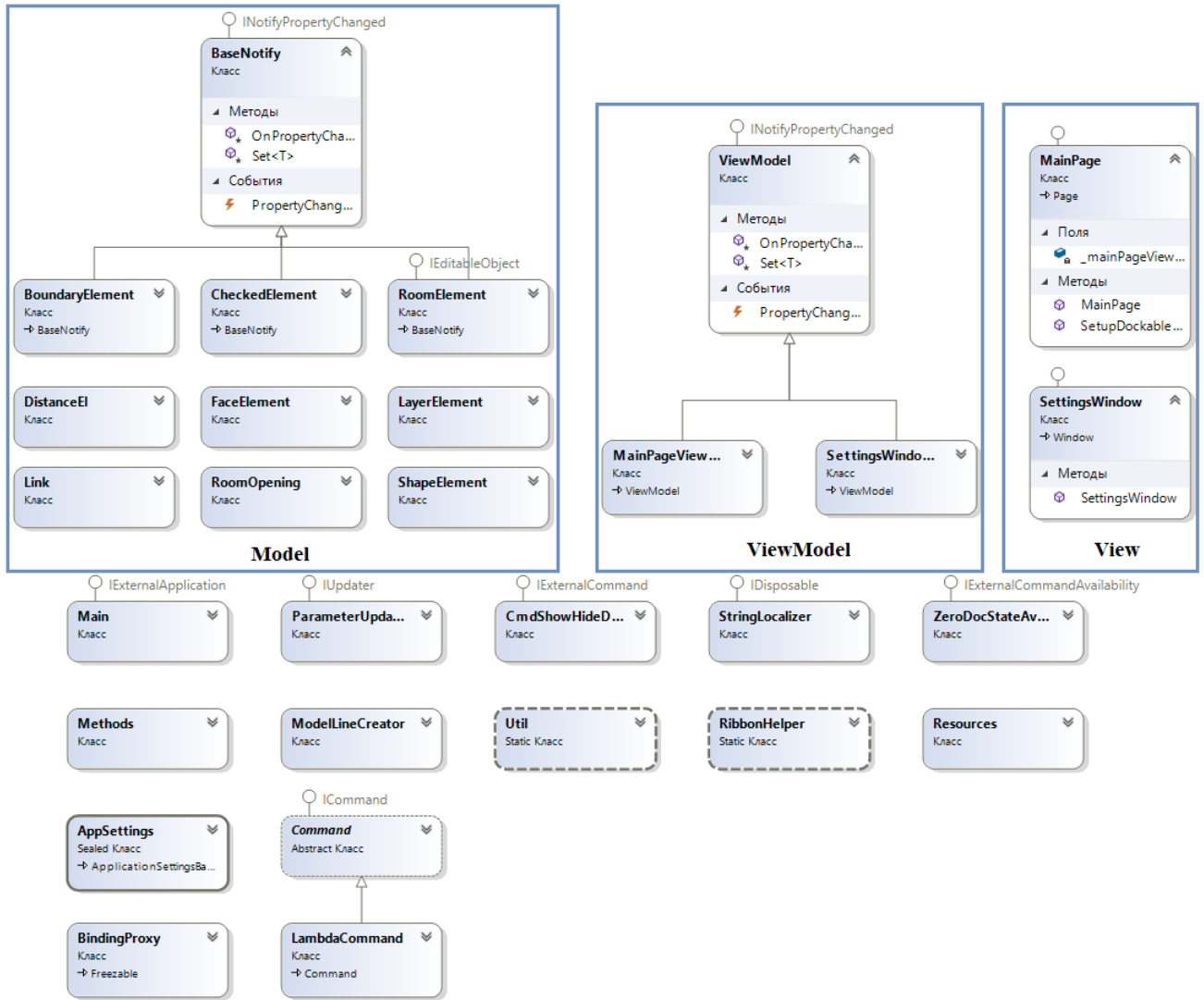


Рис. 3.2. Классы проекта с MVVM

Основные классы представлены на рисунке 3.3.

