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. 588-2

\_\_\_\_\_\_\_\_\_\_\_\_\_\_К.Д. Робканов

«\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2021 г.

Руководитель:

к.т.н., доцент каф. КСУП

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

«\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2021 г.

Томск 2021

**Оглавление**

1 Введение……………………………………………………………………………3

2 Постановка и анализ задачи………………………………………………………4

2.1 Описание предмета проектирования…………………………………….5

2.2 Выбор инструментов и средств реализации…………………………….6

2.3 Назначение плагина………………………………………………………6

3 Обзор аналогов…………………………………………………………………….7

4 Описание реализации……………………………………………………………...8

# 4.1 Диаграмма классов………………………………………………………..8

5 Описание программы для пользователя………………………………………...12

6 Тестирование программы………………………………………………………..14

6.1 Функциональное тестирование…………………………………………14

6.2 Модульное тестирование………………………………………………18

6.3 Нагрузочное тестирование……………………………………………...20

Заключение………………………………………………………………………….22

Список использованных источников……………………………………………...23

**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 «Журнального столика» для системы автоматизированного проектирования КОМПАС 3D с помощью интегрированной среды разработки Visual Studio 2019 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 КОМПАС 3D [3], построить модель «Журнального столика». Необходимо чтобы плагин позволял задавать параметры по умолчанию, а также изменять входные параметры журнального столика, такие как:

– длина столешницы;

– высота столешницы;

– ширина столешницы;

– высота ножек журнального столика;

– значение основания ножек журнального столика.

# 2.1 Описание предмета проектирования

Предметом проектирования является модель журнального столика.

Журнальный столик (также кофейный, диванный) – удлиненный низкий столик, который обычно располагают перед [диваном](https://ru.wikipedia.org/wiki/%D0%94%D0%B8%D0%B2%D0%B0%D0%BD), [кушеткой](https://ru.wikipedia.org/wiki/%D0%9A%D1%83%D1%88%D0%B5%D1%82%D0%BA%D0%B0) или группой кресел в гостиной или в холле для украшения интерьера и временного размещения книг, журналов, подсвечников и других предметов[4].

Параметры журнального столика:

– длина журнального столика L (от 400 до 800 мм);

– ширина журнального столика W (от 400 до 800 мм);

– высота журнального столика H1 (от 20 до 80 мм);

– высота ножек журнального столика H2 (от 400 до 700 мм);

– диаметр(основания) ножек журнального столика D (от 50 до 200 мм).

Зависимые параметры журнального столика:

– диаметр ножек журнального столика должен быть соответствовать этим формулам: D < W/3 и D < L/3;

– полная высота журнального столика (H1+H2) должна быть больше в 440 мм.

На рисунке 2.1 представлена 3D модель журнального столика с параметрами.



Рисунок 2.1 – 3D модель журнального столика с параметрами

# 2.2 Выбор инструментов и средств реализации

На основе требований к техническому заданию программа выполнена на языке программирования C# в среде Microsoft Visual Studio 2019 с использованием .NET Framework 4.8 , библиотеки для Kompas 3D [3].

Инструментом тестирования и создания модульных тестов был выбран тестовый фреймворк NUnit [5] версии 3.13.

Для реализации пользовательского интерфейса использовалась система для построения настольные приложения Windows Forms.

# 2.3 Назначение плагина

Назначение разрабатываемого плагина обусловлено быстрым моделированием журнального столика с разными входными параметрами. Благодаря данному расширению, проектировщики могут наглядно рассмотреть спроектированную модель, при необходимости перестроить под необходимые им параметры.

**3 Обзор аналогов**

Плагин PDF для САПР КОМПАС-3D позволяет производить экспорт моделей и сборок из КОМПАС-3D в формате PDF. Основной особенностью является возможность интерактивного взаимодействия пользователя с сохраненной 3D сценой внутри PDF файла. Например, пользователь может вращать, масштабировать, передвигать детали и сборки внутри 3D PDF файла. Также доступно создание анимации сборки и разборки изделий. Это полезно для подготовки интерактивных сборочных инструкций, создания маркетинговых материалов, презентаций, а также для налаживания взаимодействия между проектировщиками и заказчиками. В подобных ситуациях традиционным подходом являлся экспорт сборки или детали КОМПАС-3D в промежуточный формат и дальнейшее сохранение в формат 3D PDF. Используемый подход в плагине исключает использование промежуточных файлов для осуществления 3D преобразования, что существенно повышает качество выходной 3D модели в формате PDF[6]. Ключевые возможности:

– сохранение деталей и сборок в формате 3D PDF для интерактивного просмотра при помощи бесплатной программы Adobe Reader;

– создание анимаций, имитирующих естественный порядок сборки и разборки создание имитации анимации гибки листовых тел;

– вставка в существующие PDF документы, содержащие основной текст, фоновые картинки, таблицы спецификаций, эмблемы, логотипы;

– пакетный режим для поочередной конвертации всех файлов.

**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.[7]

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.[7]

На рисунке 4.1 представлена диаграмма классов.

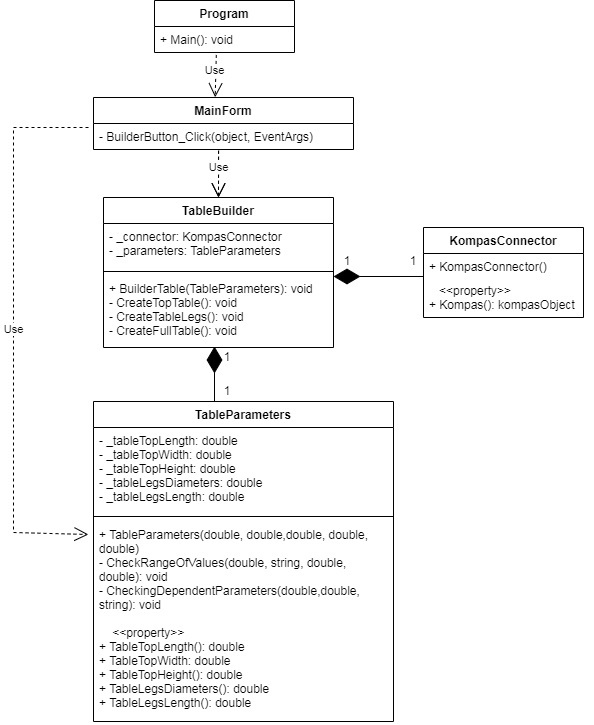


Рисунок 4.1 – Изначальная диаграмма классов

Для реализации был выбран следующий набор классов:

1)MainForm – класс диалогового окна, который обеспечивает взаимодействие между пользователем и программой;

2)KompasConnector – класс для работы с API КОМПАС 3D;

3)TableBuilder – класс, осуществляющий вызов методов API, необходимых для постройки 3D-модели;

4)TableParameters − класс, хранящий в себе все параметры проектируемой 3D-модели, осуществляет проверку зависимых параметров.

