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)

Студент гр. 586-1

\_\_\_\_\_\_\_\_\_\_\_\_\_А. Ахметов

«\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2020г.

Руководитель

к.т.н., доцент каф. КСУП

\_\_\_\_\_\_\_\_\_\_А.А. Калентьев

«\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2020г.

2020

**РЕФЕРАТ**

Пояснительная записка 34 с., 23 рис., 1 таблица, 11 источников.

КОМПАС-3D, ПЛАГИН, МОДЕЛЬ РЮМКИ, АВТОМАТИЗИРОВАННОЕ ПРОЕКТИРОВАНИЕ, САПР.

Целью данной работы является разработка плагина, автоматизирующего построение модели рюмки, для системы автоматизированного проектирования КОМПАС-3D 18.1, с помощью интегрированной среды разработки Visual Studio Сommunity 2019.

В процессе работы изучено внешнее строение рюмки и основные функции системы автоматизированного проектирования «Компас-3D» v 18.1.

В результате работы был создан плагин, автоматизирующий построение рюмки.

Отчет по пояснительной записке выполнен в текстовом редакторе Microsoft Word 2016.

**Оглавление**

[1 Введение 4](#_Toc40433153)

[2 Постановка и анализ задачи 5](#_Toc40433154)

[2.1 Описание предмета проектирования 5](#_Toc40433155)

[2.2 Выбор инструментов и средств реализации 7](#_Toc40433156)

[2.3 Назначение плагина 8](#_Toc40433157)

[2.4 Описание аналогов разрабатываемого продукта 8](#_Toc40433158)

[2.4.1 Оборудование: Трубопроводы 9](#_Toc40433159)

[2.4.2 Механика: Пружины 10](#_Toc40433160)

[3 Описание реализации 12](#_Toc40433161)

[3.1 Диаграмма прецедентов плагина 13](#_Toc40433162)

[3.2 Диаграмма классов 15](#_Toc40433163)

[4 Описание программы для пользователя 17](#_Toc40433164)

[5 Тестирование программы 21](#_Toc40433165)

[5.1 Функциональное тестирование 21](#_Toc40433166)

[5.2 Модульное тестирование 25](#_Toc40433167)

[5.3 Нагрузочное тестирование 26](#_Toc40433168)

[Заключение 29](#_Toc40433169)

[Список использованных источников 30](#_Toc40433170)

[Приложение А 31](#_Toc40433175)

# 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.

Практическая реализация методов и идей автоматизированного моделирования происходит в рамках систем автоматизированного проектирования (САПР). Однако мало создать высокопроизводительные современные САПР. Надо уметь их эффективно использовать. Для этого нужны квалифицированные инженеры-пользователи САПР. В рамках современного «компьютеризированного» общества инженер любой специальности, занимающийся разработкой технических устройств, должен уметь использовать средства автоматизированного проектирования. Это позволяет повысить эффективность моделирования, улучшить его качество, снизить материальные затраты и уменьшить число разработчиков.

Таким образом, целью данной работы является разработка плагина, автоматизирующего построение модели рюмки, для системы автоматизированного проектирования «Компас-3D» v 18.1 [1], с помощью интегрированной среды разработки Visual Studio Сommunity 2019 [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 [1], строит объект «Рюмка». Более того, требовалось, чтоб плагин позволял изменять входные параметры в соответствии с требованиями пользователей, а именно: диаметр подставки, диаметр ножки, диаметр скругления подставки и ножки, высота ножки, диаметр бокала, высота нижнего бокала, диаметр горлышка, высота верхнего бокала.

# 2.1 Описание предмета проектирования

Рюмка – небольшой, чаще всего стеклянный или хрустальный сосуд на ножке [3]. Используется для употребления спиртных напитков. Существуют несколько разновидностей рюмок, предназначенных для тех или иных напитков. Ниже, на рисунке 2.1 представлены виды рюмок и для чего они предназначены.



Рисунок 2.1 – Виды рюмок

На рисунке 2.2 представлен чертеж рюмки с обозначенными параметрами D1, D2, D3, H1, D4, H2, D5, H3.