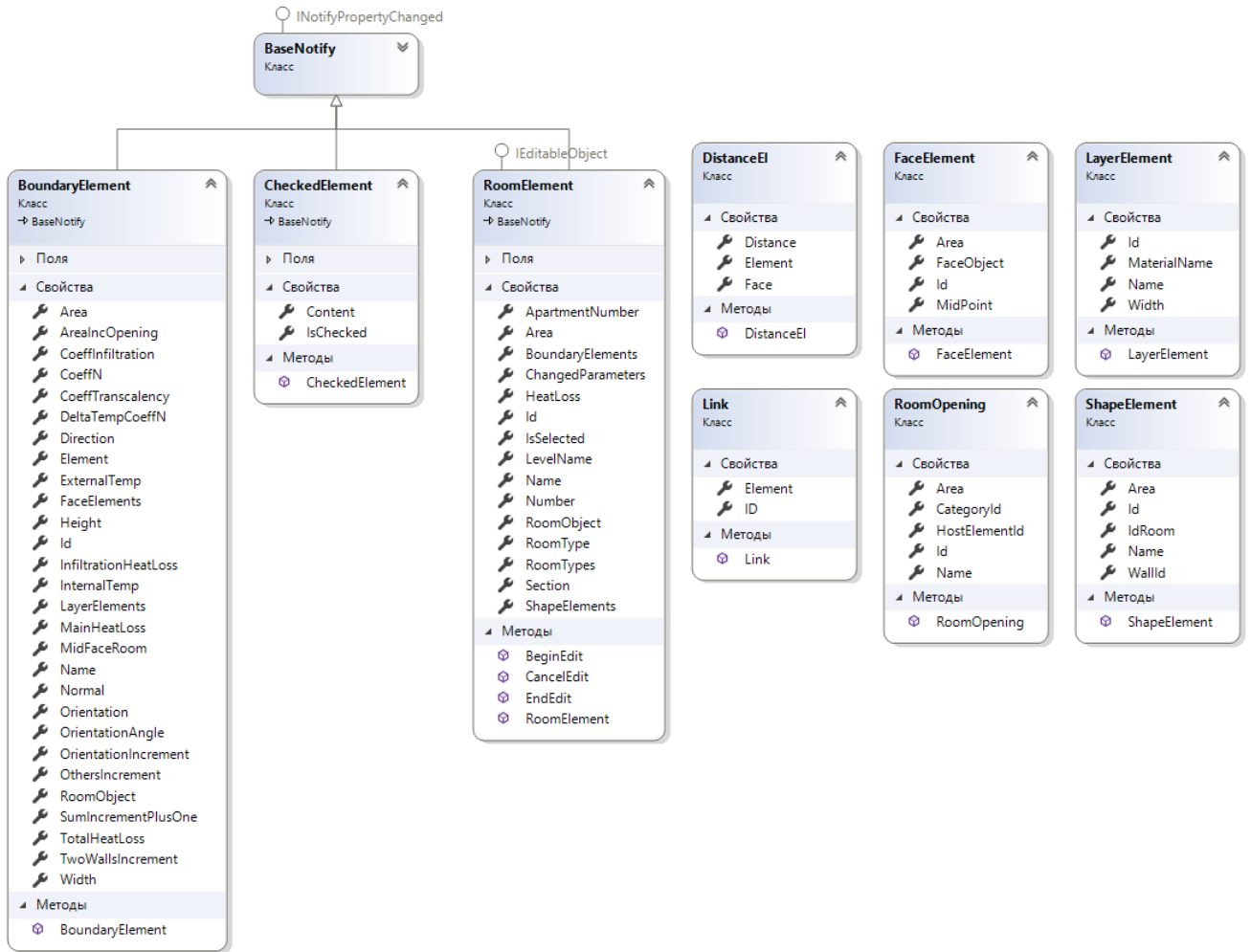


Рис. 3.3. Основные созданные классы

BoundaryElement - класс со свойствами, отражающими особенности ограждающих конструкций. Соответствующий объект заносится в `DataGrid` в панели `TwintHeat`, во вкладке «Расчет теплопотерь», каждое свойство рассчитывается с помощью разработанных методов.

