Министерство науки и высшего образования Российской Федерации Федеральное государственное бюджетное образовательное учреждение высшего образования

ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ СИСТЕМ УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ (ТУСУР)

Кафедра компьютерных систем в управлении и проектировании (КСУП)

РАЗРАБОТКА ПЛАГИНА «Вешалка для одежды»   
ДЛЯ «КОМПАС-3D V20**»**

Пояснительная записка

к лабораторной работе по дисциплине

«Основы разработки САПР»

Выполнил: Студент гр. 589-3

Марченко Н.О.

« » 20 г.

Принял: к.т.н., доцент каф. КСУП

Калентьев А.А.

« » 20 г.

Томск 2022

# Реферат

Лабораторная работа 27 с., 15 рис., 8 источников.

Ключевые слова: КОМПАС 3D V20, VISUAL STUDIO, C#, ПЛАГИН, САПР.

Целью данной работы является разработка плагина для автоматизации проектирования модели вешалки для одежды.

Результатом работы является плагин, осуществляющий построение модели вешалки по заданным пользователям параметрам.

Пояснительная записка выполнена в текстовом редакторе Google Docx.

# Оглавление

[1 Введение 4](#_Toc122637165)

[2 Постановка и анализ задачи 5](#_Toc122637166)

[2.1 Описание предмета проектирования 6](#_Toc122637167)

[2.2 Выбор инструментов и средств реализации 7](#_Toc122637168)

[2.3 Назначение плагина 7](#_Toc122637169)

[3 Обзор аналогов 8](#_Toc122637170)

[4 Описание реализации 9](#_Toc122637171)

[4.1 Диаграмма классов 10](#_Toc122637172)

[5 Описание программы для пользователя 16](#_Toc122637173)

[6 Тестирование плагина 19](#_Toc122637174)

[7 Заключение 26](#_Toc122637175)

[Список источников 27](#_Toc122637176)

# 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. Она позволяет значительно повысить точность расчетов, выбрать наилучшие варианты для реализации на основе строгого математического анализа всех или большинства вариантов проекта с оценкой технических, технологических и экономических характеристик производства и эксплуатации проектируемого объекта, значительно повысить качество конструкторской документации, существенно сократить сроки проектирования и передачи конструкторской документации в производство, эффективнее использовать технологическое оборудование с программным управлением [1].

Таким образом, целью данной работы является разработка плагина, автоматизирующего построение модели «вешалка для одежды» для системы автоматизированного проектирования Компас 3D V20 с помощью интегрированной среды разработки Visual Studio 2022 Сommunity. [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.

# Постановка и анализ задачи

В соответствии с техническим заданием требовалось разработать плагин, который на основе входных параметров, интегрируя с системой Компас 3D V20, строит модель «Вешалка для одежды» [3].

# Описание предмета проектирования

Вешалка для одежды – это вешалка с несколькими крючками для хранения верхней и другой одежды.

На рисунке 2.1 представлена схема вешалки с параметрами:

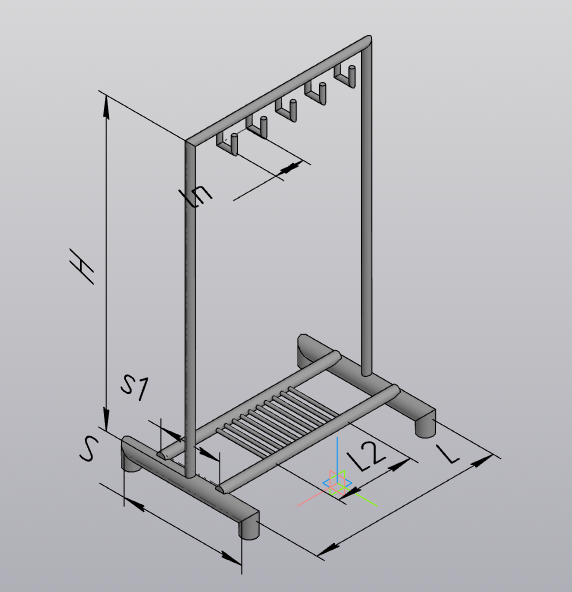


Рисунок 2.1 – Схема вешалки с параметрами

Данная модель имеет 6 основных параметров :

