Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждения

высшего образования

ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

СИСТЕМ УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ (ТУСУР)

Кафедра компьютерных систем в управлении и проектировании (КСУП)

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

по дисциплине

«Основы разработки САПР» (ОРСАПР)

Студент гр. 586-1

\_\_\_\_\_\_\_\_\_\_\_\_\_Д.А. Мех

«\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2020г.

Руководитель

К.т.н., доцент каф. КСУП

\_\_\_\_\_\_\_\_\_\_А.А. Калентьев

«\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2020г.

2020

**РЕФЕРАТ**

Пояснительная записка 46 с., 29 рис., 4 таблицы, 12 источника.

AUTOCAD, ПЛАГИН, МОДЕЛЬ КРОВАТИ, АВТОМАТИЗИРОВАННОЕ ПРОЕКТИРОВАНИЕ, САПР.

Целью данной работы является разработка плагина, автоматизирующего построение модели кровати, для системы автоматизированного проектирования AutoCAD 2020, с помощью интегрированной среды разработки Visual Studio Сommunity 2020.

В процессе работы изучено внешнее устройство кровати и основные функции системы автоматизированного проектирования «AutoCAD» 2020.

В результате работы был создан плагин, автоматизирующий построение кровати.

Отчет по пояснительной записке выполнен в текстовом редакторе Microsoft Word 2019.

**Содержание**

1 Введение 4

2 Постановка и анализ задачи 5

2.1 Описание предмета проектирования 5

2.2 Выбор инструментов и средств реализации 8

2.3 Назначение плагина 8

2.4 Описание аналогов разрабатываемого продукта 8

2.4.1 Оборудование: Металлоконструкции 8

2.4.2 Hilti BIM/CAD 10

3 Описание реализации 12

3.1 Диаграмма прецедентов плагина 13

3.2 Диаграмма классов 17

3.3 Диаграмма пакетов 21

4 Описание программы для пользователя 23

5 Тестирование программы 27

5.1 Функциональное тестирование 27

5.2 Модульное тестирование 30

5.3 Нагрузочное тестирование 31

Заключение 34

Список использованных источников 35

Приложение А 36

**1 Введение**

Автоматизация моделирования имеет огромное значение для развития науки, техники и производства в современном обществе. В настоящее время автоматизация – основной способ повышения производительности и эффективности труда инженерно-технических работников, занимающихся моделированием сложных устройств. Использование автоматизации в проектировании позволяет создавать все более сложные технические объекты и гибко реагировать на появление новых решений и технологий в той или иной области техники.

Практическая реализация методов и идей автоматизированного моделирования происходит в рамках систем автоматизированного проектирования (САПР). Однако мало создать высокопроизводительные современные САПР. Надо уметь их эффективно использовать. Для этого нужны квалифицированные инженеры-пользователи САПР. В рамках современного «компьютеризированного» общества инженер любой специальности, занимающийся разработкой технических устройств, должен уметь использовать средства автоматизированного проектирования. Это позволяет повысить эффективность моделирования, улучшить его качество, снизить материальные затраты и уменьшить число разработчиков.

Таким образом, целью данной работы является разработка плагина, автоматизирующего построение модели кровати, для системы автоматизированного проектирования «AutoCAD» 2020 [1], с помощью интегрированной среды разработки Visual Studio Сommunity 2020 [2].

Интегрированная среда разработки Visual Studio — это стартовая площадка для написания, отладки и сборки кода, а также последующей публикации приложений. Интегрированная среда разработки (IDE) представляет собой многофункциональную программу, которую можно использовать для различных аспектов разработки программного обеспечения.

**2 Постановка и анализ задачи**

В рамках лабораторных работ в соответствии с технически заданием требовалось разработать плагин, который на основе входных параметров, интегрируя с системой AutoCAD [1], строит деталь «Кровать». Более того, требовалось, чтоб плагин позволял изменять входные параметры в соответствии с требованиями пользователей, а именно: высота, длина и ширина основной части, диаметр и высота ножек, а также высота и толщина спинки.

**2.1 Описание предмета проектирования**

Кровать – это элемент мебели, как правило, выполненный из дерева или металла и имеющий форму прямоугольника или квадрата. На каркас кровати устанавливается матрас, по форме и размерам соответствующий ложу кровати, а также укладываются другие постельные принадлежности.

Разрабатываемая модель кровати состоит из трех основных частей: спинки, ножек и основной части. Данные детали сборки должны соответствовать следующим параметрам:

* Габариты основной части кровати: Длина (от 120 до 230 см), Ширина (от 60 до 230 см), Высота (от 10 до 30 см).
* Габариты ножек: Диаметр (от 5 до 10 см), Высота (от 10 до 30 см).
* Габариты спинки: Высота (от 60 до 100 см), Толщина спинки (от 5 до 30 см)

Изделие должно иметь от 1-го до 2-х спальных мест и быть пригодным для использования человеком ростом от 100 до 220 см.

На рисунке 4.2 приведена 3D модель кровати с обозначенными на нем параметрами L, H, W, D, Lh, Hb, Tb.

