

Содержание

Введение.....	9
Глава 1. Обзор систем, имеющих средства акустического моделирования	11
1.1. Dassault Systemes Simulia Abaqus Unified FEA	11
1.2. MSC Actran	11
1.3. COMSOL Multiphysics.....	11
1.4. Siemens LMS Virtual.Lab	12
1.5. Siemens NX CAE	12
1.6. ESI VA One	13
1.7. ESI Virtual Performance Solution	13
1.8. Выводы	13
Глава 2. Описание акустической модели.....	15
2.1. Допущения модели.....	15
2.2. Структура модели	15
2.3. Формулы.....	16
2.3.1. Характеристики помещений и источников шума	16
2.3.2. Уровни шума от источников	17
2.3.3. Характеристики многослойных перегородок.....	18
2.4. Пример модели.....	19
2.5. Выводы	20
Глава 3. Проектирование системы акустического моделирования	21
3.1. Архитектура системы.....	21
3.2. Выбор программных средств.....	21
3.3. База данных о материалах.....	22
3.4. Модель.....	23
3.4.1. Расчётный модуль.....	26
3.4.1.1. Описание алгоритма акустического расчёта.....	26

	8
3.5. Редактор акустической модели.....	27
3.6. Выводы	28
Глава 4. Реализация системы акустического моделирования.....	29
4.1. База данных о материалах.....	29
4.1.1. Класс MaterialsDB.....	29
4.1.2. Класс MatDBDict	29
4.2. Класс Model и компоненты модели.....	29
4.2.1. Описание формата файла хранения модели	29
4.3. Расчётный модуль	31
4.4. Редактор акустической модели.....	31
4.4.1. Видовой порт	31
4.4.2. Формы пользовательского интерфейса	32
4.5. Выводы	35
4.6. Пример работы программы.....	35
4.6.1. Описание тестовой модели	35
4.6.2. Результат расчётов.....	39
Заключение	35
Литература	41
Приложение А (обязательное) Описания основных классов	42
1. Модуль MaterialsDatabase.....	42
2. Модуль ModelLib	42
3. Модуль Viewport.....	45

ВВЕДЕНИЕ

Борьба с шумовым загрязнением на железнодорожном транспорте – важная и сложная задача по ряду экономических и социальных причин.

Низкий уровень шума – показатель высокого качества транспортного средства. Соответствие международным стандартам на акустические характеристики транспортных средств повышает их конкурентоспособность на мировом рынке [1]. На решение задачи эффективной шумозащиты в транспортных средствах производителями тратятся значительные средства. Например, в самолётах стоимость шумозащиты достигает 25% стоимости, в автомобилях – до 10%. Вследствие высокой стоимости средств шумозащиты и повышения эргономичности транспортного средства уровень шумозащиты оказывает заметное влияние на стоимость транспортного средства: каждый децибел снижения шума обеспечивает около 1% повышения стоимости продаваемого изделия [2].

Повышенный шумовой фон оказывает комплексное губительное действие на организм человека. Он воздействует на нервную (центральную и вегетативную), слуховую и сердечно-сосудистую системы человека, в число оказываемых эффектов входят: раздражительность, головная боль, головокружение, снижение памяти, повышенная утомляемость, понижение аппетита, агрессивность, шумовая болезнь (нейросенсорная тугоухость) [2,3]. Согласно [4], повышенный уровень шумового фона является вредным производственным фактором. Среди профессиональных заболеваний вибрационная болезнь и неврит слухового нерва занимают – 1-3-е места по частоте среди профессиональных, в России – более 35% от общего числа профессиональных [2].

Такое действие повышенного шумового фона на машиниста подвижного состава может привести к ошибке управления, и, как следствие, к аварии. Поэтому к уровню шума в кабинах машинистов предъявляются жёсткие требования: для рабочих мест в кабинах машинистов тепловозов, электровозов, поездов метрополитена, дизель-поездов и автомотрис – не более 80 дБА, для рабочих мест в кабинах машинистов скоростных и пригородных электропоездов – не более 75 дБА [5]. По данным из [6], применяемые на сегодняшний день средства тепло-, шумо-, виброзащиты не соответствуют требованиям существующих санитарных норм и правил.

Таким образом, задача проектирования кабин машинистов, удовлетворяющих санитарным нормам и эргономическим требованиям, является актуальной технической, социальной и экономической проблемой.

Основная доля снижения шума в кабине машиниста и пассажирском салоне вагона подвижного состава приходится на мероприятия, принимаемые на стадии проектирования за счёт выбора оптимальных мер шумозащиты. Инженерам необходимо эффективно применить средства звукоизоляции и звукопоглощения, соблюсти требования стандартов, уложиться в бюджет.

Цель данной работы – создание программного комплекса, автоматизирующего акустический расчёт конструкции железнодорожного вагона для оценки эффективности применённых средств защиты от воздушного шума на стадии проектирования.

Для акустического расчёта используется алгоритм, основанный на методе акустического расчёта, предложенного в [6]. Данный метод ориентирован на выполнение расчётов вручную, на основании определённого набора путей распространения шума, что ограничивает его применение для акустического расчёта конструкций вагонов, отличных от типовых.

В результате данной работы разработан и реализован алгоритм акустического расчёта, позволяющий рассчитывать уровень шума и не требующий задания путей распространения шума. В отличие от средств инженерного анализа, использующих метод конечного элемента для расчёта ожидаемых уровней шума, полученный алгоритм требует значительно меньшие вычислительные и временные ресурсы. Независимость алгоритма от конструкции позволяет использовать его для акустического расчёта различных типов железнодорожных вагонов: электровозов, локомотивов, вагонов метро, трамваев и т.д.

ГЛАВА 1. ОБЗОР СИСТЕМ, ИМЕЮЩИХ СРЕДСТВА АКУСТИЧЕСКОГО МОДЕЛИРОВАНИЯ

1.1. Dassault Systemes Simulia Abaqus Unified FEA

Программное обеспечение для математическое моделирования. Представляет собой пакет унифицированных решений для проведения инженерного анализа изделия с точки зрения явлений различных областей физики – гидродинамики, пьезоэлектрики, механики, акустики, теории теплопроводности. Использует метод конечных элементов, позволяет создавать, редактировать модели изделий, а также импортировать их из других САПР. Область применения – разные области машиностроения, в том числе автомобилестроение (симуляция статических нагрузок, краш-тестов).

1.2. MSC Actran

Программное обеспечение для инженерного анализа акустических свойств механических систем и их частей. Позволяет симулировать акустические, виброакустические, аэроакустические явления. Интегрирован с MSC Nastran в области виброакустической симуляции. Для расчётов используется метод конечных элементов с бесконечными элементами на краю расчётной области для моделирования открытых пространств. Приложение позволяет импортировать поверхности из различных форматов файлов, назначать им материалы из базы данных, хранимой в отдельном файле как часть модели, и производить расчёт. Применяется в разных областях машиностроения: автомобилестроении, самолётостроении, кораблестроении, электронике, в гражданской и оборонной промышленности.

1.3. COMSOL Multiphysics

COMSOL Multiphysics – платформа для компьютерного моделирования различных физических задач. За счёт модульной структуры предоставляет возможности для моделирования явлений различной природы: электрические, механические (в том числе виброакустический анализ как раздел структурного анализа), гидродинамические, химические. Также есть модули для интеграции с широким спектром САПР: AutoCAD, SOLIDWORKS, Inventor, CATIA V5 и др.

Имеет встроенные средства для создания 1-, 2- и 3-мерной модели изделия, возможна синхронизация с другими САПР. Набор атрибутов материала зависит от того, в каком модуле производится работа. Область применения – широкий спектр областей машиностроения.

1.4. Siemens LMS Virtual.Lab

LMS Virtual.Lab представляет собой программный комплекс для 3 мерного конечно-элементного анализа и проектирования многомодульных объектов, моделирования и оптимизации механических систем при анализе структурной целостности, шумов и вибраций, долговечности и динамических расчётах.

Модуль NVH (noise, vibration, harness) предназначен для инженерного анализа поведения изделия при воздействии на него шума и вибрации с возможностью объединения результатов конечно-элементного моделирования и экспериментальных данных с тестовых моделей. Позволяет анализировать реакцию изделия на внешний источник шума, строить модель изделия, основываясь на данных из различных источников (экспериментальных и расчётных), находить и анализировать отдельные пути передачи шума и вибрации, исследовать характеристики отдельных подсистем изделия. Область применения модуля NVH –транспортное машиностроение. Результаты работы модуля NVH можно использовать в качестве дополнения к модулю Acoustics для более глубокого анализа.

Модуль Acoustics предназначен для симуляции, исследования и улучшения акустических (звуковых, шумовых и вибрационных) характеристик широкого спектра механической систем. В нем используются методы конечного элемента, граничного элемента, трассировки лучей. Области применения – проектирование акустического оборудования, архитектура, транспортное машиностроение, энергетика (турбины, ветряки)

1.5. Siemens NX CAE

Siemens NX CAE – набор средств инженерного анализа от компании Siemens PLM Software. Вся работа с CAE приложениями осуществляется через интерфейс препостпроцессора NX Advanced FEM, к которому подключаются требуемые расчётные модули. К основным расчётным модулям NX относят NX Nastran, NX Thermal и NX Flow. NX Nastran – конечно-элементный решатель, позволяющий производить полный набор инженерных расчётов, включая расчёт напряжённо-деформированного состояния, собственных частот и форм колебаний, анализ устойчивости, решение задач теплопередачи, исследование установившихся и неуставившихся процессов, нелинейных статических процессов, нелинейных динамических переходных процессов, анализ частотных характеристик, отклика на динамические и случайные воздействия. В частности, NX Nastran позволяет производить NVH-анализ изделия.