1. Высота вешалки H (от 1000 до 1300 мм);
2. Ширина опоры s1 (от 200 до 300 мм);
3. Количество крючков n (от 2 до 7);
4. Расстояние между крючками ln (от 50 до 100 мм);
5. Ширина вешалки L (от (n\*ln+100) до (n\*ln+150) мм);
6. Длина опоры вешалки S (от 400 до (s1+250) мм);
7. Длина подставки для обуви L2(от 100 до (L-80) мм);

# Выбор инструментов и средств реализации

На основе требований к техническому заданию программа выполнена на языке программирования C# в среде Microsoft Visual Studio 2022 с использованием .NET Framework 4.7.2 [4].

Инструментом тестирования и создания модульных тестов был выбран тестовый фреймворк NUnit [5] версии 3.13.2.

# Назначение плагина

Назначение разрабатываемого плагина обусловлено быстрым построением вешалки, рассчитанных на различное количество крючков. Благодаря данному расширению, проектировщики могут наглядно рассмотреть спроектированную модель, при необходимости перестроить под необходимые им параметры.

API КОМПАС3D – это ориентированные на прикладного программиста инструментальные средства разработки приложений (библиотек конструктивов, прикладных САПР) на базе системы КОМПАС.

# Обзор аналогов

Базис-Мебельщик

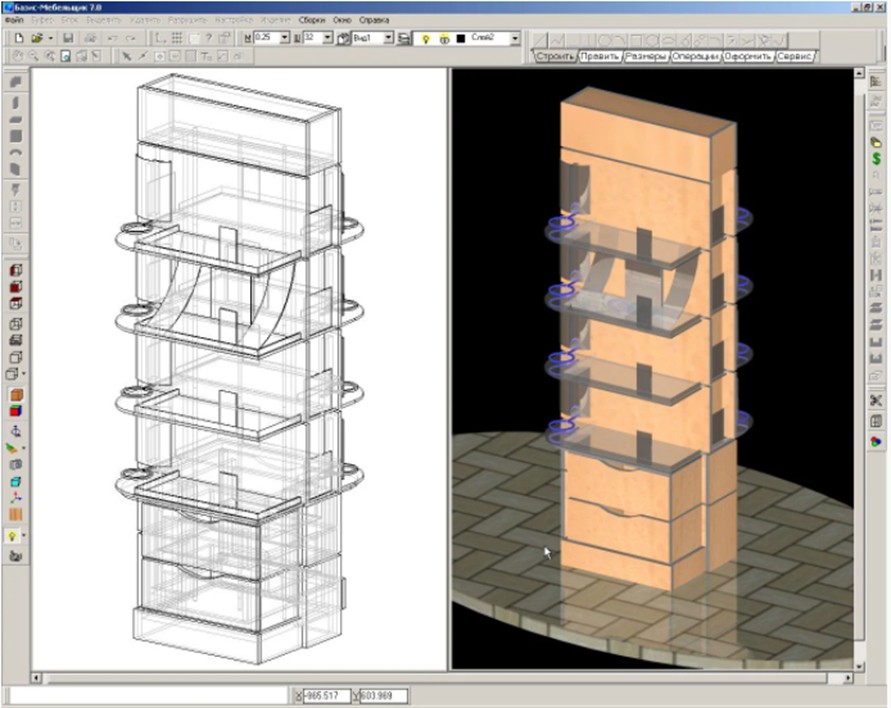
Базис-Мебельщик помогает проектировать сложную корпусную мебель массового производства. Благодаря мощному и быстрому движку вы можете в кратчайшие сроки рассчитать необходимые данные, сформулировать схемы и раскрой, создавать схемы и чертежи и запускать просмотр в режиме 3D. Точная панель управления позволяет создавать даже мелкие элементы с высочайшей деталировкой и подготавливать фотореалистичные изображения для презентации.[3]

Рисунок 3.1 – Базис-Мебельщик Библиотека помогает в решении следующих задач:

* автоматизация сложных повторяющихся процессов;
* графический редактор профессионального уровня для планов высочайшей точности;
* запуск трехмерного и двухмерного виртуального просмотра для более наглядной работы;

# Описание реализации

Для графического описания абстрактной модели проекта, а также пользовательского взаимодействия (сценарии действия) использован стандарт UML.

