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

\_\_\_\_\_\_\_\_\_\_\_\_\_\_И.Д. Бураков

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

Руководитель:

к.т.н., доцент каф. КСУП

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

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

**Оглавление**

[1 Введение 3](#_Toc122792379)

[2 Постановка и анализ задачи 4](#_Toc122792380)

[2.1 Описание предмета проектирования 5](#_Toc122792381)

[2.2 Выбор инструментов и средств реализации 6](#_Toc122792382)

[2.3 Назначение плагина 6](#_Toc122792383)

[3 Обзор аналогов 7](#_Toc122792384)

[4 Описание реализации 8](#_Toc122792385)

[4.1 Диаграмма классов 8](#_Toc122792386)

[5 Описание программы для пользователя 13](#_Toc122792387)

[6 Тестирование программы 16](#_Toc122792388)

[6.1 Функциональное тестирование 16](#_Toc122792389)

[6.2 Модульное тестирование 18](#_Toc122792390)

[6.3 Нагрузочное тестирование 19](#_Toc122792391)

[Заключение 22](#_Toc122792392)

[Список использованных источников 23](#_Toc122792393)

# 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 «Стол» для системы автоматизированного проектирования 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 Стола. [3] Необходимо чтобы плагин позволял задавать параметры по умолчанию, а также изменять входные параметры стола, такие как:

* длина стола ***L***;
* высота стола ***H***;
* ширина стола **W**;
* высота столешницы **K**;
* расстояние между ножками стола по ширине стола **w1**;
* расстояние между ножками стола по длине стола **w2**;
* расстояние от края стола по длине **v1**;
* расстояние от края стола по ширине **v2**;
* радиус закругления углов столешницы **R**.

# 2.1 Описание предмета проектирования

Стол — предмет обихода, мебельное изделие, имеющее приподнятую горизонтальную или наклонную поверхность, предназначенную для размещения на ней предметов и (или) для выполнения работ, принятия пищи, игр, рисования, обучения и другой деятельности. Широко распространены столы, имеющие прямоугольную крышку и четыре (две или одну центральную) опоры. [3]

На рисунке 2.1 представлен чертеж стола.

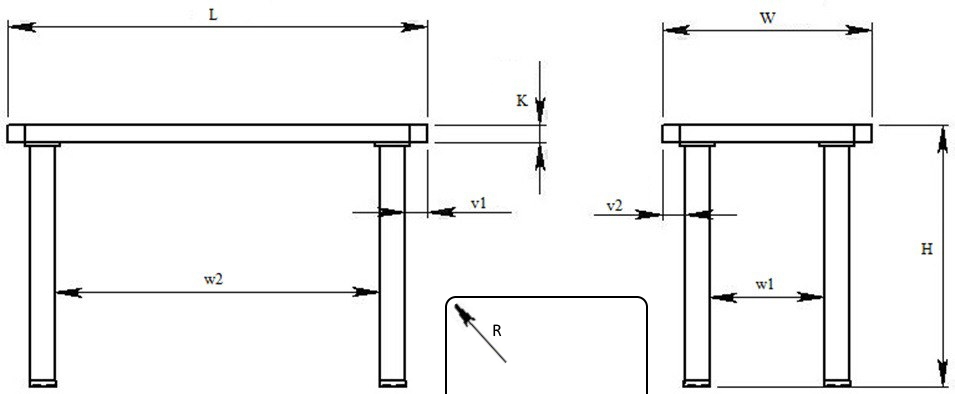


Рисунок 2.1 – Чертеж стола.

Параметры стола:

* длина стола ***L*** (минимум – 300 мм, максимум – 2000 мм);
* высота стола ***H*** (минимум – 300 мм, максимум – 1200 мм);
* ширина стола **W** (минимум – 300 мм, максимум – 2000 мм);
* высота столешницы **K** (минимум – 10 мм, максимум – 50 мм);
* расстояние между ножками стола по ширине стола **w1** (минимум – ширина стола **W**/2 мм, максимум – ширина стола **W** мм);
* расстояние между ножками стола по длине стола **w2** (минимум – длина стола **L**/2 мм, максимум – длина стола **L**);
* расстояние от края стола по длине **v1**. Определяется автоматически по формуле: длина стола **L** – **w2** расстояние между ножками стола по длине;
* расстояние от края стола по ширине **v2**. Определяется автоматически по формуле: длина стола **W** – **w1** расстояние между ножками стола по длине;
* радиус закругления углов столешницы **R** (минимум – 0 мм, максимум – 2 \* (**v1** + **v2**) /2 мм).

# 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 6 [2], библиотеки для Kompas 3D [5].

Инструментом тестирования и создания модульных тестов был выбран тестовый фреймворк NUnit [7] версии 3.13.3.

Для реализации пользовательского интерфейса использовалась система для построения настольных приложений WPF [8].

# 2.3 Назначение плагина

Назначение разрабатываемого плагина обусловлено быстрым моделированием столов различных размеров. Благодаря данному расширению, мастера по столам могут наглядно рассмотреть спроектированную модель, при необходимости перестроить под необходимые им параметры.

