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 «ПИСЬМЕННЫЙ СТОЛ» ДЛЯ «КОМПАС-3D v19»**

Пояснительная записка по лабораторному проекту

по дисциплине «ОСНОВЫ РАЗРАБОТКИ САПР»

«Построение письменного стола в системе КОМПАС-3D v19»

Выполнил:

студент гр.587-3

\_\_\_\_\_\_\_\_\_\_\_\_\_\_Якупов Д. Ф.

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

Руководитель:

к.т.н., доцент каф. КСУП

\_\_\_\_\_\_\_\_\_\_\_\_\_\_Калентьев А. А.

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

Томск 2021

Оглавление

[1 Описание САПР 3](#_Toc36331825)

[2 Описание API 4](#_Toc36331827)

[3 Плагин PDF для САПР КОМПАС-3D 7](#_Toc36331829)

[4 Описание предмета проектирования 8](#_Toc36331830)

[5 Диаграмма классов 10](#_Toc36331834)

[6 Макет пользовательского интерфейса 1](#_Toc36331835)3

[7 Тестирование плагина 1](#_Toc36331835)5

[7.1 Модульное тестирование 1](#_Toc36331835)5

[7.2 Функциональное тестирование 1](#_Toc36331835)6

[7.3 Нагрузочное тестирование 21](#_Toc36331835)

[Заключение](#_Toc36331836) 23

[Список литературы](#_Toc36331836) 24

[Приложение А](#_Toc36331836) 25

# 1 Описание САПР

# В настоящее время проектирование в своем понимании представляет собой автоматизированный процесс и в некотором роде программно-аппаратный. Проектировщику, который занимается разработкой сложного механизма подходят системы автоматизации проектных решений — САПР [1].

САПР позволяют уменьшить финансовые затраты на разработку макета, а также сократить время, которое тратит проектировщик на создание модели объекта и составление проектной документации.

В каждой крупной САПР есть свой средства для разработки, которые предоставляются с целью дать возможность разработчикам расширить функционал данной системы под свои конкретные нужды. Данным средством является API — программный интерфейс приложения [2]. Это набор готовых средств: классов, процедур, функций, структур, констант и т.д. API позволяет определить функциональность, которую предоставляет приложение, при этом абстрагируясь от того, как она реализована.

Расширение функциональности, в основном, подразумевает разработку плагина или библиотеки на основе предоставленного API. Плагин — независимо компилируемый программный модуль, динамически подключаемый к основной программе, предназначенный для расширения или использования ее возможностей [3].

В качестве системы, которая предоставляет API и для которой стоит задача разработать плагин, была выбрана САПР «КОМПАС-3D» версии 19.

КОМПАС-3D – это российская система трехмерного проектирования, ставшая стандартом для тысяч предприятий и десятков тысяч профессиональных пользователей. Система КОМПАС-3D широко используется для проектирования изделий основного и вспомогательного производств в таких отраслях промышленности, как машиностроение, приборостроение, станкостроение и т. д.

# Описание API

Главным интерфейсом API системы КОМПАС-3D является KompasObject. Получить указатель на этот интерфейс можно при работе под управлением внешнего приложения – после вызова стандартной системной функции. Методы этого интерфейса реализуют наиболее общие функции работы с документами системы, системными настройками, файлами, а также дают возможность получить указатели на другие интерфейсы [4].

Ниже в таблицах 1.1-1.3 представлены основные свойства и методы интерфейсов KompasObject, IPart и ksDocument3D.

Таблица 1.1 – Методы интерфейса KompasObject.

|  |  |  |  |
| --- | --- | --- | --- |
| Метод | Входные параметры | Возвращаемое значение | Описание |
| Document3D() |  | Указатель на интерфейс документа трёхмерной модели ksDocument3D | Даёт возможность получить указатель на интерфейс трёхмерного документа (детали или сборки) |
| Visible |  |  | Свойство видимости приложения |
| GetMathematic2D() |  | Указатель на интерфейс ksMathematic2D | Метод для получения указателя на интерфейс для работы с математическими функциями |
| GetParamStruct(short structType) | structType – тип интерфейса параметров | Указатель на интерфейс указаного типа из StructType2D | Метод для получения указателя на интерфейс графического документа (чертежа или фрагмента) |

Продолжение таблицы 1.1

|  |  |  |  |
| --- | --- | --- | --- |
| Метод | Входные параметры | Возвращаемое значение | Описание |
| GetDynamicArray(long type) | ext – расширение имени файла,  filter – фильтр поиска (0 – фильтр формируется автоматически),  preview – признак подключения окна предварительного просмотра:  с подключением окна, без подключения окна  typeDir – стартовая папка | Строка с именем файла | Возвращает указатель на интерфейс динамического массива |

Таблица 1.2 – Методы интерфейса IPart.

|  |  |  |  |
| --- | --- | --- | --- |
| Метод | Входные параметры | Возвращаемое значение | Описание |
| EntityCollection (short objType) | |  |  | | --- | --- | | objType | - тип объектов, содержащихся в массиве. | | Указатель на интерфейс [ksEntityCollection](mk:@MSITStore:D:\INSTAL\KOMPAS-3D%20V17.1\KOMPAS\SDK\SDK.chm::/ksEntityCollection.htm) или [IEntityCollection](mk:@MSITStore:D:\INSTAL\KOMPAS-3D%20V17.1\KOMPAS\SDK\SDK.chm::/ksEntityCollection.htm) | Формирует массив объектов и возвращает указатель на его интерфейс |
| GetDefaultEntity  (short objType) | |  |  | | --- | --- | | objType | - тип объекта. | | |  | | --- | | Указатель на интерфейс [ksEntity](mk:@MSITStore:D:\INSTAL\KOMPAS-3D%20V17.1\KOMPAS\SDK\SDK.chm::/ksEntity.htm) или [IEntity](mk:@MSITStore:D:\INSTAL\KOMPAS-3D%20V17.1\KOMPAS\SDK\SDK.chm::/ksEntity.htm). |   Типы объектов (objType):   |  |  |  | | --- | --- | --- | | o3d\_planeXOY | 1 | - плоскость XOY | | o3d\_planeXOZ | 2 | - плоскость XOZ | | o3d\_planeYOZ | 3 | - плоскость YOZ | | o3d\_pointCS | 4 | - точка начала системы координат | | o3d\_axisOX | 71 | - ось OX | | o3d\_axisOY | 72 | - ось OY | | o3d\_axisOZ | 73 | - ось OZ | | Получить указатель на интерфейс объекта, создаваемого системой по умолчанию |