UML язык графического описания для объектного моделирования в области разработки программного обеспечения. UML является языком широкого профиля, это – открытый стандарт, использующий графические обозначения для создания абстрактной модели системы, называемой UML – моделью. UML был создан для определения, визуализации, проектирования и документирования, в основном, программных систем. UML не является языком программирования, но на основании UML возможна генерация кода и наоборот. [6]

При использовании UML была построена диаграмма классов.

# Диаграмма классов

Диаграмма классов определяет типы классов системы и различного рода статические связи, которые существуют между ними. На диаграммах классов изображаются также атрибуты классов, операции классов и ограничения, которые накладываются на связи между классами.[7] Целью создания диаграммы классов является графическое представление статической структуры элементов системы. Диаграмма классов плагина представлена на рисунке 4.1.

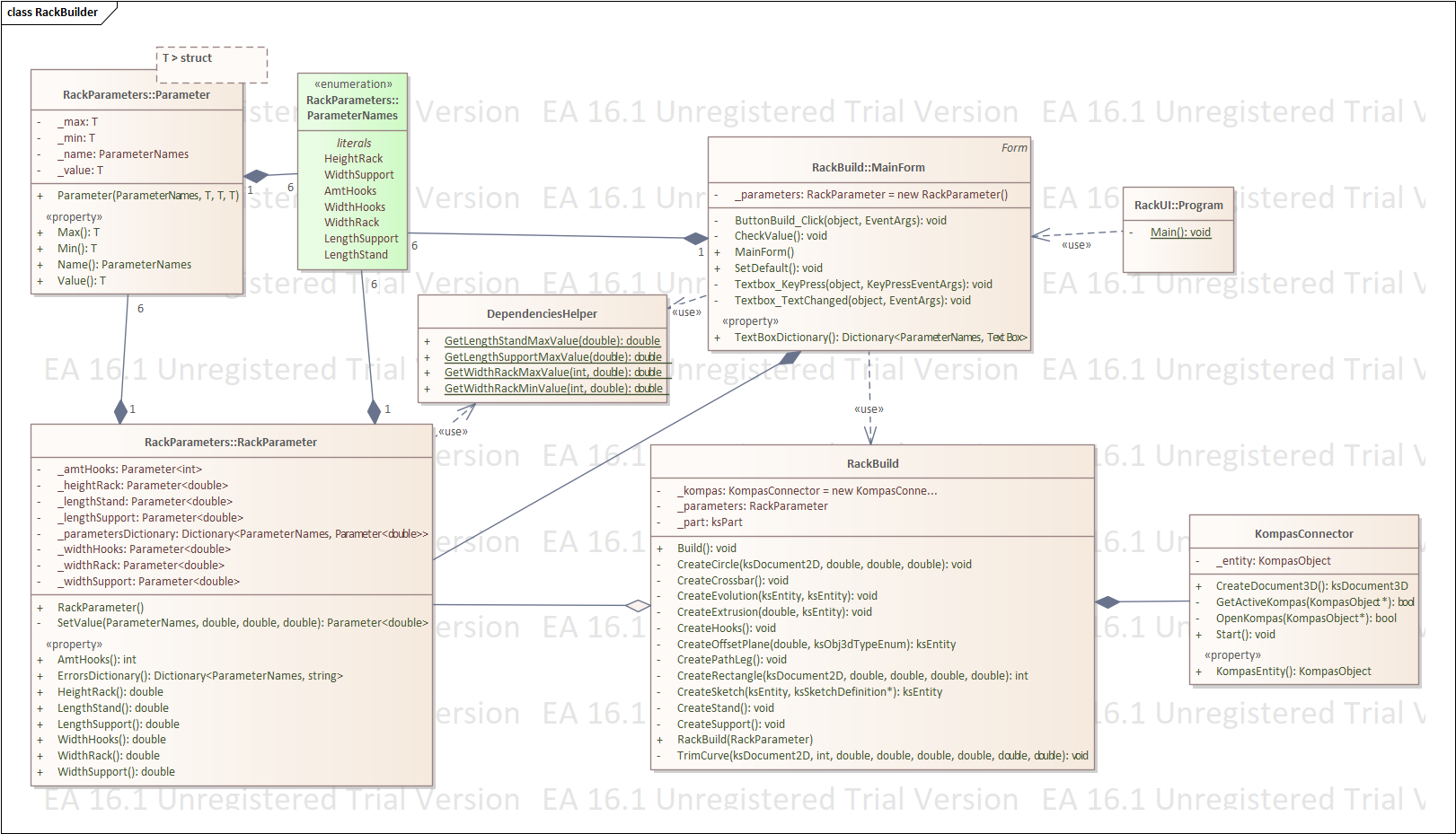


Рисунок 4.1 – Диаграмма классов

В таблицах 4.1 - 4.4 представлены описания полей, свойств и методов.

Таблица 4.1 – Описание класса Parameter

|  |  |  |
| --- | --- | --- |
| Название | Тип данных | Описание |
| \_min | T | Закрытое поле, минимальное значение параметра |
| \_max | T | Закрытое поле, максимальное значение параметра |
| \_value | T | Закрытое поле, значение параметра |
| \_name | ParameterNames | Закрытое поле, имя параметра |
| Name | ParameterNames | Свойство, устанавливает или возвращает имя |
| Value | T | Свойство, устанавливает или возвращает значение |
| Min | T | Свойство, устанавливает или возвращает минимальное значение |
| Max | T | Свойство, устанавливает или возвращает максимальное значение |
| Parameter (ParameterNames, T, T, T) |  | Конструктор класса |