RoomElement содержит информацию по выбранному помещению, служит для хранения значений необходимых нам свойств объекта.

RoomOpening содержит информацию по проемам в помещении.

ShapeElement - класс, который хранит информацию граням помещения, по которым был создан объект `Shape`.

WallLayerElement - класс, хранящий информацию по слоям стены.

Все операции, связанные с объектами, документами, классами, методами и свойствами Revit, используют Revit API. Именно этот инструмент позволяет

обработать все данные, которые необходимо задействовать для достижения задач. В таблице 3.1 приведены основные методы Revit API, используемые при разработке плагина, и их применение.

Таблица 3.1

Применение методов Revit API

Название метода	Применение
GetBoundarySegments	Для извлечения граничных сегментов помещения
GetGeometry	Получение Solid (твердое тело) из пространственного элемента
GetElement	Получение элемента по ID
GetRoomAtPoint	Получение помещения, содержащего указанную точку
GetSideFaces	Получение боковых граней стены
GetBoundingBox	Получение ограничивающую рамку грани
Project	Построение проекции указанной точки на кривую
FindInserts	Поиск ID проемов в стене
LookupParameter	Поиск параметра для элемента с заданным именем
Triangulate	Получение треугольной сетки грани
CreateBound	Создание экземпляра линии по двум точкам
GetParameter	Получение параметра по ID
Evaluate	Вычисление центральной точки грани
GetEndPoint	Получение точки в начале или конце кривой

3.2 ПОЛЬЗОВАТЕЛЬСКИЙ ИНТЕРФЕЙС

Для пользовательского интерфейса плагина использовалась технология WPF [6]. Была выбрана данная технология, так как WPF использует новую модель построения пользовательских приложений и дает возможность применения развитых графических эффектов с сохранением производительности в отличие от технологии Windows Forms.

При запуске Revit с установленным плагином в верхней части окна программы появится новая вкладка. При переходе на данную вкладку отображаются две кнопки. При нажатии на одну отображается панель расчета параметров площадей помещений для квартирографии. Вторая кнопка позволяет

открыть панель, используемую для расчета тепловых потерь помещений. При повторном нажатии этих кнопок закрываются соответствующие панели.

Квартирография

Интерфейс панели, с помощью которой производился расчет площадей помещений для квартирографии изображен на рисунке 3.4. Есть возможность выбора уровня для отображения помещений, расположенных на соответствующем уровне. Список помещений имеет такие параметры, как ID помещения, имя помещения, номер помещения, площадь, тип помещения, номер квартиры, секция, количество несовпадений в рассчитанных площадях и записанных в модели.

В данной панели предусмотрены следующие возможности:

- фильтрация данных помещения для удобства поиска необходимых помещений;
- отображение рассчитанных площадей;
- выбор точности округления площади, чтобы соответствующая площадь всех помещений округлялась до указанной точности;
- выбор способа округления площадей, то есть будет округляться каждое слагаемое из которых рассчитывается значение или итоговая сумма;
- загрузка марок помещений;
- запись рассчитанных площадей в модель;
- обнуление площадей, существующих в модели;

При записи в модели у помещений создаются параметры площадей. Если определенных параметров нет в проекте, то перед записью они автоматически создаются.

Панель TWINT AREA

AREA

Помещения

Этаж 01

Только в текущем виде

ID	Имя	Номер	S	Тип	№ кв.	Секция	Ошибки
577249	Спальня	11	10.321	1	2	1	4
577252	Кухня-гостиная	10	17.847	2	2	1	4
577255	Спальня	12	10.643	1	2	1	4

Расчётные площади для квартиры 2

ADSK_Площадь квартиры общая	71,1	0.1
ADSK_Площадь квартиры жилая	20,96	0.01
ADSK_Количество комнат	2	
ADSK_Коэффициент площади	1	
ADSK_Площадь с коэффициентом	10,321	0.001
ADSK_Площадь квартиры	59,75	0.01
ADSK_Площадь квартиры общая с коэфф.	63,1	0.1
ITKS_Площадь комбинированная	10,32	0.01
Площадь комбинированная	10,32	0.01

31.05.2022

Рис. 3.4. Расчет площадей помещений

Также в плагине есть возможность добавления поквартирных значений площадей в марках помещений, как того требует нормативная документация. При этом плагин позволяет выбирать тип добавляемой марки помещения. По нажатию на кнопку «Разместить метки» находится центр каждого из выбранных помещений, и в данном месте размещаются марки с площадями (Рисунок 3.5).