Также возможности для NVH анализа изделия предоставляет модуль Multiphysics, однако расчёт производится тем же NX Nastran.

Имеет встроенную базу данных материалов.

1.6. ESI VA One

VA One — программный продукт, разработанный ESI Group для моделирования виброакустических явлений. Для расчётов используются методы конечных элементов (FE), граничных элементов (BEM), и статистического анализа энергии (SEA), также есть возможность комбинировать эти методы в единой модели. Область применения — транспортное машиностроение. Симуляция охватывает весь спектр слышимых частот, есть функция оптимизации изделия. Поддерживаются пористые и эластичные материалы.

1.7. ESI Virtual Performance Solution

ESI Virtual Performance Solution — решение для виртуального прототипирования, позволяющее исследовать процессы, относящиеся к разным областям физики и протекающие в изделии. Основная особенность — использование единой модели для проведения всех симуляций.

Для изучения акустических характеристик в пакете есть 3 модуля:

- Linear perturbation response of mechanical systems;
- NVH analysis: Eigen mode, harmonic and transient response;
- Interior acoustics analysis.

Эти модули позволяют исследовать реакцию механических систем на внешние возмущения, анализировать собственные колебательные частоты изделия, анализировать акустические характеристики интерьера транспортного средства. Модули поддерживают работу с пористыми и эластичными материалами и многослойными композитами. Применяется в транспортном машиностроении.

1.8. Выводы

На данный момент существует большое количество развитых программных средств для изучения свойств проектируемого изделия путём математического моделирования различных физических процессов, протекающих в этом изделии, в том числе акустических. Большая часть этих средств инженерного анализа предоставляет средства для имитации не одного, а нескольких физических явлений, относящихся к различным областям физики, таким например, как химические, оптические, механические, электромагнитные, акустические явления. Поэтому можно выделить такую тенденцию в развитии средств инженерного анализа, как переход к использованию единой, «ключевой» модели изделия при проведении всех требуемых пользователю симуляций, вместо практики создания отдельной

модели для каждого симулируемого физического процесса после каждого изменения в проекте. Основной математический метод, используемый при математическое моделировании – метод конечного элемента, однако существуют реализации и других методов, которые можно использовать в частных случаях.

ГЛАВА 2. ОПИСАНИЕ АКУСТИЧЕСКОЙ МОДЕЛИ

2.1. Допущения модели

При проектировании алгоритма акустического расчёта предполагаются следующие допущения:

- Все источники шума, имеющиеся в модели, точечные ненаправленные, однако стены помещений, через которые проникает шум, рассматриваются как плоские направленные источники шума;
- Рассматривается только воздушный шум;
- Все акустические перегородки – прямоугольной формы;
- Отношение интенсивности поглощённого звука к интенсивности падающего при прохождении через акустическую перегородку пропорционально толщине этой перегородки и не зависит от частоты звука;
- Штучные поглотители шума не учитываются.

2.2. Структура модели

Акустическая модель вагона включает в себя следующие сущности:

- Помещения;
- Перегородки помещений (Акустические перегородки);
- Источники шума (например, аэродинамический шум или шум от силовой установки);
- Линейные источники шума (например, рельсы);
- Расчётные точки – точки, в которых производится расчёт уровня звукового поля;
- Отражающие поверхности (например, земля или стены тоннеля).

Помещения формируются акустическими перегородками и могут содержать источники шума и расчётные точки.

Акустические перегородки имеют следующие свойства:

- Название;
- Расположение (координаты);
- Ограждаемые помещения;
- Строение - набор слоёв материалов, имеющих различные шумозащитные характеристики, различную толщину и образующих перегородку.

Источники шума имеют следующие свойства:

- Название;
- Расположение (координаты);
- Помещение, в котором расположен источник;
- Мощность источника по октавным полосам.

Линейные источники шума имеют следующие свойства:

- Название;
- Координаты концов источника;
- Мощностью по октавным частотам.

Расчётные точки имеют следующие свойства:

- Название;
- Расположение (координаты);
- Помещение, в котором расположена расчётная точка.

Отражающие поверхности характеризуются:

- Названием;
- Расположением;
- Отражающим материалом.

2.3. Формулы

Формулы, используемые для вычислений, взяты из [2,6].

2.3.1. Характеристики помещений и источников шума

A - эквивалентная площадь звукопоглощения помещения, м^2 , определяется по формуле:

$$A = \sum_{i=1}^n \alpha_i S_i + \sum_{j=1}^m A_j n_j \quad (2.1)$$

где α_i - коэффициент звукопоглощения i -й поверхности; S_i - площадь i -й поверхности, м^2 ; A_j - эквивалентная площадь звукопоглощения j -го штучного поглотителя, м^2 ; n_j - количество j -х штучных поглотителей, шт (в данной работе штучные поглотители не учитываются).

$\bar{\alpha}$ - средний коэффициент звукопоглощения, определяется по формуле

$$\bar{\alpha} = \frac{A}{S_{\text{общ}}} \quad (2.2)$$

где $S_{\text{общ}}$ – суммарная площадь перегородок помещения.

k - коэффициент, учитывающий нарушение диффузности звукового поля в помещении, вычисляется по формуле:

$$k = \frac{1}{1 - \bar{\alpha}} \quad (2.3)$$

B - постоянная помещения, определяется по формуле:

$$B = \frac{A}{1 - \bar{\alpha}} \quad (2.4)$$

Φ - фактор направленности источника шума, находится по формуле:

$$\Phi = \frac{4\pi}{\Omega} \quad (2.5)$$

где Ω - пространственный угол излучения источника в радианах.

2.3.2. Уровни шума от источников

Уровень шума (в децибелах), создаваемого в помещении одиночным источником шума, определяется по формуле:

$$L = L_w + 10 \cdot \lg \left(\frac{\chi \Phi}{\Omega r^2} + \frac{4}{kB} \right) \quad (2.6)$$

где L_w – октавный уровень звуковой мощности источника шума, χ - коэффициент, учитывающий влияние ближнего акустического поля в тех случаях, когда расстояние r меньше удвоенного максимального габарита источника ($r/l_{\text{max}} < 2$). Находится графически в зависимости от отношения расстояния от акустического центра источника к его максимальному габариту.

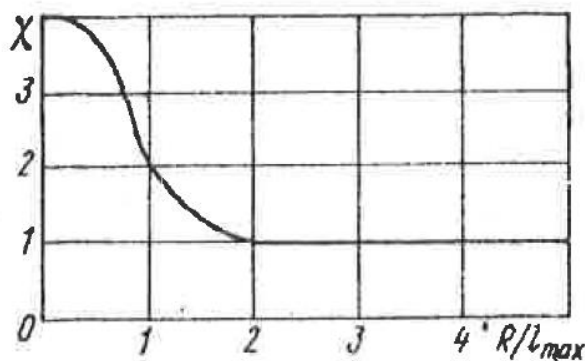


Рис. 2.1. График определения коэффициента χ как функции отношения r/l_{max}

Суммарный уровень шума, создаваемый несколькими источниками в помещении:

$$L = 10 \cdot \lg \left(\sum_{i=1}^m \frac{10^{0.1L_{wi}} \chi_i \Phi_i}{\Omega_i r_i^2} + \frac{4}{kB} \sum_{i=1}^n 10^{0.1L_{wi}} \right) \quad (2.7)$$

где m - число источников шума, ближайших к расчётной точке (находящихся на расстоянии $r_i \leq 5r_{\min}$, где r_{\min} - расстояние от расчётной точки до акустического центра ближайшего источника шума); n - общее число источников шума в помещении.

Уровень шума, создаваемого точечным внешним источником в помещении при прохождении шума через акустическую перегородку:

$$L = L_w - R + 10 \lg(S) - 10 \lg(B_{\Pi}) - 10 \lg(k) \quad (2.8)$$

где R - звукоизоляция перегородки, дБ, S - площадь перегородки, B_{Π} - постоянная помещения, в которое проникает шум, L_w - уровень шума, действующий на перегородку

Если рассматривать акустическую перегородку, через которую в помещение проникает шум, как источник шума, уровень мощности такого источника рассчитывается по формуле:

$$L = L_{\text{пад}} + 10 \lg(p) + 10 \lg(S) \quad (2.9)$$

где $L_{\text{пад}}$ - уровень шума, действующего на перегородку (в точке, совпадающей с акустическим центром перегородки), p - коэффициент звукопроводности.

Уровень шума от источника, отражённый от отражающей поверхности:

$$L = L_w - 20 \lg\left(\frac{r}{2}\right) + 10 \lg(1 - \alpha_f) \quad (2.10)$$

где r - расстояние между расчётной точкой и зеркальным (относительно поверхности) отражением источника шума, α_f - частотно-зависимый коэффициент звукопоглощения материала отражающей поверхности.

2.3.3. Характеристики многослойных перегородок

Рассматривается задача нахождения значений коэффициентов отражения (отношения интенсивности отражённого звука к интенсивности падающего), прохождения (отношения интенсивности прошедшего через перегородку звука к интенсивности падающего) и поглощения (отношения интенсивности поглощённого звука к интенсивности падающего) акустической перегородки, состоящей из 2 и более слоёв материалов.