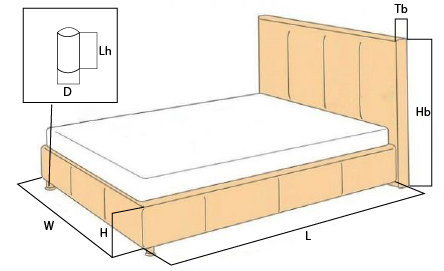


Рисунок 4.2 – 3D модель кровати с обозначенными параметрами

Обозначенные параметры:

H – Высота основной части;

L – Длинна основной части;

W – Ширина основной части;

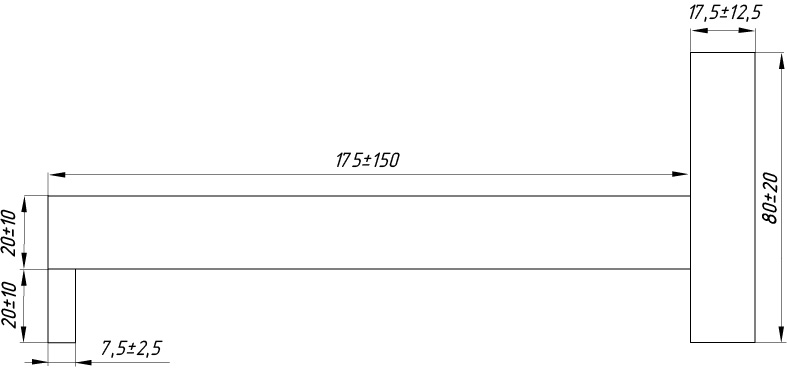
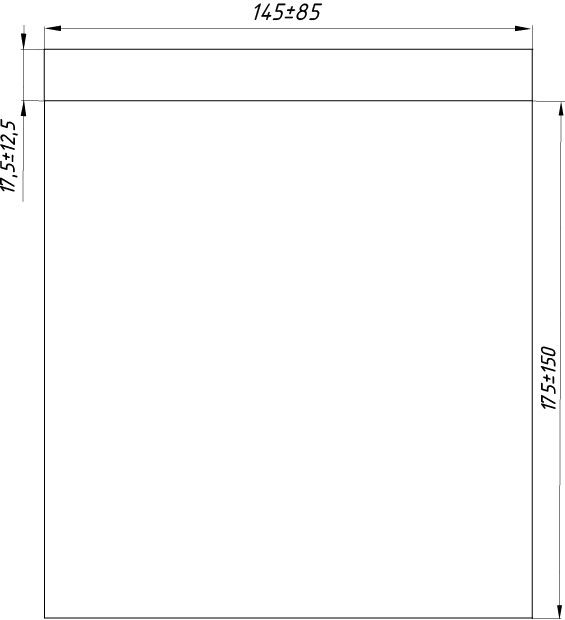
D – Диаметр ножек;

Lh – Высота ножек;

Hb – Высота спинки;

Tb – Толщина спинки;

Рассмотрим чертеж кровати со средними размерами.



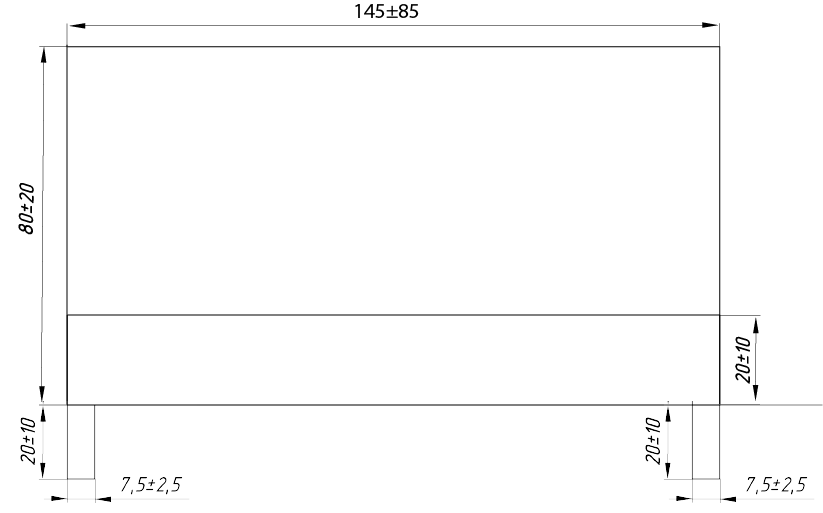


Рисунок 4.3 – Чертеж кровати в 3-х проекциях

**2.2 Выбор инструментов и средств реализации**

В связи с требованием технического задания программа выполнена на языке C# в среде Microsoft Visual Studio 2020 [2], с использованием .NET Framework 4.7.0, для системы AutoCAD 2020. Инструментом тестирования и создания модульных тестов был выбран стандартный обозреватель тестов среды Microsoft Visual Studio 2020 с тестовым фреймворком NUnit версии 3.12.0.

Для реализации пользовательского интерфейса использовался WinForm.

Взаимодействие плагина с системой AutoCAD 2020 [1] осуществляется посредством интерфейсов, называемого API. Явных преимуществ между версиями нет, поскольку обе версии реализуют различные функции системы и взаимно дополняют друг друга. Для выполнения лабораторных использовался AutoCAD .NET API.

