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)

Выполнил:

студент гр. 589-3

\_\_\_\_\_\_\_\_\_\_\_\_\_\_А.М. Потлог

«\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2022 г.

Руководитель:

к.т.н., доцент каф. КСУП

\_\_\_\_\_\_\_\_\_\_\_\_\_\_А.А. Калентьев

«\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2022 г.

**Оглавление**

[1 Введение 3](#_Toc90061417)

[2 Постановка и анализ задачи 4](#_Toc90061418)

[2.1 Описание предмета проектирования 5](#_Toc90061419)

[2.2 Выбор инструментов и средств реализации 6](#_Toc90061420)

[2.3 Назначение плагина 6](#_Toc90061421)

[3 Обзор аналогов 7](#_Toc90061422)

[4 Описание реализации 8](#_Toc90061423)

[4.1 Диаграмма классов 8](#_Toc90061424)

[5 Описание программы для пользователя 11](#_Toc90061425)

[6 Тестирование программы 14](#_Toc90061426)

[6.1 Функциональное тестирование 14](#_Toc90061427)

[6.2 Модульное тестирование 17](#_Toc90061428)

[6.3 Нагрузочное тестирование 18](#_Toc90061429)

[Заключение 21](#_Toc90061430)

[Список использованных источников 22](#_Toc90061431)

# 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. 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 «Стеллаж» для системы автоматизированного проектирования Inventor 2022 и Компас 3D 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.

# 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 САПР Kompas 3D, строит модель Стеллаж. [3] Также нужно было сделать поддержку работы плагина для Inventor 2022 (дополнительное задание).[4] Необходимо чтобы плагин позволял задавать параметры по умолчанию, а также изменять входные параметры стеллажа, такие как:

* длина стеллажа ***L***;
* высота стеллажа ***H***;
* ширина стеллажа ***W***;
* расстояние между полками стеллажа ***h1***;
* ширина стойки ***w1***;
* ширина полки ***w2***;
* расстояние верхнего отступа ***h2***;
* расстояние нижнего отступа ***h3***;

# 2.1 Описание предмета проектирования

Стеллаж — складское специализированное мебельное оборудование для хранения предметов/грузов, состоящее из металлических стоек, балок и поперечных балок либо многоярусных настилов (полок), закрепленных на балках, либо состоящее из закреплённых на стойках консолей (консольные стеллажи).[5]

На рисунке 2.1 представлен чертеж стеллажа.

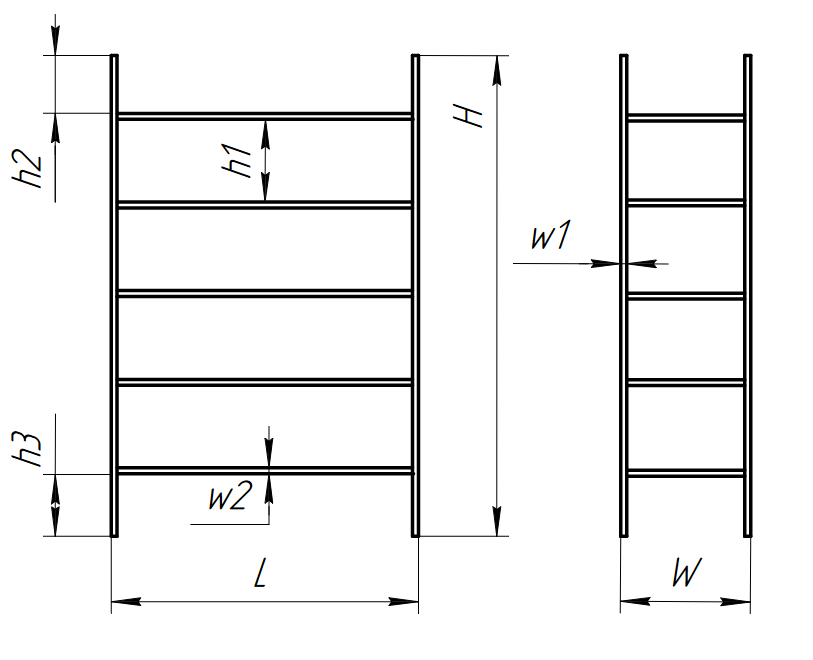


Рисунок 2.1 – Чертеж стеллажа

Параметры стеллажа:

* длина стеллажа ***L*** (минимум – 1000мм, максимум – 2000мм);
* высота стеллажа ***H*** (минимум – 1000мм, максимум – 3000мм);
* ширина стеллажа ***W*** (минимум – 500мм, максимум – 1000мм);
* расстояние между полками стеллажа ***h1*** (не больше высоты стеллажа ***H*** и не меньше ширины полки ***w2,*** минимум – ***w2***, максимум – ***H***. Количество полок будет определяться автоматически по формуле, количество полок = (***H*** – ***h2*** - ***h3***)/***h1***);
* ширина стойки ***w1*** (минимум – 20мм, максимум – 30мм);
* ширина полки ***w2*** (минимум – 20мм, максимум – 30мм);
* расстояние верхнего отступа ***h2*** (не больше расстояния между полками стеллажа ***h1***, минимум – 0, максимум – ***h1***);
* расстояние нижнего отступа ***h3*** (не больше расстояния между полками стеллажа ***h1***, минимум – 0, максимум – ***h1***);

# 2.2 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 core 6 [2], библиотека «Inventor» [6] для основных операций в САПР Inventor, библиотеки для Kompas 3D [7].

Инструментом тестирования и создания модульных тестов был выбран тестовый фреймворк NUnit [8] версии 3.13.3.

Для реализации пользовательского интерфейса использовалась система для построения настольные приложения WPF [9].

# 2.3 Назначение плагина

Назначение разрабатываемого плагина обусловлено быстрым моделированием стеллажей разных типов. Благодаря данному расширению, мастера по стеллажам могут наглядно рассмотреть спроектированную модель, при необходимости перестроить под необходимые им параметры.

# 3 Обзор аналогов

**Плагин**

Конфигуратор стеллажного оборудования с 2D-планом и 3D-конфигуратором стеллажей для B2B-площадки международной компании ARNEG[10].