Рассмотрим падение звука на перегородку из 2 слоёв, для каждого из которых известны коэффициенты отражения и поглощения:

В модели имеется 8 внешних источника шума: 2 рельса и 6 тяговых электрических двигателей.

В тамбуре на высоте человеческого роста и в кабине машинистов на местах машинистов расположены расчётные точки.

Более подробное описание модели см. в главе 4.5.

2.5. Выводы

Построенная модель достаточно универсальна для того, чтобы представлять вагоны подвижного железнодорожного состава различного назначения и конфигурации. В последующих главах предложен и реализован алгоритм акустического расчёта, основанный на данной модели.

ГЛАВА 3. ПРОЕКТИРОВАНИЕ СИСТЕМЫ АКУСТИЧЕСКОГО МОДЕЛИРОВАНИЯ

3.1. Архитектура системы

Для построения вагонов подвижного состава используется множество материалов с различными акустическими характеристиками, причём набор материалов может со временем пополняться, а их характеристики - уточняться. Кроме того, информация о материалах может потребоваться одновременно нескольким пользователям, поэтому рационально хранить информацию об этих материалах в базе данных. Исходя из этого, решено реализовать 3-х уровневую архитектуру.

Уровню базы данных соответствует модуль MaterialsDatabase, осуществляющий функции хранения информации о материалах, используемых для построения железнодорожного вагона.

Уровню логики соответствует модуль ModelLib, реализующий акустическую модель.

Уровню интерфейса соответствует модуль ModelEditor, содержащий формы пользовательского интерфейса.

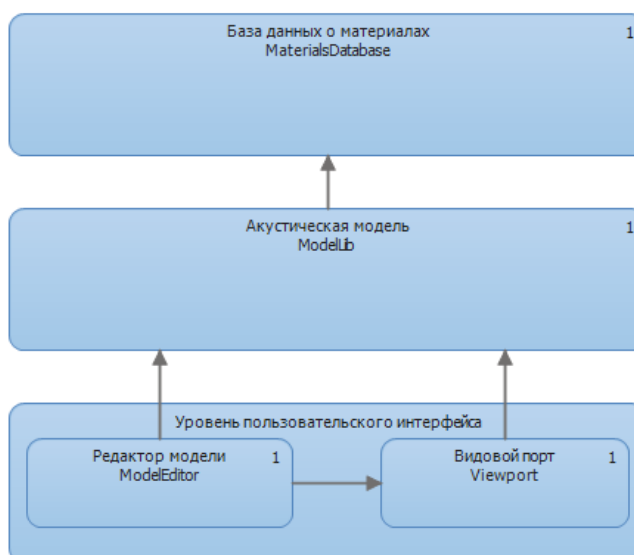


Рис. 3.1. Иллюстрация архитектуры программного комплекса

3.2. Выбор программных средств

В качестве среды программирования решено использовать среду Visual Studio, язык программирования - C#. Преимущества выбора:

- Встроенные в .Net средства работы с БД;
- Встроенные средства Visual studio для создания собственных элементов управления;
- встроенные развитые средства генерации кода на основе диаграмм классов, автоматической валидации архитектуры.

В качестве СУБД решено использовать MySQL по причине свободного распространения.

Для работы с графикой решено использовать графическую библиотеку – OpenGL, библиотеку для использования OpenGL в C# - OpenTK.

Для загрузки полигональной модели вагона решено использовать библиотеку Open Asset Library (AssImp) из-за – большого списка поддерживаемых форматов файлов.

3.3. База данных о материалах

База данных материалов должна содержать информацию о материалах, из которых строится вагон, их шумопоглощающих и шумоотражающих свойствах.

Каждый материал имеет следующие свойства:

- Название (строка);
- Коэффициент отражения – отношение интенсивности отражённого поверхностью материала звука к интенсивности падающего (вещественное число);
- Параметр поглощения – отношение коэффициента поглощения к толщине перегородки из этого материала (вещественное число);
- Цвет, которым в редакторе модели отображаются перегородки, имеющие на лицевой стороне слой из этого материала (кодируется целым числом).

За получение данных и внесение изменений в базу данных отвечает специальный класс, осуществляющий также функции создания соединения, авторизации и получения данных с сервера базы данных о материалах - MaterialsDB.

Для упрощения доступа к информации о материалах по их идентификатору решено создать класс, инкапсулирующий функции выборки значений из базы данных через класс MaterialsDB – MatDBDict.

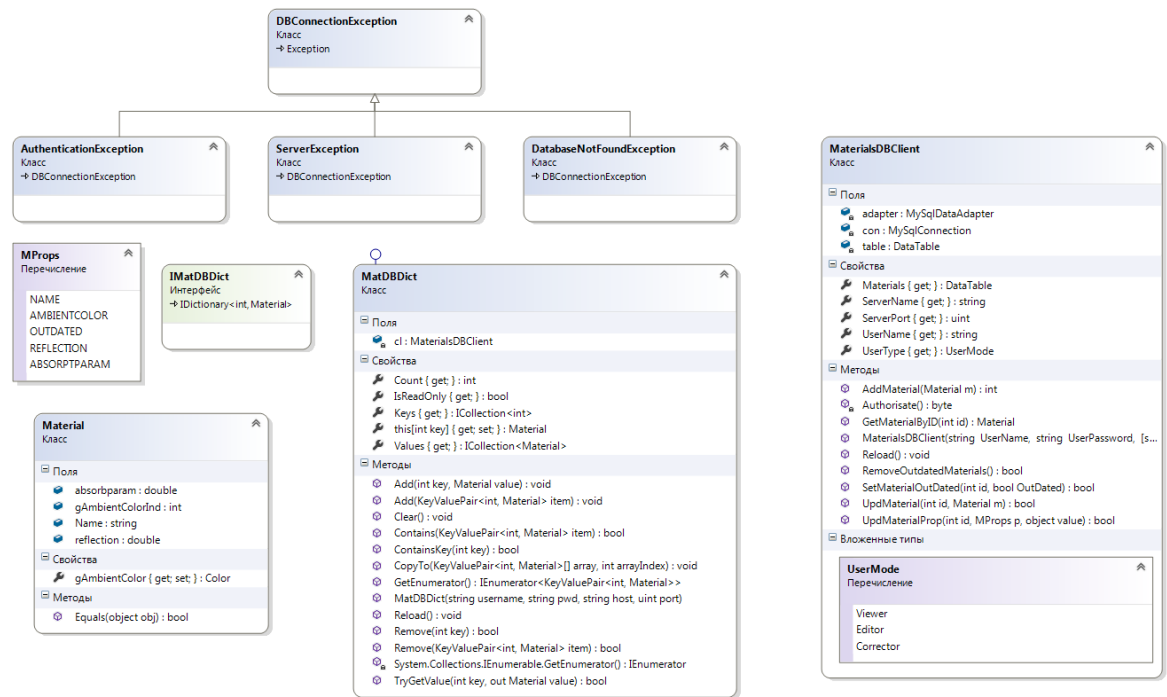


Рис. 3.2 Диаграмма классов модуля MaterialsDatabase

3.4. Модель

Модель представляется классом Model. Свойства:

- коллекция источников шума sources (тип элементов – NoiseSource);
- коллекция перегородок Borders (тип элементов – Border);
- коллекция помещений Rooms (тип элементов – room);
- коллекция расчётных точек calcpnts (тип элементов – calcpoint);
- коллекция отражающих поверхностей Reflectors (тип элементов – Reflector);
- MeshModel – полигональная модель, по образу которой пользователем строится модель, используемая для акустического расчёта (тип – PolygonalModel).

Методы:

- LoadFromFile(string filename) – загрузка модели из файла, статичный, возвращаемое значение – загруженная модель;
- SaveToFile(string filename) – сохранение модели в файл, возвращаемое значение – логическое: истина при успешном сохранении, ложь – в противном случае;
- calc() – выполнение акустического расчёта модели.

Класс Border представляет акустическую перегородку. Свойства:

- Name – название (текстовая строка);
- Rectangle – прямоугольник, моделирующий положение перегородки в пространстве (объект класса spacerect);

- Room1 – помещение с лицевой стороны перегородки (room);
- Room2 – помещение с обратной стороны перегородки (room);
- Layers – список слоёв, из которых построена перегородка (коллекция элементов типа border_layer).

Класс border_layer представляет 1 слой акустической перегородки. Свойства:

- materialId – идентификатор материала, из которого состоит слой (целое число);
- thickness – толщина слоя в метрах (вещественное число).

Класс Reflector представляет отражающую поверхность. Свойства:

- Name – название (текстовая строка);
- Rectangle – прямоугольник, определяющий положение перегородки в пространстве (объект класса spacerect);
- Matid – идентификатор материала, из которого сделана отражающая поверхность: 0 – асфальт, бетон, 1 – песок, трава.

Класс spacerect представляет прямоугольник, расположенный в пространстве. Свойства:

- lt, rt, lb, rb – левая верхняя, правая верхняя, левая нижняя и правая нижняя вершины соответственно (тип – Vector3d из библиотеки OpenTK);
- facenorm – вектор единичной длины, перпендикулярный прямоугольнику и направленный в направлении лицевой стороны прямоугольника (Vector3d);
- Square – площадь прямоугольника (вещественное число).