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

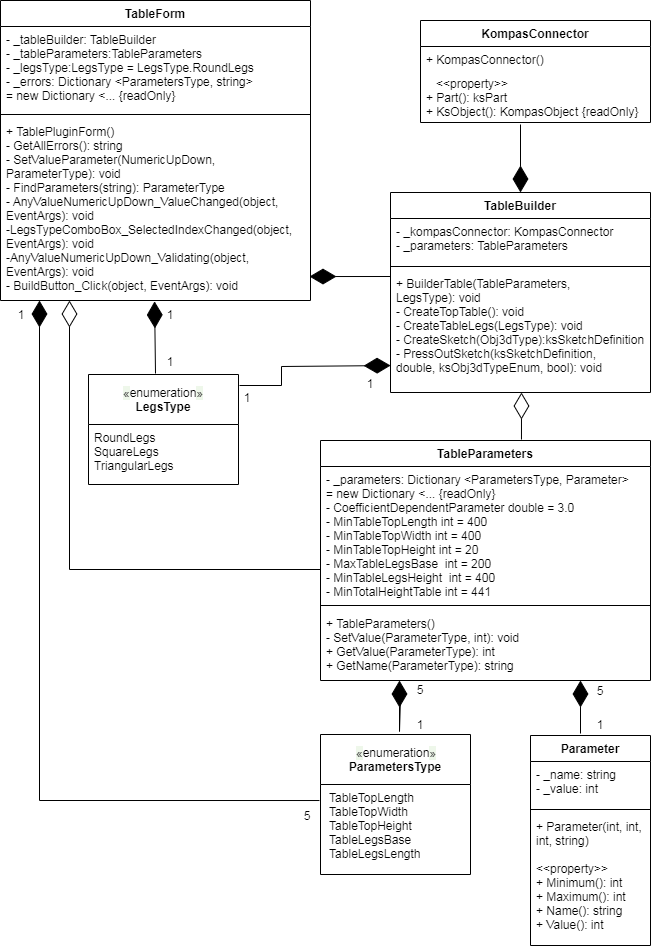


Рисунок 4.2 – Итоговая диаграмма классов

Опишем некоторые изменения, возникшие после добавления дополнительной функциональности (добавлена возможность выбора типа основания ножек) и переработки кода в ходе разработки плагина.

Архитектура переписана с использованием словарей для сокращения дублирования кода и создание гибкой бизнес-логики программы.

Добавлено перечисление ParameterType и LegsType. Перечисление ParameterType используется как ключ для словарей.

Добавлен класс Parameter, в котором хранится дополнительные параметры стола.

В классе TableBuiilder создан отдельный метод для выдавливания и метод для построения эскизов в САПР «Компас-3D v19».

# 5 Описание программы для пользователя

Макет пользовательского интерфейса (рисунок 5.1) представляет собой форму для ввода параметров журнального столика. Построение модели осуществляется путем нажатия на кнопку «Построить». После ввода некорректных данных (например, введена пустая строка или значение параметра не входит диапазон), на форме будет выделено поле красным цветом с некорректным параметром, так же на форме будет показан текст ошибки (рисунок 5.2).

Если пользователь не исправит ошибки и нажмёт на кнопку «Построить» будет показано окно с текстами всех ошибок (рисунок 5.3).

Плагин состоит из диалогового окна, которое имеет пять полей ввода параметров, одну кнопку и одно поле для выбора типа основания ножек.

Так же плагин имеет параметры по умолчанию, которые показаны на рисунке 5.1.