**2.3 Назначение плагина**

Назначение, разрабатываемого плагина обусловлено выбором сферы его применения. В данном проекте выбором сферы применения стал подбор параметров кровати, удовлетворяющих методу ее эксплуатации, так как данная сфера, одна из наиболее популярных, ввиду повышенного спроса на аудио выводящее оборудование. Для поддержание пользовательского интереса и спроса на продукцию, производителям необходимо часто выпускать товар с обновленным дизайном.

# 2.4 Описание аналогов разрабатываемого продукта

**2.4.1 Оборудование: Металлоконструкции**

Оборудование: Металлоконструкции [3] — приложение для КОМПАС-3D, предназначенное для автоматизации работ по проектированию конструкций из профильного металлопроката. Приложение позволяет быстро проектировать всевозможные рамы и каркасы, автоматически создавать комплект документации.

Создание металлоконструкции в приложении начинается с построения Трехмерного каркаса — геометрических осей, которые являются эскизом конструкции. После чего для каждой из осей назначается профиль. Сортамент профиля может выбираться из нового Каталога профилей, входящего в комплект поставки приложения, либо из Справочника Материалы и Сортаменты для КОМПАС. Для удобства построения и редактирования металлоконструкции в приложении реализован механизм Характерных точек, который позволяет задавать длину и угол поворота профиля непосредственно в окне построения. При изменении Трехмерного каркаса металлоконструкция перестроится автоматически.

После назначения профилей необходимо проработать отдельные узлы металлоконструкции. Для этого в приложении есть специальные инструменты. Можно корректировать длины деталей, задавать угловую или стыковую разделки, строить дополнительные элементы в виде ребер жесткости или фасонок.

Для созданной с помощью приложения конструкции можно автоматически получить спецификацию либо любые другие виды настраиваемых отчетов. Металлоконструкция, спроектированная в приложении, может быть проверена на наличие пересечений. Трехмерная модель позволяет сразу выявить возможные нестыковки. Оборудование: Металлоконструкции позволяет избежать дополнительных затрат на материал и инструмент.

На рисунке 3.1 представлен интерфейс, каталога выбора структуры швеллера, приложения “Оборудование: Металлоконструкции”.

Рисунок 3.1 – Интерфейс, каталога выбора структуры швеллера, приложения “Оборудование: Металлоконструкции”

**2.4.2 Hilti BIM/CAD**

Существует похожая сборка Hilti BIM/CAD. Скачать можно с официального сайта разработчика HILTI [4].

Данный плагин представляет собой библиотеку загружаемых 2D и 3D моделей монтажных изделий Hilti. Все объекты содержат BIM информации, атрибуты и спецификации. Также есть возможность интеграции в различные CAD системы и представление модели в разных форматах. Главное окно библиотеки представлено на рисунке 3.1.