Таблица 4.2 – Описание класса DependenciesHelper

|  |  |  |
| --- | --- | --- |
| Название | Тип данных | Описание |
| GetWidthRackMaxValue(int,double) | double | Считает и возвращает максимальное значение ширины вешалки |
| GetWidthRackMaxValue(int,double) | double | Считает и возвращает минимальное значение ширины вешалки |
| GetLengthSupportMaxValue(double) | double | Считает и возвращает максимальное значение длины опоры вешалки |
| GetLengthStandMaxValue(double) | double | Считает и возвращает максимальное значение длины подставки вешалки |

Таблица 4.3 – Описание класса KompasConnector

|  |  |  |
| --- | --- | --- |
| Название | Тип данных | Описание |
| \_entity | KompasObject | Закрытое поле, экземпляр API КОМПАС 3D |
| KompasEntity | KompasObject | Свойство, возвращает экземпляр API КОМПАС 3D |
| Start() | void | Устанавливает связь с КОМПАС 3D |
| GetActiveKompas (KompasObject\*) | bool | Подключается к открытому КОМПАС 3D |
| OpenKompas (KompasObject\*) | bool | Открывает КОМПАС 3D |
| CreateDocument3D() | ksDocument3D | Создает документ в КОМПАС 3D |

Таблица 4.4 – Описание класса RackParameter

|  |  |  |
| --- | --- | --- |
| Название | Тип данных | Описание |
| \_heightRack | Parameter<double> | Закрытое поле, высота вешалки |
| \_widthSupport | Parameter<double> | Закрытое поле, ширина вешалки |
| \_amtHooks | Parameter<int> | Закрытое поле, количество крючков |
| \_widthHooks | Parameter<double> | Закрытое поле, расстояние между крючками |
| \_widthRack | Parameter<double> | Закрытое поле, ширина вешалки |
| \_lengthSupport | Parameter<double> | Закрытое поле, длина опоры вешалки |
| \_parametersDictionary | Dictionary  <ParameterNames, Parameter<double>> | Закрытое поле, словарь типа “Имя параметра - поле параметра” |
| \_lengthStand | Parameter<double> | Закрытое поле, длина подставки под обувь |