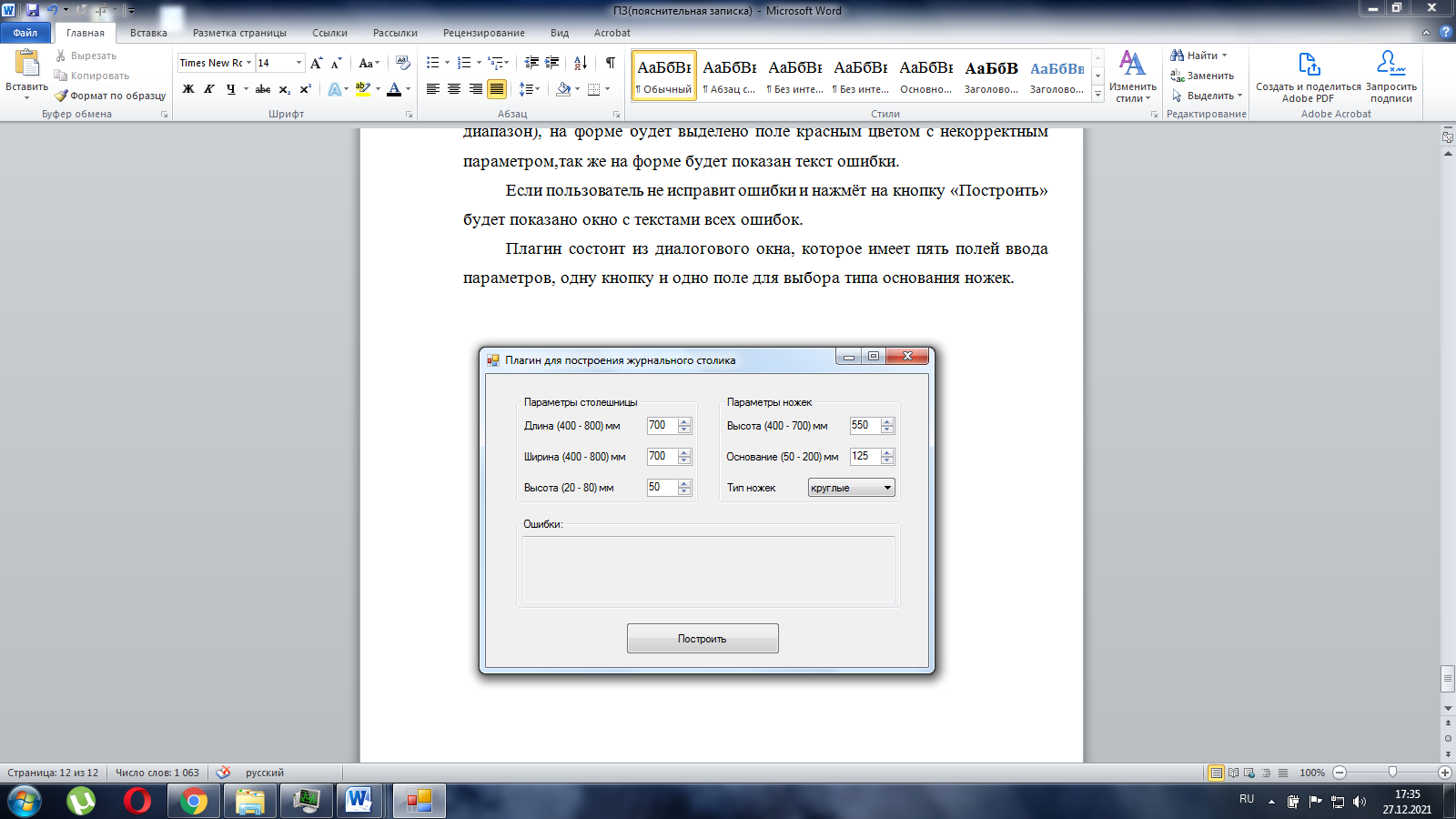


Рисунок 5.1 – Макет пользовательского интерфейса

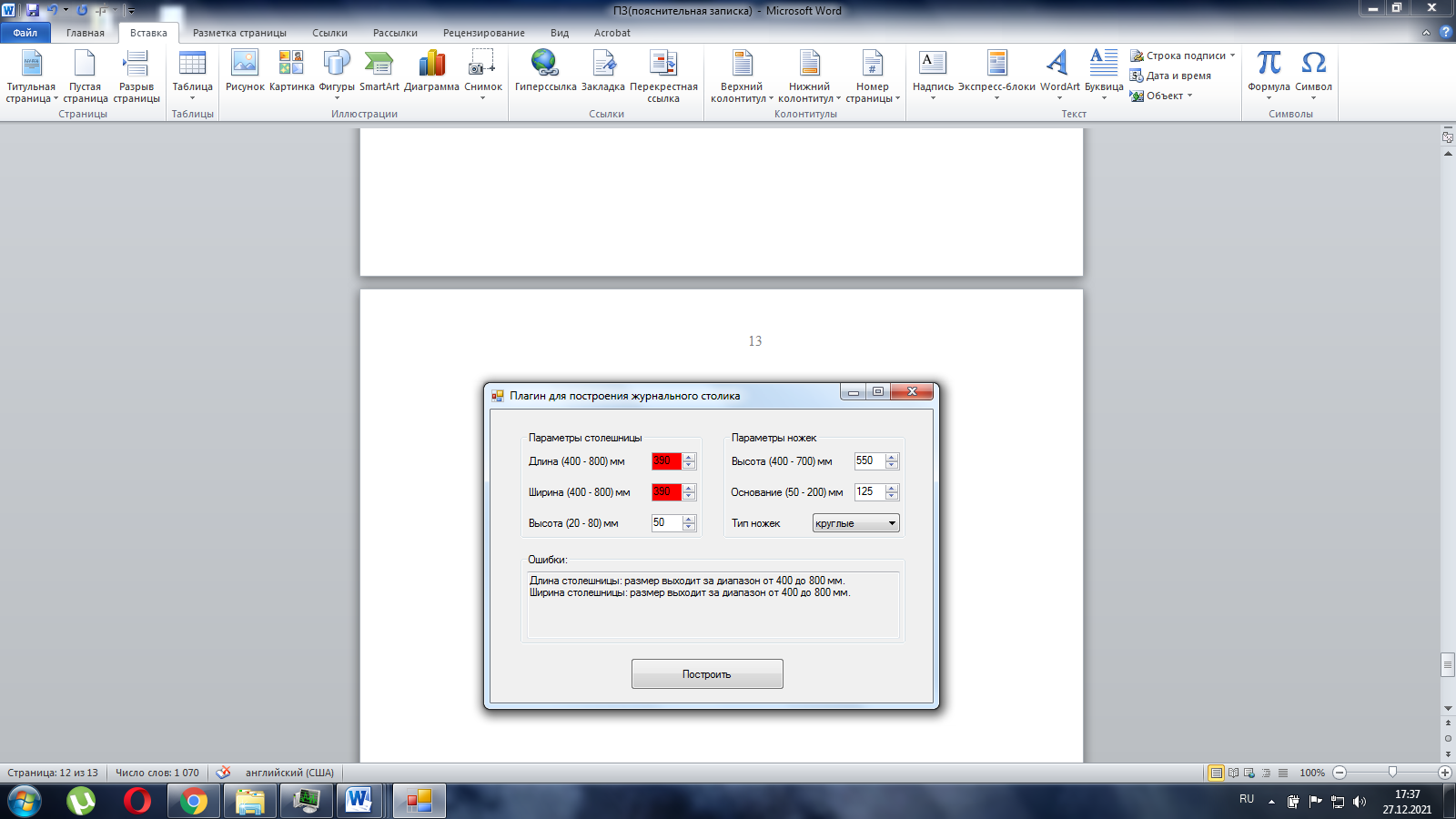


Рисунок 5.2 – Макет пользовательского интерфейса с некорректными данными

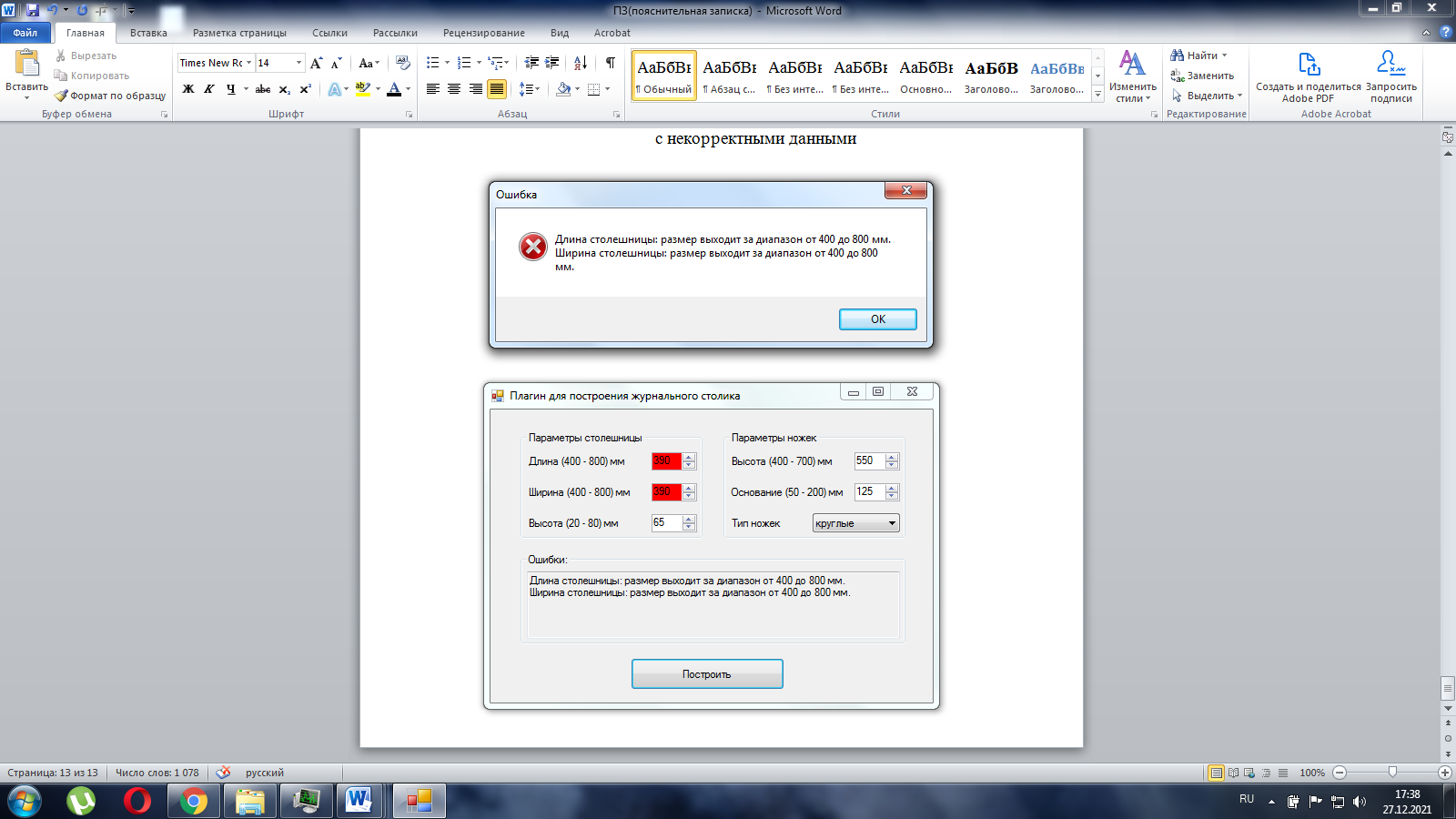


Рисунок 5.3 – Окно ошибки

**6 Тестирование программы**

Тестирование позволяет убедиться в работоспособности программы, выявлять ошибки при изменении какой-либо функциональности.