Продолжение таблицы 1.2

|  |  |  |  |
| --- | --- | --- | --- |
| Метод | Входные параметры | Возвращаемое значение | Описание |
| GetPart(int type) | |  |  | | --- | --- | | type | - тип компонента. | | указатель на интерфейс компонента [ksPart](mk:@MSITStore:D:\INSTAL\KOMPAS-3D%20V17.1\KOMPAS\SDK\SDK.chm::/ksPart.htm) или [IPart](mk:@MSITStore:D:\INSTAL\KOMPAS-3D%20V17.1\KOMPAS\SDK\SDK.chm::/ksPart.htm). | Получить указатель на интерфейс компонента в соответствии с заданным типом |
| NewEntity(short objType) | |  |  | | --- | --- | | objType | - [тип объекта](mk:@MSITStore:D:\INSTAL\KOMPAS-3D%20V17.1\KOMPAS\SDK\SDK.chm::/Obj3dType_NewEntil_Part.htm). | | указатель на интерфейс [ksEntity](mk:@MSITStore:D:\INSTAL\KOMPAS-3D%20V17.1\KOMPAS\SDK\SDK.chm::/ksEntity.htm) или [IEntity](mk:@MSITStore:D:\INSTAL\KOMPAS-3D%20V17.1\KOMPAS\SDK\SDK.chm::/ksEntity.htm). | Создать новый интерфейс объекта и получить указатель на него |

|  |  |  |  |
| --- | --- | --- | --- |
| Метод | Входные параметры | Возвращаемое значение | Описание |
| Create (bool invisible, bool \_typeDoc) | invisible – признак режима редактирования документа  (TRUE – невидимый режим,  FALSE – видимый режим),  typeDoc – тип документа  (TRUE – деталь,  FALSE – сборка). | TRUE – в случае успешного завершения. | Дает возможность создать пустой документ (деталь или сборку) |
| UpdateDocumentParam() |  | TRUE – в случае успешного завершения. | Активизировать измененные параметры документа |
| GetPart(int type) | type – тип компонента из перечисления Типы компонентов. |  | Получить указатель на интерфейс компонента в соответствии с заданным типом |

Таблица 1.3 Методы интерфейса ksDocument3D.

# Обзор аналогов плагина

# Плагин PDF для САПР КОМПАС-3D.

Данный плагин позволяет производить экспорт моделей и сборок из КОМПАС-3D в формат PDF формат [5]. 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. Ключевые возможности:

* сохранение деталей и сборок в формате 3D PDF для интерактивного просмотра при помощи 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 предмета проектирования

Предметом проектирования является письменный стол.

Письменные столы используются практически всеми людьми, это один из самых востребованных видов мебели. Зачастую, письменные столы используются не только как предмет, на который можно разложить тетради или учебники, но и вместо специализированных компьютерных столов. Даже в недорогие письменные столы имеют специальное отверстие в столешнице, которое позволяет удобно провести провода, при использовании монитора, ноутбука или просто зарядки телефона.

В большинстве случаем, столы имеют несложную конструкцию, имеют квадратную или прямоугольную столешницу, могут иметь несколько разновидностей ножек. В зависимости от желания человека (покупателя), столы могут иметь отверстие под провода, различное сочетание цветов и оттенков.

Параметры стола:

* Длина столешницы L (от 1000 до 2000 мм);
* Ширина столешницы S (от 600 до 800 мм);
* Высота столешницы H (от 30 до 40 мм);
* Количество ножек n (от 4 до 5 шт);
* Высота ножек h1 (от 600 до 700 мм);
* Наличие или отсутствие сквозного отверстия для проводов;
* Радиус сквозного отверстия для проводов r (от 20 до 30 мм);
* Вид ножек (с круглым или квадратным основанием);

Плагин имеет зависимые параметры:

* Количество ножек n = 5, при L = 2000 мм;
* Расстояние от верхнего левого угла столешницы по длине стола до центра окружности сквозного отверстия a ≤ L – r – 100 мм;
* Расстояние от верхнего левого угла столешницы по ширине стола до центра окружности сквозного отверстия b ≤ S – r – 70 мм;
* Внешний диаметр ножек D (от 40 до 60 мм), если выбрано круглое основание;
* Длина ребра основания ножки A (от 40 до 60 мм), если выбрано квадратное основание.

Пример модели приведен на рисунке 4.1.



Рисунок 4.1 – 3D модель письменного стола с параметрами

# 5 Диаграмма классов

Диаграмма классов описывает типы объектов системы и различного рода статические отношения, которые существуют между ними. На диаграммах классов отображаются также свойства классов, операции классов и ограничения, которые накладываются на связи между объектами [7]. На рисунке 5.1 представлена диаграмма классов.