Класс calcpoint представляет расчётную точку. Свойства:

- Name – название (текстовая строка);
- Pos – координаты в пространстве (Vector3d);
- Room – помещение, в котором расположена расчётная точка (room).

Класс NoiseSource представляет точечный источник шума. Свойства:

- Name – название (текстовая строка);
- Pos – координаты в пространстве (Vector3d);
- Room – помещение, в котором расположен источник шума (room);
- Power – уровни мощности источника по октавным полосам (объект класса noise).

Класс LinearNoiseSrc представляет линейный источник шума. Свойства:

- Name – название (текстовая строка);
- A, B – координаты начальной и конечной точек (Vector3d);

- Room – помещение, в котором расположен источник шума (room);
- Power – уровни мощности источника по октавным полосам (объект класса noise).

Класс room представляет помещение. Свойства:

- Название (текстовая строка).

Для упрощения функций сохранения и загрузки моделей из файлов, а также организации программного интерфейса между моделью и элементами графического интерфейса (видовыми портами) необходим класс Document, связывающий модель с видовыми портами и файлом, в котором сохраняется модель. Свойства:

- Model – текущая модель (Model);
- Saved – флаг, отражающий состояние модели (сохранена или нет, логический тип);
- FileName – название файла, в котором сохранена модель (текстовая строка).

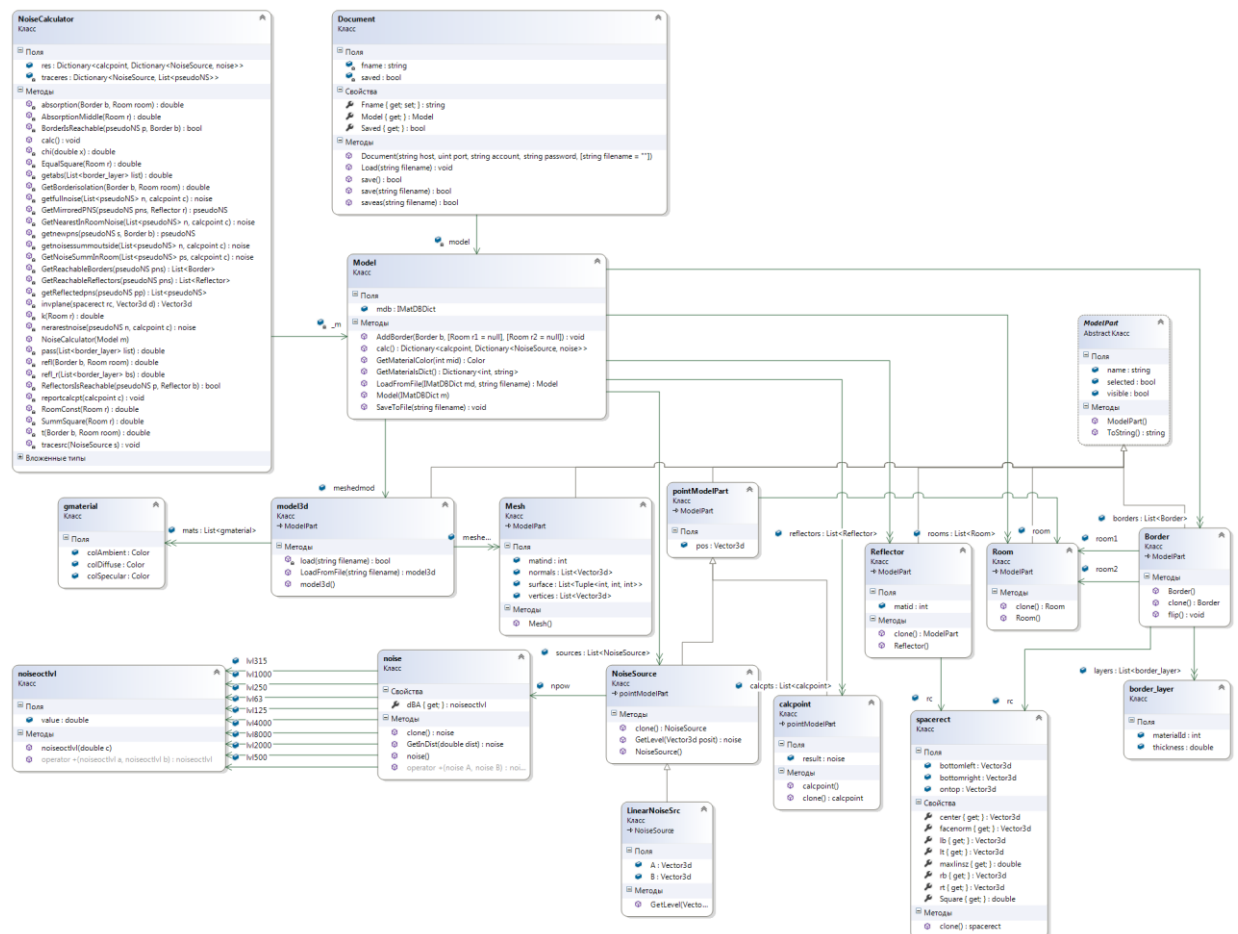


Рис. 3.3. Диаграмма классов модуля ModelLib

3.4.1. Расчётный модуль Решено выделить расчётный модуль в отдельный от модели класс NoiseCalc. Это позволяет упростить реализацию класса модели и классов компонент модели, отделив свойства и методы, значимые для расчётов, от исходных данных модели. Класс NoiseCalc имеет:

- Конструктор, принимающий модель как параметр;
- Метод calc(), выполняющий акустический расчёт модели;
- Свойство res - таблицу результатов акустического расчёта.

3.4.1.1. Описание алгоритма акустического расчёта

Для проведения акустического расчёта используется алгоритм, основанный на понятии вторичного источника. Вторичный источник – абстракция источника шума. Вторичным источником может быть:

- акустическая перегородка, через которую передаётся шум от другого источника и рассматриваемая как источник шума. В этом случае такой источник шума будет иметь направление, ортогональное акустической перегородке и сонаправленное шумовому воздействию на перегородку, расположение (координаты акустического центра) в центре акустической перегородки, максимальный габарит как у перегородки и уровень звуковой мощности, определяемый по формуле (2.9);
- обычный источник шума.

Предполагается, что шумовой фон в помещении получается в результате сложения действия всех вторичных источников, расположенных в этом помещении, по формуле (2.7).

Описание алгоритма:

1. Для каждого источника шума N:
 - а. Создать эквивалентный вторичный источник;
 - б. Добавить его в очередь вторичных источников Q;
 - в. Очистить список D;
 - г. Пока Q не пуста:
 1. Выбрать из очереди верхний вторичный источник P;
 2. Найти для P все достижимые перегородки (с учётом отражения звука от шумоотражающих поверхностей, в этом случае уровень падающего на них шума корректируется согласно формуле (2.10));
 3. Добавить P в список итоговых вторичных источников D;

4. Для каждой достижимой P перегородки B :

- а. создать новый (дочерний) вторичный источник S , уровень мощности которого определяется по формуле (2.9);
- б. добавить B в список перегородок, через которые запрещено проходить вторичным источникам, порождённым от S ;
- в. добавить новый вторичный источник S в Q .
- д. D содержит все вторичные источники, полученные из N . D сохраняется в контейнере, хранящем пары источников шума и списков порождённых им вторичным источникам.

2. Для каждой расчётной точки P уровень шума от источника N складывается из уровней шума всех вторичных источников, находящихся в одном с S помещении и порождённых от N , по формуле (2.7).

3.5. Редактор акустической модели

Для визуального контроля над моделью необходимо иметь элемент управления, способный графически отображать вносимые в модель изменения, а также класс, осуществляющий синхронное обновление всех таких элементов управления, привязанных к данной модели, при внесении в неё изменений.

Для графического отображения модели используется элемент управления `ViewportControl`. Для связи экземпляров класса `ViewportControl` с моделью (через класс `Document`) используется класс `ViewportList`. Он имеет:

- Свойство `ownDoc` – документ, модель из которого отображается в элементах `ViewportControl`;
- Метод `Redraw()`, вызывающий обновление изображения во всех связанных элементах `ViewportControl`.