# 6.1 Функциональное тестирование

При функциональном тестировании [8] проверялось корректность работы плагина для построения журнального столика, а именно, соответствие полученного результата в виде трехмерной модели, с входными параметрами.

Проведено тестирование максимальных и минимальных параметров модели.

На рисунках 6.1 – 6.3 представлены проверки размеров модели с минимальным введенными параметрами в САПР «Компас-3D v19» (ширина столешницы 400 мм, длина столешницы 400 мм, высота столешницы 21 мм, высота ножек 420, значение основания ножек 50 мм).

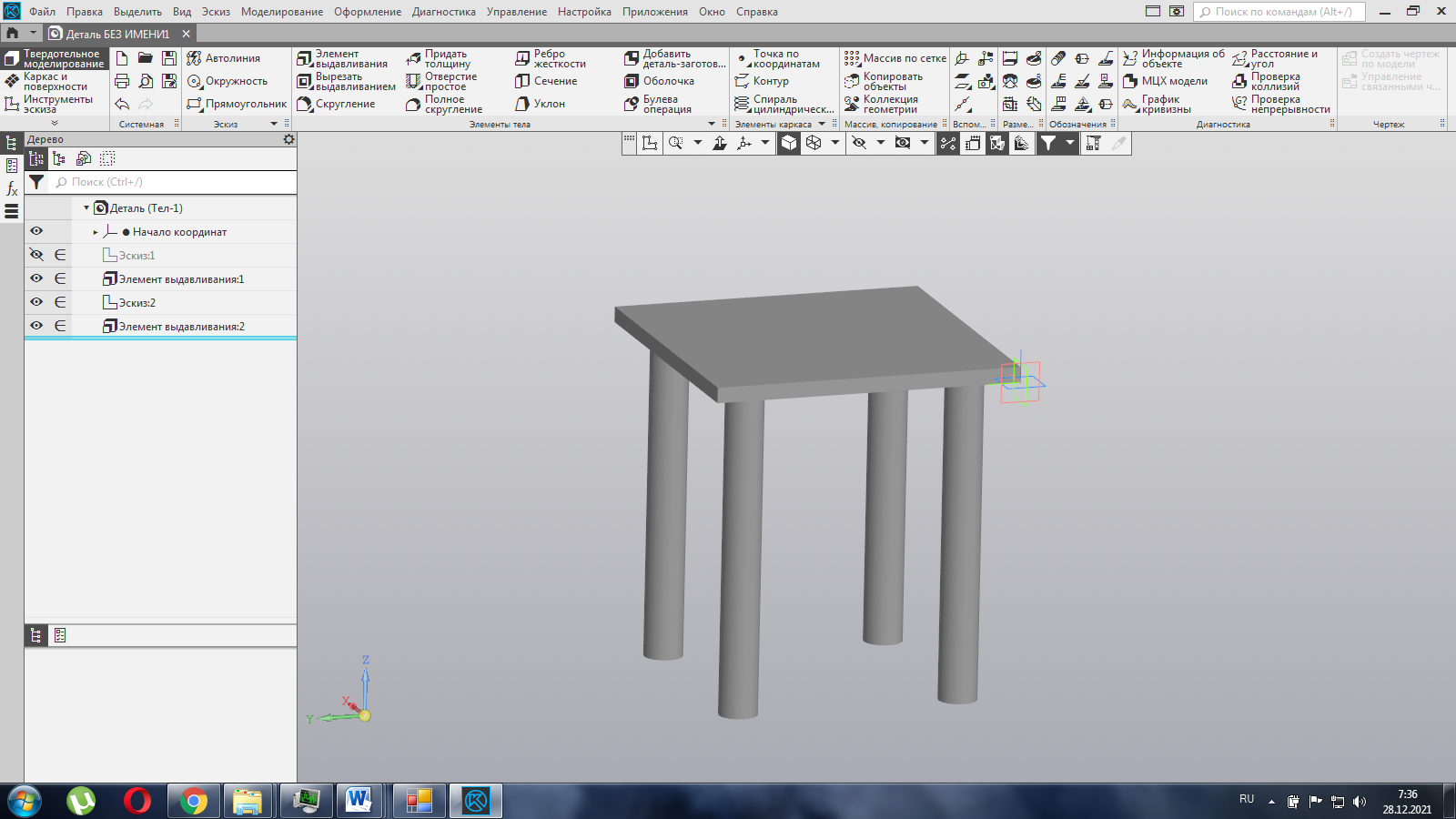


Рисунок 6.1 – Модель журнального столика с минимальными входными данными с круглыми ножками