# 3 Обзор аналогов

**SketchUp**

SketchUp — программа для 3D дизайна и архитектурного проектирования. В основном используется для моделирования жилых домов, мебели, интерьера. Есть инструменты для проектирования лестниц, электропроводки, санитарно-технических коммуникаций и оборудования. [4]

Данный плагин содержит более трехсот предустановленных стилей, разделенных на библиотеки. Пользователи могут создавать собственные стили и сохранять изображения и значки. Применяя этот плагин, пользователи могут моделировать все заборы или перила в 3D, 2D линиях, расположенных с основной стороны или 2D скрытых линиях.

Пример стола в программе и интерфейс программы представлен на рисунке 3.1.

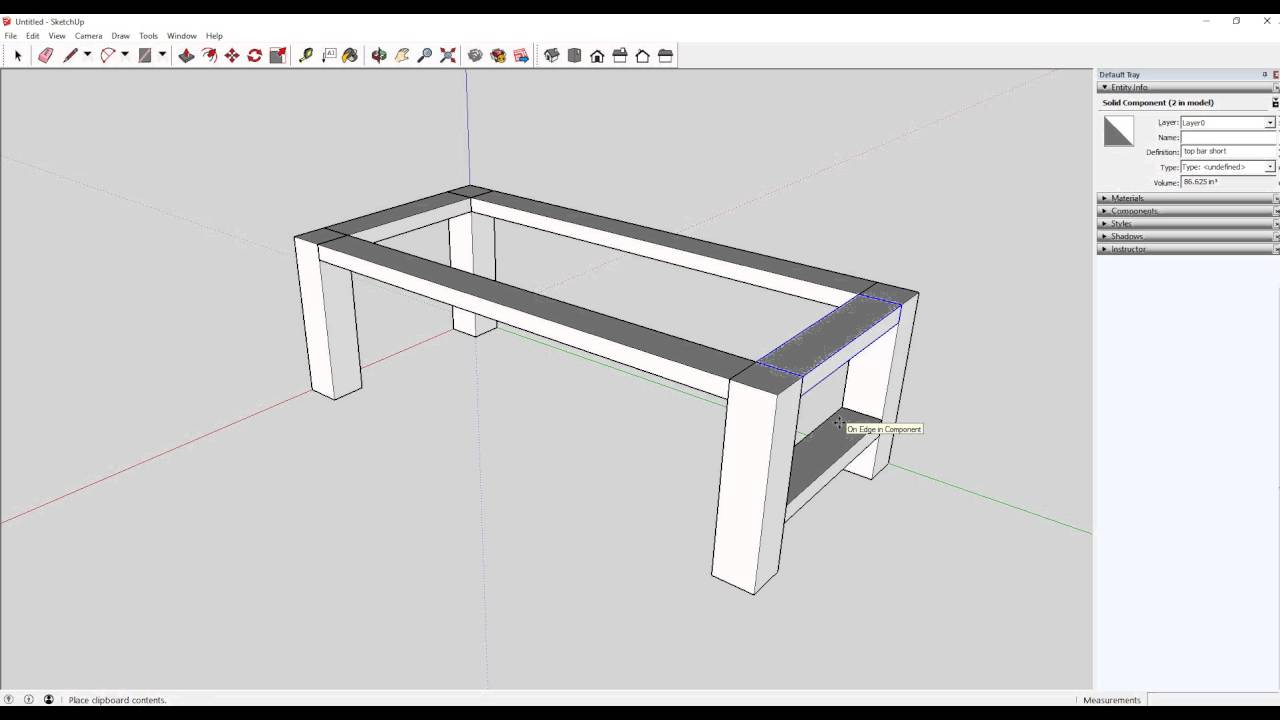


Рисунок 3.1 — Интерфейс программы SketchUp

# 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. [9]

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. [9]

На рисунке 4.1 представлена диаграмма классов.