Продолжение таблицы 4.4

|  |  |  |
| --- | --- | --- |
| Название | Тип данных | Описание |
| ErrorsDictionary | Dictionary  <ParameterNames, Parameter<double>> | Cловарь типа “Имя параметра  - текст ошибки” |
| HeigthRack | double | Свойство, устанавливает или возвращает высоту вешалки |
| WidthSupport | double | Свойство, устанавливает или возвращает ширину опоры вешалки |
| AmtHooks | int | Свойство, устанавливает или возвращает количество крючков |
| WidthHooks | double | Свойство, устанавливает или возвращает расстояние между крючками |
| WidthRack | double | Свойство, устанавливает или возвращает ширину вешалки |
| LengthSupport | double | Свойство, устанавливает или возвращает длину опоры вешалки |
| SetValue (PrameterNames, double, double, double) | Parameter<double> | Устанавливает значения параметров |
| RackParameter() |  | Конструктор класса |
| LengthStand | double | Свойство, устанавливает или возвращает длину подставки для обуви |

Таблица 4.5 – Описание класса RackBuild

|  |  |  |
| --- | --- | --- |
| Название | Тип данных | Описание |
| \_parameters | RackParamete r | Закрытое поле. Содержит размеры модели |
| \_part | ksPart | Закрытое поле. Объект детали |
| \_kompas | KompasConne ctor | Закрытое поле. Устанавливает связь с КОМПАС-3D |
| Build() | void | Строит деталь |
| CreatePathLeg() | void | Строит основание |
| CreateCrossbar() | void | Строит верхнюю перекладину |
| CreateSeparators() | void | Строит верхнюю перекладину |
| CreateSupport() | void | Строит опору |
| CreateSketch(ksEntity, ksSketchDefinition\*) | ksEntity | Создаёт эскиз |
| CreateOffsetPlane(double, ksObj3dTypeEnum) | ksEntity | Создаёт смещённую плоскость |
| CreateRectangle(ksDocumen t2D, double, double, double, double) | int | Создаёт прямоугольник |
| CreateCircle(ksDocument2D  , double, double, double) | void | Создаёт окружность |
| TrimCurve(ksDocument2D, int, double, double, double, double, double, double) | void | Усекает указанную кривую |
| CreateEvolution(ksEntity, ksEntity) | void | Операция «Элемент по траектории» |
| CreateExtrusion(double, ksEntity) | void | Операция «Выдавливание» |
| RackBuild(RackParameter) |  | Конструктор класса |
| CreateStand() | void | Строит подставку |

# Описание программы для пользователя

Плагин представляет собой пользовательскую форму с полями для ввода 6 геометрических параметров: «Высота вешалки», «Ширина опоры», «Количество разделителей», «расстояние между крючками», «Ширина вешалки», «Длина опоры вешалки». Ниже располагается кнопка для запуска построения. Справа находится изображение, демонстрирующее размеры.

Макет пользовательского интерфейса представлен на рисунке 5.1.

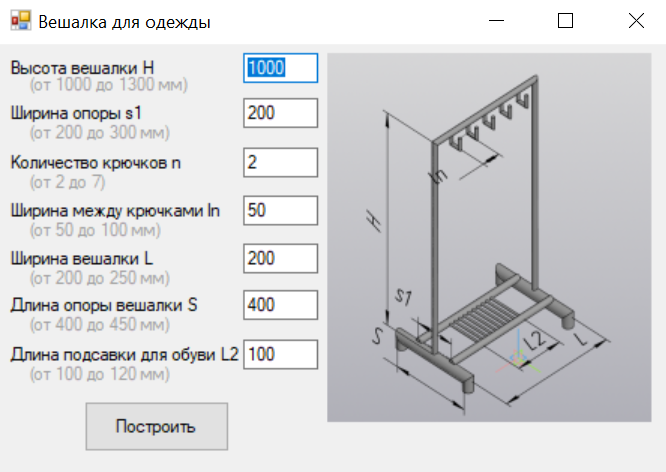


Рисунок 5.1 – Макет пользовательского интерфейса

При вводе параметров, выходящих за диапазон, соответствующее поле подсвечивается красным цветом (рис. 5.2). При нажатии на кнопку «Построить запускается построение модели. При попытке запустить построение с неверными параметрами будет выведено соответствующее сообщение. Пример сообщения представлен на рисунке 5.3.