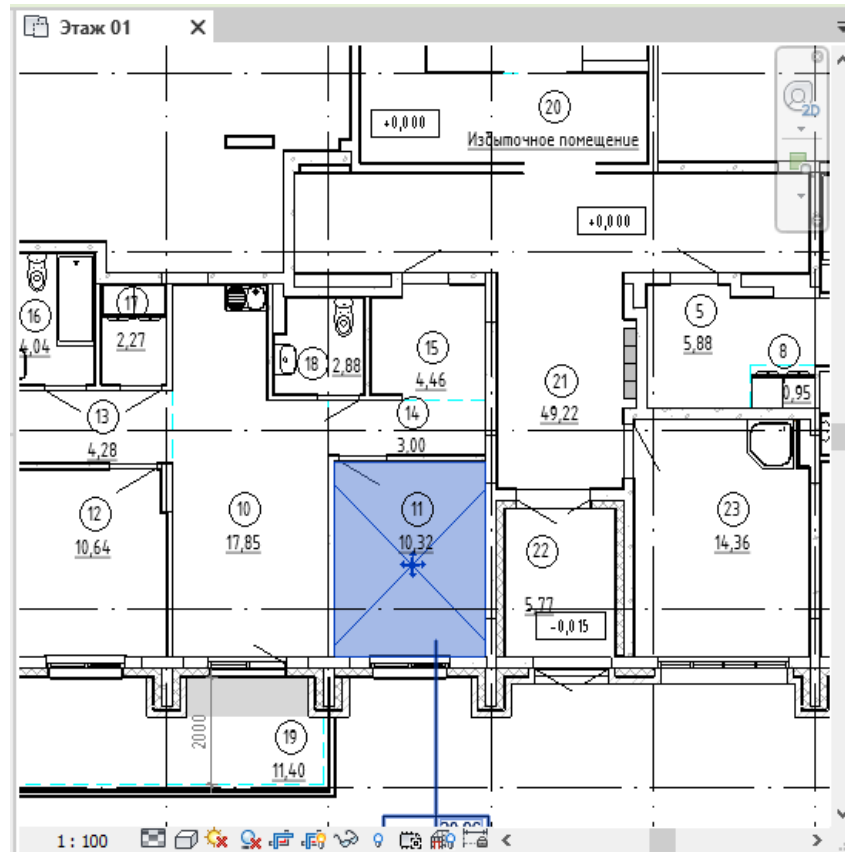


Рис. 3.5. Добавленные марки помещений

Теплопотери

На рисунках 3.6, 3.7 представлен интерфейс панели расчета теплопотерь. В данной панели во вкладке «Помещения» есть возможность выбора помещения, для которого необходимо произвести расчет тепловых потерь. Таблица с помещениями аналогична расположенной в панели по квартирографии.

На вкладке «Расчет теплопотерь» отображается наименование выбранного помещения, таблица ограждающих конструкций с рассчитанными параметрами. Часть кода выполнения расчета тепловых потерь представлен на рисунке в Приложении 1.

Панель TWINT HEAT

Помещения | Расчет теплопотерь | Поиск

Тамбур 32

Пересчет параметров

Ограждающие конструкции

ID	тэн	Наименование	Ориентация	Ширина	Высота	Площадь	Площадь-пр.	Козфф. теплопров.	n	Δt·n
3943669	22	05_МОП Утеплитель 150_штукатурка 10 белая	с	4.87	2.585	13.31	13.31	0.220962	1	57
3944169	22	04_штукатурка 10_Белый	з	2.96	2.585	7.71	2.44	0.220962	1	57
3944085	22	05_МОП Утеплитель 150_штукатурка 10 белая	ю	5.41	2.585	14.66	4.307	0.220962	1	57
3944018	22	05_МОП Утеплитель 150_штукатурка 10 белая	в	2.8	2.585	7.41	2.14	0.220962	1	57
3943673	22	04_Гипс 30+12 МОП	с	0.55	2.585	31.15	31.15	0.220962	1	57
3943672	22	04_Гипс 30+12 МОП	в	0.17	2.585	0.3	0.3	0.220962	1	57