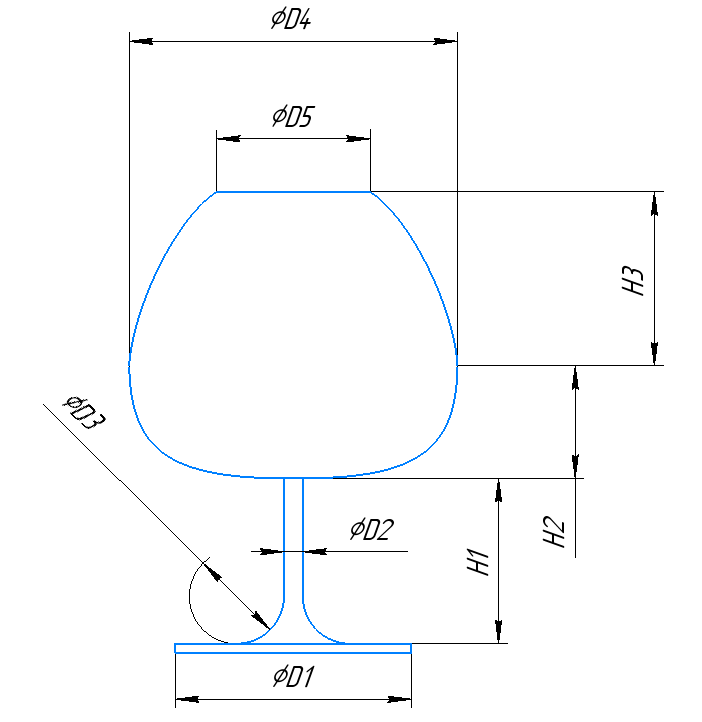


Рисунок 2.2 – Чертеж рюмки с обозначенными параметрами

Обозначенные параметры:

D1 – диаметр подставки;

D2 – диаметр ножки;

D3 – крутизна скругления подставки и ножки;

H1 – высота ножки;

D4 – диаметр бокала;

H2 – расстояние между вершиной ножки и диаметром бокала;

D5 – диаметр горлышка;

H3 – расстояние между диаметрами бокала и горлышка;

На рисунке 2.3 представлена 3D-модель рюмки

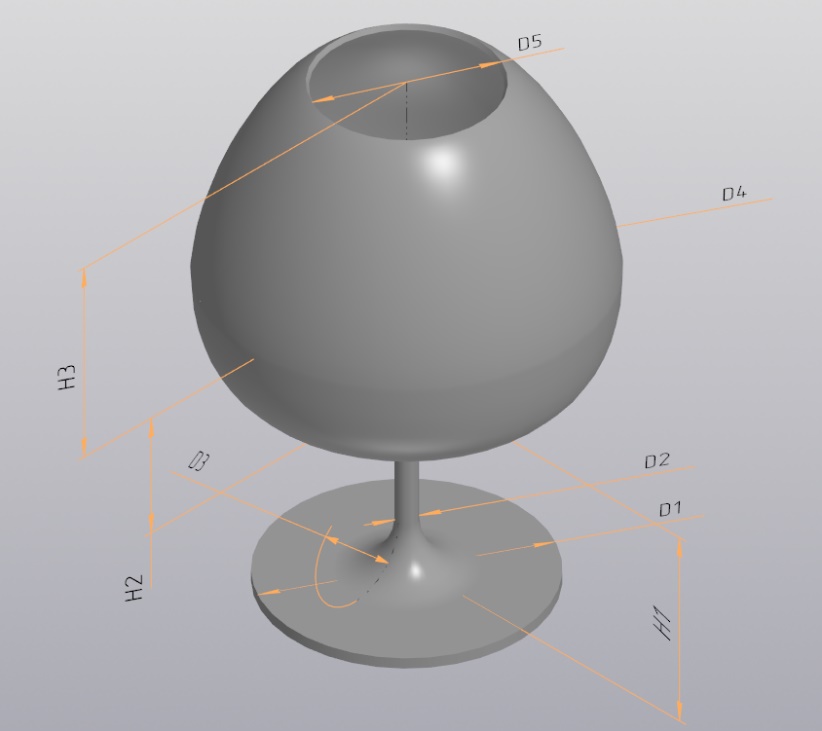


Рисунок 2.3 – 3D-модель рюмки

Для рюмок существует ГОСТ 5.121-69 – Изделия из бесцветного хрусталя. Требования к качеству аттестованной продукции [4].