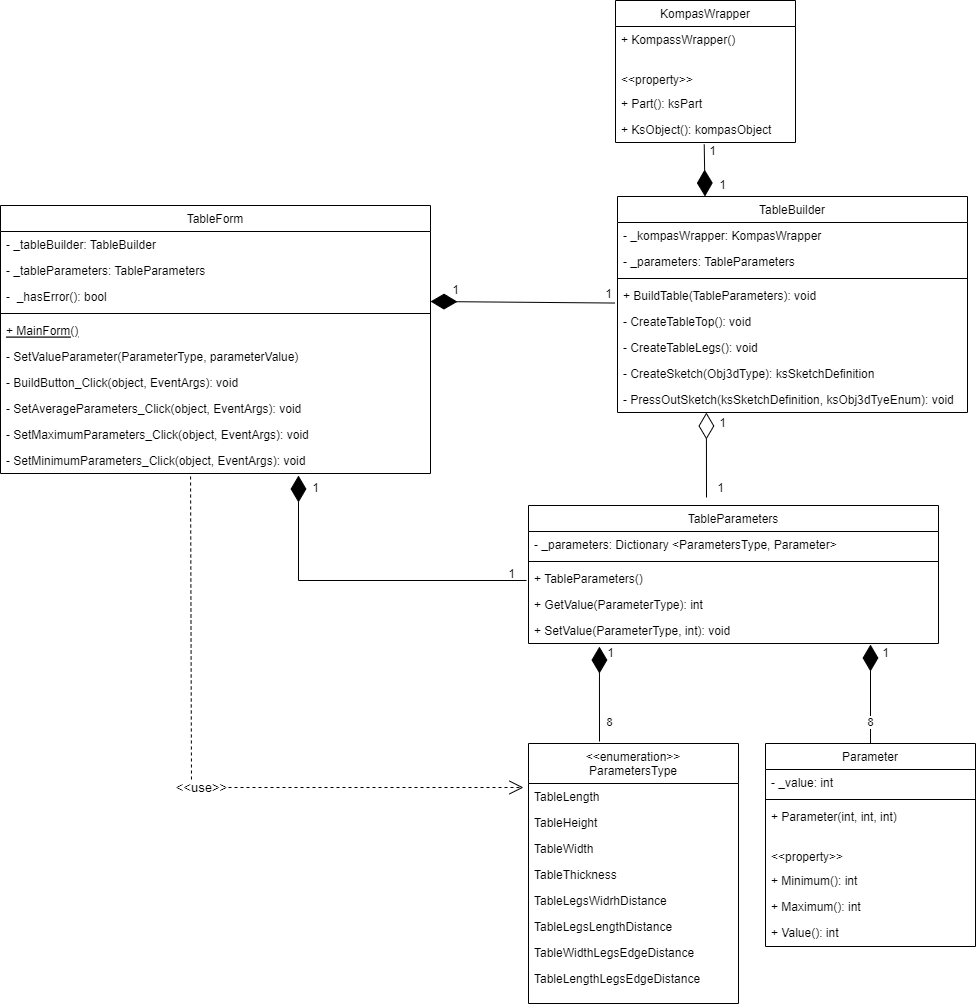


Рисунок 4.1 – Изначальная диаграмма классов

При проектировании системы подразумевалось использовании WinForms, однако в процессе разработки было решено использовать более новую технологию такую как WPF, поэтому диаграмма приобрела другой вид, однако основные классы остались почти без изменений.

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

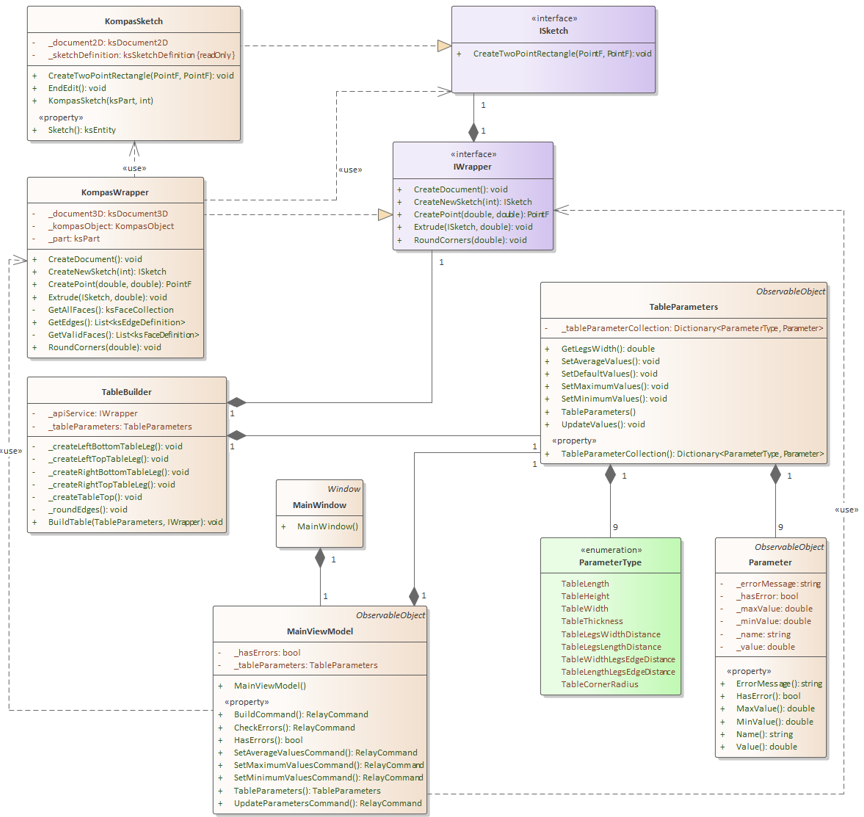


Рисунок 4.2 – Итоговая диаграмма классов

Был добавлен интерфейс (IWrapper) для того, чтобы программа могла поддерживать несколько САПР для построения детали «Стол», а также для простоты тестирования. Также были реализован класс от этого интерфейса — KompasWrapper.

Так как я не знал, как правильно пользоваться Mock объектами в языке C# был создан интерфейс для эскизов – это сделало проще тестирование классов, связанных с эскизом. С помощью этого интерфейса можно упростить расширение программы и реализовать в ней поддержку нескольких САПР. Также был реализован класс от этого интерфейса – KompasSketch.