Пирог

ID	Название	Толщина	Материал
529131	01_Блок силикатный 250	250	Стены 280
3943669	05_МОП Утеплитель 150_штукатурка 10 белая	10	ADSK_Штукатурка_Гладкая белая
3943669	05_МОП Утеплитель 150_штукатурка 10 белая	150	ЭковерЭкофасад Стандарт

Проемы

ID	Название	Элемент основа	Площадь проема
2780364	Витраж_1 этаж без имп. низ 95	3944169	5.27
4628509	Д7.2 2100x1125 Л Остекл.	3944085	10.353
2779827	Витраж_1 этаж без имп. низ 110	3944018	5.27

Рис. 3.6. Общий интерфейс панели расчета теплопотерь

Δt·n	Надб. на ориент.	Надб. на 2 стены	Прочие надб.	Σнадб.+1	Козфф.Инф	Основн. потери	Потери на инфильтр.	Общие потери	Направление	Ориент., град.
57	0.1	0	0	1.1	0.09	184.401	16.596	200.997	внутренняя	0
57	0.05	0	0	1.05	0.09	101.961	9.176	111.137	внешняя	270
57	0	0	1.98	1	0.09	184.64	16.618	201.258	внутренняя	180
57	0.05	0	0	1.05	0.09	97.994	8.819	106.813	внутренняя	90
57	0.1	0	0	1.1	0.09	431.562	38.841	470.403	внутренняя	0
57	0.05	0	0	1.05	0.09	3.967	0.357	4.324	внутренняя	90

Рис. 3.7. Интерфейс панели расчета теплопотерь

При выборе определенной ограждающей конструкции отображается таблица со слоями, из которых состоит данная конструкция. В таблице отображается ID элемента, толщина элемента и наименование материала. Далее есть таблица со всеми проемами помещения. Здесь по каждому проему приводится ID, наименование, ID элемента с ссылкой, к которому принадлежит данный проем, площадь проема в метрах квадратных.

На рисунке 3.8 можно увидеть слои ограждающей конструкции с Id 3943669 в плане Revit. Весь состав слоев для конкретной ограждающей конструкции представлен в таблице «Пирог» (рисунок 3.6).