# 2.2 Выбор инструментов и средств реализации

В связи с требованием технического задания программа выполнена на языке C# в среде Microsoft Visual Studio 2019 [2], с использованием .NET Framework 4.7.0, для системы КОМПАС 3D V18.1. Инструментом тестирования и создания модульных тестов был выбран стандартный обозреватель тестов среды Microsoft Visual Studio 2019 [2] с тестовым фреймворком NUnit версии 3.12.0.

Для реализации пользовательского интерфейса использовался WinForm.

Взаимодействие плагина с системой КОМПАС 3D [1] осуществляется посредством интерфейсов, называемых API. В КОМПАС 3D на данный момент существует API двух версий API 5 [5] и API 7 [5]. Явных преимуществ между версиями нет, поскольку обе версии реализуют различные функции системы и взаимно дополняют друг друга. Для выполнения лабораторных была выбрана версия API 5, так как для полноценной реализации плагина «Рюмка» достаточно методов и свойств интерфейсов API 5.

# 2.3 Назначение плагина

Назначение, разрабатываемого плагина обусловлено выбором сферы его применения. В данном проекте выбором сферы применения стало употребление спиртных напитков, так как данная сфера, одна из наиболее популярных, ввиду повышенного спроса на алкогольную продукцию. Для поддержания пользовательского интереса и спроса на продукцию, производителям необходимо часто выпускать товар с обновленным дизайном.

# 2.4 Описание аналогов разрабатываемого продукта

КОМПAС-3D [1] является открытой системой, что позволяет создавать дополнительные программные модули (пользовательские библиотеки) и применять их во время работы над документами. Таким образом, стандартные возможности чертежно-графического редактора и трехмерного моделирования могут быть дополнены исходя из тех специальных задач, которые приходится решать пользователю.

В настоящее время система КОМПAС-3D непрерывно расширяется плагинами и модулями, которые можно разделить на два условных класса: плагины или модули, реализующие (моделирующие) конкретные трехмерный объекты в зависимости от сферы применения.

# 2.4.1 Оборудование: Трубопроводы

Оборудование: Трубопроводы [6] – специализированное приложение системы КОМПАС-3D, предназначенное для быстрого проектирования гидравлических и пневматических систем, различных инженерных коммуникаций, обвязок машин и оборудования, автоматического создания комплекта документации для изготовления трубопроводов.

Оборудование: Трубопроводы избавляет проектировщика от выполнения рутинных действий и исключает подавляющее большинство ошибок. Приложение позволяет строить трубопроводы по самым сложным траекториям.

В процессе построения траектории автоматически строится и сам трубопровод, для задания параметров которого предусмотрена специальная панель. В зависимости от геометрии осевой линии выполняется автоматический подбор деталей трубопровода. Если траектория совершает поворот, автоматически размещается отвод либо сгиб трубы, а если труба пересекается с другой трубой, в этом месте появляется тройник либо врезка. Радиусы поворотов определятся автоматически в зависимости от выбранного диаметра трубы

Трубопровод отображается в дереве построения модели отдельным объектом, что делает навигацию по проекту более удобной.

Приложение также позволяет размещать элементы трубопроводов (запорную арматуру, фланцы, фильтры, клапаны и т. д.), подрезать торцы труб, редактировать диаметр труб и толщину стенки.

В качестве деталей и арматуры трубопровода можно использовать как компоненты из каталога приложения, так и компоненты из Библиотеки Стандартные Изделия. Работа конструктора значительно облегчается в части составления проектной документации. Все добавленные в модель элементы автоматически переносятся в чертежи, спецификации, отчеты.

На рисунке 2.4 представлен пример настройки параметров стиля трубопровода.