В классе TableBuilder появился новый метод \_roundEdges, который необходим для выполнения дополнительного задания, а именно закругление граней столешницы.

Также класс TableParameters приобрел несколько новых методов, а именно:

1. SetDefaultValues, который устанавливает параметры по умолчанию, чтобы затрачивать меньше времени для заполнения параметров на UI и сразу была возможность продемонстрировать, что плагин способен построить модель стола в Kompas 3D;
2. SetAverageValues, который устанавливает средние значения для параметров, чтобы пользователь из UI построил стол со средними значениями;
3. SetMaximumValues, который устанавливает максимальные значения для параметров, чтобы пользователь из UI построил стол с максимальными значениями;
4. SetMinimumValue, который устанавливает минимальные значения для параметров, чтобы пользователь из UI построил стол с минимальными значениями;
5. GetLegsWidth, который определяет ширину ножек в зависимости от заданных параметров;
6. UpdateValues, которые обновляет параметры при изменении одного из них.

В класс KompasWrapper были добавлены методы, для выполнения дополнительного задания, такие как:

1. GetAllFaces, который возвращает все поверхности 3D объекта;
2. GetValidFaces, который возвращает все валидные поверхности из тех, что вернул метод GetAllFaces;
3. GetEdges, который возвращает список граней для закругления, которые получает из списка валидных поверхностей из метода GetValidFaces;
4. RoundCorners, которые закругляет все грани с определенным радиусом, которые получает из метода GetEdges;

# 5 Описание программы для пользователя

Макет пользовательского интерфейса представляет собой форму для ввода параметров стола. Построение модели осуществляется путем нажатия на кнопку «Build Table»).

На рисунке 5.1 представлен изначальный макет пользовательского интерфейса.

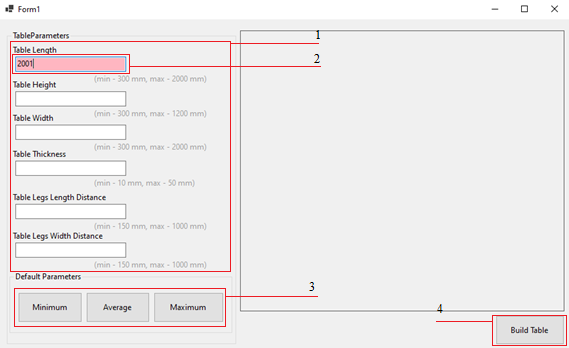


Рисунок 5.1 – Изначальный макет пользовательского интерфейса.

Описание пользовательского интерфейса:

1 – поля для ввода параметров стола;

2 – пример ввода некорректного значения;

3 – кнопки для установления стандартных значений в параметры стола;

4 – кнопка для построения 3D модели стола.

С помощью данного окна пользователь может изменять параметры будущей 3D модели стола:

Напротив полей ввода находится название компонента, за который поле отвечает и корректные размеры – минимальный и максимальный.

При нажатии на кнопку «Minimum» будет создана 3D модель с минимальными корректными размерами. При нажатии на кнопку «Average» будет создана 3D модель со средними корректными размерами. При нажатии на кнопку «Maximum» будет создана 3D модель с максимальными корректными размерами. При нажатии на кнопку «Build Table» будет произведено построение 3D модели с заданными параметрами. В поле справа будет располагаться модель стола, для лучшего понимания размеров и их влияния на конечную модель.

Когда пользователь вводит некорректные данные TextBox подсвечивается.

В ходе разработки были произведены некоторые изменения в пользовательском интерфейс. Итоговый макет пользовательского интерфейса представлен на рисунке 5.2.

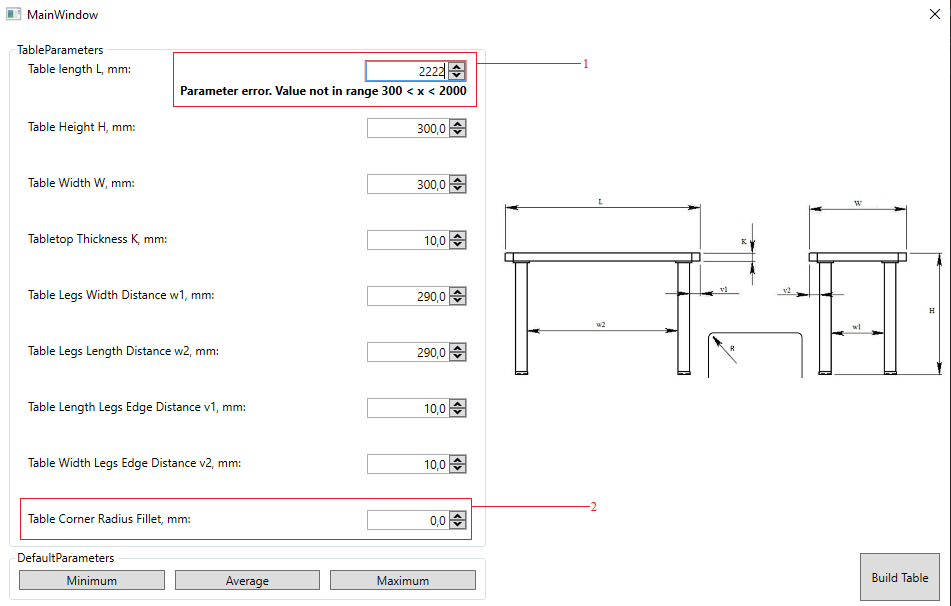


Рисунок 5.2 – Итоговый макет пользовательского интерфейса.