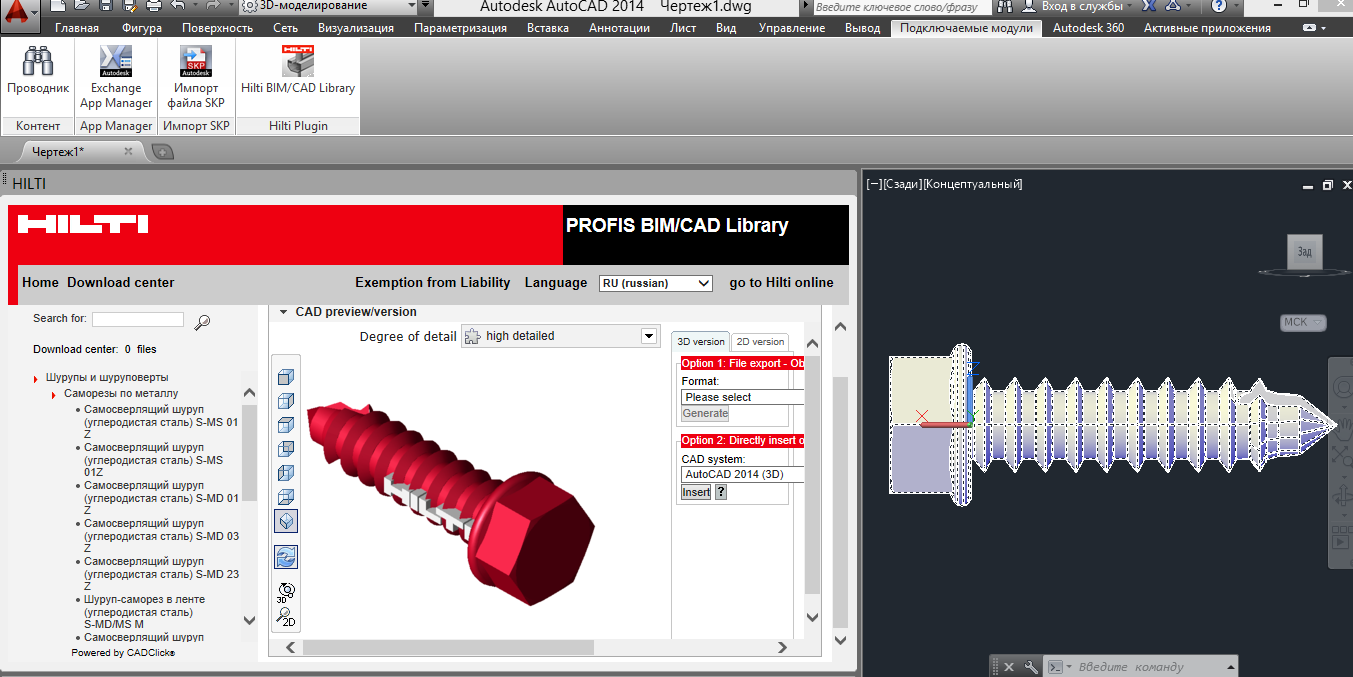


Рисунок 3.2 – Главное окно библиотеки Hilti BIM/CAD

**3 Описание реализации**

На этапе разработки проекта программы для формального описания архитектурной особенности, пользовательского сценария системы был выбран унифицированный язык моделирования (UML) [5]. На основе UML построены: диаграммы вариантов использования диаграммы классов, диаграммы пакетов.

В процессе реализации диаграммы классов, вариантов использования и пакетов были дополнены.

Дополнения были обусловлены нарастанием функционала в реализации мелких детальных особенностей и изменений требований заказчика.

**3.1 Диаграмма прецедентов плагина**

Прецеденты [5] – это технология определения функциональных требований к системе. Работа прецедентов заключается в описании типичных взаимодействий между пользователем системы и самой системой.

Прецеденты представляют собой ценный инструмент для понимания функциональных требований к системе.

Рассмотрим диаграмму прецедентов плагина на рисунке 3.1. Действующим лицом выступает пользователь.

Для пользователя доступны следующие возможности:

* Ввести параметры;
* Создать модель кровати;
* Нарисовать превью модели кровати;
* Запустить плагин;
* Закрыть плагин.



Рисунок 3.1 – Диаграмма прецедентов плагина

В результате изменений, внесенных заказчиком, диаграмма прецедентов подверглась изменениям.

Добавлены возможности для задания количества и формы динамиков. Рассмотрим изначальную версию диаграммы прецедентов на рисунке 3.2

****

Рисунок 3.2 – Изначальная диаграмма прецедентов плагина

**3.2 Диаграмма классов**

Диаграмма классов [5] описывает типы объектов системы и различного рада статические отношения, которые существуют между ними. На диаграммах классов отображаются также свойства классов, операции классов и ограничения, которые накладываются на связи между объектами. В UML термин функциональность применяется в качестве основного термина, описывающего и свойства, и операции класса. Целью создания диаграммы классов является графическое представление статической структуры декларативных элементов системы.

Рассмотрим диаграмму классов плагина на рисунке 3.3.



Рисунок 3.3 – Диаграмма классов плагина

Класс Parameter хранит информацию об одном параметре проектируемой модели.

Класс Parameters хранит словарь со всеми параметрами модели, также поддерживаются методы расчета средних значений параметров в зависимости от максимальных и минимальных значений.

Класс ModelBuilder хранит параметры модели, введенные пользователем. Реализуются методы используемы для построения 3D модели, данные методы используют параметры модели.

Класс CommandManager хранит объект класса построителя 3D модели (ModelBuilder), реализует методы для связи программы с САПР.

Класс ModelParametersForm является формой пользовательского интерфейса. Реализует методы, используемые для взаимодействия с пользователем.Рассмотрим изначальную версию диаграммы классов на рисунке 3.4.

Класс ParametersConstants был добавлен для того, чтобы в него можно было вынести граничные и некоторые стандартные значения параметров кровати.

Класс ModelDrawer был добавлен для того, чтобы рисовать на переданном ему PictureBox объекте модель кровати.

Рассмотрим изначальную версию диаграммы классов на рисунке 3.4



Рисунок 3.4 – Изначальная диаграмма классов плагина

**3.3 Диаграмма пакетов**

Пакет (package) [5] – это инструмент группирования, который позволяет взять любую конструкцию UML объединить ее элементы в единицы высокого уровня. В основном пакеты служат для объединения классов в группы, но могут применяться для любой другой конструкции языка UML.

Рассмотрим диаграмму пакетов плагина на рисунке 3.5.



Рисунок 3.5 – Диаграмма пакетов плагина

Пакеты Builder и ParametersAndTools содержат классы бизнес логики плагина.

Пользовательский интерфейс содержится в пакете UI, при работе данный пакет использует бизнес логику приложения и графический интерфейс (пакет WinForms).

Пакет UnitTests хранит классы с юнит-тестами приложения, для работы используется вспомогательный пакет NUnit.

Пакет ParametersAndTools содержат классы для хранения параметров модели и удобной работы с ними.

Пакет Drawer содержит классы для отрисовки модели.

В результате изменений, внесенных заказчиком, диаграмма пакетов подверглась изменениям. В результате были добавлены новые классы и перечисления в пакет Parameters и переделан пакет UnitTests, а также был добавлен пакет для отрисовки модели на форме.