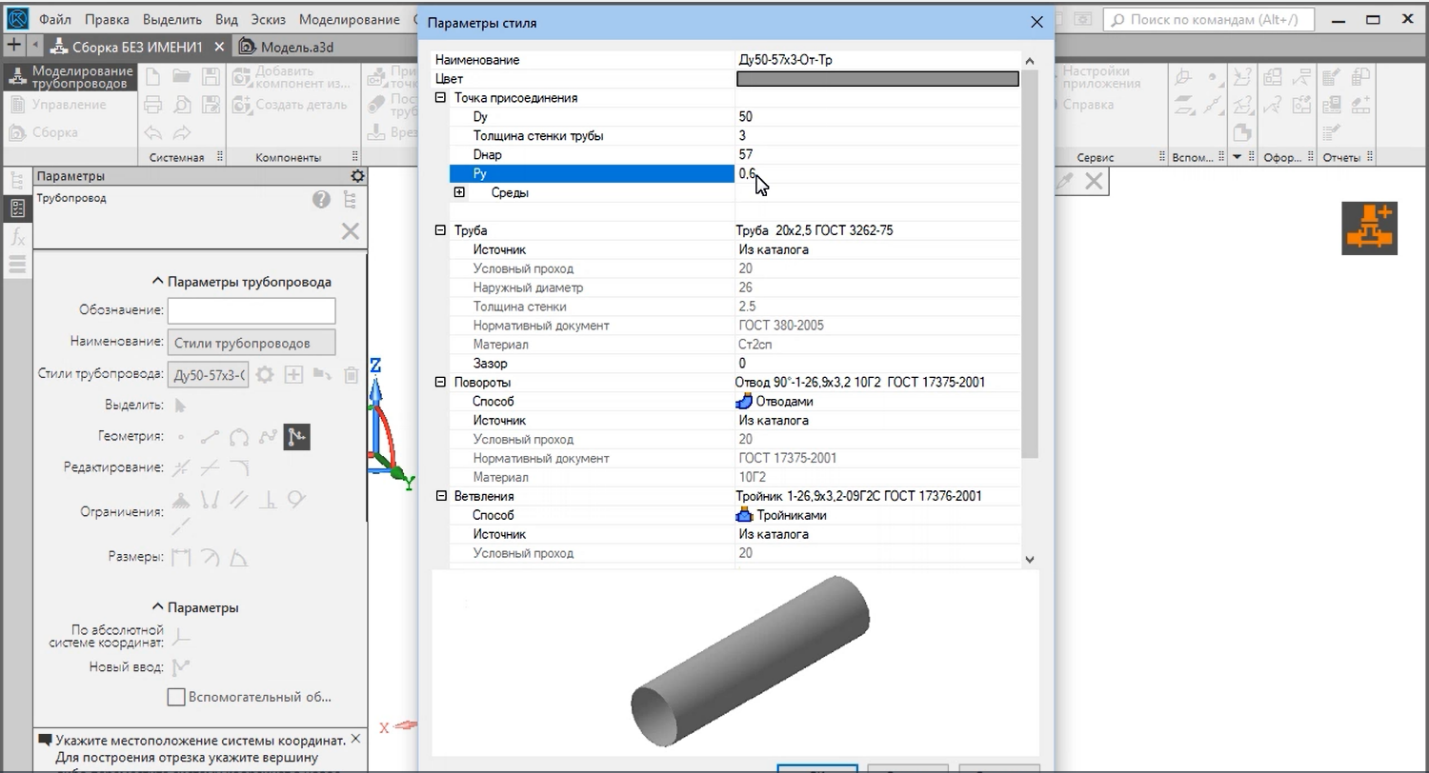


Рисунок 2.4 – Настройка параметров стиля трубопровода.

# 2.4.2 Механика: Пружины

Приложение позволяет выполнять проектные и проверочные расчеты пружин сжатия, растяжения, кручения, а также тарельчатых, конических и фасонных пружин. По результатам расчетов автоматически формируются чертежи и 3D-модели [7].

В основу приложения положены следующие методики расчета:

* пружины сжатия и растяжения — ГОСТ 13764-86, ГОСТ 13765-86;
* тарельчатые пружины — ГОСТ 3057-90;
* пружины кручения — методика из книги В.И. Анурьев ”Справочник конструктора-машиностроителя” том 3;
* конические и фасонные пружины — методика из книги С. Д. Пономарёв, Л. Е. Андреева «Расчет упругих элементов машин и приборов».

В результате проектного расчета система предлагает множество решений, удовлетворяющих исходным данным, из которых конструктор может выбрать оптимальное по одному или нескольким критериям.

При создании чертежа пружины возможны выбор типа зацепов, автоматическая постановка размеров, выносных видов, диаграмм деформаций или усилий.

После вставки модели пружины в сборку можно изменять длину пружины, что позволяет выставить деталь в рабочее состояние или промежуточное.

Как показывает практика, использование приложения Пружины позволяет в 15–20 раз повысить скорость проектирования и выпуска конструкторской документации пружин.