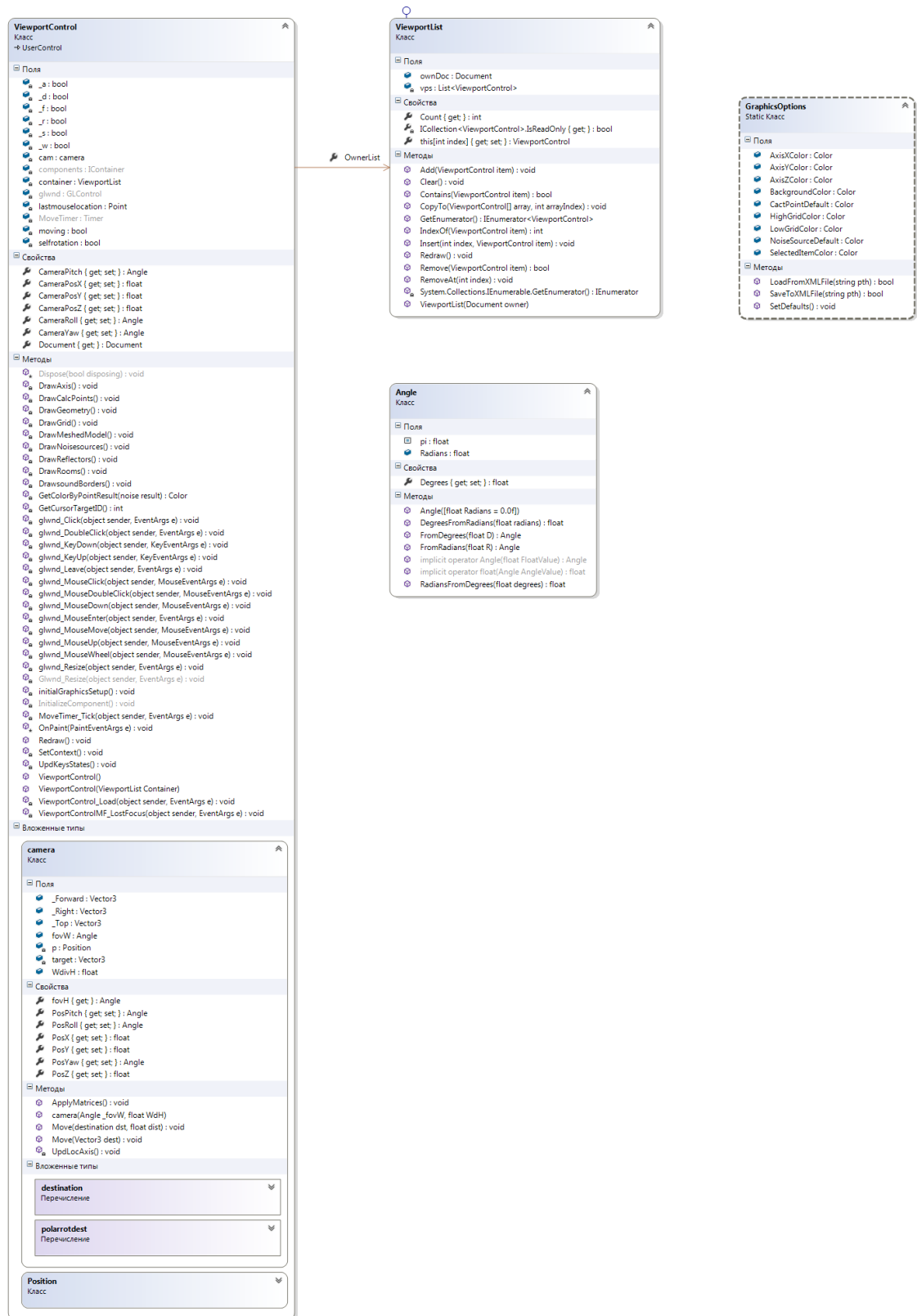


Рис. 3.4. Диаграмма классов видowego порта

3.6. Выводы

Получен перечень классов, которые необходимо реализовать, требований к их функциональности, и логических связей между ними. В следующей главе рассказывается об особенностях реализации компонент системы.

ГЛАВА 4. РЕАЛИЗАЦИЯ СИСТЕМЫ АКУСТИЧЕСКОГО МОДЕЛИРОВАНИЯ

4.1. База данных о материалах

4.1.1. Класс MaterialsDB

Конструктор принимает параметры авторизации. В случае провала авторизации возникает исключение, соответствующее причине провала.

Данные загружаются в поле DataTable Materials класса MaterialsDB.

Обновление и добавление записей о материалах производится специальными методами UpdMaterial(int id, Mprops prop, object value) и AddMaterial(Material m) соответственно.

Для удаления записи о материале необходимо

1. Отметить материал как устаревший методом SetMaterialOutDated(Int32 id, bool OutDated);
2. Удалить устаревшие материалы, вызвав метод RemoveOutdatedMaterials().

4.1.2. Класс MatDBDict

Содержит поле класса MaterialsDB. Для доступа к материалам по их идентификатору класс реализует интерфейс IDictionary<Int32,Material>.

Класс Model и классы частей модели реализованы в соответствии с описаниями.

4.2. Класс Model и компоненты модели

Классы, представляющие модель и её компоненты, реализованы в соответствии с описанием.

4.2.1. Описание формата файла хранения модели

Для хранения модели в файле используется следующий формат (двоичный):

1. Блок помещений:
 - а. Количество помещений;
 - б. Представления UTF-8 названий помещений (здесь и далее – структура из: длины массива байт, представляющей название в кодировке UTF-8 и самого массива байт)
2. Блок акустических перегородок:
 - а. Количество перегородок;
 - б. Соответствующее количество раз:

1. UTF-8-представление названия перегородки;
 2. Индекс помещения 1; -1, если не назначено;
 3. Индекс помещения 2; -1, если не назначено;
 4. Координаты точки на верхней грани;
 5. Координаты нижней левой точки;
 6. Координаты нижней правой точки;
 7. Количество слоёв;
 8. Соответствующее число раз:
 - а. Идентификатор материала;
 - б. Толщина слоя.
3. Блок отражающих поверхностей:
- а. Количество отражающих поверхностей
 - б. Соответствующее количество раз:
 1. UTF-8-представление названия перегородки;
 2. Материал отражающей поверхности(0-асфальт/бетон, 1-трава/песок);
 3. Координаты точки на верхней грани;
 4. Координаты нижней левой точки;
 5. Координаты нижней правой точки.
4. блок источников шума:
- а. Количество источников шума;
 - б. Соответствующее количество раз:
 1. UTF-8-представление названия источника шума;
 2. Индекс помещения, -1 если не назначено;
 3. Уровни мощности источника:
 - а. Уровень мощности в октавной полосе со среднегеометрической частотой $f=31,5$ Гц;
 - б. $f=63$ Гц;
 - в. $f=125$ Гц;
 - г. $f=250$ Гц;
 - д. $f=500$ Гц;
 - е. $f=1000$ Гц;
 - ж. $f=2000$ Гц;
 - з. $f=4000$ Гц;
 - и. $f=8000$ Гц.
 4. true, если источник шума линейный, false – если точечный;

5. Если источник шума линейный:
 - а. Координаты начальной точки;
 - б. Координаты конечной точки.
 6. В противном случае:
 - а. Координаты источника шума.
5. Блок расчётных точек:
- а. Количество расчётных точек;
 - б. Соответствующее количество раз:
 1. UTF-8-представление названия расчётной точки;
 2. Индекс помещения, -1 если не назначено;
 3. Координаты расчётной точки.

4.3. Расчётный модуль

Конструктор класса `NoiseCalculator` принимает в качестве параметра объект класса `Model`, расчёт производится методом `calc()`, результаты расчётов представлены свойством `res`.

Алгоритм акустического расчёта реализован согласно описанию.

4.4. Редактор акустической модели

4.4.1. Видовой порт

Классы `ViewportControl` и `ViewportList` реализованы в соответствии описанием.

Класс `ViewportList` содержит список связанных с ним видовых портов, что позволяет производить их синхронное обновление при внесении в модель изменений.

Класс `ViewportControl` использует графическую библиотеку `OpenGL` для вывода графического изображения модели на экран.

4.4.2. Формы пользовательского интерфейса

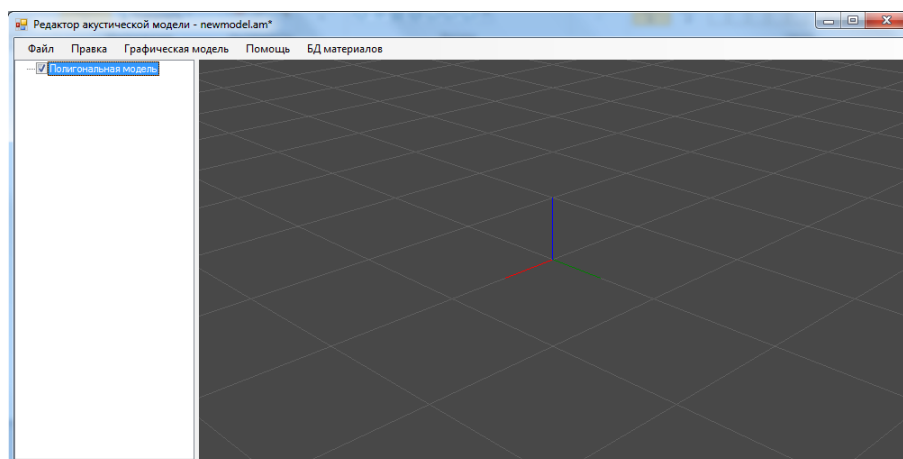


Рис. 4.1. Главная форма редактора акустической модели

Название: Помещение 1: Помещение 2:

Координаты точки на верхней грани: x: y: z:

Координаты левого нижнего угла: x: y: z:

Координаты правого нижнего угла: x: y: z:

Слой перегородки

	Материал	Толщина (в метрах)
▶	Сталь	0.002
	Минеральная вата	0.04
	Сталь	0.002
*		

Рис. 4.2. Форма редактирования акустической перегородки

Название: Помещение:

Координаты: x: y: z:

Рис. 4.3. Форма редактирования расчётной точки

Результаты акустического расчёта

Название расчётной точки	Название источника шума	f=31.5	f=63	f=125	f=250	f=500	f=1000	f=2000	f=4000	f=8000	Уровень по шкале A	Координата x расчётной точки	Координата y расчётной точки	Координата z расчётной точки
Расчётная ...	Источник ...	142.000	65.000	34.000	14.000	0.000	0.000	0.000	0.000	0.000	100.000	0.000	0.000	0.000

Рис. 4.4. Форма с результатами акустического расчёта:
 последовательно идут столбцы с названием расчётной точки, названием источника шума, столбцы с уровнями шума, создаваемого в указанной расчётной точке указанным источником шума для октавных полос со среднегеометрическими частотами f от 31,5 Гц до 8000 Гц, уровень шума по шкале A, координаты расчётной точки.