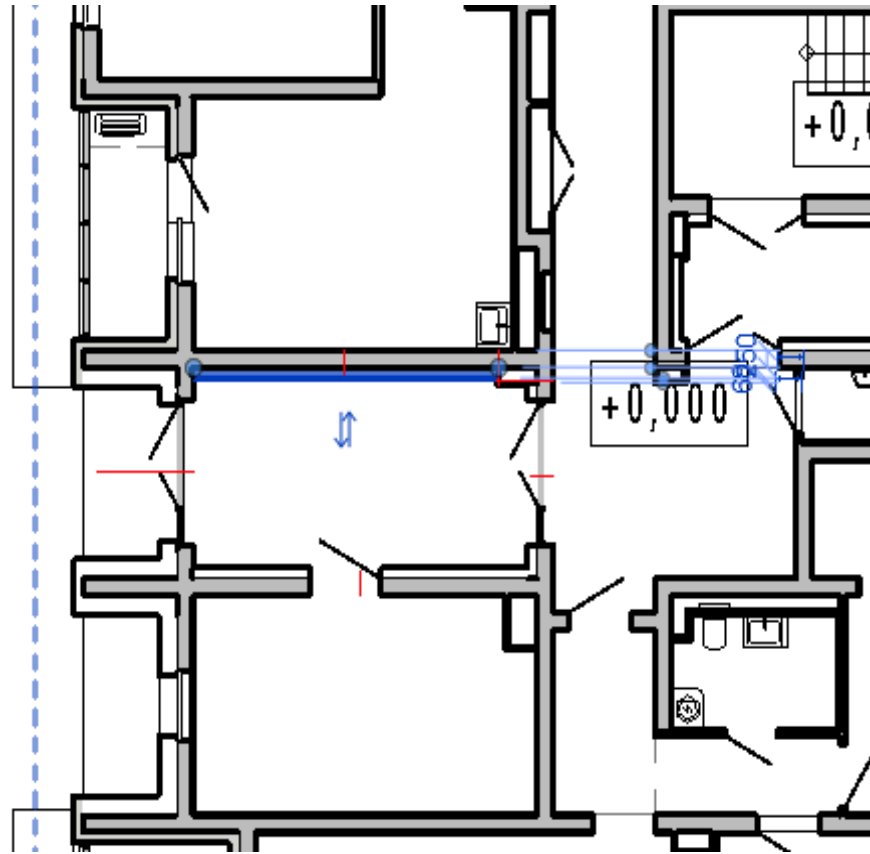


Рис. 3.8. Ограждающая конструкция

ЗАКЛЮЧЕНИЕ

В процессе решения поставленных задач были определены, систематизированы и учтены особенности моделирования элементов здания. Были разработаны следующие алгоритмы: определение пирога стен (метод сканирующих отрезков), визуализация формы помещений (метод коробки со стенками с нулевой толщиной) расчета чистых площадей (метод смежной грани), расположения стены относительно пространства.

Таким образом, была выполнена автоматизация работы процессов, тем самым уменьшено время и устранена возможность ошибок при подсчете теплопотерь и параметров квартирографии. Использование разработанного инструмента позволит инженерам проще производить расчеты для подбора нагревательных элементов и составления декларации недвижимости за счет быстрого доступа к актуальным и необходимым показателям в условиях постоянного изменения проектируемых моделей зданий.

Плагин используется на предприятии и прошел апробацию на реальных проектах строительства.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Autodesk [сайт]. URL: <https://www.autodesk.ru> (дата обращения: 21.06.2022).
2. Divin, N. V. BIM by using Revit API and Dynamo. A review. URL: https://alfabuild.spbstu.ru/userfiles/files/AlfaBuild/AlfaBuild_2020_14/1404-1.pdf (дата обращения: 21.06.2022).
3. GitLab [сайт]. URL: <https://gitlab.com> (дата обращения: 21.06.2022).
4. Marcus Kim, Lance Kirby. Mastering Autodesk Revit 2018. 2017. 1019 с.
5. Revit: BIM software for designers, builders, and doers [сайт]. URL: <https://www.autodesk.com/products/revit> (дата обращения: 21.06.2022).
6. The complete WPF tutorial [сайт]. URL: <https://www.wpf-tutorial.com> (дата обращения: 21.06.2022).
7. VisualStudio LiveShare [сайт]. URL: <https://visualstudio.microsoft.com/ru/services> (дата обращения: 21.06.2022).
8. Векторная геометрия для разработчиков Revit API [сайт]. URL: <https://habr.com/ru/post/336502> (дата обращения: 21.06.2022).
9. Джереми Таммик. Строительный кодер: блог [сайт]. URL: <https://thebuildingcoder.typepad.com/blog> (дата обращения: 21.06.2022).
10. Документация для Revit API [сайт]. URL: <https://www.revitapidocs.com> (дата обращения: 21.06.2022).
11. Заварзин, Б. Б. Методика расчета теплотерь для помещений / Б. Б. Заварзин, Р. В. Рюмин, А. Г. Чукарев. Текст: непосредственный // Молодой ученый. 2017. № 43 (177). С. 40-43. URL: <https://moluch.ru/archive/177/46071> (дата обращения: 21.06.2022).
12. Инженерные системы [сайт]. URL: <https://gotika1.ru/ventilyaciya/infiltratsiya> (дата обращения: 21.06.2022).
13. Колчин, В. Н. Применение BIM-технологий в строительстве и проектировании // Инновации и инвестиции. 2019. № 2.2019. С. 209-213.