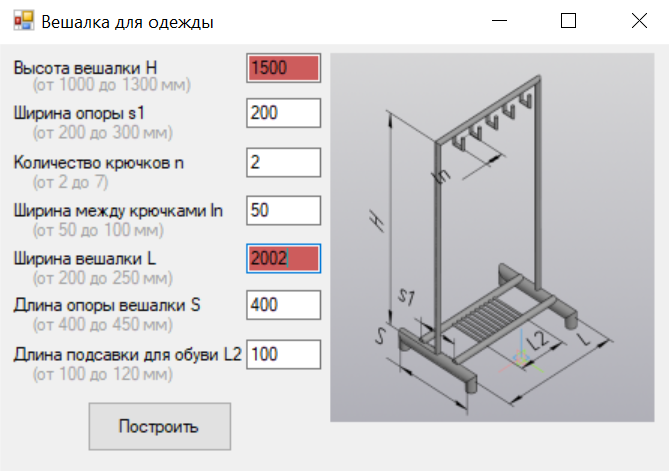


Рисунок 5.2 – Пример ввода некорректных данных

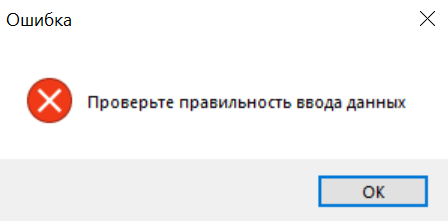


Рисунок 5.3 – Сообщение о вводе некорректных данных

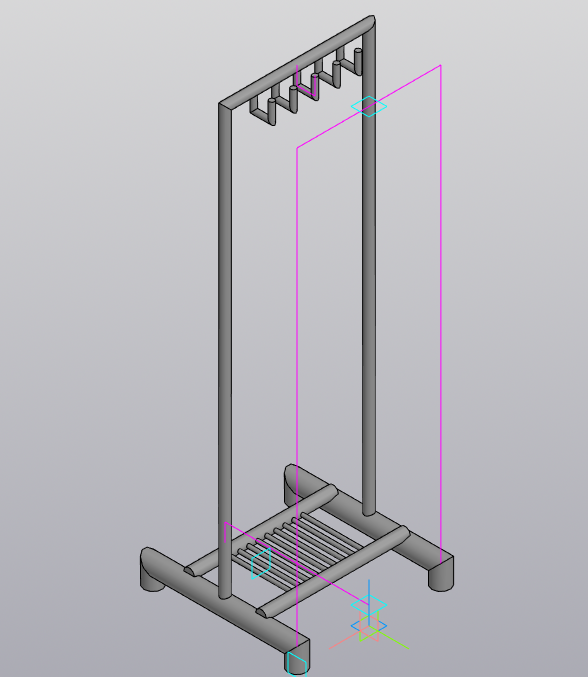


Рисунок 5.4 – Построенная модель Вешалки

# Тестирование плагина

Далее было произведено юнит-тестирование плагина.

Юнит-тестирование (англ. «unit-testing», или модульное тестирование) — тестирование отдельного элемента изолированно от остальной системы. Относительно парадигмы объектно-ориентированного программирования системой является вся программа, а отдельным элементом — класс или его метод. Юнит-тестирование предназначено для проверки правильности работы отдельно взятого элемента. Чтобы исключить из результатов тестирования влияние потенциальных ошибок других элементов, тестируемый элемент должен быть максимально изолирован, то есть не использовать объекты и методы других классов [8].

Ниже, на рисунке 6.1, изображен список тестов класса «RackParametr».