Рассмотрим изначальную версию диаграммы пакетов на рисунке 3.6

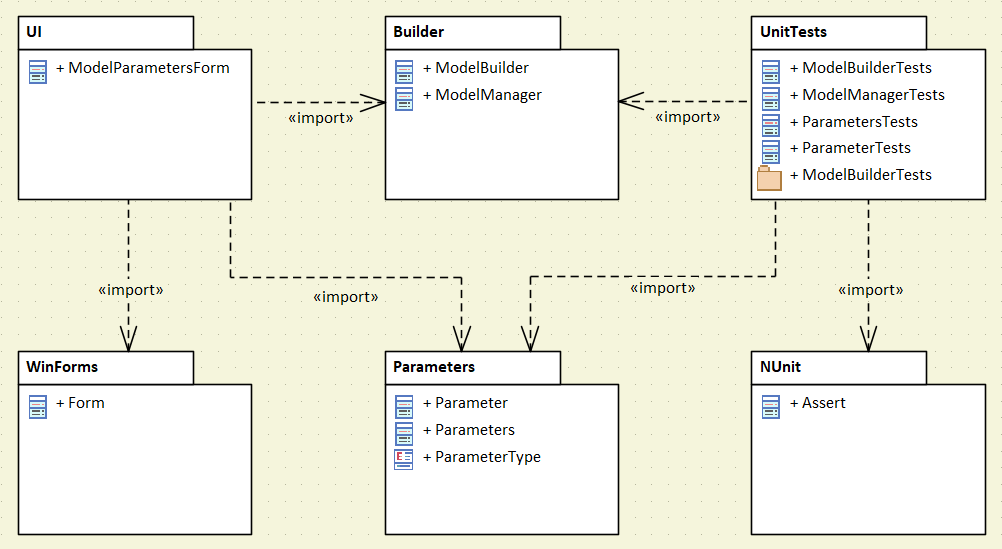


Рисунок 3.6– Изначальная диаграмма пакетов плагина

**4 Описание программы для пользователя**

Плагин состоит из одного диалогового окна с условным разделением на функциональные блоки: “Основная часть”, “Ножки”, “Спинка”, “Параметры пользователя”, “Превью модели”. На рисунке 4.1. изображен вид диалогового окна плагина.

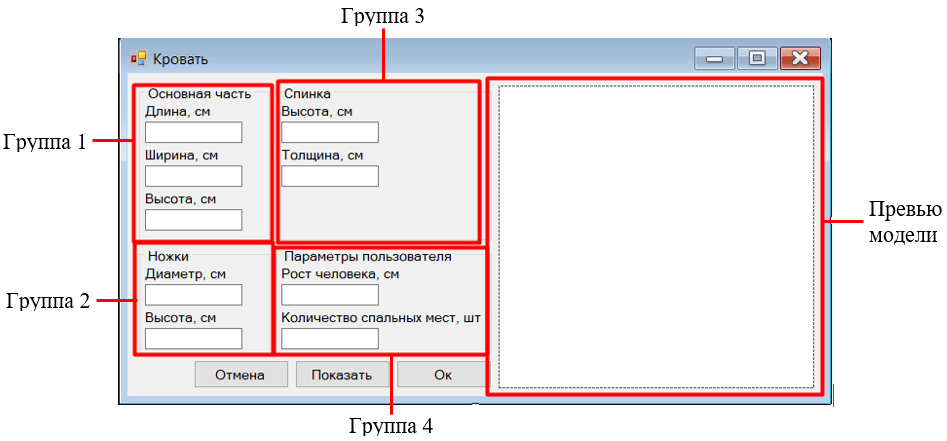


Рисунок 4.1 – Вид диалогового окна плагина

При запуске программы все параметры будут заполнены средними значениями, кроме параметров пользователя которые будут содержать константные начальные значения.

При попытке ввести в поля параметров кровати что-либо кроме цифр и точки/запятой, ввод блокируется и ввод нажатого символа не происходит. При вводе параметров пользователя допускается ввод только целых положительных чисел и работает оно по такому же принципу.

При попытке построить или показать превью модели кровати с некорректными параметрами, будет выведено модальное окно с ошибкой, и построение не будет выполнено, а превью не будет отрисовано.

Модальное окно, которое показывается при вводе некорректного значения изображено на рисунке 4.2.

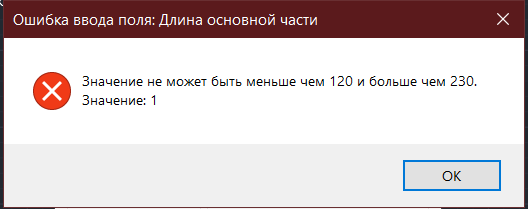


Рисунок 4.2 – Модальное окно с ошибкой

Блок “Основная часть” состоит из 3 полей для ввода значений, названий пунктов и единиц измерения.

Блок “Ножки” состоит из 2 полей для ввода значений, названий пунктов и единиц измерения.

Блок “Спинка” состоит из 2 полей для ввода значений, названий пунктов и единиц измерения.

Блок “Ножки” состоит из 2 полей для ввода значений, названий пунктов и единиц измерения.