Описание нововведений в пользовательском интерфейсе:

1 – теперь максимальные и минимальные значение не подписаны рядом сполем ввода, а высвечиваются, если пользователь вводит значения, выходящие за диапазон;

2 – новое поле ввода для задания радиуса скругления у углов столешницы.

После ввода необходимых параметров, построить деталь в САПР Kompas 3D можно с помощью кнопки «Build Table». Стол, построенный по заданным параметрам по умолчанию в САПР Kompas 3D, представлен на рисунке 5.3. Стол, построенный с закруглениями представлен на рисунке 5.4.

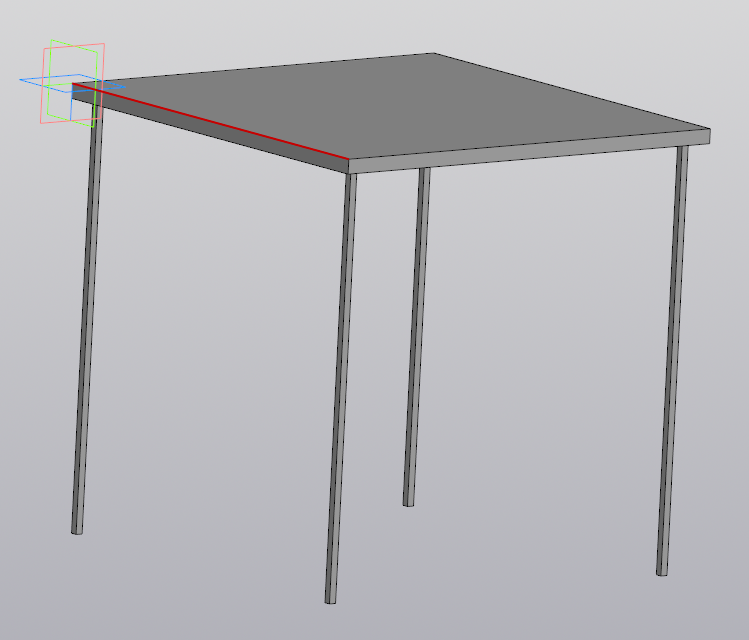


Рисунок 5.3 — Стол, построенный по заданным параметрам по умолчанию в САПР Kompas 3D.

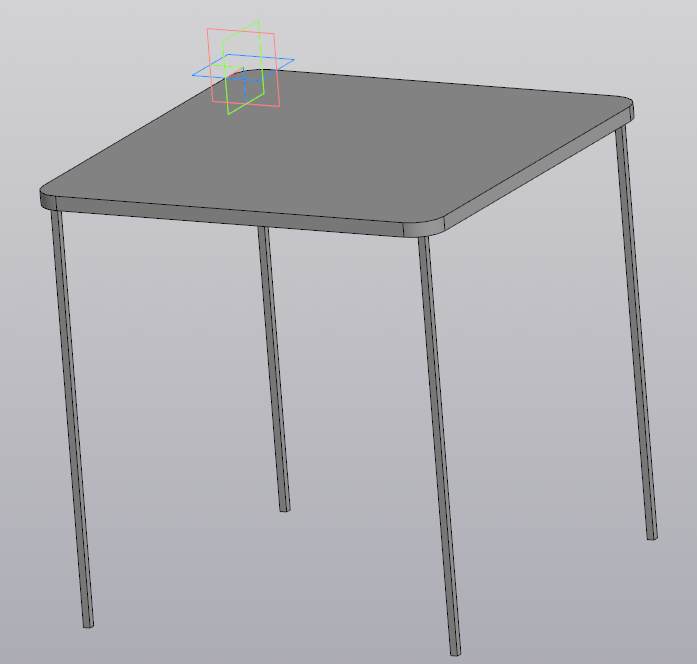


Рисунок 5.4 — Стол, построенный по заданным параметрам по умолчанию и закруглениями в САПР Kompas 3D.

# 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 «Стол», а именно, соответствие полученного результата в виде трехмерной модели, с входными параметрами. [10]

Проведено тестирование максимальных и минимальных параметров модели.

На рисунке 6.1 представлена проверки размеров модели с минимальным введенными параметрами в САПР Kompas 3D.