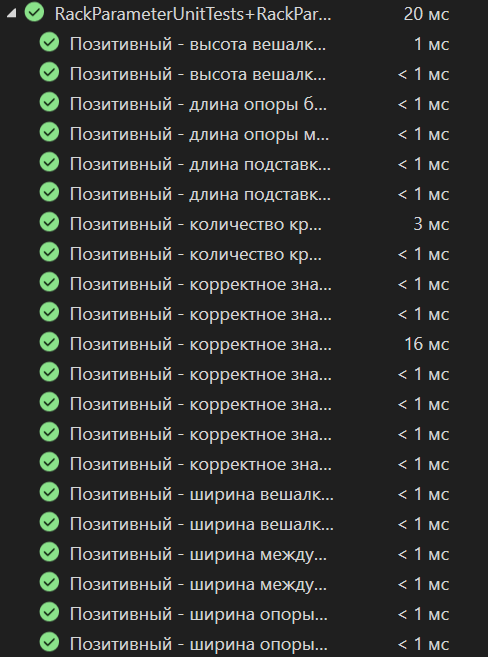


Рисунок 6.1 – Список тестов класса «RackParametr»

На рисунке 6.2 изображен список тестов класса «Parameter».

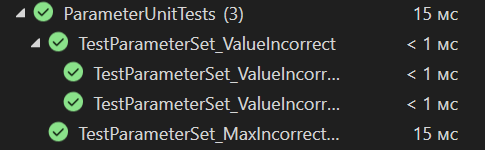


Рисунок 6.2 – Список тестов класса «Parameter»

Результат покрытия классов «Parameter» и «RackParameter» тестами приведен на рисунке 6.3.

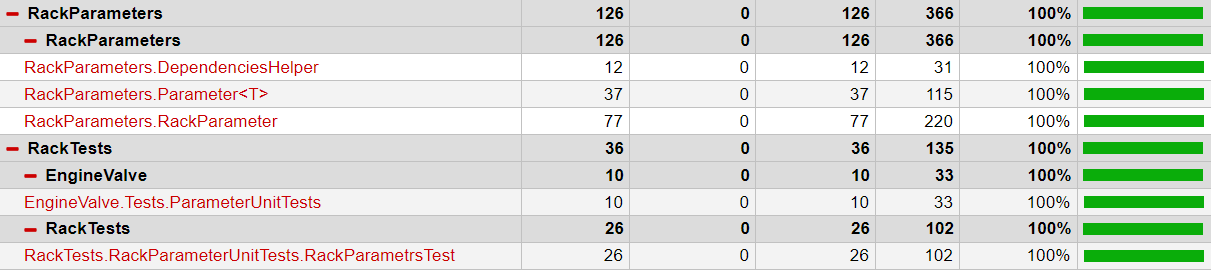


Рисунок 6.3 – Степень покрытия тестами бизнес-логики плагина

Далее, было выполнено нагрузочное тестирование.

Поскольку потребление памяти приложением статично, а скорость построения зависит от скорости работы «КОМПАС-3D» и от конфигурации компьютера, измерению в стресс-тесте будет подвергаться потребление памяти при многократных построениях. Стоит заранее отметить, что программой не допускается построение нескольких моделей одновременно, а только по очереди.

Конфигурация компьютера, на котором выполнялся тест:

1. X86-совместимый CPU Intel Core i5-8250U @ 1.60GHz.
2. 8 ГБ DDR4 ОЗУ на частоте 2133 МГц в одноканальном режиме.
3. Графический процессор Intel UHD Graphics 620.
4. ОС Windows 10 x64.

Нагрузочное тестирование изначально предполагало построение 1000 моделей и удержание их открытыми в «КОМПАС-3D». Тест завершился досрочно на построении восьмидесяти второй модели, так как «КОМПАС-3D» аварийно завершил свою работу без вывода информации об ошибке.

Потребление ОЗУ самой программой составляет в среднем 20 МБ, в то время как потребление памяти пакетом «КОМПАС-3D» сильно увеличивается со временем.