Конфигуратор представляет собой трехуровневую систему – проект, группа, секция, на каждом уровне используется соответствующий функционал. Работа с проектами аналогичная традиционным заказам или заявкам - создание и редактирование названия проекта, текстовые комментарии, привязка к пользователю и филиалу.

Реализована возможность изменения цвета деталей (цвет меняется на всем проекте для определенной группы элементов - например, цвет всех полок или цвет всех стоек в проекте), некоторые элементы имеют дополнительный выбор материала - хром. Для задних стенок присутствует возможность замены на перфорированные. Возможно изменение числа и типа полок стеллажа, возможное количество полок определяет высота стоек. Для полок можно менять высоту ценникодержателя либо убрать его совсем. Также доступна установка делителем и ограничителей на полки.

Интерфейс программы показан на рисунке 3.1.

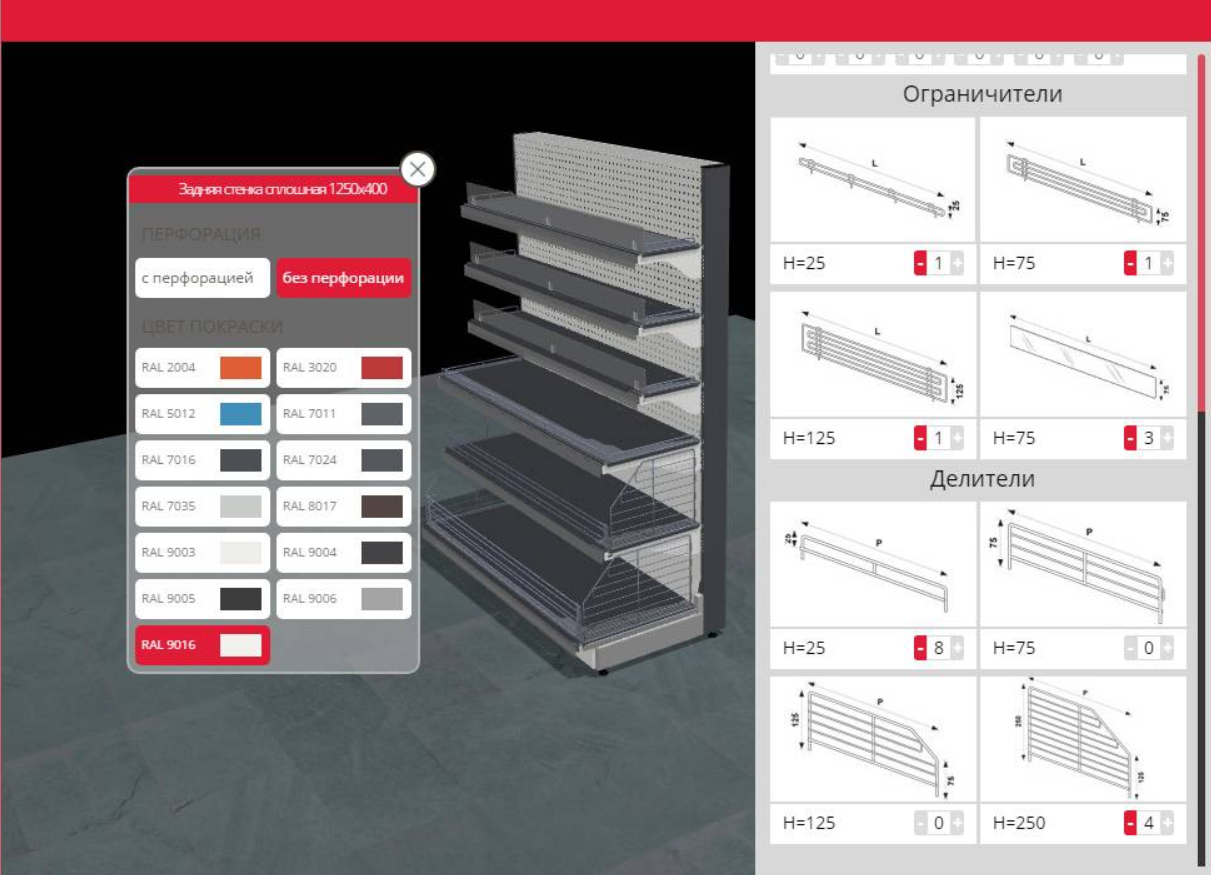


Рисунок 3.1 — Интерфейс программы для моделирования стеллажей ARNEG

# 4 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.[11]

При использовании UML были простроена диаграмма классов.

# 4.1 Диаграмма классов

Диаграмма классов – структурная диаграмма языка моделирования UML, демонстрирующая общую структуру иерархии классов системы, их коопераций, атрибутов (полей), методов, интерфейсов и взаимосвязей между ними.[11]

На рисунке 4.1 представлена диаграмма классов.