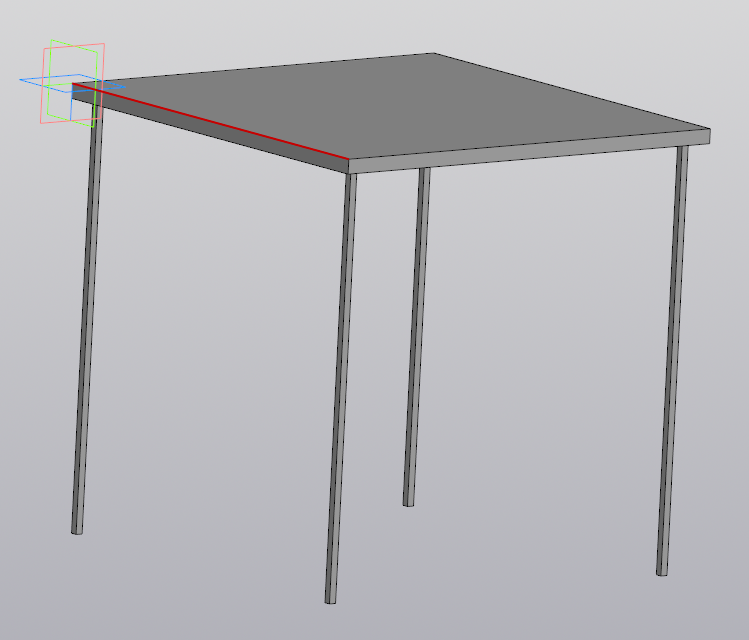


Рисунок 6.1 – Модель с минимальными введенными параметрами в Kompas 3D

Ниже на рисунке 6.2 представлена проверка размеров модели с максимальными введенными параметрами в САПР Kompas 3D. А также на рисунке 6.3 представлена проверка построения скруглений.

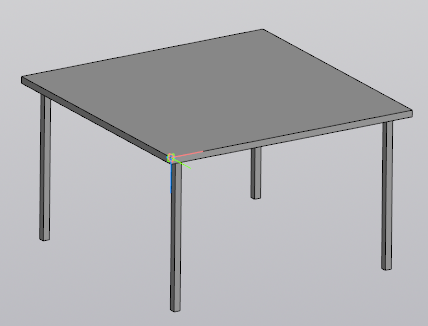


Рисунок 6.2 — Модель с максимальными параметрами в Kompas 3D

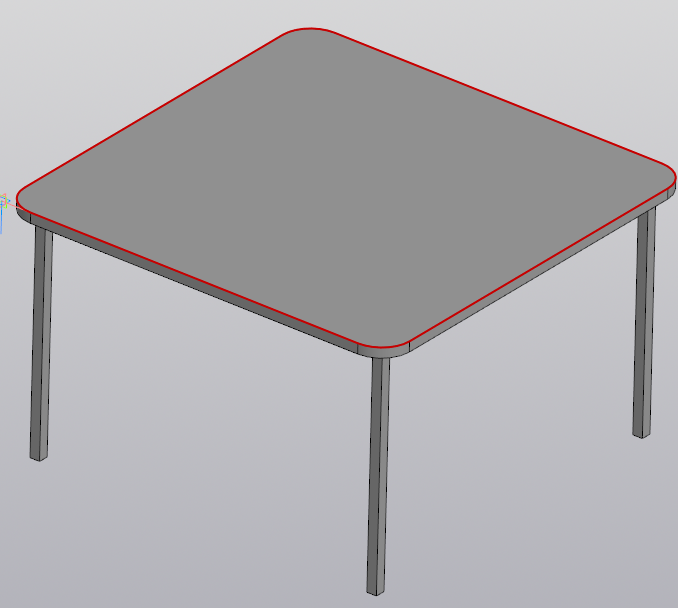
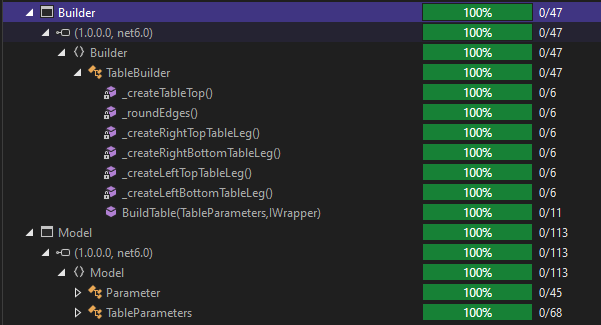


Рисунок 6.3 — Модель со скруглениями в Kompas 3D

# 6.2 Модульное тестирование

В целях проверки корректности работы методов и свойств классов при помощи тестового фреймворка NUnit версии 3.13 проведено модульное тестирование [11], проверялись открытые поля и методы. На рисунке 6.4 представлено тестирование классов проектов: Model, Builder. Степень покрытия проектов — сто процентов.

  
Рисунок 6.4 – Тестирование классов

Перечень тестов для Builder и Model, а также и их описание представлено в таблице 6.1.