График зависимости потребления ОЗУ «КОМПАС-3D» от количества построенных моделей представлен на рисунке 6.4.

Рисунок 6.4 – Зависимость потребления ОЗУ от количества моделей при стандартных параметрах

Скорость построения напрямую зависит от свободного времени CPU и пропускной способности памяти. Косвенная зависимость от количества памяти заключается в том, что при её нехватке операционная система будет вынуждена предоставлять пакету «КОМПАС-3D» области файла подкачки, что существенно снизит его производительность. Такой ситуации не удалось достичь на вышеуказанной конфигурации по причине избытка памяти и преждевременного завершения работы «КОМПАС-3D».

Рисунок 6.5 – Зависимость затрат времени от количества моделей при стандартных параметрах

На нижеприведённом графике легко видеть, что количество времени, затрачиваемого на построение, растёт с количеством открытых моделей.

Пиковое потребление ОЗУ было при удержании восьмидесяти одной модели, оно составило 6,7 ГБ

При построении вешалок с минимальными параметрами программа смогла построить восемьдесят девять моделей, после аварийно завершилась. График потребления ОЗУ и времени построения представлены на рисунках 6.6 и 6.7

Рисунок 6.6 – Зависимость потребления ОЗУ от количества моделей при минимальных параметрах

Рисунок 6.7 – Зависимость затрат времени от количества моделей при минимальных параметрах

При построении вешалок с максимальными параметрами программа смогла построить семьдесят одну модель, после чего аварийно завершилась. График потребления ОЗУ и времени построения представлены на рисунках 6.8 и 6.9.

Рисунок 6.8 – Зависимость потребления ОЗУ от количества моделей при максимальных параметрах

Рисунок 6.9 – Зависимость затрат времени от количества при максимальных параметрах

Исходя из результатов тестирования, можно сделать вывод, что программа выполняет заявленную функциональность в полном объёме. Использовать её для построения сразу нескольких моделей и удержания их в памяти не представляется возможным, так как среда «КОМПАС-3D» V.20 не выдерживает нагрузки подобного рода. Рекомендуется строить модель вешалки для разового применения.

# Заключение

В ходе выполнения лабораторных работ были изучены предметная область проектирования, предмет проектирования, аналоги предмета проектирования, API, функциональное и нагрузочное тестирование и на основании полученных данных были спроектированы UML диаграммы классов, разработан плагин для создания 3D моделей «Вешалка для одежды» в САПР КОМПАС 3D V20 и проведено функциональное и нагрузочное тестирование плагина.

# Список источников

1. Система трёхмерного моделирования КОМПАС-3D [электронный ресурс]. – URL: <https://ascon.ru/products/7/review/>(дата обращения 11.12.2022).
2. Базовые интерфейсы API системы КОМПАС [электронный ресурс]. – URL: <https://it.wikireading.ru/23741>(дата обращения 11.12.2022).
3. Конструкторы мебели для самостоятельного проектирования: обзор лучшего софта. [Электронный ресурс]. – URL: <https://amssoft.ru/repair/programmy-dlya-proektirovaniya-mebeli.php> (дата обращения 13.12.2022).
4. Фаулер М. UML. Основы. 3-е издание / М.Фаулер. – 3-е изд., пер. с англ. – СПб.: Символ-Плюс, 2019. – 192 с. (дата обращения 14.12.2022).
5. Visual Studio [Электронный ресурс]. – URL: https://visualstudio.microsoft.com/ru/ (дата обращения: 10.12.2022).
6. Пресс-формы 3D express [Электронный ресурс]. - Режим доступа: https://kompas.ru/kompas-3d/application/machinery/press-formy-3d-express/ Дата обращения (27.10.2022)
7. UML. [Электронный ресурс]. – Режим доступа: <http://www.uml.org/>(дата обращения 10.12.2022).
8. Новые технологии в программировании: учебное пособие / А.А.Калентьев, Д.В.Гарайс, А.Е.Горяинов – Томск: Эль Контент, 2018.—176 с.