Все вышеперечисленные блоки содержат поля значений, в которые можно ввести только положительные числа. Допускается ввод значений с плавающей точкой.

Блок “Параметры пользователя” состоит из 2 полей для ввода значений, названий пунктов и единиц измерения. Для ввода допускаются только положительные числа.

Блок “Превью модели” состоит из полотна, на котором отрисовывается превью модели кровати по заданным параметрам.

При изменении значения поля “Рост человека, см” происходит перерасчет длины основной части кровати.

При изменении значения поля “Количество спальных мест, шт” происходит перерасчет ширины основной части кровати.

Рассчитанные значения являются минимальными удовлетворяющими введенным параметрам и при дальнейшем вводе параметров меньше выставленных и попытке построить модель, будет выведено модальное окно с ошибкой.

Нажатие кнопки “Отмена” закрывает окно плагина и сбрасывает параметры к начальным значениям. Нажатие кнопки доступно с момента запуска программы.

Нажатие кнопки “Ок” производит построение детали с учетом заданных параметров в текущем документе программы AutoCAD, после чего производится закрытие окна плагина. Нажатие кнопки доступно с момента запуска программы.

Нажатие кнопки “Показать” производит отрисовку модели с учетом заданных параметров на главном диалоговом окне плагина. Нажатие кнопки доступно с момента запуска программы.

В результате изменений, внесенных заказчиком, интерфейс программы подвергся изменениям. В результате были добавлены новый блок и кнопка для управления им.

Рассмотрим изначальную версию диалогового окна плагина на рисунке 4.3

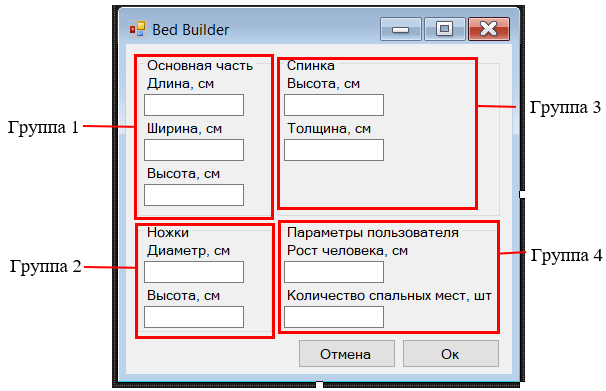


Рисунок 4.3 – Изначальная версия диалогового окна плагина.

**5 Тестирование программы**

Тестирование позволяет выявлять ошибки в программе в процессе разработки и при выпуске промежуточных и финальных версий приложения.

**5.1 Функциональное тестирование**

При функциональном тестировании [6] проверялась корректность работы плагина “Кровать”, а именно, соответствие полученного результата в виде трехмерной модели, с входными параметрами.

Проведем тестирование максимальных и минимальных параметров модели.

Модель с минимальными параметрами представлена на рисунке 5.1.

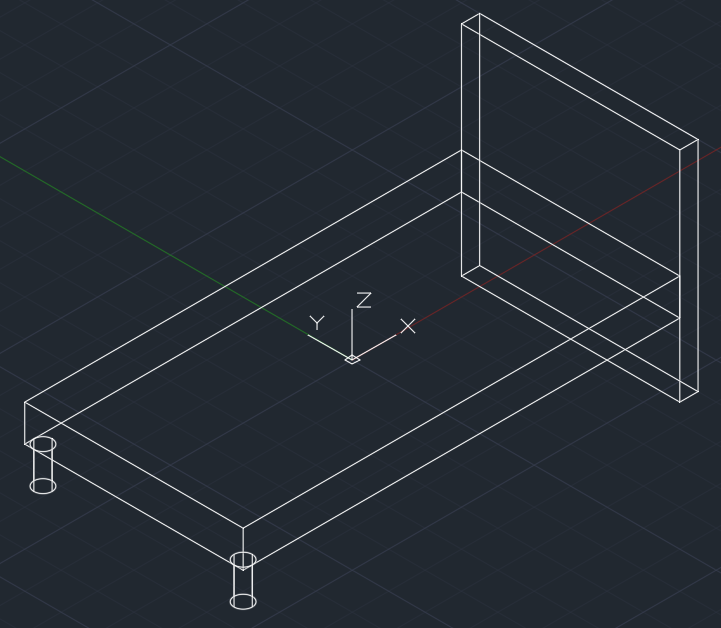


Рисунок 5.1 – Модель кровати с минимальными возможными параметрами

Модель с максимальными параметрами представлена на рисунке 5.2.