Таблица 6.1 – Тестовые случаи метода TestGetValue\_CorrectGetValue.

|  |  |  |
| --- | --- | --- |
| Тестируемое свойство | Параметр | Описание |
| TableHeight | 1000 | Проверка корректного получения значения свойства TableHeight |
| TableLength | 1000 | Проверка корректного получения значения свойства TableLength |
| TableWidth | 1000 | Проверка корректного получения значения свойства TableWidth |

Продолжение таблицы 6.1.

|  |  |  |
| --- | --- | --- |
| TableThikness | 20 | Проверка корректного получения значения свойства TableThikness |
| TableLegsLengthDistance | 900 | Проверка корректного получения значения свойства TableLegsLengthDistance |
| TableLegsWidthDistance | 900 | Проверка корректного получения значения свойства TableLegsWidthDistance |
| TableLengthLegsEdgeDistance | 50 | Проверка корректного получения значения свойства TableLengthLegsEdgeDistance |
| TableWidthhLegsEdgeDistance | 50 | Проверка корректного получения значения свойства TableWidthhLegsEdgeDistance |

Таблица 6.2 – Тестовые случаи метода TestGetValue\_CorrectSеtValue.

|  |  |  |
| --- | --- | --- |
| Тестируемое свойство | Параметр | Описание |
| TableHeight | 1000 | Проверка корректного присваивания значения свойства TableHeight |
| TableLength | 1000 | Проверка корректного присваивания значения свойства TableLength |
| TableWidth | 1000 | Проверка корректного присваивания значения свойства TableWidth |

Продолжение таблицы 6.2.

|  |  |  |
| --- | --- | --- |
| TableThikness | 20 | Проверка корректного присваивания значения свойства TableThikness |
| TableLegsLengthDistance | 900 | Проверка корректного присваивания значения свойства TableLegsLengthDistance |
| TableLegsWidthDistance | 900 | Проверка корректного присваивания значения свойства TableLegsWidthDistance |
| TableLengthLegsEdgeDistance | 50 | Проверка корректного присваивания значения свойства TableLengthLegsEdgeDistance |
| TableWidthhLegsEdgeDistance | 50 | Проверка корректного присваивания значения свойства TableWidthhLegsEdgeDistance |

Таблица 6.3 – Тестовые случаи, метод TestSetValue\_IncorrectSetValue.

|  |  |  |
| --- | --- | --- |
| Тестируемое свойство | Параметр | Описание |
| TableLength | 9 | Проверка некорректной передачи значения свойства TableLength. Меньше минимального. |

Продолжение таблицы 6.3.

|  |  |  |
| --- | --- | --- |
| TableLength | 10000 | Проверка некорректной передачи значения свойства TableLength. Больше максимального. |
| TableHeight | 9 | Проверка некорректной передачи значения свойства TableHeight. Меньше минимального. |
| TableHeight | 10000 | Проверка некорректной передачи значения свойства TableHeight. Меньше минимального. |
| TableWidth | 9 | Проверка некорректной передачи значения свойства TableWidth  Меньше минимального. |
| TableWidth | 10000 | Проверка некорректной передачи значения свойства TableWidth. Больше максимального. |
| TableLegsLengthDistance | 9 | Проверка некорректной передачи значения свойства TableLegsLengthDistance. Меньше минимального. |

Продолжение таблицы 6.3.

|  |  |  |
| --- | --- | --- |
| TableLegsLengthDistance | 10000 | Проверка некорректной передачи значения свойства TableLegsLengthDistance. Больше максимального. |
| TableLegsWidthDistance | 9 | Проверка некорректной передачи значения свойства TableLegsWidthDistance. Меньше минимального. |
| TableLegsWidthDistance | 10000 | Проверка некорректной передачи значения свойства TableLegsWidthDistance. Больше максимального. |
| TableLengthLegsEdgeDistance | -9 | Проверка некорректной передачи значения свойства TableLengthLegsEdgeDistance. Меньше минимального. |

Продолжение таблицы 6.3.

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

Таблица 6.4 - Метод TestSetCorrect\_DefaultParameters.

|  |  |  |
| --- | --- | --- |
| Тестируемое свойство | Параметр | Описание |
| TableHeight | 300, 600, 1200 | Проверки установки через команды SetMinimumValues, SetAverageValues, SetMaximumValues |

Продолжение таблицы 6.4.

|  |  |  |
| --- | --- | --- |
| TableLength | 300, 1000, 2000 | Проверки установки через команды SetMinimumValues, SetAverageValues, SetMaximumValues |
| TableWidth | 300, 1000, 2000 | Проверки установки через команды SetMinimumValues, SetAverageValues, SetMaximumValues |
| TableThikness | 10, 20, 50 | Проверки установки через команды SetMinimumValues, SetAverageValues, SetMaximumValues |
| TableLegsWidthDistance | 290, 990, 1990 | Проверки установки через команды SetMinimumValues, SetAverageValues, SetMaximumValues |
| TableLegsLengthDistance | 290, 990, 1990 | Проверки установки через команды SetMinimumValues, SetAverageValues, SetMaximumValues |

Продолжение таблицы 6.4.

|  |  |  |
| --- | --- | --- |
| TableLengthLegsEdgeDistance | 10 | Проверки установки через команды SetMinimumValues, SetAverageValues, SetMaximumValues |
| TableWidthLegsEdgeDistance | 10 | Проверки установки через команды SetMinimumValues, SetAverageValues, SetMaximumValues |

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 |

Таблица 6.6 - Тестирование класса TableBuilder.

|  |  |
| --- | --- |
| Название теста | Описание |
| TestBuildTable\_DoesNotThrowException | Тестирование построения корректного стола |

# 6.3 Нагрузочное тестирование

В целях проверки производительности работы плагина, было проведено нагрузочное тестирование [12]. Тестирование производилось на ПК со следующей конфигурацией:

* ЦП Intel i5 3.2ГГц;
* 16 ГБ ОЗУ;
* графический процессор объемом памяти 2 ГБ.