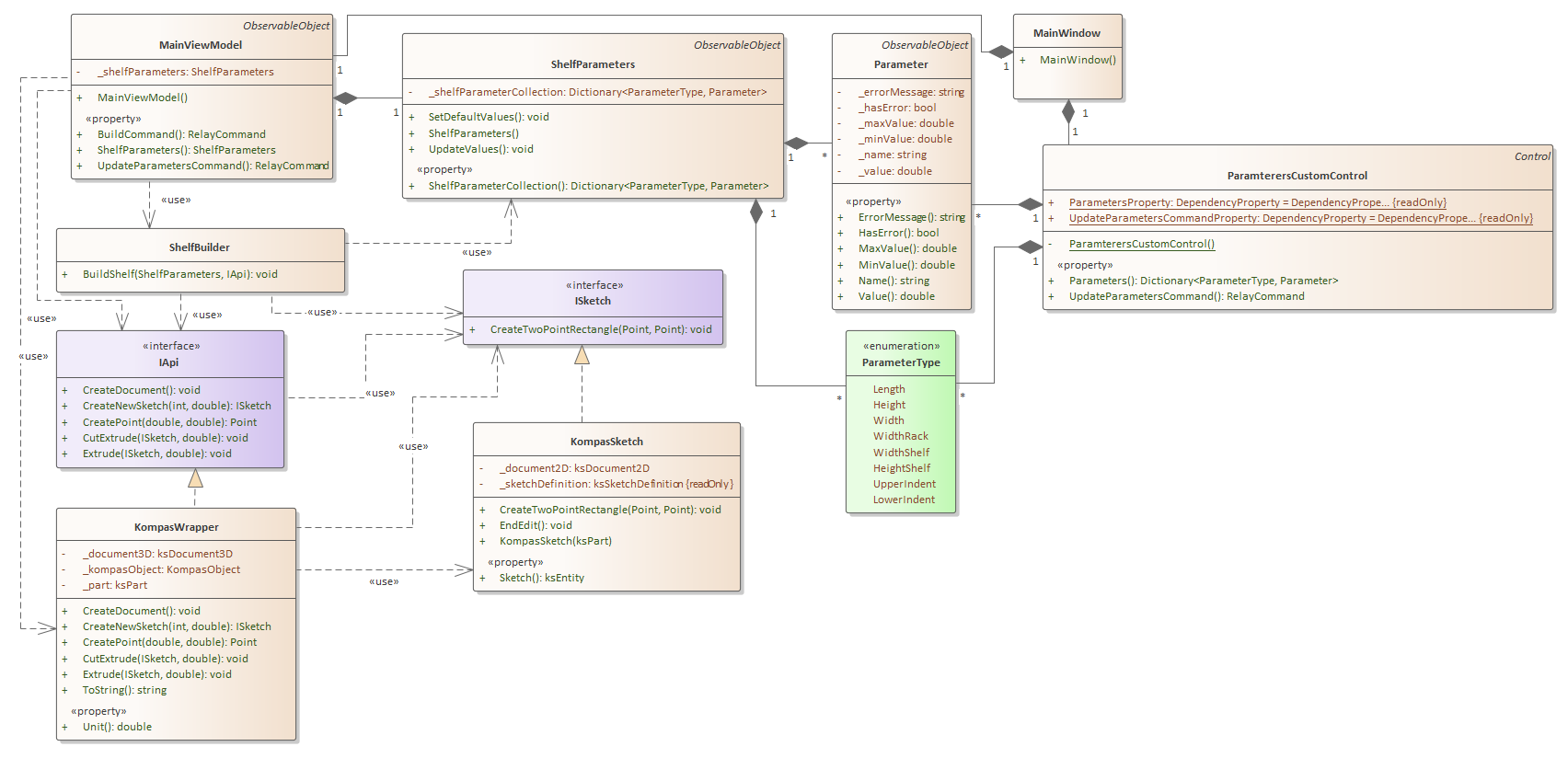


Рисунок 4.1 – Изначальная диаграмма классов

Главное окно связывается с MainViewModel для управления данными. MainViewModel содержит экземпляр класса ShelfParameters.

ParametersControl отвечает за отображение параметров модели, которые определены как словарь в классе Shelfparamteres, так же данный элемент принимает команду для обновления зависимых параметров.

Будет использоваться дополнительная библиотека **Microsoft.Xaml.Behaviors.Wpf** для привязки команды, к триггерам.

В итоговом проекте созданы следующие классы и методы, которые отображены на итоговой диаграмме классов (рисунок 4.2).

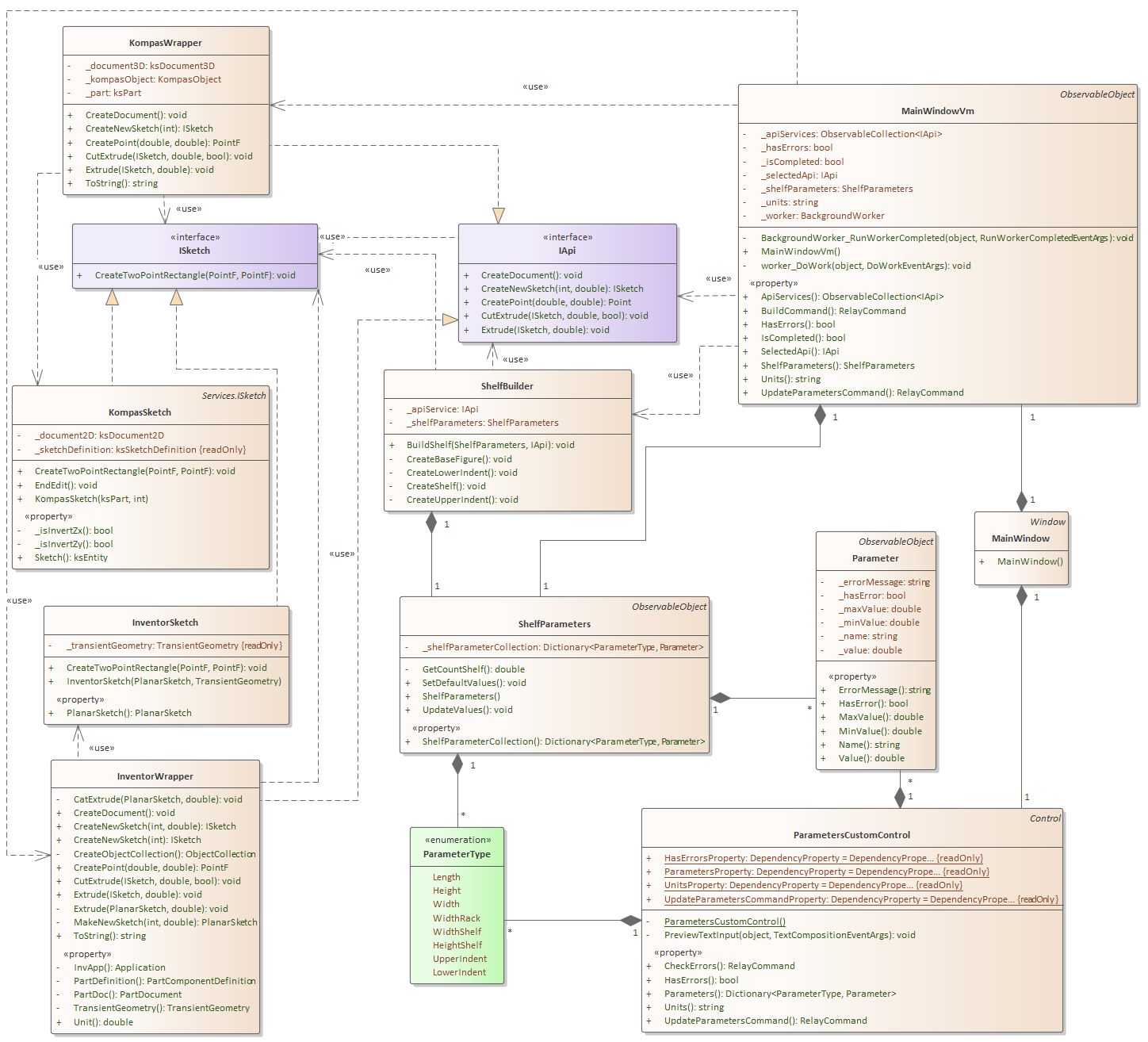


Рисунок 4.2 – Итоговая диаграмма классов

В класс MainViewModel, были добавлены вспомогательные поля и методы background worker, для блокировки кнопки «Построить», во время построения детали.