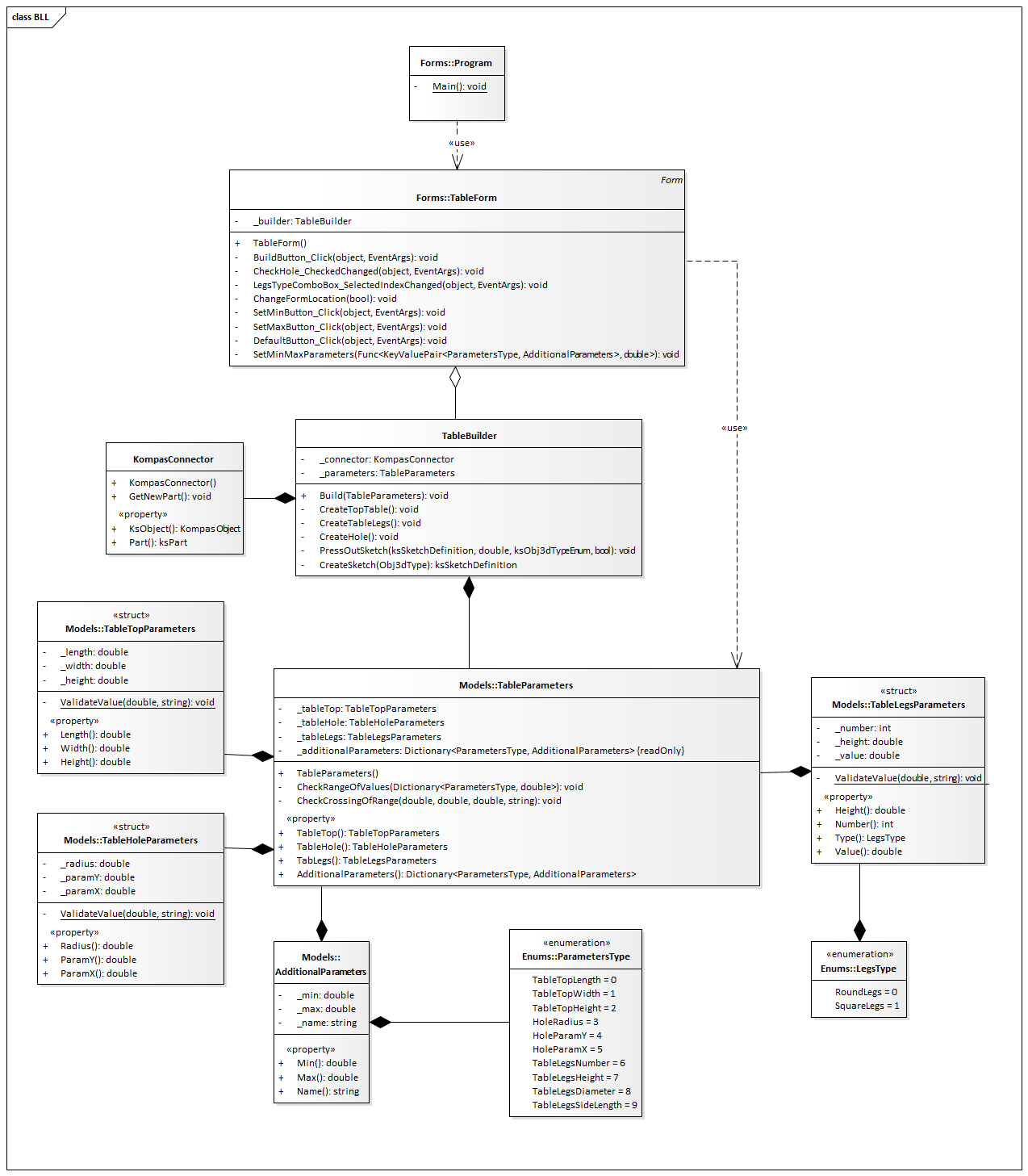


Рисунок 5.1 – Диаграмма классов

Для реализации был выбран следующий набор классов и структур:

* TableForm – класс диалогового окна, который обеспечивает взаимодействие между пользователем и программой;
* TableParameters − класс, хранящий в себе все параметры проектируемой 3D-модели, осуществляет проверку зависимых параметров;
* AdditionalParameters − класс, хранящий в себе допольнительные параметры 3D-модели;
* KompasConnector – класс для работы с API КОМПАС 3D.
* TableBuilder – класс, осуществляющий вызов методов API, необходимых для постройки 3D-модели.
* TableTopParameters – структура данных, хранящая в себе параметры столешницы.
* TableHoleParameters – структура данных, хранящая в себе параметры функционального отверстия для проводов.
* TableLegsParameters – структура данных, хранящая в себе

На рисунке 5.2 изображена диаграмма классов приложения после завершения разработки дополнительного функционала и рефактронига.

После добавления новой функциональности и рефакторинга в структуре приложения произошли следующие изменения:

* Было добавлено создание нескольких выдвижных ящиков, где количество указывается отдельным свойством TableBoxNumber в классе TableParameters.
* В Builder был добавлены методы CreateBox и BuildBox, для создание ящиков в зависимости от их количества.
* Был произведен рефакторинг методов валидации. Общие методы проверки были вы перенесены в отдельный статический класс ParametersValidation.
* В класс TableParameters был добавлен метод SetMinMaxParameters для установки минимальных/максимальных параметров в форме. Это позволило уменьшить количество повторяющего кода.