- URL:<https://cyberleninka.ru/article/n/primenenie-bim-tehnologiy-v-stroitelstve-i-proektirovanii/viewer> (дата обращения: 21.06.2022).
14. Паттерн MVVM [сайт]. URL: https://professorweb.ru/my/WPF/documents/WPF/level36/36_5.php (дата обращения: 21.06.2022).
 15. Расчет теплотерь помещения как основа подбора отопительной системы [сайт]. URL: <https://gidrolux.ru/information/40/188> (дата обращения: 21.06.2022).
 16. СП 131.13330.2020 “СНиП 23-01-99* Строительная климатология”. Актуализированная редакция. – М.: Минстрой России, 2020.
 17. СП 50.13330.2012 “СНиП 23-02-2003. Тепловая защита зданий”. Актуализированная редакция. – М.: Минрегион России, 2012.
 18. Справочная информация о Revit [сайт]. URL: <https://knowledge.autodesk.com> (дата обращения: 21.06.2022).
 19. Талапов В. В. Технология BIM: суть и особенности внедрения информационного моделирования зданий. - М.: ДМК Пресс, 2015. - 410 с.: ил.
 20. Что значит плагин, определение понятия простыми словами. URL: <https://std17.com/blog/dlya-novichkov/chto-takoe-plagin-dlya-sajta-zachem-nuzhen-i-chto-on-oznachaet-opredelenie-ponyatiya-prostymi-slovami> (дата обращения: 21.06.2022).
 21. Шаблон model-view-ViewModel [сайт]. URL: <https://docs.microsoft.com/ru-ru/xamarin/xamarin-forms/enterprise-application-patterns/mvvm> (дата обращения: 21.06.2022)

Код расчета теплопотерь помещения

```

private void InitializeBoundariesTab() {
    try {
        if (SelectedRoomElement == null) {
            TaskDialog.Show("Расчет теплопотерь", "Выберете помещение");
            return;
        }
        //получение ограждающих конструкций
        SelectedRoomElement.BoundaryElements?.Clear();
        SelectedRoomElement.BoundaryElements =
Methods.GetRoomBoundaries(SelectedRoomElement.RoomObject);
        var floorsBoundaries = Methods.GetRoomFloors(SelectedRoomElement.RoomObject);

        foreach (var floorBD in floorsBoundaries) {
            SelectedRoomElement.BoundaryElements.Add(floorBD);
        }
        //соответствие площадей
        var listBoundArea = Methods.GetListBoundArea(SelectedRoomElement.RoomObject);
        if (SelectedRoomElement.BoundaryElements == null) {return;}

        foreach (var boundary in SelectedRoomElement.BoundaryElements) {
            FaceElement faceElement = listBoundArea.SingleOrDefault(num => num.Id ==
boundary.Id);
            //стены
            if (boundary.Element is Wall) {
                Wall wall = boundary.Element as Wall;
                Curve wallCurve = (wall.Location as LocationCurve).Curve;
                //присваивание огр.конструкции площади и центр.т
                if (faceElement != null) {
                    boundary.MidFaceRoom = faceElement.MidPoint;
                    boundary.Area = faceElement.Area;
                }
                else {
                    boundary.MidFaceRoom = (wallCurve.GetEndPoint(0) +
wallCurve.GetEndPoint(1)) / 2;
                }

                boundary.LayerElements = Methods.GetWallLayerElements(boundary);

                if (Math.Abs(boundary.Normal.Z) == 0 || Math.Abs(boundary.Normal.Z) == 1) {
                    boundary.CoeffN = 1.0;
                }
                else {
                    boundary.CoeffN = 1.1;
                }
            }
            //ориентация огр. конструкции