В окно приложения был добавлен combobox, для выбора Inventor или Компас 3D.

Так как были реализованы интерфейсы ISketch и IApi, при выполнении дополнительного задания, нужно было написать только два новых класса InventorSketch и InventorWrapper, которые позволили добавить поддержку построения детали в Inventor, не меняя класс ShelfBuilder.

# 5 Описание программы для пользователя

Макет пользовательского интерфейса представляет собой форму для ввода параметров стеллажа. Построение модели осуществляется путем нажатия на кнопку «Построить». При попытке ввода недопустимых символов, они не будут вводиться в строку (если необходимо ввести цифры, то невозможно будет ввести другие символы).

На рисунке 5.1 представлен макет пользовательского интерфейса.

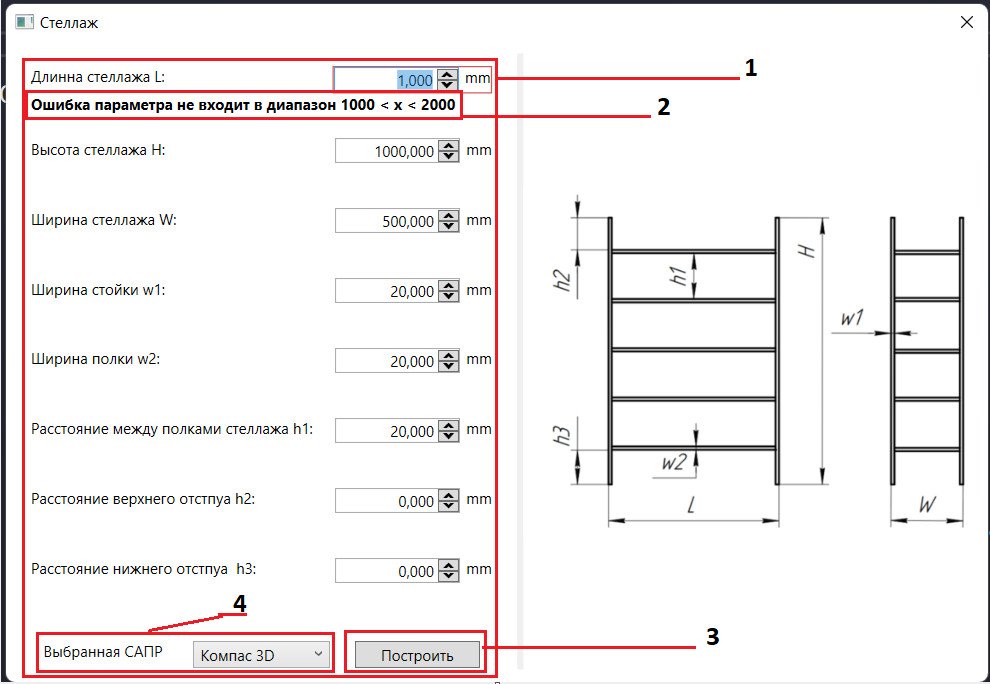


Рисунок 5.1 – Макет пользовательского интерфейса

Описание пользовательского интерфейса:

1 – поле ввода параметров стеллажа;

2 – описание ошибки введенного параметра;

3 – кнопка для построения модели стеллажа;

4 – combobox, для выбора САПР;

При вводе некорректных значений пользователь увидит сообщение об ошибке, которое отображает допустимое значение параметра.

Во время построения детали, кнопка «построить» становится неактивной

После нажатия на кнопку «Построить» при введенных некорректных значениях, появится окно, приведенный на рисунке 5.2.

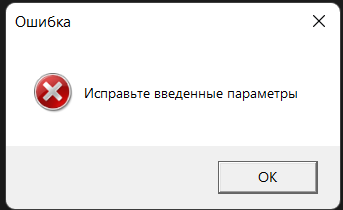


Рисунок 5.2 — Окно ошибки

После ввода необходимых параметров, построить деталь в САПР Inventor 2022 можно с помощью кнопки «Построить». Стеллаж, построенный по заданным параметрам в САПР Kompas 3D и Inventor 2022, представлен на рисунках 5.3-5.4.