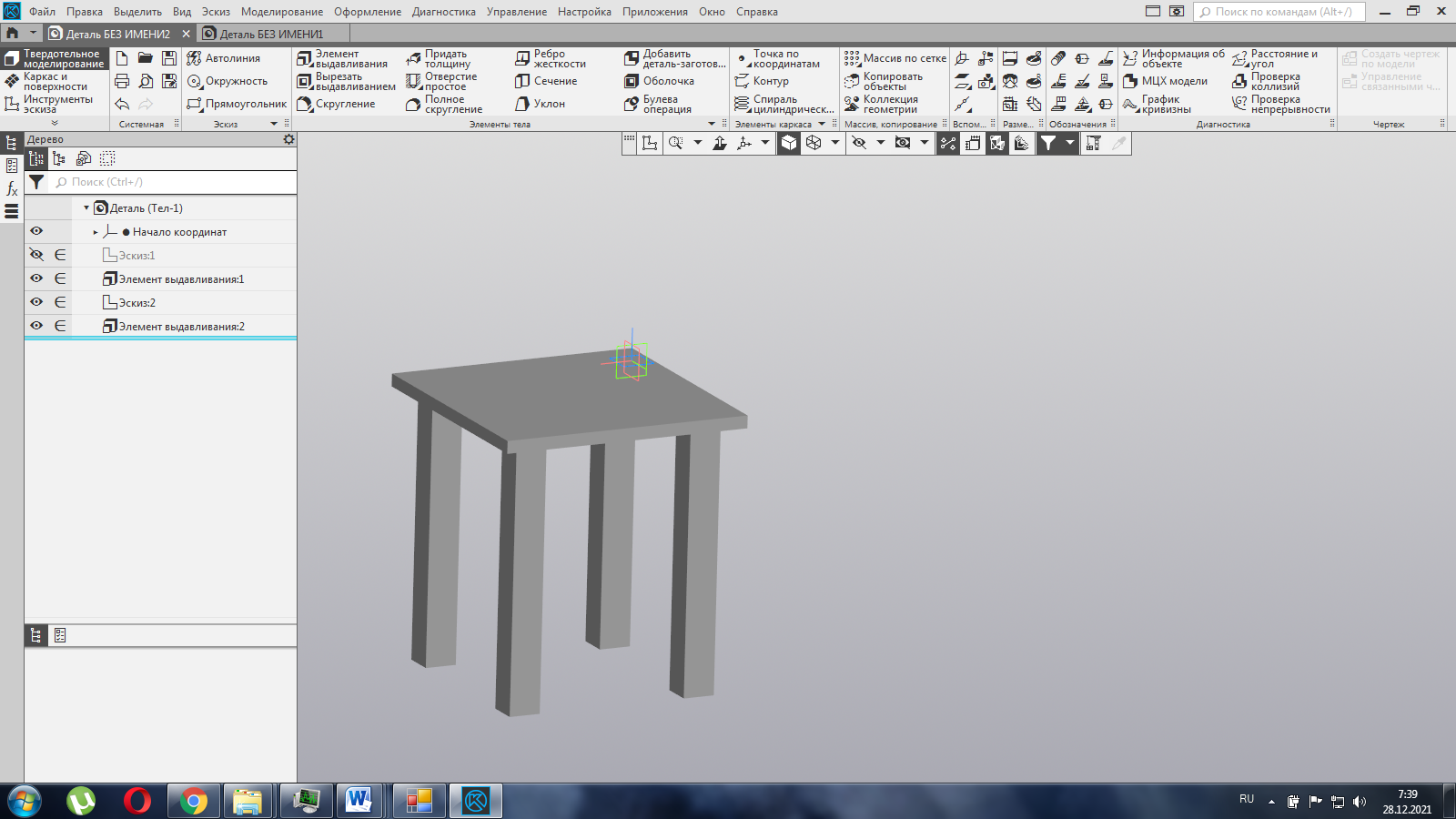


Рисунок 6.2 – Модель журнального столика с минимальными входными данными с квадратными ножками

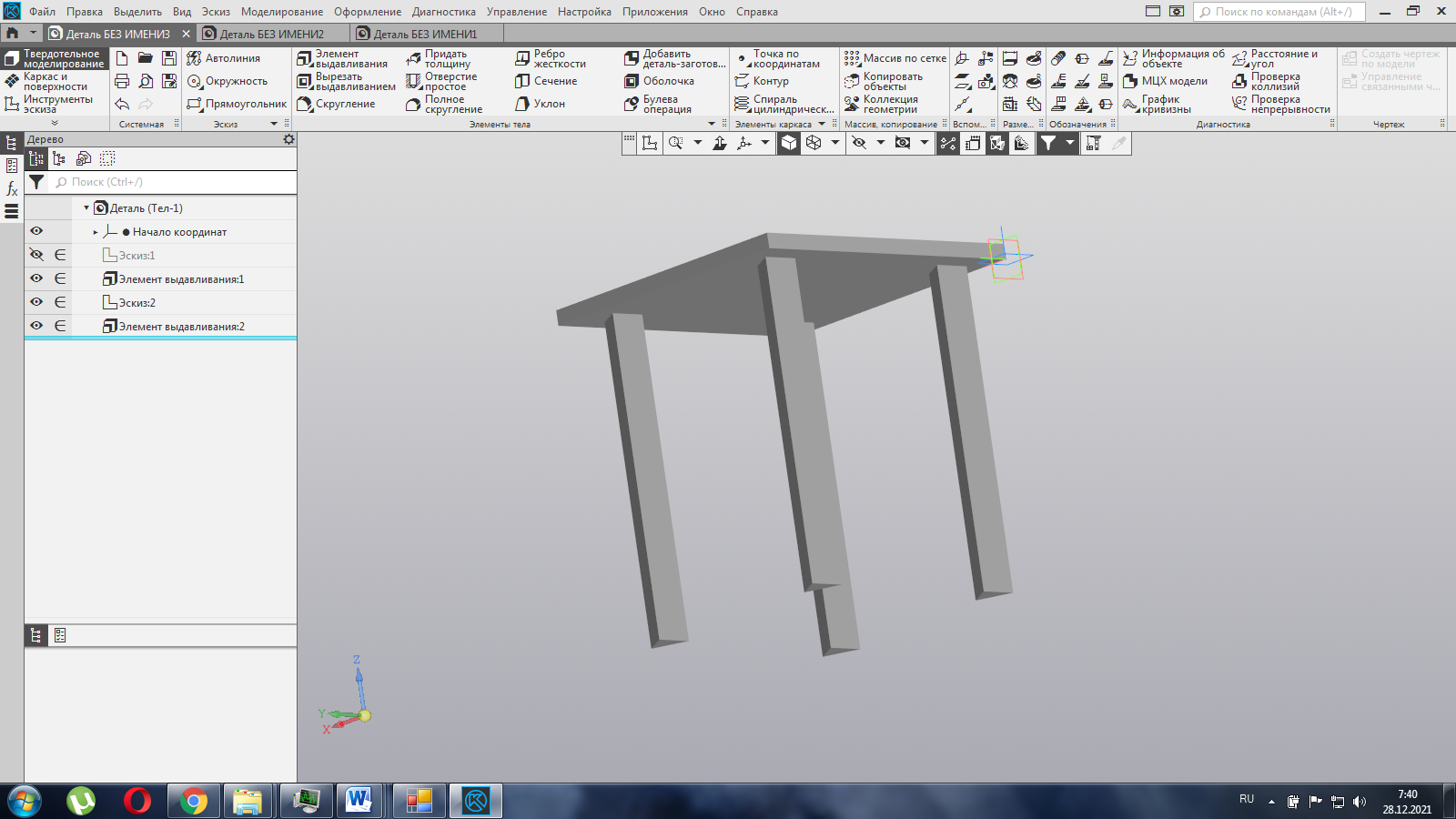


Рисунок 6.3 – Модель журнального столика с минимальными входными данными с треугольными ножками

На рисунках 6.4 – 6.6 представлены проверки размеров модели с максимально введенными параметрами в САПР «Компас-3D v19» (ширина столешницы 800 мм, длина столешницы 800 мм, высота столешницы 80 мм, высота ножек 700, значение основания ножек 200 мм).