Редактирование линейного источника шума

Название: Полный уровень по шкале A, dBA=70.0003256576812

Уровень шума по октавным полосам со среднегеометрической частотой f :

f=31,5 Гц f=63 Гц f=125 Гц f=250 Гц f=500 Гц f=1000 Гц f=2000 Гц f=4000 Гц f=8000 Гц

Координаты начальной точки

x: y: z:

Координаты конечной точки

x: y: z:

112 55 21 0 0 0 0 0 0

Рис. 4.5. Форма редактирования линейного источника шума

Редактирование источника шума

Название: Полный уровень по шкале A, dBA=64.791204194496

Уровень шума по октавным полосам со среднегеометрической частотой f :

f=31,5 Гц f=63 Гц f=125 Гц f=250 Гц f=500 Гц f=1000 Гц f=2000 Гц f=4000 Гц f=8000 Гц

Помещение:

Координаты

x: y: z:

85 42 45 22 37 31 42 55 63

Рис. 4.6. Форма редактирования точечных источников шума

Редактирование отражающих поверхностей

Название:

Материал:

Координаты точки на верхней грани

x: y: z:

Координаты левого нижнего угла

x: y: z:

Координаты правого нижнего угла

x: y: z:

Готово Отмена

Рис. 4.7. Форма редактирования отражающих поверхностей

Редактирование помещения

Название:

Готово Отмена

Рис. 4.8. Форма редактирования помещений

БД материалов

Соединение: Правка:

Поиск:

ID	Outdated	Name	AMBIENTCOLOR	REFLECTION	ABSORPTPARAM
0	0	Сталь	C0C0C0	0.3	0.2
2	0	Минеральная в...	76824D	0.2	0.6
4	0	Стекло	D0D0D0	0.2	0.05

Рис. 4.9. Форма просмотра и редактирования базы данных материалов

Редактирование материала

Название:

Коэффициент отражения:

Параметр поглощения:

Сбросить Сохранить Отмена

Рис. 4.10. Форма редактирования материала в базе данных материалов

4.5. Пример работы программы

4.5.1. Описание тестовой модели

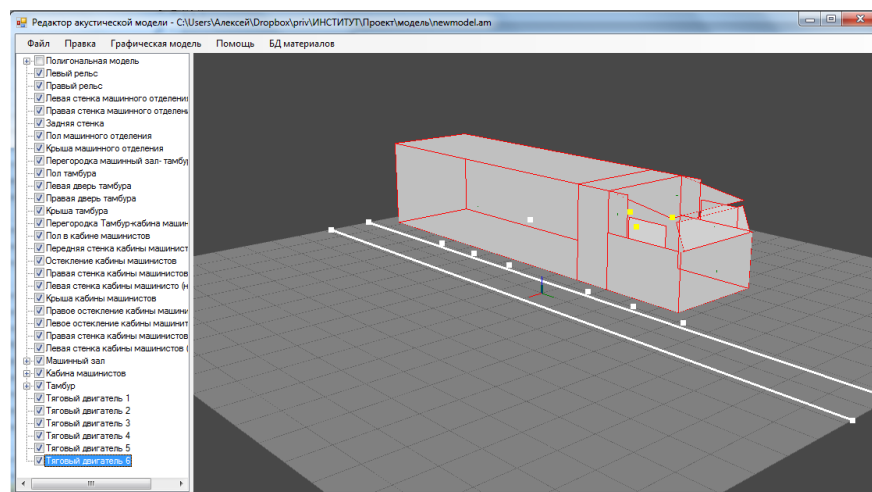


Рис. 4.11. Тестовая акустическая модель

В качестве тестовой модели используется модель из примера в главе (2.4)

Далее представлены формы, демонстрирующие параметры элементов модели (для однотипных элементов модели представлено по 1 форме, демонстрирующей общие характеристики).

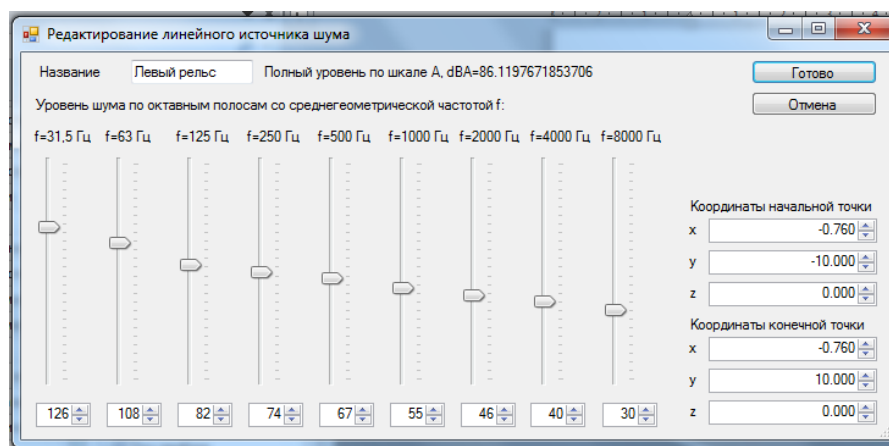


Рис. 4.12. Акустические характеристики рельсов как источников шума

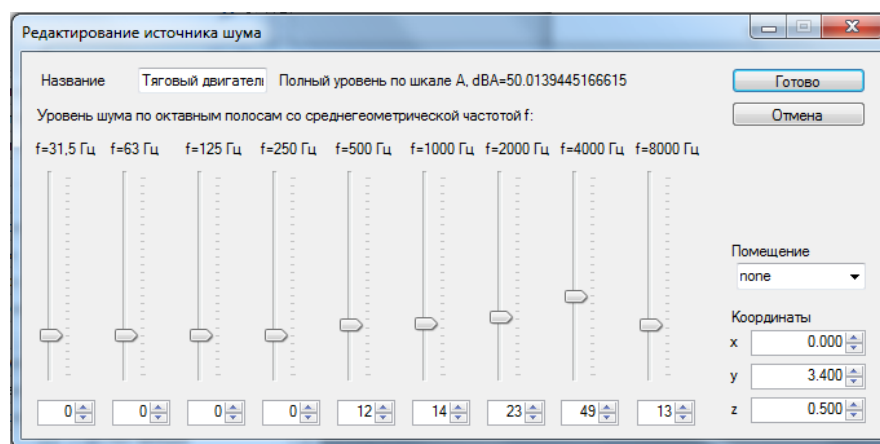


Рис. 4.13. Акустические характеристики одного из тяговых двигателей

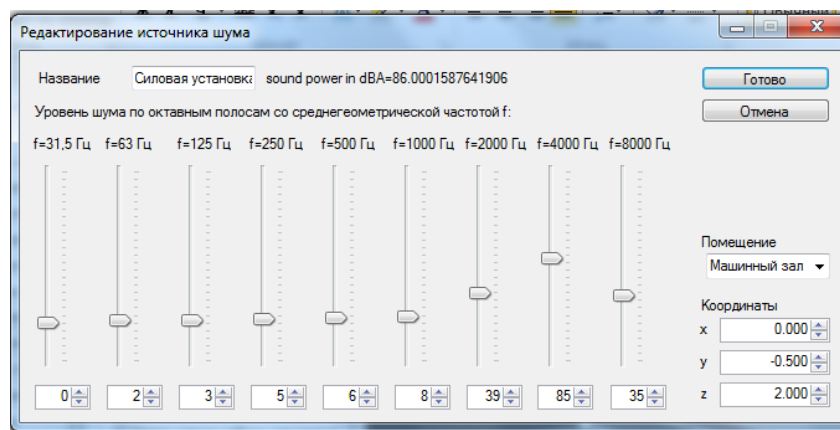


Рис. 4.14. Акустические характеристики силовой установки

Редактирование перегородки

Название: Остекление кабины м. Помещение 1: Кабина машиниста. Помещение 2: none

Координаты точки на верхней грани:

x	0.000
y	5.650
z	3.300

Координаты левого нижнего угла:

x	-1.200
y	5.900
z	2.700

Координаты правого нижнего угла:

x	1.200
y	5.900
z	2.700

Слой перегородки:

Материал	Толщина (в метрах)
▶ Стекло	0.02
*	

Перевернуть

Рис. 4.15. Характеристики лобового остекления

Редактирование перегородки

Название: Перегородка Тамбур- Помещение 1: Кабина машиниста- Помещение 2: Тамбур

Готово Отмена

Координаты точки на верхней грани

x	0.000
y	3.700
z	3.700

Координаты левого нижнего угла

x	1.300
y	3.700
z	1.300

Координаты правого нижнего угла

x	-1.300
y	3.700
z	1.300

Слой перегородки

	Материал	Толщина (в метрах)
▶	Сталь	0.002
	Минеральная вата	0.1
	Сталь	0.002
*		

Перевернуть

Рис. 4.16. Характеристики перегородки между тамбуром и кабиной машинистов

Редактирование перегородки

Название: Правая стенка маши- Помещение 1: Машинный зал Помещение 2: попе

Готово Отмена

Координаты точки на верхней грани

x	1.300
y	0.000
z	3.700

Координаты левого нижнего угла

x	1.300
y	2.700
z	1.300

Координаты правого нижнего угла

x	1.300
y	-5.000
z	1.300

Слой перегородки

	Материал	Толщина (в метрах)
▶	Сталь	0.002
	Минеральная вата	0.04
	Сталь	0.002
*		

Перевернуть

Рис. 4.17. Характеристики одной из внешних стен вагона

Редактирование перегородки

Название: Пол в кабине машины Помещение 1: Кабина машиниста Помещение 2: попе

Готово Отмена

Координаты точки на верхней грани

x	0.000
y	5.900
z	1.300

Координаты левого нижнего угла

x	-1.300
y	3.700
z	1.300

Координаты правого нижнего угла

x	1.300
y	3.700
z	1.300

Слой перегородки

	Материал	Толщина (в метрах)
▶	Сталь	0.02
*		

Перевернуть

Рис. 4.18. Характеристики пола в кабине машинистов

Редактирование расчётной точки

Название: Место машиниста 1

Помещение: Кабина машинистов

Координаты:

x: -0.650

y: 4.000

z: 2.800

Готово Отмена

Рис. 4.19. Характеристики одной из расчётных точек в кабине машиниста

Редактирование расчётной точки

Название: Точка в тамбуре

Помещение: Тамбур

Координаты:

x: 0.000

y: 3.200

z: 2.900

Готово Отмена

Рис. 4.20. Характеристики расчётной точки в тамбуре

Редактирование отражающих поверхностей

Название: Бетонное основание

Материал: асфальт, бетон

Координаты точки на верхней грани

Координаты левого нижнего угла

Координаты правого нижнего угла

x: 0.000 x: 10.000 x: -10.000

y: -10.000 y: 10.000 y: 10.000

z: 0.000 z: 0.000 z: 0.000

Готово Отмена

Рис. 4.21. Характеристика отражающей поверхности

4.5.2. Результат расчётов

Результаты акустического расчёта														
Название расчётной точки	Название источника шума	f=31.5	f=63	f=125	f=250	f=500	f=1000	f=2000	f=4000	f=8000	Уровень по шкале А	Координата x расчётной точки	Координата y расчётной точки	Координата z расчётной точки
Место машиниста 1	Тяговый двигатель 2	1.179	1.179	1.179	1.170	13.170	15.161	24.152	50.143	14.143	51.157	-0.650	4.000	2.800
Место машиниста 1	Тяговый двигатель 6	-9.849	-9.849	-9.849	-9.869	2.131	4.111	13.090	39.069	3.069	40.083	-0.650	4.000	2.800
Место машиниста 1	Тяговый двигатель 5	-6.013	-6.013	-6.013	-6.031	5.969	7.952	16.935	42.917	6.917	43.931	-0.650	4.000	2.800
Место машиниста 1	Тяговый двигатель 4	-0.050	-0.050	-0.050	-0.059	11.941	13.933	22.924	48.915	12.915	49.929	-0.650	4.000	2.800
Место машиниста 1	Тяговый двигатель 3	-2.922	-2.922	-2.922	-2.938	9.062	11.046	20.029	46.013	10.013	47.027	-0.650	4.000	2.800
Место машиниста 1	Силовая установка	-6.216	-4.216	-3.216	-1.216	-0.216	1.783	32.783	78.782	28.782	79.783	-0.650	4.000	2.800
Место машиниста 1	Тяговый двигатель 1	-0.766	-0.766	-0.766	-0.777	11.223	13.213	22.202	48.191	12.191	49.205	-0.650	4.000	2.800
Место машиниста 1	Левый рельс	122.199	104.199	78.199	70.177	63.177	51.154	42.132	36.110	26.110	82.318	-0.650	4.000	2.800
Место машиниста 1	Правый рельс	122.199	104.199	80.199	70.177	63.177	51.154	42.132	36.110	26.110	82.342	-0.650	4.000	2.800
Место машиниста 2	Тяговый двигатель 6	-9.849	-9.849	-9.849	-9.869	2.131	4.111	13.090	39.069	3.069	40.083	0.650	4.000	2.800
Место машиниста 2	Левый рельс	122.199	104.199	78.199	70.177	63.177	51.154	42.132	36.110	26.110	82.318	0.650	4.000	2.800
Место машиниста 2	Тяговый двигатель 4	-0.050	-0.050	-0.050	-0.059	11.941	13.933	22.924	48.915	12.915	49.929	0.650	4.000	2.800
Место машиниста 2	Тяговый двигатель 5	-6.013	-6.013	-6.013	-6.031	5.969	7.952	16.935	42.917	6.917	43.931	0.650	4.000	2.800
Место машиниста 2	Тяговый двигатель 2	1.179	1.179	1.179	1.170	13.170	15.161	24.152	50.143	14.143	51.157	0.650	4.000	2.800
Место машиниста 2	Тяговый двигатель 3	-2.922	-2.922	-2.922	-2.938	9.062	11.046	20.029	46.013	10.013	47.027	0.650	4.000	2.800
Место машиниста 2	Силовая установка	-6.216	-4.216	-3.216	-1.216	-0.216	1.783	32.783	78.782	28.782	79.783	0.650	4.000	2.800
Место машиниста 2	Тяговый двигатель 1	-0.766	-0.766	-0.766	-0.777	11.223	13.213	22.202	48.191	12.191	49.205	0.650	4.000	2.800
Место машиниста 2	Правый рельс	122.199	104.199	80.199	70.177	63.177	51.154	42.132	36.110	26.110	82.342	0.650	4.000	2.800
Точка в тамбуре	Тяговый двигатель 1	3.748	3.748	3.748	3.740	15.740	17.732	26.724	52.716	16.716	53.730	0.000	3.200	2.900
Точка в тамбуре	Силовая установка	-4.896	-2.896	-1.896	0.103	1.103	3.103	34.102	80.102	30.102	81.102	0.000	3.200	2.900
Точка в тамбуре	Тяговый двигатель 3	-3.102	-3.102	-3.102	-3.120	8.880	10.861	19.843	45.825	9.825	46.839	0.000	3.200	2.900
Точка в тамбуре	Тяговый двигатель 2	0.640	0.640	0.640	0.627	12.627	14.615	23.602	49.589	13.589	50.603	0.000	3.200	2.900
Точка в тамбуре	Тяговый двигатель 5	-4.743	-4.743	-4.743	-4.760	7.240	9.223	18.205	44.188	8.188	45.202	0.000	3.200	2.900
Точка в тамбуре	Тяговый двигатель 4	1.233	1.233	1.233	1.225	13.225	15.216	24.207	50.199	14.199	51.213	0.000	3.200	2.900
Точка в тамбуре	Правый рельс	123.261	105.261	81.261	71.239	64.239	52.217	43.195	37.172	27.172	83.405	0.000	3.200	2.900
Точка в тамбуре	Тяговый двигатель 6	-8.564	-8.564	-8.564	-8.584	3.416	5.395	14.375	40.354	4.354	41.368	0.000	3.200	2.900
Точка в тамбуре	Левый рельс	123.261	105.261	79.261	71.239	64.239	52.217	43.195	37.172	27.172	83.381	0.000	3.200	2.900

Рис. 4.22. Форма с результатами акустического расчёта

4.6. Выводы

Приложение успешно реализовано. В следующей главе демонстрируется пример акустического расчёта тестовой модели.

ЗАКЛЮЧЕНИЕ

Создана начальная версия системы акустического моделирования железнодорожных транспортных средств, позволяющая определить уровень шума в заданных расчётных точках подвижного состава. Приложение реализовано в результате выполнения следующих этапов:

- Разработана архитектура системы акустического моделирования.
- Предложена модель акустического расчёта.
- Разработан формат данных для представления модели транспортного средства.
- Разработана и реализована реляционная модель базы данных Материалы.
- Выполнено проектирование и реализация классов компонентов приложения, обеспечивающих управление базой данной, акустический расчёт, управление параметрами модели, ввод данных, отображение результатов и другие функции приложения;
- Выполнен акустический расчёт в различных точках модели электровоза ВЛ-15, обнаружено значительное превышение норм шума на рабочих местах машинистов.

На данный момент в модели не учитываются такие факторы, как структурный шум, штучные поглотители шума в помещениях и зависимость коэффициентов поглощения и отражения шума акустических перегородок от частоты звука. Расширение модели и усовершенствование алгоритма, позволяющие учесть эти факторы, а также более тесная интеграция с системами автоматизированного проектирования – возможное направление дальнейшей работы над полученной системой акустического моделирования.

Подпись автора _____ (Апарнев А.Н.)