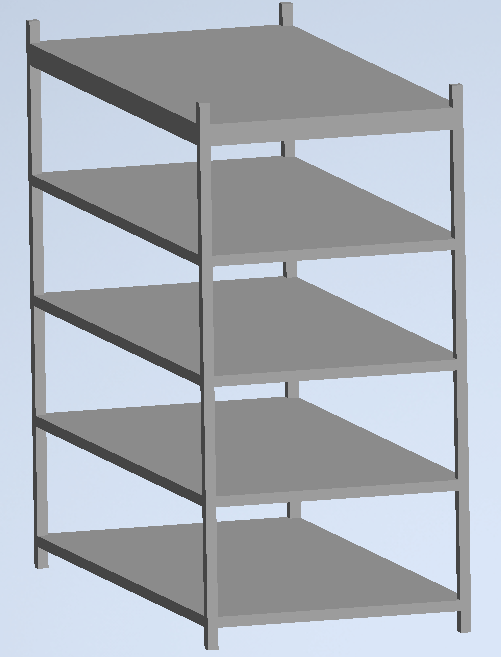


Рисунок 5.3 — Стеллаж, построенный по заданным параметрам в САПР Inventor 2022

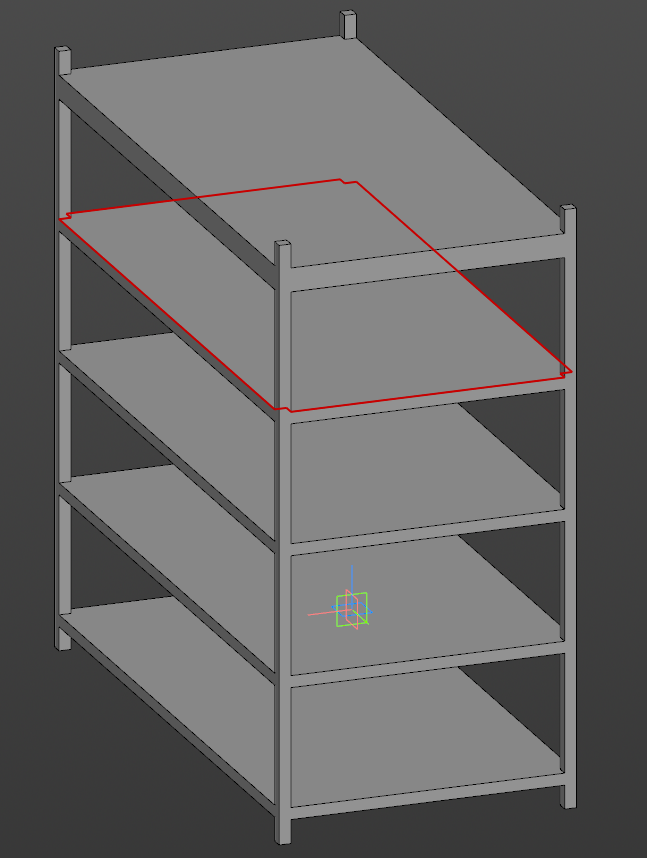


Рисунок 5.4 — Стеллаж, построенный по заданным параметрам в САПР Kompas 3D

# 6 Тестирование программы

Тестирование позволяет убедиться в работоспособности программы, выявлять ошибки при изменении какого-либо функционала.

# 6.1 Функциональное тестирование

При функциональном тестировании проверялось корректность работы плагина «Стеллаж», а именно, соответствие полученного результата в виде трехмерной модели, с входными параметрами. [12]

Проведено тестирование максимальных и минимальных параметров модели.

На рисунках 6.1 и 6.2 представлены проверки размеров модели с минимальным введенными параметрами в САПР Inventor 2022 и Kompas 3D.

Минимальные параметры стеллажа:

Длинна 1000 мм;

Высота 1000 мм;

Ширина 500 мм;

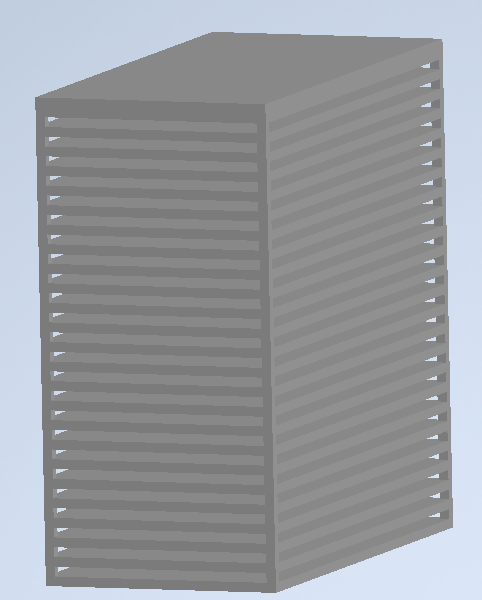
Ширина стойки 20 мм;

Ширина полки 20 мм;

Расстояние между полками 20 мм;

Расстояние верхнего отступа 0 мм;

Расстояние нижнего отступа 0 мм;

  
Рисунок 6.1 – Модель с минимальными введенными параметрами в Inventor 2022

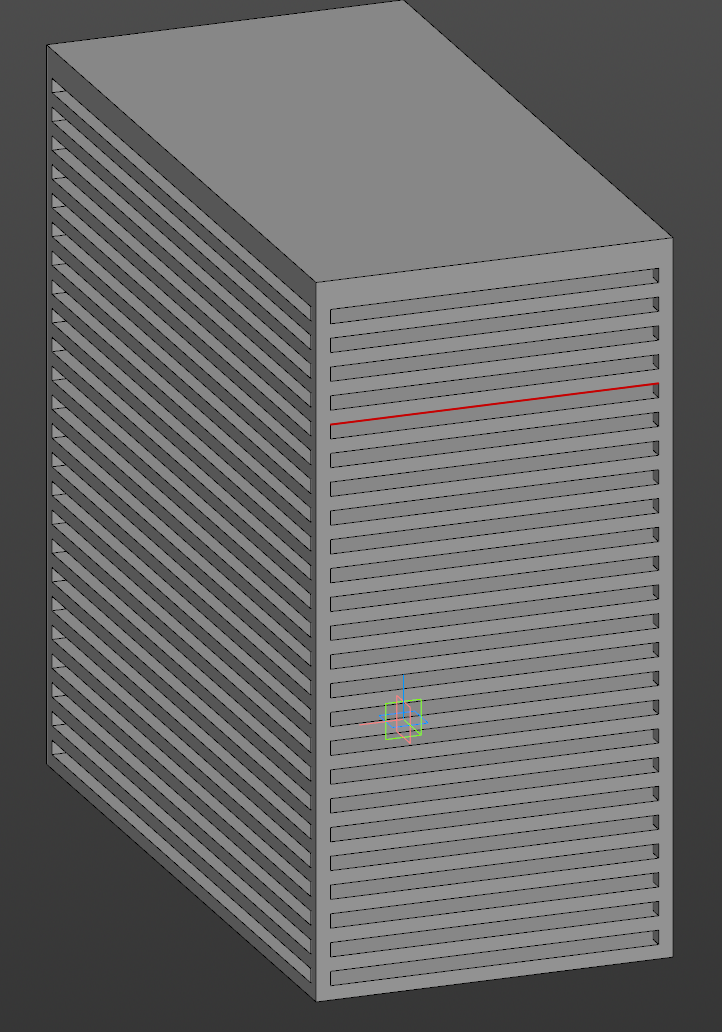


Рисунок 6.2 – Модель с минимальными введенными параметрами в Kompas 3D

Ниже на рисунках 6.3 и 6.4 представлены проверки размеров модели с максимальными введенными параметрами в САПР Inventor 2022 и Kompas 3D