На рисунке 6.6 для проведения нагрузочного, с каждым успешным построением фигуры производилась запись результатов в текстовый файл «log.txt».

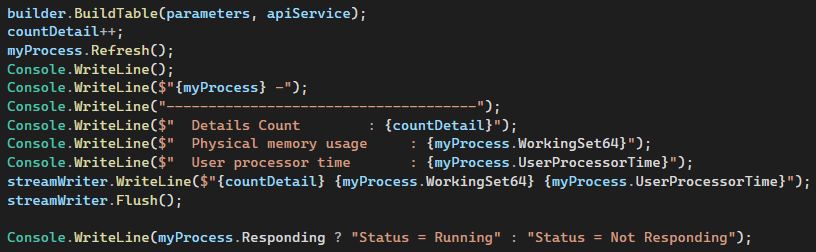


Рисунок 6.5 – Зацикливание построения фигуры.

На графике, изображенном на рисунке 6.6 ось «Х» - количество построенных деталей, ось «Y» - количество потребляемой оперативной памяти. На графике, изображенном на рисунке 6.7, ось «X» – время в минутах, ось «Y» – количество построенных деталей. На протяжении всех тестов общая загруженность процессора была в пределах 38 процентов.



Рисунок 6.6 – График зависимости загруженности памяти от количества деталей

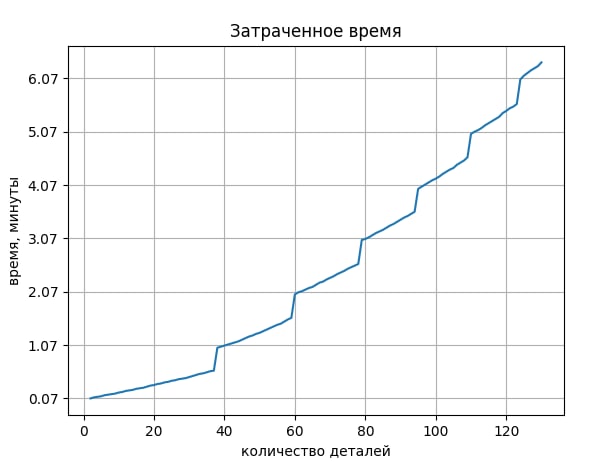


Рисунок 6.7 – График зависимости времени от количества построенных деталей с параметрами по умолчанию.

Исходя из графика, представленного на рисунке 6.6 можно сделать вывод, что зависимость загруженности памяти от кол-ва деталей является почти линейной.

Исходя из графика, представленного на рисунке 6.7 можно сделать вывод, что периодически САПР Kompas 3D начинает строить детали быстрее (резкий вертикальный подъем линии на графике). Можно предположить, что это происходит из-за того, что освобождается память, которая была задействована при открытии и прогрузки предыдущих документов и моделей и, возможно, очистки памяти от различных объектов, в которых больше нет необходимости.

В ходе нагрузочных тестов было замечено, что САПР Kompas 3D, экстренно прекращает свою работу по ошибке, при общем количестве построенных деталей от 120 до 130.

# 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 «Стол» в 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: 24.12.2022).
3. Стол — Википедия. [Электронный ресурс]. — Режим доступа: https://ru.wikipedia.org/wiki/Стол (дата обращения 20.10.2022).
4. SketchUp. [Электронный ресурс]. – Режим доступа: https://www.sketchup.com/ru/ (дата обращения 20.10.2022).
5. КОМПАС-3D. Официальный сайт САПР КОМПАС [Электронный ресурс]. — Режим доступа: https://kompas.ru/ (дата обращения 20.10.2022).
6. КОМПАС-3D для разработчиков [Электронный ресурс]. – URL: https://kompas.ru/solutions/developers/ (дата обращения: 20.10.2022).
7. NUnit [Электронный ресурс]. – URL: <https://nunit.org/> (дата обращения: 11.12.2022).
8. 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: 11.12.2022).
9. UML. [Электронный ресурс]. – Режим доступа: http://www.uml.org/ (дата обращения 11.12.2022).
10. Функциональное тестирование [Электронный ресурс]. – URL: https://daglab.ru/funkcionalnoe-testirovanie-programmnogo-obespechenija/ (дата обращения: 11.12.2022).
11. Юнит-тестирование для чайников [Электронный ресурс]. – URL: https://habr.com/ru/post/169381/ (дата обращения: 11.12.2022).
12. Нагрузочное тестирование: с чего начать и куда смотреть [Электронный ресурс]. – URL: https://habr.com/ru/company/jugru/blog/329174/ (дата обращения: 11.12.2022).