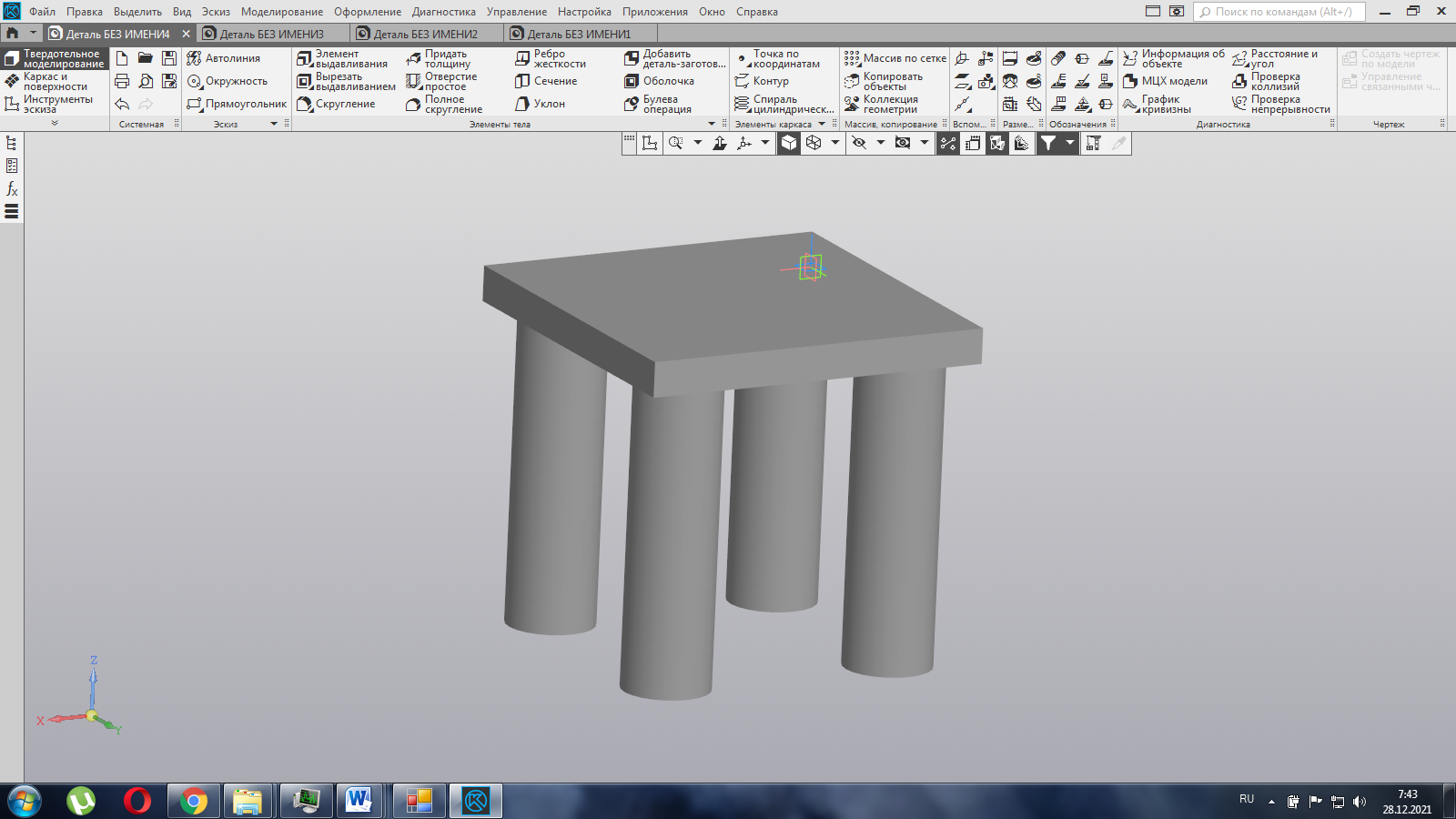


Рисунок 6.4 – Модель журнального столика с максимально входными данными с круглыми ножками

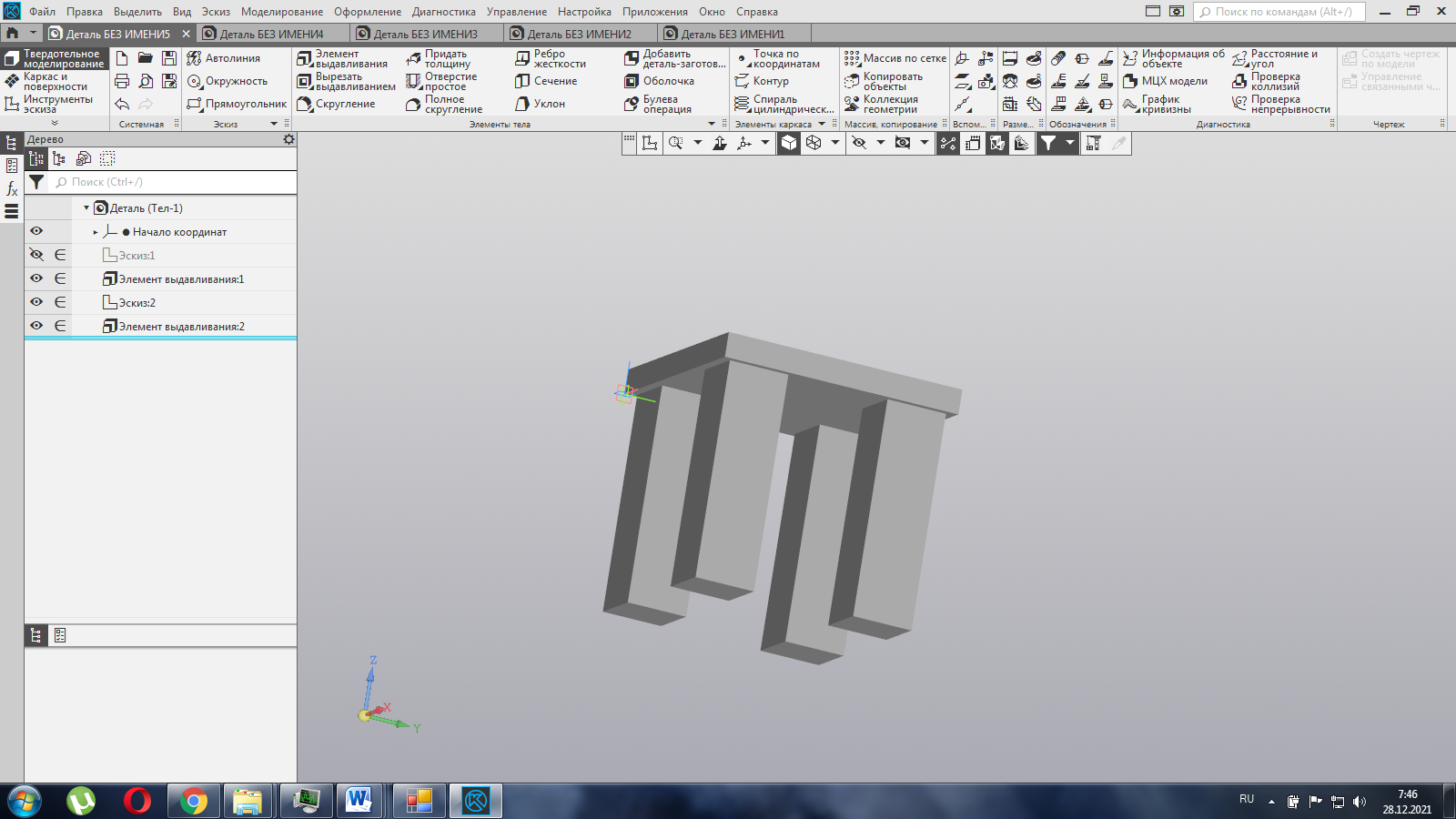


Рисунок 6.5 – Модель журнального столика с максимально входными данными с квадратными ножками