Максимальные параметры стеллажа:

Длинна 1000 мм;

Высота 1000 мм;

Ширина 500 мм;

Ширина стойки 20 мм;

Ширина полки 20 мм;

Расстояние между полками 20 мм;

Расстояние верхнего отступа 0 мм;

Расстояние нижнего отступа 0 мм;



Рисунок 6.3 — Модель с максимально веденными параметрами в Inventor 2022

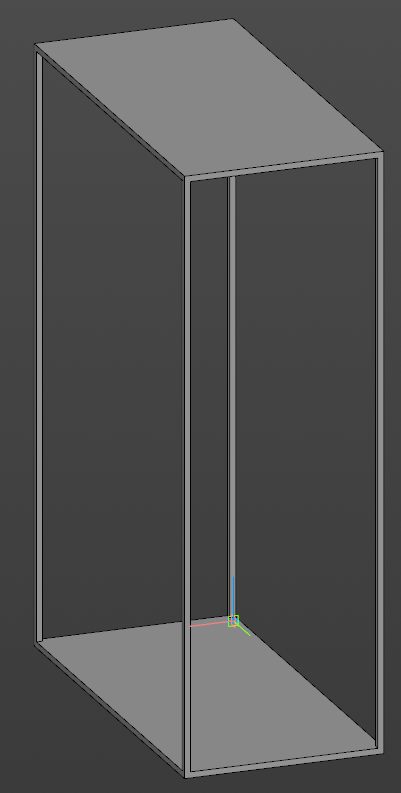
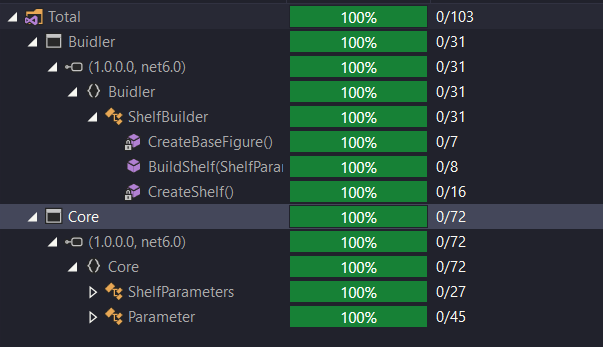


Рисунок 6.4 — Модель с максимально веденными параметрами в Kompas 3D

# 6.2 Модульное тестирование

В целях проверки корректности работы методов и свойств классов при помощи тестового фреймворка NUnit версии 3.13 проведено модульное тестирование [13], проверялись открытые поля и методы. На рисунке 6.5 представлено тестирование классов проектов: Core, Builder. Степень покрытия проектов — сто процентов.

  
Рисунок 6.5 – Тестирование классов

Перечень тестов с их описание представлено в таблицах 6.1-6.6

Тестирование класса ShelfParameters.

Таблица 6.1 – Тестовые случаи, метод TestGetValue\_CorrectGetValue.

|  |  |  |
| --- | --- | --- |
| Тестируемое свойство | Параметр | Описание |
| Length | 1000 | Проверка корректного получения значения свойства Length |
| Height | 1000 | Проверка корректного получения значения свойства Height |
| Width | 500 | Проверка корректного получения значения свойства Width |
| WidthRack | 20 | Проверка корректного получения значения свойства WidthRack |

Продолжение таблицы 6.1.

|  |  |  |
| --- | --- | --- |
| WidthShelf | 20 | Проверка корректного получения значения свойств WidthShelf |
| HeightShelf | 20 | Проверка корректного получения значения свойства HeightShelf |
| UpperIndent | 0 | Проверка корректного получения значения свойства UpperIndent |
| LowerIndent | 0 | Проверка корректного получения значения свойства LowerIndent |

Таблица 6.2 – Тестовые случаи, метод TestSetValue\_CorrectSetValue.

|  |  |  |
| --- | --- | --- |
| Тестируемое свойство | Параметр | Описание |
| Length | 1000 | Проверка корректной передачи значения свойства Length |
| Height | 1000 | Проверка корректной передачи значения свойства Height |
| Width | 500 | Проверка корректной передачи значения свойства Width |
| WidthRack | 20 | Проверка корректной передачи значения свойства WidthRack |

Продолжение таблицы 6.2.

|  |  |  |
| --- | --- | --- |
| WidthShelf | 20 | Проверка корректной передачи значения  свойства WidthShelf |
| HeightShelf | 20 | Проверка корректной передачи значения свойства HeightShelf |
| UpperIndent | 0 | Проверка корректной передачи значения свойства UpperIndent |
| LowerIndent | 0 | Проверка корректной передачи значения свойства LowerIndent |

Таблица 6.3 – Тестовые случаи, метод TestSetValue\_IncorrectSetValue.

|  |  |  |
| --- | --- | --- |
| Тестируемое свойство | Параметр | Описание |
| Length | 9 | Проверка некорректной передачи значения свойства Length. Меньше минимального. |
| Length | 10000 | Проверка некорректной передачи значения свойства Length. Больше максимального. |
| Height | 9 | Проверка некорректной передачи значения свойства Heigth. Меньше минимального. |

Продолжение таблицы 6.3.

|  |  |  |
| --- | --- | --- |
| Height | 10000 | Проверка некорректной передачи значения свойства Height. Меньше минимального. |
| Width | 9 | Проверка некорректной передачи значения свойства Width  Меньше минимального. |
| Width | 10000 | Проверка некорректной передачи значения свойства Width. Больше максимального. |
| WidthRack | 9 | Проверка некорректной передачи значения свойства WidthRack. Меньше минимального. |
| WidthRack | 10000 | Проверка некорректной передачи значения свойства WidthRack. Больше максимального. |

Продолжение таблицы 6.3.