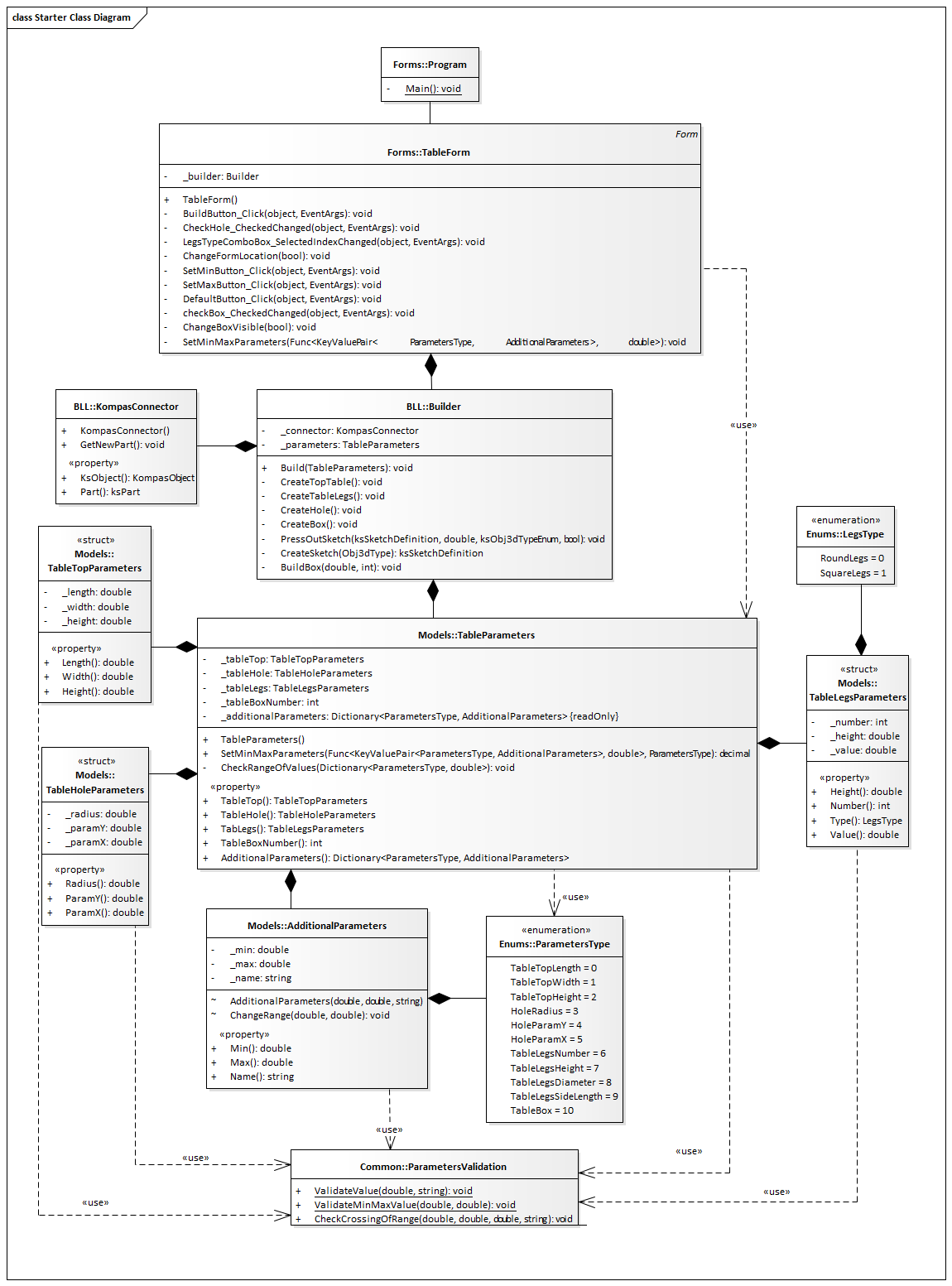


Рисунок 5.1 – Диаграмма классов после добавления функциональности

# 6 Макет пользовательского интерфейса

Макет пользовательского интерфейса представляет собой форму для ввода параметров. Построение модели осуществляется нажатием на кнопку «Построить 3D модель». На рисунках 6.1 и 6.2 представлен макет пользовательского интерфейса.

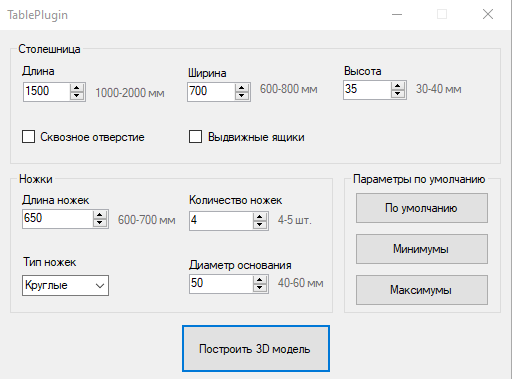


Рисунок 6.1 – Макет пользовательского интерфейса

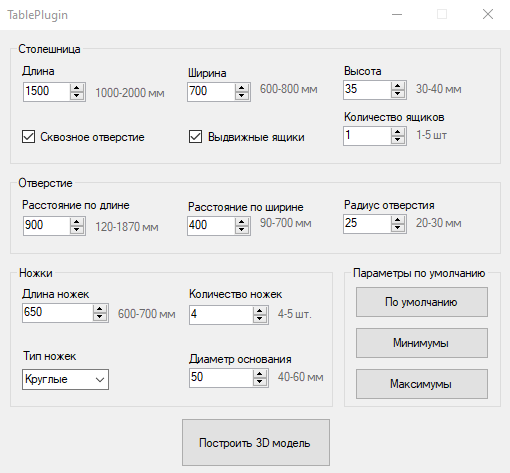


Рисунок 6.2 – Макет пользовательского интерфейса c включенным сквозным отверстием

После ввода некорректных параметров и нажатия кнопки построения модели отображается окно предупреждения с описанием ошибок (рис. 6.3).

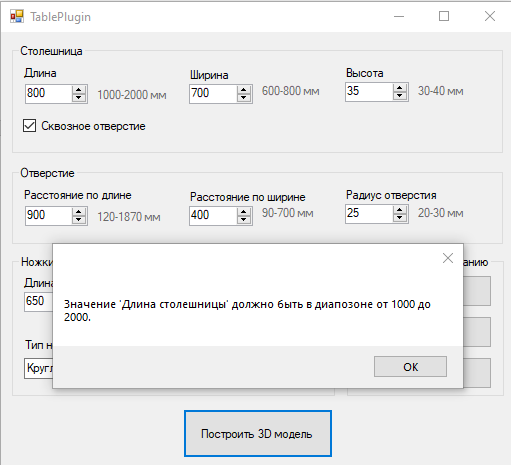


Рисунок 6.3 – Предупреждение при вводе несовместимых параметров

# 7 Тестирование плагина

# 7.1 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 и выполнить рефакторинг на уменьшение связности классов. [7]

Тестирование проводилось с помощью фреймворка модульного тестирования NUnit 3.13.1. Тестовые сценарии приведены в приложении А. На рисунке 7.1 показаны результаты тестов, а на рисунке 7.2 показана степень покрытия кода тестами.