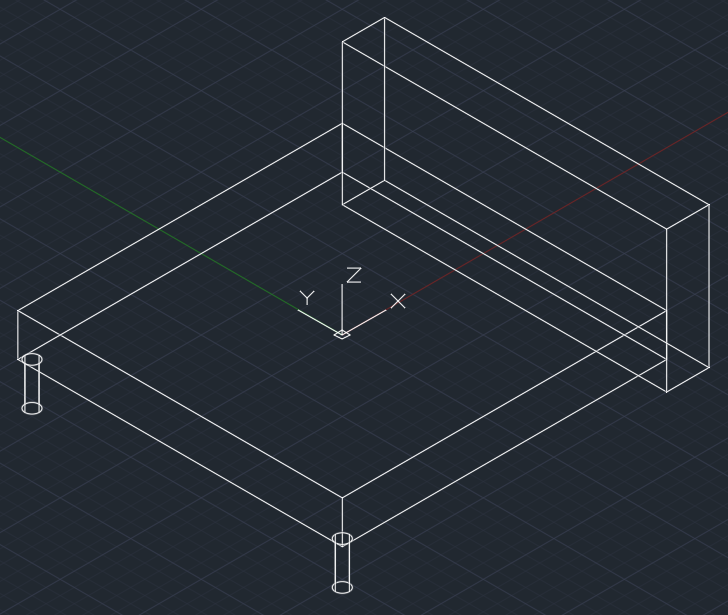


Рисунок 5.2 – Модель музыкальной колонки с максимальными возможными параметрами

Проведем тестирование построения модели кровати для одного человека среднего роста. Результаты тестирования представлены на рисунке 5.3.

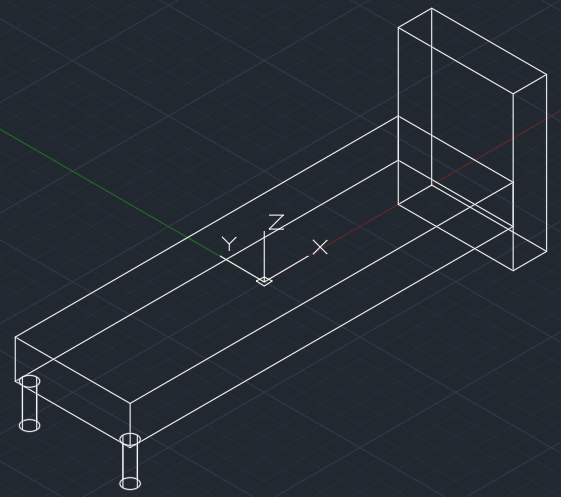


Рисунок 5.3 – Модель кровати, рассчитанная на одного человека ростом 170 см

Проведем тестирование построения кровати для двух человек ростом 170см. Результаты тестирования представлены на рисунке 5.4.

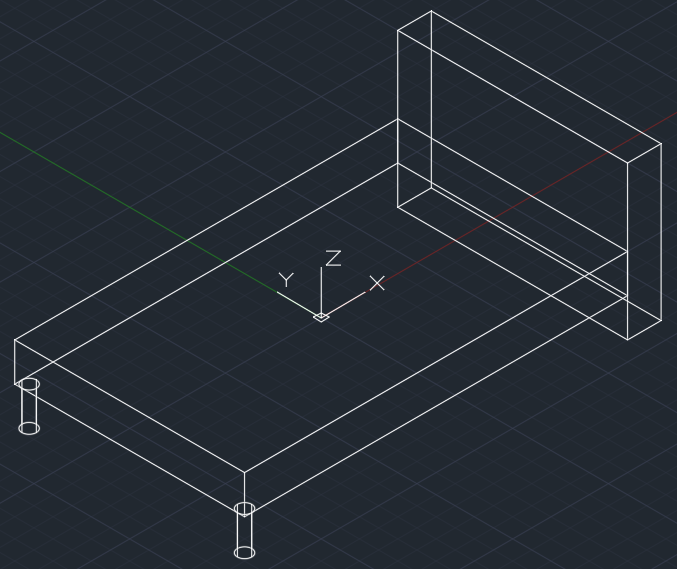


Рисунок 5.4 – Модель музыкальной колонки с 2 динамиками

**5.2 Модульное тестирование**

В целях проверки корректности работы методов и свойств классов при помощи обозревателя тестов Visual Studio было проведено модульное тестирование [7], проверялись открытые поля и методы, для этого были созданы тестовые классы:

* ParametersTests – класс тестирующий свойства и методы класса ParametersTests. Описание класса ParametersTests в Приложении А (Таблица А.1);
* ParameterTests – класс тестирующий свойства и методы класса ModelElements. Описание класса ParameterTests в Приложении А (Таблица А.2);

Диалоговое окно состояний запущенных тестов для классов ParametersTests и ParameterTests изображены на рисунке 5.5.

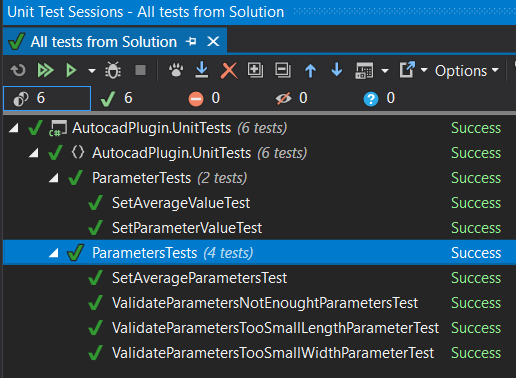


Рисунок 5.5 - Диалоговое окно состояний запущенных тестов для классов ParametersTests и ParameterTests

**5.3 Нагрузочное тестирование**

Нагрузочное тестирование – это тестирование производительности, сбор показателей и определение производительности и времени отклика программно-технической системы или устройства в ответ на внешний запрос с целью установления соответствия требованиям, предъявляемым к данной системе (устройству) [8].