|  |  |  |
| --- | --- | --- |
| WidthShelf | 9 | Проверка некорректной передачи значения свойства WidthShelf. Меньше минимального. |
| WidthShelf | 10000 | Проверка некорректной передачи значения свойства WidthShelf. Больше максимального. |
| HeightShelf | 9 | Проверка некорректной передачи значения свойства HeightShelf. Меньше минимального. |
| HeightShelf | 10000 | Проверка некорректной передачи значения свойства HeightShelf. Больше максимального. |
| UpperIndent | -9 | Проверка некорректной передачи значения свойства UpperIndent.  Меньше минимального. |
| UpperIndent | 10000 | Проверка некорректной передачи значения свойства UpperIndent. Больше максимального. |

Продолжение таблицы 6.3.

|  |  |  |
| --- | --- | --- |
| LowerIndent | -9 | Проверка некорректной передачи значения свойства LowerIndent.  Меньше минимального. |
| LowerIndent | 10000 | Проверка некорректной передачи значения свойства LowerIndent. Больше максимального. |

Таблица 6.4 - Метод TestSetCorrect\_DefaultParameters.

|  |  |
| --- | --- |
| Название теста | Описание |
| TestUpdateValues\_CorrectSetValue | Проверка независимых параметров |

Таблица 6.5 - Тестирование класса Parameter.

|  |  |
| --- | --- |
| Название теста | Описание |
| TestNegativeValueToZero\_InCorrectValue | Проверка ввода негативного значения |
| TestGetValueName\_CorrectValue | Проверка получения Name |
| TestGetValueErrorMessage\_CorrectValue | Проверка получения ErrorMessage |
| TestBuildShelf\_DoesNotThrowException | Тестирование построения корректного стеллажа |

Таблица 6.6 - Тестирование класса ShelfBuilder.

|  |  |
| --- | --- |
| Название теста | Описание |
| TestBuildShelf\_DoesNotThrowException | Тестирование построения корректного стеллажа |

# 6.3 Нагрузочное тестирование

В целях проверки производительности работы плагина, было проведено нагрузочное тестирование [14]. Тестирование производилось на ПК со следующей конфигурацией:

* ЦП Ryzen 5 3600 3.6ГГц;
* 32 ГБ ОЗУ;
* графический процессор объемом памяти 6 ГБ.

На рисунке 6.6 для проведения нагрузочного, с каждым успешным построением фигуры производилась запись результатов в текстовые файлы «logInventor.txt» и «logKompas3D.txt».

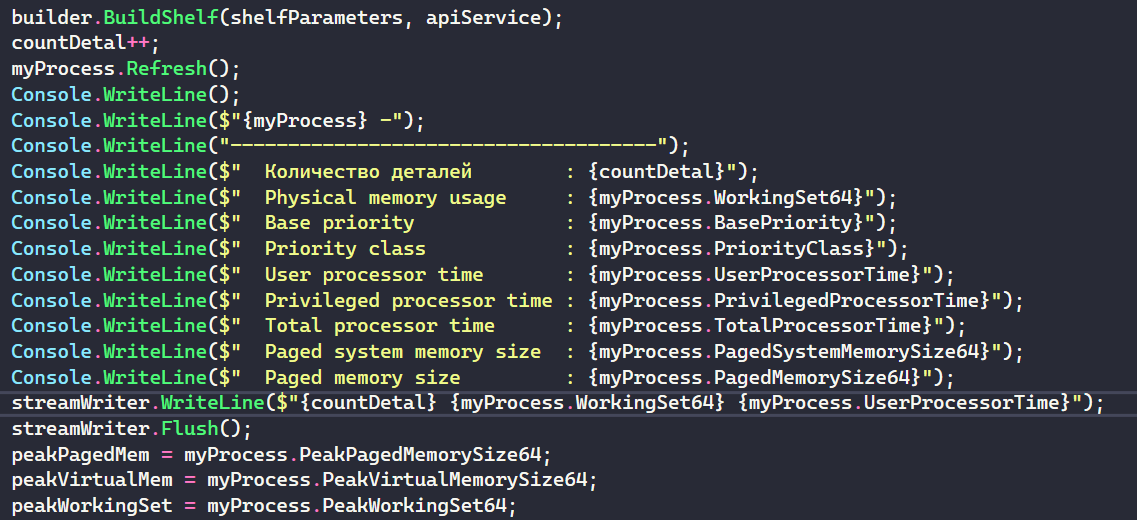


Рисунок 6.6 – Зацикливание перестроения фигуры

На графике, изображенном на рисунке 6.7 ось «Х» - количество построенных деталей, ось «Y» - количество потребляемой оперативной памяти. На графике, изображенном на рисунке 6.8 в текущей главе, ось «X» – время в минутах, ось «Y» – количество построенных деталей. На протяжении всех тестов (продолжительностью до сбоя Kompas 3D и Inventor) общая загруженность процессора была в пределах 22 процентов, для тестирования на САПР Inventor. Для САПР Kompas 3D общая загруженность процессора была в пределах 20 процентов.

На рисунке 6.7 представлено тестирование зацикленного перестроения фигуры:

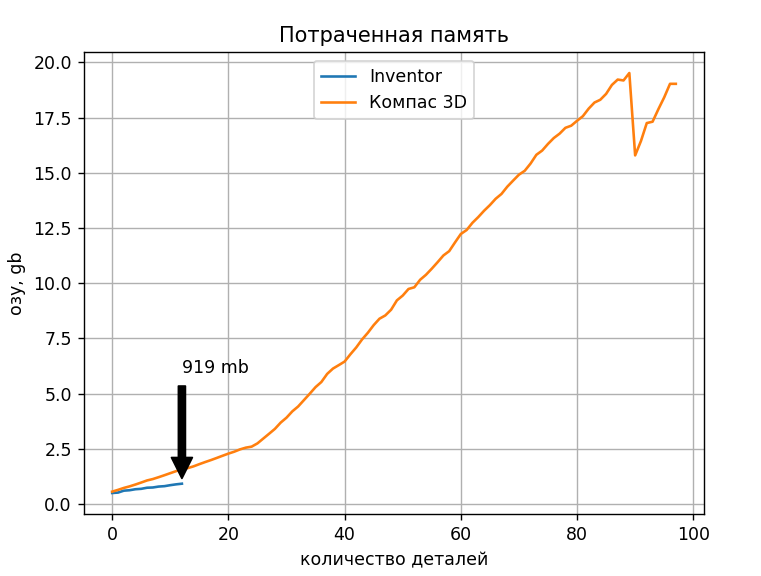
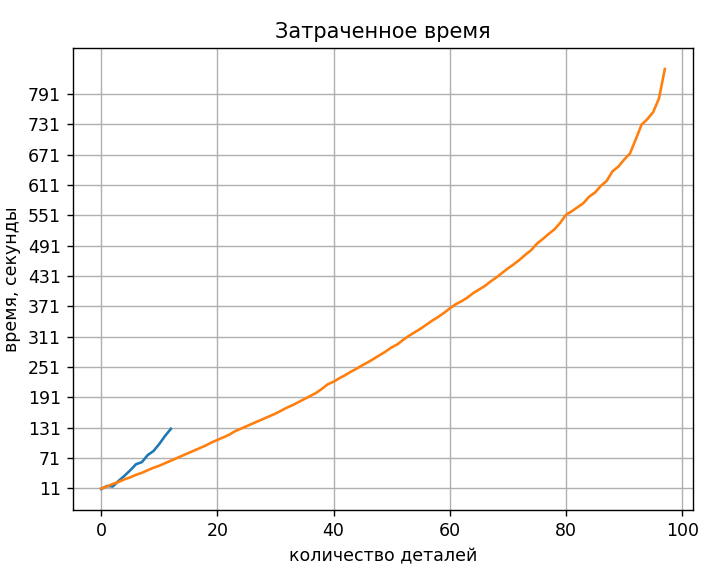