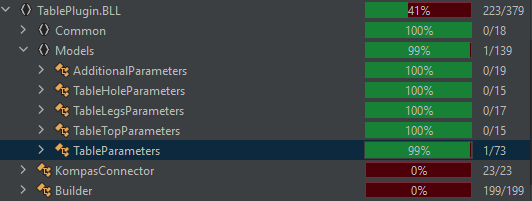


Рисунок 7.1 – Результаты модульного тестирования

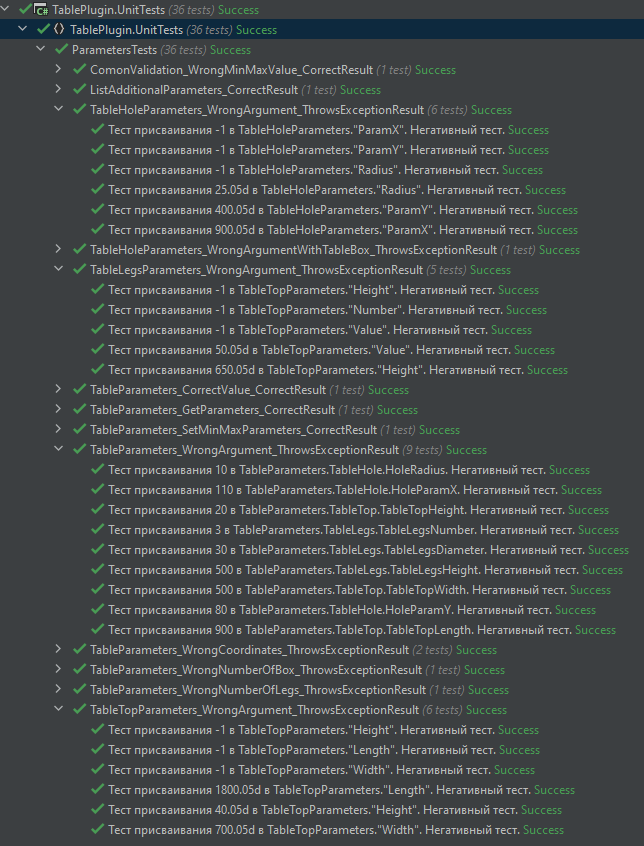


Рисунок 7.2 – Результат проверки покрытия кода тестами

## 

## **7.2 Функциональное тестирование**

Функциональное тестирование – это тестирование функциональности и поведения программы на соответствие требованиям функциональной спецификации. Функциональная спецификация определяет, что именно делает ПО, какие задачи оно решает. Для проведения функционального тестирования необходимо проверить правильность построения детали при минимальных и максимальных значениях входных параметров.

Минимальные входные параметры:

* Длина столешницы = 1000 мм;
* Ширина столешницы = 600 мм;
* Ширина столешницы = 30 мм;
* Длина ножек = 600 мм;
* Количество ножек = 4 шт;
* Диаметр/длина стороны основания ножек = 40 мм;
* Расстояние до центра отверстия по длине столешницы = 120 мм;
* Расстояние до центра отверстия по ширине столешницы = 90 мм;
* Радиус отверстия = 20 мм;
* Количество ящиков = 1 шт.

Максимальные входные параметры:

* Длина столешницы = 200 мм;
* Ширина столешницы = 800 мм;
* Ширина столешницы = 340 мм;
* Длина ножек = 700 мм;
* Количество ножек = 5 шт;
* Диаметр/длина стороны основания ножек = 60 мм;
* Расстояние до центра отверстия по длине столешницы = 1870 мм;
* Расстояние до центра отверстия по ширине столешницы = 700 мм;
* Радиус отверстия = 30 мм;
* Количество ящиков = 5 шт.

Результаты построения при минимальных и максимальных параметрах, а также новая функциональность показаны на рисунках 7.3 – 7.8.

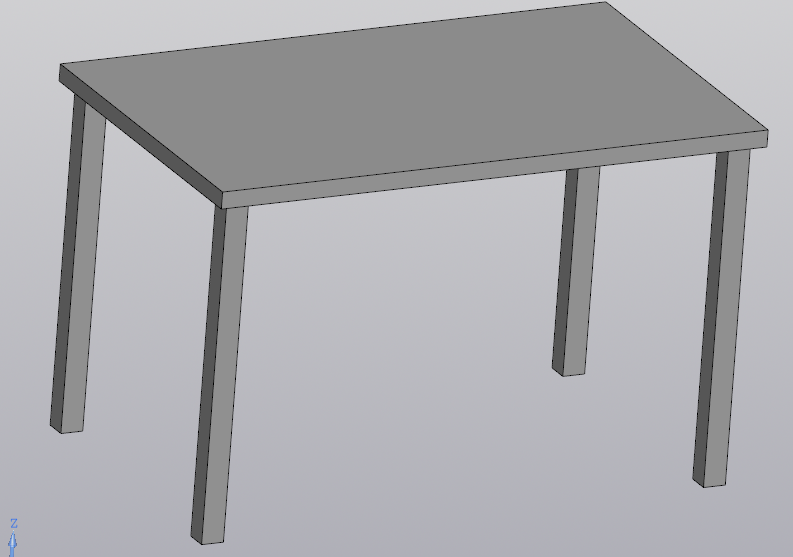


Рисунок 7.3 – Построение стола с квадратными ножками

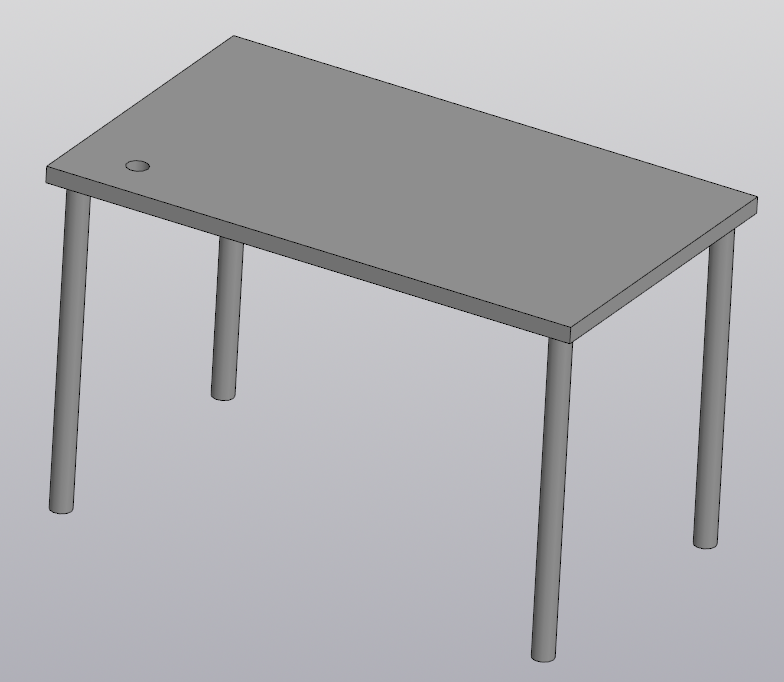


Рисунок 7.4 – Построение стола с круглыми ножками, отверстием и минимальными параметрами