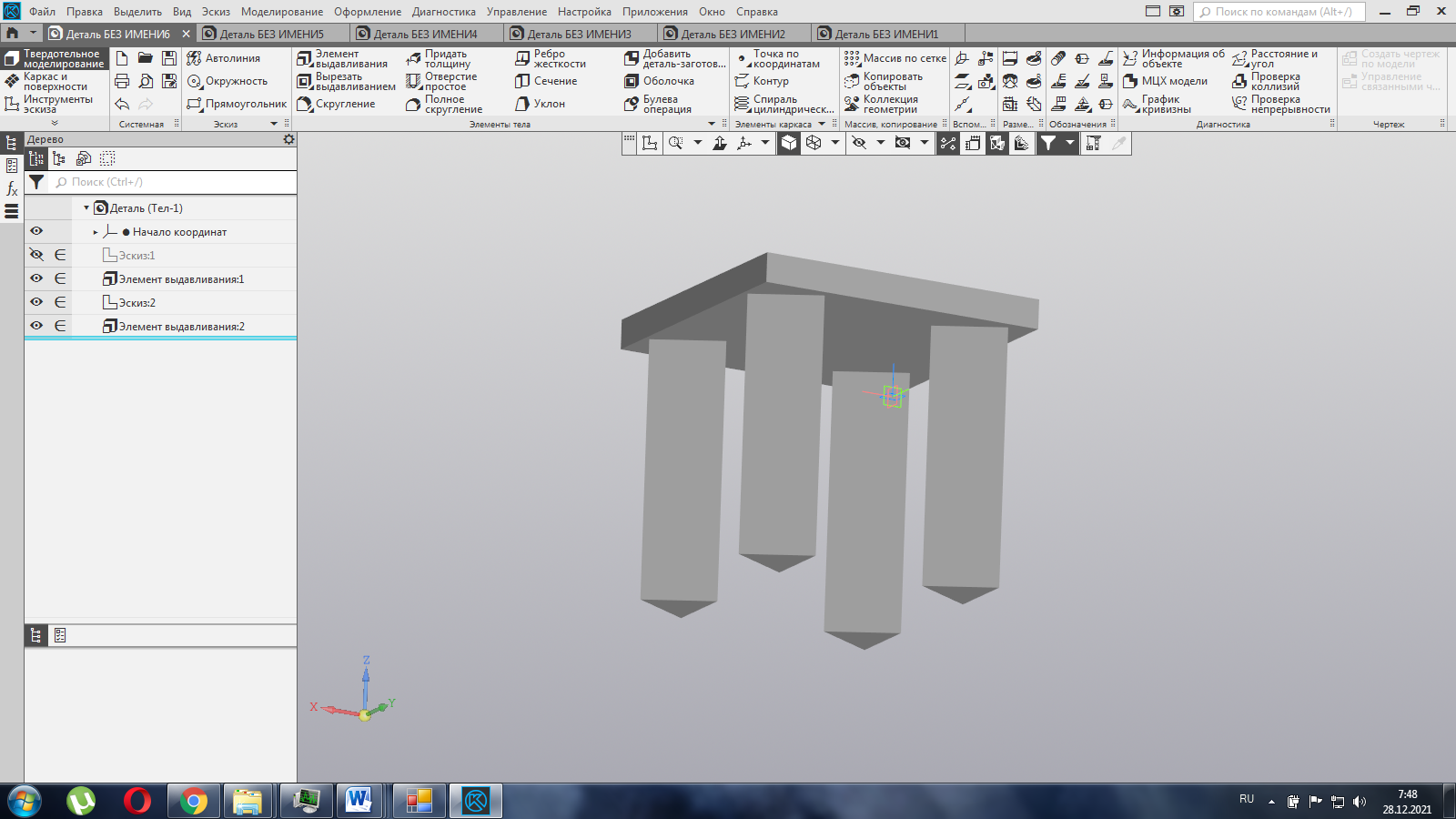


Рисунок 6.6 – Модель журнального столика с максимально входными данными с треугольными ножками

**6.2 Модульное тестирование**

Модульное тестирование – это тестирование отдельного элемента изолированно от остальной системы. Такое тестирование предназначено для проверки правильности работы отдельно взятого класса. Чтобы исключить из результатов тестирования влияние потенциальных ошибок других классов, тестируемый класс должен быть максимально изолирован, т. е. не использовать объекты и методы других классов. Данное требование в итоге позволяет иначе взглянуть на взаимодействие классов[9].

Тестирование проводилось с помощью фреймворка модульного тестирования NUnit 3.13 . На рисунке 6.7 показана степень покрытия кода тестами для классов Parameter и TableParameters, а на рисунке 6.8 показаны результаты тестов для этих же классов. Всего было написано 34 теста.