Рисунок 6.7 – График зависимости загруженности памяти от количества деталей

  
Рисунок 6.8 – График зависимости времени от количества построенных деталей с параметрами по умолчанию

Исходя из вышеуказанных графиков на рисунках 6.7 и 6.8, скорость построения в САПР Kompas 3D выше, чем на САПР Inventor.

Так же на рисунке 6.7 видно, что САПР Kompas 3D на 90 детали очищает оперативную память.

САПР Kompas 3D потребляет больше оперативной памяти, чем Inventor.

САПР Inventor при нагрузочном тестировании выдает ошибку на 13 детали Рисунок 6.9. Предполагается это связано с тем, что Inventor не успевает построить деталь и начинает построение следующей.

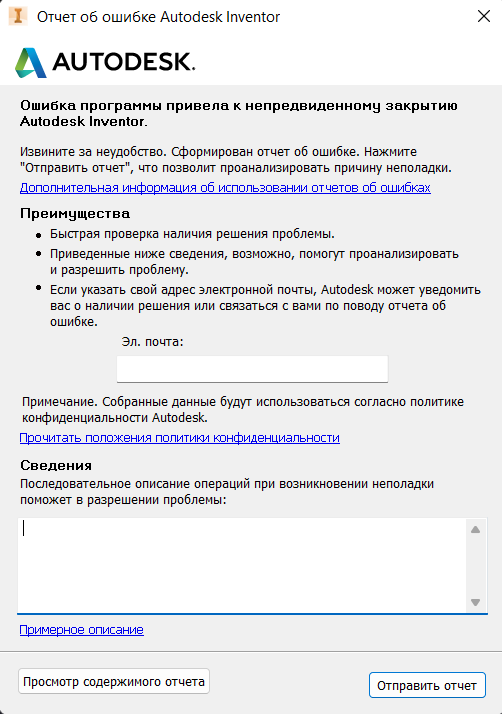


Рисунок 6.9 – Ошибка Inventor при нагрузочном тестировании

# 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 «Стеллаж» в САПР Inventor 2022 и Kompas 3D, и проведено функциональное и нагрузочное тестирование плагина.

# Список использованных источников

1. Автоматизация вычислительных процедур в прикладных задачах инженерного проектирования [Электронный ресурс]. – URL: https://scienceforum.ru/2014/article/2014000201 (дата обращения: 09.12.2022).
2. Visual Studio [Электронный ресурс]. – URL: https://visualstudio.microsoft.com/ru/ (дата обращения: 09.12.2022).
3. Autodesk Inventor— Википедия. [Электронный ресурс]. — Режим доступа: https://ru.wikipedia.org/wiki/Autodesk\_Inventor (дата обращения 09.12.2022).
4. КОМПАС-3D. Официальный сайт САПР КОМПАС [Электронный ресурс]. — Режим доступа: https://kompas.ru/ (дата обращения 09.12.2022).
5. Стеллаж — Википедия. [Электронный ресурс]. — Режим доступа: https://ru.wikipedia.org/wiki/%D0%A1%D1%82%D0%B5%D0%BB%D0%BB%D0%B0%D0%B6 (дата обращения 09.12.2022).
6. Разработка приложений для Inventor - Autodesk. [Электронный ресурс]. – URL: https://www.autodesk.ru/autodesk-developer-network/software-platform-russian/develop-inventor (дата обращения: 09.12.2022).
7. КОМПАС-3D для разработчиков [Электронный ресурс]. – URL: https://kompas.ru/solutions/developers/ (дата обращения: 09.12.2022).
8. NUnit [Электронный ресурс]. – URL: <https://nunit.org/> (дата обращения: 09.12.2022).
9. Что такое Windows Presentation Foundation (WPF) [Электронный ресурс]. – URL: https://docs.microsoft.com/ru-ru/visualstudio/designers/getting-started-with-wpf?view=vs-2022 (дата обращения: 09.12.2022).
10. 3D-конфигуратор стеллажей ARNEG. [Электронный ресурс]. – Режим доступа: https://anykey-design.ru/portfolio/razrabotka-web-saytov/konfiguratory/3d-konfigurator-stellazhey-arneg/ (дата обращения 30.09.2021).
11. UML. [Электронный ресурс]. – Режим доступа: http://www.uml.org/ (дата обращения 09.12.2022).
12. Функциональное тестирование [Электронный ресурс]. – URL: https://daglab.ru/funkcionalnoe-testirovanie-programmnogo-obespechenija/ (дата обращения: 09.12.2022).
13. Юнит-тестирование для чайников [Электронный ресурс]. – URL: https://habr.com/ru/post/169381/ (дата обращения: 09.12.2022).
14. Нагрузочное тестирование: с чего начать и куда смотреть [Электронный ресурс]. – URL: https://habr.com/ru/company/jugru/blog/329174/ (дата обращения: 09.12.2022).