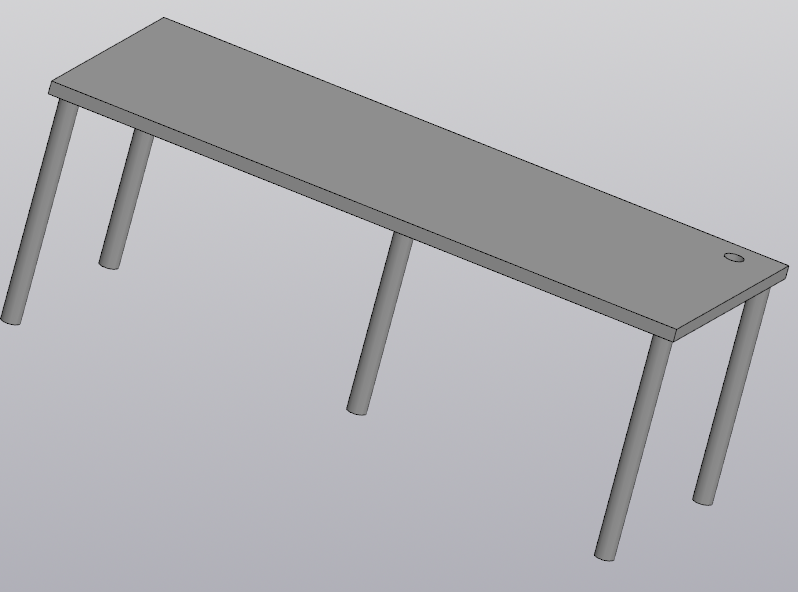


Рисунок 7.5 – Построение стола с круглыми ножками, отверстием и максимальными параметрами

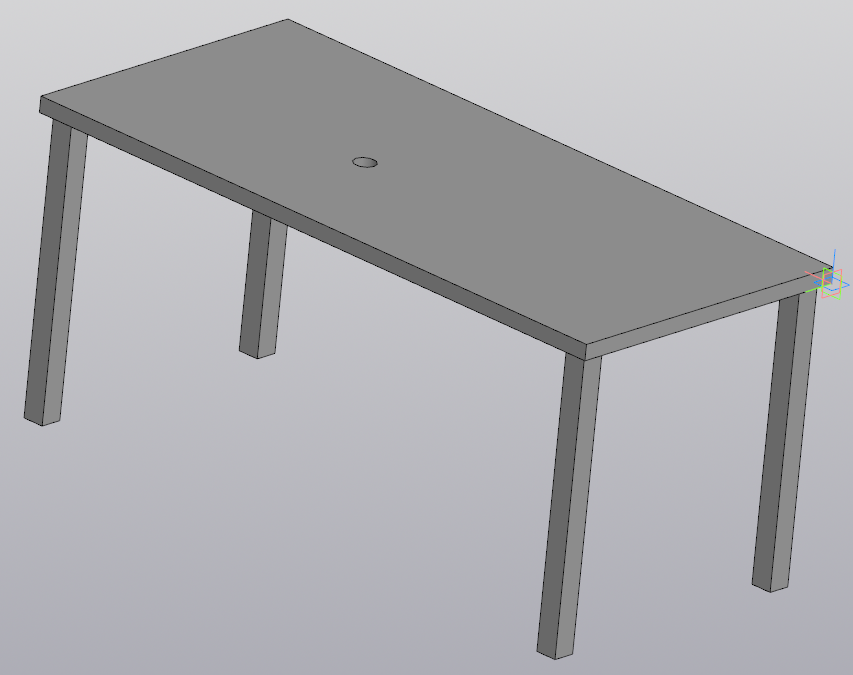


Рисунок 7.6 – Построение стола с квадратными ножками, отверстием и параметрами по умолчанию

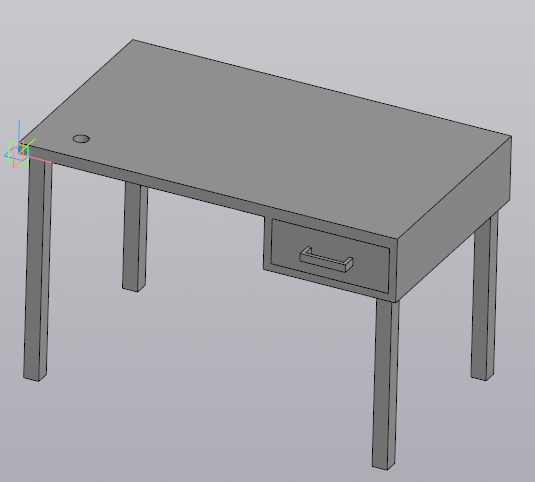


Рисунок 7.7 – Построение стола с квадратными ножками, отверстием, выдвижным ящиком и минимальными параметрами

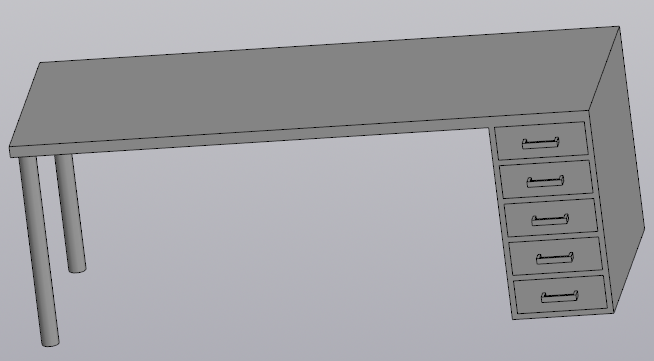


Рисунок 7.8 – Построение стола с круглыми ножками, 5 выдвижными ящиками и максимальными параметрами

## **7.3 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.[8]

Тестирование проводилось на компьютере с следующими техническими параметрами:

Процессор: AMD Ryzen 5 3600 6-Core Processor 3.60 GHz

Оперативная память: 16Gb DDR4 DIMM 3600MHz

Видеоадаптер: NVIDIA GeForce GTX 750

Операционная система: Windows 10 Pro 64-bit

На рисунке 7.9 представлено затраченное время при построении 200 столов, а на рисунке 7.10 количество ОЗУ потребляемое при построение.