ЛИТЕРАТУРА

1. Белякин С.К. Разработка метода акустического расчёта комбинированных глушителей шума транспортных средств: диссертация ... кандидата технических наук : 05.05.03./ Белякин, С.К.; [Место защиты: Курганский государственный университет]. - Курган, 2000. - 189 с.
2. Трунова И.Г. Выбор и расчет средств защиты от шума и вибрации: учебное пособие по выполнению дипломных, курсовых и практических работ для студентов /И.Г. Трунова А.Б. Елькин, В.М. Смирнова. - Нижний Новгород: НГТУ им. Р.Е. Алексеева, 2012. - 116 с.
3. Шишелова Т.И., Малыгина Ю.С., Нгуен Суан Дат ВЛИЯНИЕ ШУМА НА ОРГАНИЗМ ЧЕЛОВЕКА [Электронный ресурс] // Успехи современного естествознания. 2009. № 8. с. 14-15. URL: <http://www.natural-sciences.ru/ru/article/view?id=14048> (дата обращения: 15.06.2016)
4. ГОСТ 12.0.003-74 ССБТ. Опасные и вредные производственные факторы. Классификация. – введ. 1974-01-01. – Москва: ИПК издательство стандартов, 2000. – 4 с.
5. ГОСТ 12.1.003-2014 ССБТ. Шум. Общие требования безопасности. – взамен ГОСТ 12.1.003-83, ГОСТ 12.1.023-80 – введ. 2015-11-01. –Москва: ФГУП Стандартиформ, 2015. – 23 с.
6. Пронников Ю.В. Совершенствование методов виброакустических расчетов и проектирования кабин локомотивов : диссертация ... кандидата технических наук : 05.02.02, 05.26.01 / Пронников Ю.В.; [Место защиты: Дон. гос. техн. ун-т]. - Ростов-на-Дону, 2012. - 132 с.

Приложение А

(обязательное)

Описания основных классов

1. Модуль MaterialsDatabase

```

public class DBConnectionException : Exception { }
public class ServerException : DBConnectionException { }
public class DatabaseNotFoundException : DBConnectionException { }
public class AuthenticationException : DBConnectionException { }
public class MaterialsDBClient
{
    public MaterialsDBClient(string UserName, string UserPassword,
string ServerName = "localhost", uint ServerPort = 3306);
    public enum UserMode { Viewer, Editor, Corrector }
    public DataTable Materials { get; }
    public Material GetMaterialByID(Int32 id);
    public Int32 AddMaterial(Material m);
    public bool RemoveOutdatedMaterials();
    public bool UpdMaterialProp(Int32 id, MProps p, object value);
    public bool UpdMaterial(Int32 id, Material m);
    public bool SetMaterialOutDated(Int32 id, bool OutDated);
}
public interface IMatDBDict:IDictionary<Int32,Material> { }
public class MatDBDict : IMatDBDict
{
    public MatDBDict(string username, string pwd, string host, uint
port);
    public void Reload();
    public void Add(int key, Material value);
    public bool ContainsKey(int key);
    public ICollection<int> Keys { get; }
    public bool Remove(int key);
    public bool TryGetValue(int key, out Material value);
    public ICollection<Material> Values;
    public Material this[int key] { get; set; }
    public void Add(KeyValuePair<int, Material> item);
    public void Clear();
    public bool Contains(KeyValuePair<int, Material> item);
    public void CopyTo(KeyValuePair<int, Material>[] array, int ar-
rayIndex);
    public int Count{ get; }
    public bool IsReadOnly { get; }
    public bool Remove(KeyValuePair<int, Material> item);
    public IEnumerator<KeyValuePair<int, Material>> GetEnumerator();
    System.Collections.IEnumerator Sys-
tem.Collections.IEnumerable.GetEnumerator();
}

```

2. Модуль ModelLib

```

public class Model
{
    public Model(IMatDBDict m);
    public void AddBorder(Border b, Room r1 = null, Room r2 = null);
}

```

```

    public IMatDBDict mdb;
    public PolygonalModel MeshModel;
    public List<calcpoint> calcps;
    public List<Room> rooms;
    public List<Border> borders;
    public List<Reflector> reflectors;
    public List<NoiseSource> sources;
    public void SaveToFile(string filename);
    public Dictionary<int,string> GetMaterialsDict();
    public Dictionary<calcpoint,Dictionary<NoiseSource,noise>> calc();
    public static Model LoadFromFile(IMatDBDict md, string filename);
    public Color GetMaterialColor(int mid);
}
public abstract class ModelPart
{
    public ModelPart();
    public bool visible;
    public bool selected;
    public string name;
    public override String ToString();
}
public class pointModelPart:ModelPart
{
    public Room room;
    public Vector3d pos;
}
public class Border : ModelPart
{
    public Border();
    public Border clone();
    public void flip();
    public spacerect Rectangle;
    public Room room1, room2;
    public List<border_layer> layers;
}
public class border_layer
{
    public int materialId;
    public double thickness;
}
public class spacerect
{
    public OpenTK.Vector3d ontop, bottomleft, bottomright;
    public Vector3d lt { get; }
    public double Square { get; }
    public double maxlinsz { get; }
    public Vector3d center { get; }
    public Vector3d facenorm { get ; }
    public spacerect clone();
    public Vector3d rt { get; }
    public Vector3d lb { get; }
    public Vector3d rb { get; }
}
public class calcpoint:pointModelPart
{
    public calcpoint();
    public calcpoint clone();
}

```

```

}
public class Document
{
    public Document(String host, UInt32 port, String account, String
password, string filename = "");
    public string FName { get; set; }
    public Model Model { get; }
    public bool Saved { get; }
    public bool save(string filename);
    public bool saveas(string filename);
    public bool save();
    public void Load(string filename);
}
public class LinearNoiseSrc:NoiseSource
{
    public Vector3d A, B;
    public override noise GetLevel(Vector3d posit);
}
public class PolygonalModel : ModelPart
{
    public PolygonalModel();
    public List<Mesh> meshes;
    public List<gmaterial> mats;
    public static PolygonalModel LoadFromFile(string filename);
}
public class gmaterial
{
    public Color colAmbient;
    public Color colDiffuse;
    public Color colSpecular;
}
public class Mesh:ModelPart
{
    public Mesh();
    public List<Vector3d> vertices;
    public List<Tuple<int, int, int>> surface;
    public List<Vector3d> normals;
    public int matind;
}
class NoiseCalculator
{
    public NoiseCalculator(Model m);
    public void calc();
    public Dictionary<calcpoint, Dictionary<NoiseSource, noise>> res;
}
public class NoiseSource:pointModelPart
{
    public NoiseSource();
    public noise power;
    public NoiseSource clone();
    public virtual noise GetLevel(Vector3d posit);
}
public class noise
{
    public noise();
    public noise clone();
}

```

```

        public noiseoctlvl lvl315, lvl63, lvl125, lvl250, lvl500,
        lvl1000, lvl2000, lvl4000, lvl8000;
        public noiseoctlvl dBA { get;}
        public static noise operator +(noise A, noise B);
        public noise GetInDist(double dist);
    }
    public class noiseoctlvl
    {
        public noiseoctlvl(double c);
        public double value;
        public static noiseoctlvl operator +(noiseoctlvl a, noiseoctlvl
b);
    }
    public class Reflector : ModelPart
    {
        public spacerect Rectangle;
        public int matid;
        public Reflector();
        public ModelPart clone();
    }
    public class reflmat
    {
        public string name;
        public double d315, d63, d125, d250, d500, d1000, d2000, d4000,
d8000;
        public Color col;
        public override String ToString();
    }
    public static class ReflMatColl
    {
        public static Dictionary<int, reflmat> d;
        public static bool initied = false;
        public static void init();
    }
    public class Room : ModelPart
    {
        public Room();
        public Room clone();
    }

```

3. Модуль Viewport

```

public class Angle
{
    public Angle(float Radians = 0.0f);
    public float Radians;
    public const float pi = (float)Math.PI;
    public float Degrees { get; set; }
    public static float DegreesFromRadians(float radians);
    public static float RadiansFromDegrees(float degrees);
    public static Angle FromRadians(float R);
    public static Angle FromDegrees(float D);
    public static implicit operator Angle(float FloatValue);
    public static implicit operator float(Angle AngleValue);
}
public static class GraphicsOptions
{
    public static Color BackgroundColor;
}

```

```

    public static Color HighGridColor;
    public static Color LowGridColor;
    public static Color SelectedItemColor;
    public static Color AxisXColor;
    public static Color AxisYColor;
    public static Color AxisZColor;
    public static Color CactPointDefault;
    public static Color NoiseSourceDefault;
    public static void SetDefaults();
    public static bool LoadFromXMLFile(string pth);
    public static bool SaveToXMLFile(string pth);
}
public partial class ViewportControl : UserControl
{
    public ViewportControl();
    public ViewportControl(ViewportList Container);
    public ViewportList OwnerList { get; set; }
    public Document Document { get; }
    public float CameraPosX { get; set; }
    public float CameraPosY { get; set; }
    public float CameraPosZ { get; set; }
    public Angle CameraYaw { get; set; }
    public Angle CameraPitch { get; set; }
    public Angle CameraRoll { get; set; }
    protected override void OnPaint(PaintEventArgs e);
    public void Redraw();
}
public enum projectionMode { axonometric = 0, perspective = 1 }
public enum defaultView { Custom = 0, Bottom = 2, Top = 3, Left = 4,
    Right = 5, Back = 8, Front = 9, }
public class ViewportList : IList<ViewportControl>
{
    public ViewportList(Document owner);
    public Document ownDoc;
    public int IndexOf(ViewportControl item);
    public void Insert(int index, ViewportControl item);
    public void RemoveAt(int index);
    public ViewportControl this[int index] { get; set; }
    public void Add(ViewportControl item);
    public void Clear();
    public bool Contains(ViewportControl item);
    public void CopyTo(ViewportControl[] array, int arrayIndex) ;
    public int Count { get; }
    public bool Remove(ViewportControl item);
    public void Redraw();
    bool ICollection<ViewportControl>.IsReadOnly { get; }
    public IEnumerator<ViewportControl> GetEnumerator();
    System.Collections.IEnumerator Sys-
tem.Collections.IEnumerable.GetEnumerator();
}

```