Нагрузочное тестирование проводилось со следующими параметрами кровати:

* Длина основной части – 200 см;
* Ширина основной части – 60 см;
* Высота основной части – 20 см;
* Диаметр ножек – 7,5 см;
* Высота ножек – 20 см;
* Высота спинки – 80 cм;
* Толщина спинки – 17,5 см;
* Рост человека – 170 см;
* Количество спальных мест – 1шт.

После построения 91 детали программа завершилась со сбоем из-за нехватки памяти.

На рисунке 5.6 представлен график потребляемой оперативной памяти относительно количества построенных деталей.

Рисунок 5.6 – График потребляемой оперативной памяти относительно числа созданных деталей

График зависимости загрузки программой центрального процессора от количества построенных деталей изображен на рисунке 5.7.

Рисунок 5.7 – График зависимости загрузки программой центрального процессора от количества построенных деталей

По полученным графикам можно сделать вывод, что зависимость прямая, т.е. используемая оперативная память и загрузка центрального процессора при увеличении количества деталей увеличивается линейно.**Заключение**

В ходе выполнения лабораторных работ были изучены предметная область проектирования, предмет проектирования, аналоги предмета проектирования, API и на основании полученных данных были спроектированы архитектура и макет системы, создан плагин «Кровать», проведены модульные, функциональные и нагрузочные тесты.

**Список использованных источников**

1. AutoCAD for Mac и Windows | САПР | Autodesk [Электронный ресурс]. – URL: <https://www.autodesk.ru/products/autocad/overview> (дата обращения: 11.03.2020);
2. Visual Studio IDE, редактор кода, Azure DevOps и App Center - Visual Studio [Электронный ресурс]. – URL: <https://visualstudio.microsoft.com/ru/> (дата обращения: 04.05.2020);
3. Приложение “Оборудование: Металлоконструкции” [Электронный ресурс]. – URL: <https://kompas.ru/kompas-3D/application/machinery/steel-constructions-3d/> (дата обращения: 11.03.2020);

# Hilti BIM/CAD Library | AutoCAD | Autodesk App Store [Электронный ресурс]. – URL: https://apps.autodesk.com/ACD/ru/Detail/Index?id=4171643372823727407&appLang=en&os=Win32\_64 (дата обращения: 11.03.2020);

1. Мартин Фаулер UML Основы. Краткое руководство по стандартному языку объектного моделирования | Файловый архив для студентов. StudFiles. [Электронный ресурс]. – URL: <https://studfile.net/preview/6354103/> (дата обращения: 11.03.2020);
2. Функциональное тестирование [Электронный ресурс]. – URL: <https://daglab.ru/funkcionalnoe-testirovanie-programmnogo-obespechenija/> (дата обращения: 13.04.2020);
3. Модульное тестирование [Электронный ресурс]. – URL: <http://espressocode.top/unit-testing-software-testing/> (дата обращения: 13.04.2020);
4. Нагрузочное тестирование [Электронный ресурс]. – URL: <https://daglab.ru/nagruzochnoe-testirovanie-programmnogo-obespechenija/> (дата обращения: 13.04.2020);

**Приложение А**

(Справочное)

В таблицах приложения для обозначения модификаторов доступа полей приняты следующие условные знаки:

* “ – ” обозначение private (закрытого) поля;
* “ + ” обозначение public (открытого) поля;

Описание полей и методов используемых для проверки тестовых случаев класса Parameters представлено в таблице А.1

|  |  |
| --- | --- |
| Наименование | Описание |
| − \_parameters | Поле хранит объект Parameters |
| − CreateParameters() | Метод заполняет \_parameters тестовыми значениями |
| + SetAverageParametersTest() | Метод для проверки корректности выставляемых значений параметров методом SetAverageParameters() |
| + ValidateParametersTest() | Метод для проверки валидации корректных параметров методом ValidateParameters() |
| + ValidateParametersNotEnoughtParametersTest() | Метод для проверки вызова исключения, в случае если параметров не хватает для построения модели. |
| + ValidateParametersTooSmallWidthParameterTest() | Метод для проверки вызова исключения, в случае если ширина кровати не удовлетворяет количеству спальных мест. |

Продолжение таблицы А.1.

|  |  |
| --- | --- |
| + ValidateParametersTooSmallLengthParameterTest() | в случае если длина кровати не удовлетворяет росту человека. |

Таблица А.1 – Описание полей и методов класса ParametersTests

Описание полей и методов используемых для проверки тестовых случаев класса Parameter представлено в таблице А.2

|  |  |
| --- | --- |
| Наименование | Описание |
| + SetAverageValueTest() | Метод для проверки функции установки среднего значения параметра. |
| + SetParameterValueTest() | Метод для проверки работы свойства значения параметра. Проверяет |
| + SetParameterValueOutOfRangeTest() | Метод для проверки вызова исключения при попытке внести значение вне установленного интервала. |