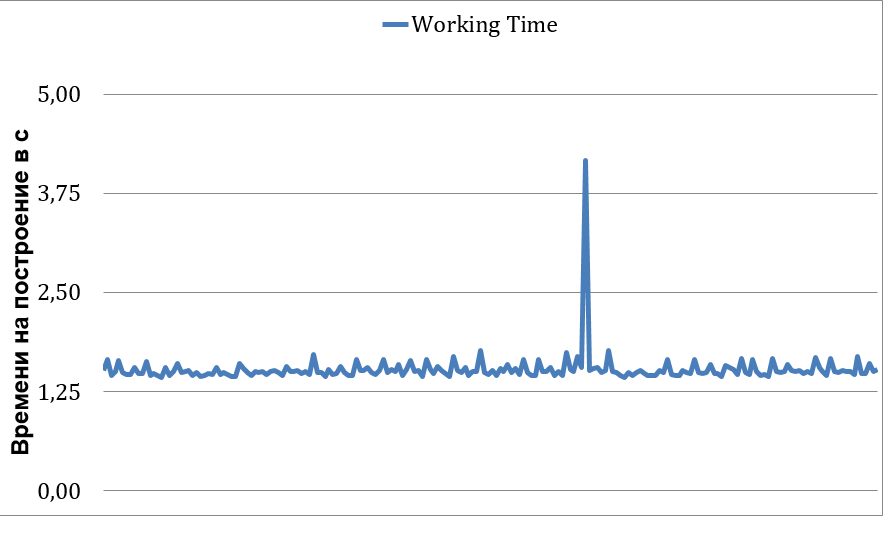


Рисунок 7.9 – Затраченное время (в секундах) на построение 200 столов

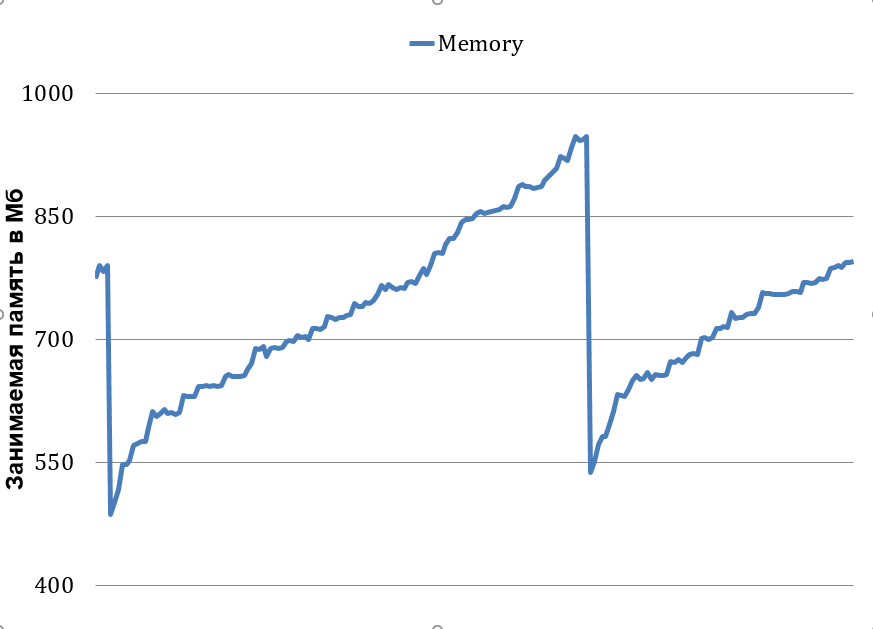


Рисунок 7.10 – Затраченное ОЗУ на построение 200 столов

На построение 200 столов примерно было затрачено чуть более 5 минут. Среднестатистическое время построение стола составляет 1,5 секунды. Высокая скорость построения столов объясняется тем, что модель стола имеет немного деталей и содержит легкие операции. Максимальное количество потребляемой оперативной памяти при работе приложения составило 947 Мб. При 5 и 131 построением стола происходит резкое уменьшение потребление оперативной памяти, предположительно из-за работы сборщика мусора. Скорость построения 125 стола сильно ниже (рис. 7.9), в сравнение с другими. Предположительно, это связано с началом работы сборщика мусора или же временной загрузкой компьютера другими системными процессами.

# 8 Заключение

В ходе курса ОРСАПР был изучен API Компас3D, основные этапы проектирования и составления проекта системы плагина, изучена предметная область объекта построения. Были составлены такие документы как техническое задание, проект системы и пояснительная записка. Были составлены UML диаграммы классов. Также была Проверена работа плагина на ОС Windows 10.

Результатом работы является плагин для САПР «КОМПАС-3D». Который выполняет построение этажерки для обуви с различными переменными параметрами.

# Список литературы

Норенков И.П. «Основы автоматизированного проектирования». Издательство: МГТУ; Москва:, 2002 – 336 с.

API – Википедия. [Электронный ресурс]. — Режим доступа: <https://ru.wikipedia.org/wiki/API> (дата обращения 13.04.2021)

Плагин – Википедия. [Электронный ресурс]. – Режим доступа: <https://ru.wikipedia.org/wiki/Плагин> (дата обращения 30.03.2021)

Кидрук Максим. КОМПАС-3D V10 на 100% / М. Кидрук. – СПб.: Питер, 2009 – 560 с.

Экспорт в формате 3D PDF из КОМПАС-3D. [Электронный ресурс]. – Режим доступа: <http://isicad.ru/ru/news.php?news=16278> (дата обращения 14.04.2021)

Новые технологии в программировании: учебное пособие / А.А.Калентьев, Д.В.Гарайс, А.Е.Горяинов— Томск: Эль Контент, 2014.—176 с.

Фаулер М. UML. Основы. 3-е издание / М.Фаулер. – 3-е изд., пер. с англ. ­– СПб.: Символ-Плюс, 2019. – 192 с.