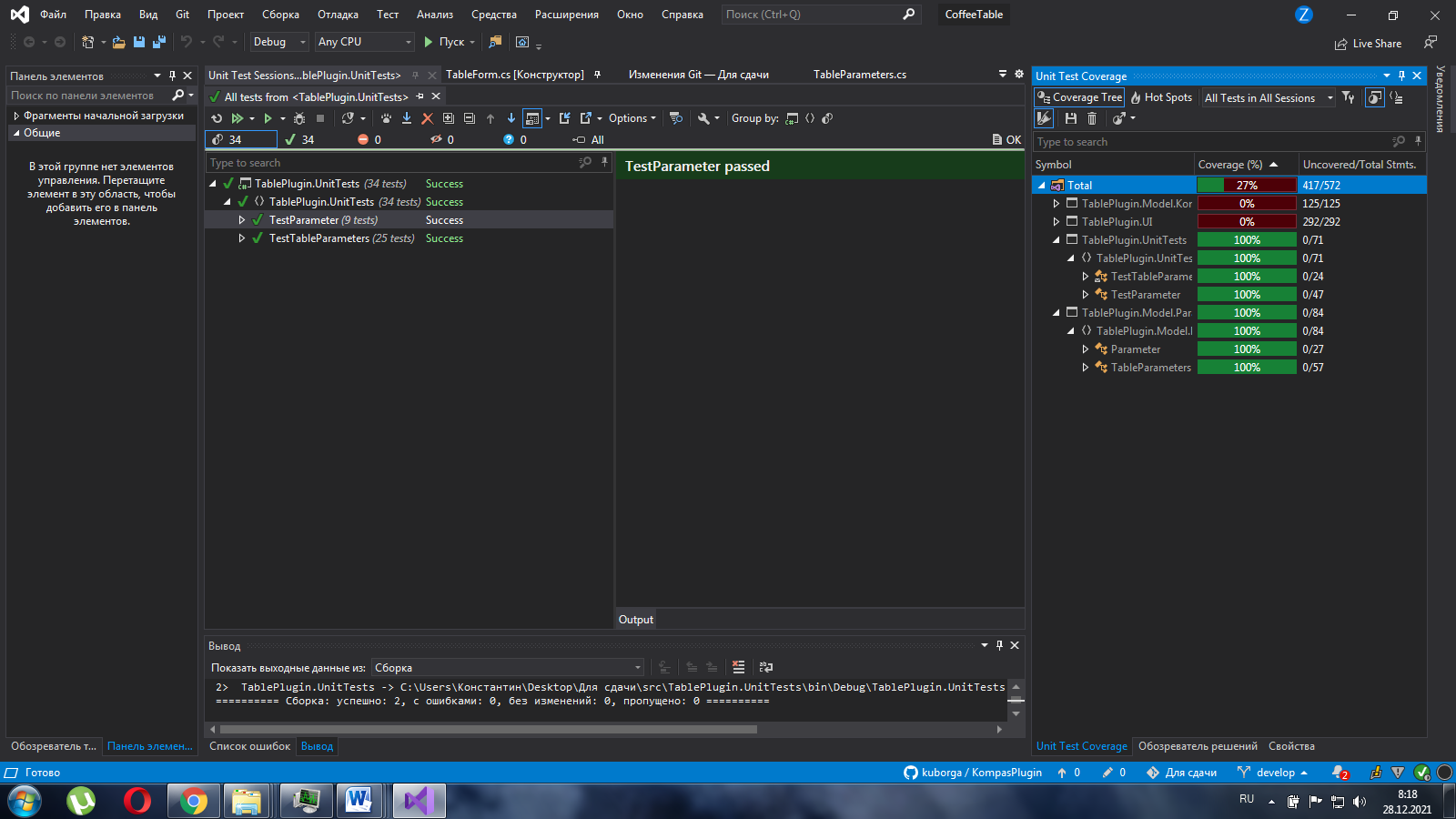


Рисунок 6.7 – Степень покрытия кода модульным тестированием

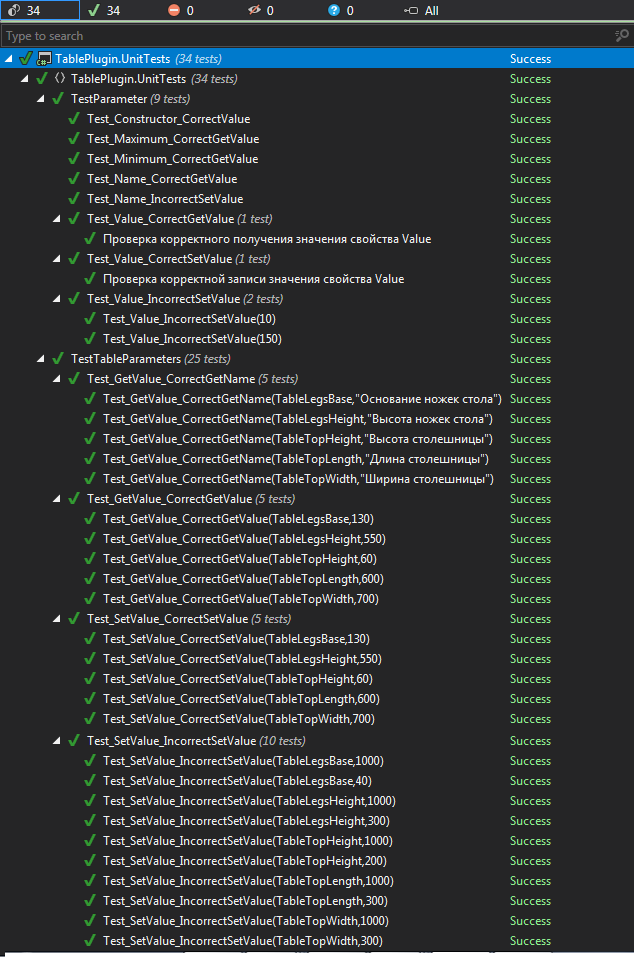


Рисунок 6.8 – Тестирование классов

# 6.3 Нагрузочное тестирование

Нагрузочное тестирование ̶ это автоматизированные испытания информационной системы (или приложения), имитирующие различные нагрузочные модели, с целью комплексной оценки производительности, проверки качественной и бесперебойной работы системы, а также проверки соответствия требованиям, предъявляемым к конкретному объекту тестирования.

Тестирование проводилось на компьютере со следующими техническими параметрами:

– процессор: Intel® Core(TM) i3-31110M CPU 2.40 GHz;

– оперативная память: 4Gb DDR3;

– видеоадаптер: NVIDIA GeForce GTX 630M;

– операционная система: Windows 7 Максимальная 64-разрядная.

На графике, изображенном на рисунке 6.9 в текущей главе, ось «X» – время в минутах, ось «Y» – количество построенных деталей.

На графике, изображенном на рисунке 6.10 ось «Х» – количество построенных деталей, ось «Y» – количество потребляемой оперативной памяти.

Рисунок 6.9 – График зависимости времени от количества построенных деталей с параметрами по умолчанию

Рисунок 6.10 – График зависимости загруженности памяти от количества деталей

Исходя из вышеуказанных графиков на рисунках 6.9 и 6.10, построение последних деталей занимало больше времени, чем вначале.

Анализируя рисунки 6.10 и 6.11 заметно, что сначала графики росли линейно, а потом экспоненциально. Такое потребление оперативной памяти связано с использованием файла подкачки, а точнее переход в виртуальную память [11]. Виртуальная память – метод управления памятью компьютера, позволяющий выполнять программы, требующие больше оперативной памяти, чем имеется в компьютере, путём автоматического перемещения частей программы между основной памятью и вторичным хранилищем (например, твердотельным накопителем).

**Заключение**

В ходе курса ОРСАПР был изучен API Компас3D, основные этапы проектирования и составления проекта системы плагина, изучена предметная область объекта построения. Были составлены UML диаграммы классов. Были составлены такие документы как техническое задание, проект системы и пояснительная записка.

Также была проверена работа плагина на ОС Windows 7.

Результатом работы является плагин для САПР «КОМПАС-3D», который выполняет построение журнального столика с различными переменными параметрами.

# Список использованных источников

1.Автоматизация вычислительных процедур в прикладных задачах инженерного проектирования [Электронный ресурс]. – URL: https://scienceforum.ru/2014/article/2014000201 (дата обращения: 28.12.2021).

2.Visual Studio [Электронный ресурс]. – URL: <https://visualstudio.microsoft.com/ru/> (дата обращения: 28.12.2021).

3.КОМПАС-3D. Официальный сайт САПР КОМПАС [Электронный ресурс]. – Режим доступа: https://kompas.ru/ (дата обращения 28.12.2021).

4.Журнальный стол – Википедия. [Электронный ресурс]. – Режим доступа: <https://ru.wikipedia.org/wiki/Журнальный_стол> (дата обращения: 28.12.2021).

5.NUnit [Электронный ресурс]. – URL: <https://nunit.org/> (дата обращения: 28.12.2021).

6.Экспорт в формате 3D PDF из КОМПАС-3D. [Электронный ресурс]. – Режим доступа: <http://isicad.ru/ru/news.php?news=16278> (дата обращения 28.12.2021).

7.UML. [Электронный ресурс]. – Режим доступа: http://www.uml.org/ (дата обращения 28.12.2021).

8.Функциональное тестирование. [Электронный ресурс]. – URL: https://daglab.ru/funkcionalnoe-testirovanie-programmnogo-obespechenija/ (дата обращения: 28.12.2021).

9.Юнит-тестирование для чайников. [Электронный ресурс]. – URL: <https://habr.com/ru/post/169381/> (дата обращения: 28.12.2021).

10.Нагрузочное тестирование: с чего начать и куда смотреть [Электронный ресурс]. – URL: <https://habr.com/ru/company/jugru/blog/329174/> (дата обращения: 28.12.2021).

11. Виртуальная память — Википедия. [Электронный ресурс] — Режим доступа: https://ru.wikipedia.org/wiki/Виртуальная\_память (дата обращения 28.12.2021).