```

```

        boundary.OrientationAngle = Methods.GetWallOrientationAngle(wall,
boundary.Normal);
        double oa = boundary.OrientationAngle;
        if ((oa >= 0 && oa <= 45) || (oa >= 315 && oa <= 360)) {
            boundary.Orientation = "с";
            boundary.OrientationIncrement = 0.1;
        }
        if (oa > 45 && oa <= 135) {
            boundary.Orientation = "в";
            boundary.OrientationIncrement = 0.05;
        }
        if (oa > 135 && oa <= 225) {
            boundary.Orientation = "ю";
            boundary.OrientationIncrement = 0.0;
        }
        if (oa > 225 && oa < 315) {
            boundary.Orientation = "з";
            boundary.OrientationIncrement = 0.05;
        }
    }
    //перекрытия
    if (boundary.Element is Floor) {
        var ListStructureFloor=Methods.GetFloorLayerElements(boundary.Id);
        foreach (var Structure in ListStructureFloor) {
            boundary.LayerElements.Add(Structure);
        }
        if (listBoundArea.Where(num => num.Id == boundary.Id).Count() > 0) {
            boundary.Area = faceElement.Area;
        }
        boundary.AreaIncOpening = boundary.Area;
    }

    boundary.DeltaTempCoeffN = boundary.InternalTemp - boundary.ExternalTemp;
    boundary.SumIncrementPlusOne = boundary.OrientationIncrement +
boundary.OthersIncrement + 1;
    double mainHeatLoss =
        boundary.Area * boundary.CoeffTranscalency * boundary.DeltaTempCoeffN *
boundary.SumIncrementPlusOne;
    boundary.MainHeatLoss = Util.RoundUp(mainHeatLoss, 3);
    double infiltrationHeatLoss = boundary.MainHeatLoss * boundary.CoeffInfiltration;
    boundary.InfiltrationHeatLoss = Util.RoundUp(infiltrationHeatLoss, 3);
    double totalHeatLoss = boundary.MainHeatLoss + boundary.InfiltrationHeatLoss;
    boundary.TotalHeatLoss = Util.RoundUp(totalHeatLoss, 3);
}
//проемы
RoomOpenings = new ObservableCollection<RoomOpening>
(Methods.OpeningRoom(SelectedRoomElement.RoomObject));
foreach (var boundary in SelectedRoomElement.BoundaryElements) {
    //чистая S огр. констр. с учетом проемов
    FaceElement faceElementE = listBoundArea.SingleOrDefault(num => num.Id ==
boundary.Id);
    double AreaOPsum = 0.0;

```

```

try {
    if (RoomOpenings != null) {
        foreach (var openings in RoomOpenings) {
            if (openings.HostElementId == boundary.Id) {
                AreaOPsum += openings.Area;
            }
        }
        var doorOpenings = RoomOpenings.Where(be => be.HostElementId ==
boundary.Id &&
        (BuiltInCategory)be.CategoryId.IntegerValue == BuiltInCategory.OST_Doors);
        if (doorOpenings.Count() > 0) {
            boundary.OthersIncrement = 1.98;
        }
    }
    boundary.AreaIncOpening = boundary.Area - AreaOPsum;
}
catch {
    boundary.AreaIncOpening = faceElementE.Area;
}
}
//определение стены относительно внеш.пространства и расчет надбавки на две
стены
var orientations = SelectedRoomElement.BoundaryElements.Where(be => be.Direction
== "внешняя").Select(be => be.Orientation);
if (orientations.Contains("з") || orientations.Contains("в")) {
    foreach (var el in SelectedRoomElement.BoundaryElements) {
        if (el.Direction == "внешняя") {
            if (el.Orientation == "с") {
                el.TwoWallsIncrement = 0.1;
                el.SumIncrementPlusOne += el.TwoWallsIncrement;
            }
        }
    }
}
if (orientations.Contains("с") || orientations.Contains("ю")) {
    foreach (var el in SelectedRoomElement.BoundaryElements) {
        if (el.Direction == "внешняя") {
            if (el.Orientation == "з" || el.Orientation == "в") {
                el.TwoWallsIncrement = 0.05;
                el.SumIncrementPlusOne += el.TwoWallsIncrement;
            }
        }
    }
}
//ограждающие конструкции
BoundaryElements = new ObservableCollection<BoundaryElement>
(SelectedRoomElement.BoundaryElements);
}
catch (Exception ex) {
    TaskDialog.Show("Ошибка", ex.Message);
}
}

```