# Приложение А

Таблица А ̶ Тестовые сценарии

|  |  |  |
| --- | --- | --- |
| Метод | Входные параметры | Описание тестового случая |
| TableParameters  \_CorrectValue  \_CorrectResult | “” | Тест присваиваения параметров в TableParameters. Позитивный тест. |
| TableParameters  \_GetParameters  \_CorrectResult | “” | Тест получения параметров в класс TableParameters. Позитивный тест. |
| ListAdditionalParameters  \_CorrectResult | “” | Тест на получение списка с дополнительными параметрами. Позитивный тест. |
| TableParameters  \_SetMinMaxParameters  \_CorrectResult | “” | Тест на установку минимальных и максимальных значений параметров. Позитивный тест. |
| TableHoleParameters  \_WrongArgument  \_ThrowsExceptionResult | “25.05d”, "Radius" | Тест присваивания 25.05d в TableHoleParameters. Radius. Негативный тест. |
| “25.05d” , "ParamX" | Тест присваивания 25.05d в TableHoleParameters. ParamX. Негативный тест. |

Продолжение таблицы А

|  |  |  |
| --- | --- | --- |
| Метод | Входные параметры | Описание тестового случая |
| TableHoleParameters  \_WrongArgument  \_ThrowsExceptionResult | “400.05d”, "ParamY" | Тест присваивания 400.05d в TableHoleParameters. ParamY. Негативный тест. |
| “-1” , "Radius" | Тест присваивания -1 в TableHoleParameters. Radius. Негативный тест. |
| “-1” , "ParamX" | Тест присваивания -1 в TableHoleParameters. ParamX. Негативный тест. |
| “-1” , "ParamY" | Тест присваивания -1 в TableHoleParameters. ParamY. Негативный тест. |
| TableHoleParameters  \_WrongArgumentWithTableBox  \_ThrowsExceptionResult | “” | Тест присваивания неправильного значения в TableHoleParameters.ParamX при наличие выдвижных ящиков. Негативный тест. |
| TableTopParameters  \_WrongArgument  \_ThrowsExceptionResult | “1800.05d”, "Length" | Тест присваивания 1800.05d в TableTopParameters.Length. Негативный тест. |
| “700.05d” , "Width" | Тест присваивания 700.05d в TableTopParameters.Width. Негативный тест. |

Продолжение таблицы А

|  |  |  |
| --- | --- | --- |
| Метод | Входные параметры | Описание тестового случая |
| TableTopParameters  \_WrongArgument  \_ThrowsExceptionResult | “40.05d”, "Height " | Тест присваивания 40.05d в TableTopParameters. Height. Негативный тест. |
| “-1” , "Length" | Тест присваивания -1 в TableTopParameters.Length. Негативный тест. |
| “-1” , "Width" | Тест присваивания -1 в TableTopParameters.Width. Негативный тест. |
| “-1” , "Height " | Тест присваивания -1 в TableTopParameters.Height. Негативный тест. |
| TableLegsParameters  \_WrongArgument  \_ThrowsExceptionResult | “650.05d” , "Height" | Тест присваивания 700.05d в TableLegsParameters. Height. Негативный тест. |
| “50.05d”, "Value" | Тест присваивания 40.05d в TableLegsParameters. Value. Негативный тест. |
| “-1” , "Height " | Тест присваивания -1 в TableLegsParameters.Length. Негативный тест. |
| “-1” , "Number" | Тест присваивания -1 в TableLegsParameters.Number. Негативный тест. |

Продолжение таблицы А

|  |  |  |
| --- | --- | --- |
| Метод | Входные параметры | Описание тестового случая |
| TableLegsParameters  \_WrongArgument  \_ThrowsExceptionResult | “-1” , "Value" | Тест присваивания -1 в TableLegsParameters.Value. Негативный тест. |
| TableParameters  \_WrongArgument  \_ThrowsExceptionResult | “900”, "TableTopLength" | Тест присваивания 900 в TableParameters.TableTop.{1}. Негативный тест. |
| “500” , "TableTopWidth" | Тест присваивания 900 в TableParameters.TableTop.{1}. Негативный тест. |
| “20” , "TableTopHeight" | Тест присваивания 900 в TableParameters.TableTop.{1}. Негативный тест. |
| “500” "TableLegsHeight" | Тест присваивания 900 в TableParameters. TableLegs. Негативный тест. |
| “3” , "TableLegsNumber" | Тест присваивания 900 в TableParameters. TableLegs.{1}. Негативный тест. |
| “30” , "TableLegsDiameter" | Тест присваивания 900 в TableParameters. TableLegs.{1}. Негативный тест. |

Продолжение таблицы А

|  |  |  |
| --- | --- | --- |
| Метод | Входные параметры | Описание тестового случая |
| TableParameters  \_WrongArgument  \_ThrowsExceptionResult | “10” "HoleRadius" | Тест присваивания 900 в TableParameters. TableHole.{1}. Негативный тест. |
| “110"HoleParamX" | Тест присваивания 900 в TableParameters. TableHole.{1}. Негативный тест. |
| “80” , "HoleParamY" | Тест присваивания 900 в TableParameters. TableHole.{1}. Негативный тест. |
| TableParameters  \_WrongNumberOfLegs  \_ThrowsExceptionResult | “” | Тест на присваивание неправильного количества ножек. Негативный тест. |
| TableParameters  \_WrongNumberOfBox  \_ThrowsExceptionResult | “” | Тест на присваивание неправильного количества ящиков. Негативный тест. |
| TableParameters  \_WrongCoordinates  \_ThrowsExceptionResult | “1000”, ParametersType  .HoleParamX”, "Расстояние по длине", new[] { 935, 1065 } | Тест на пересечение недопустимого диапозона в поле {3}. Негативный тест." |

Продолжение таблицы А

|  |  |  |
| --- | --- | --- |
| Метод | Входные параметры | Описание тестового случая |
|  | “350”, “ParametersType. HoleParamY”, "Расстояние по ширине", new[] { 285, 415 } | Тест на пересечение недопустимого диапозона в поле {3}. Негативный тест. |
| ComonValidation  \_WrongMinMaxValue  \_CorrectResult | “” | Тест проверки присваивания неправильных min/max значений. Позитивный тест. |