На рисунке 2.5 представлено проектирование пружины

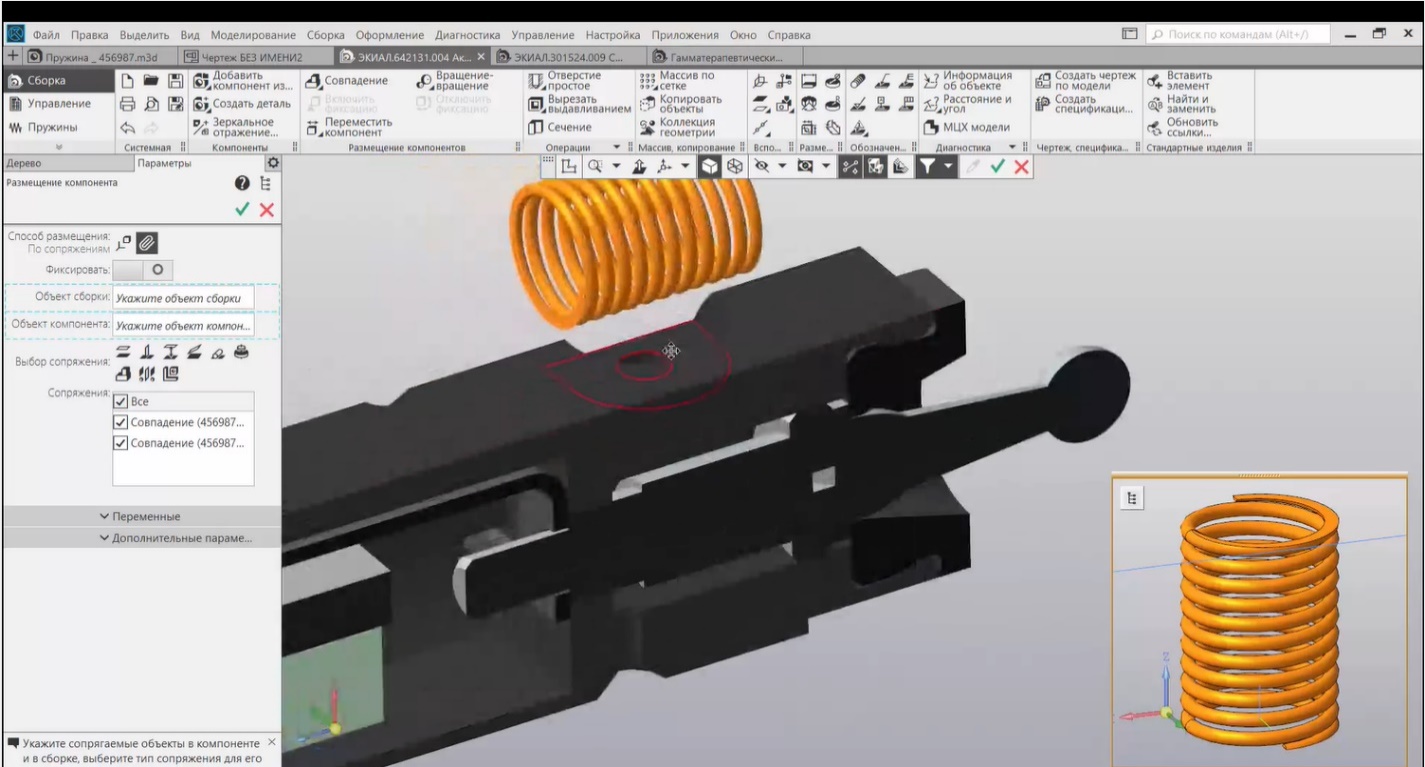


Рисунок 2.5 – Проектирование пружины

# 3 Описание реализации

На этапе разработки проекта программы для формального описания архитектурной особенности, пользовательского сценария системы был выбран унифицированный язык моделирования (UML) [8]. На основе UML построены: диаграммы вариантов использования диаграммы классов.

В процессе реализации диаграммы классов и вариантов использования были дополнены.

Дополнения были обусловлены нарастанием функционала в реализации мелких детальных особенностей и изменений требований заказчика.

# 3.1 Диаграмма прецедентов плагина

Прецеденты [8] – это технология определения функциональных требований к системе. Работа прецедентов заключается в описании типичных взаимодействий между пользователем системы и самой системой.

Прецеденты представляют собой ценный инструмент для понимания функциональных требований к системе.

Изначальная диаграмма прецедентов плагина представлена на рисунке 3.1.

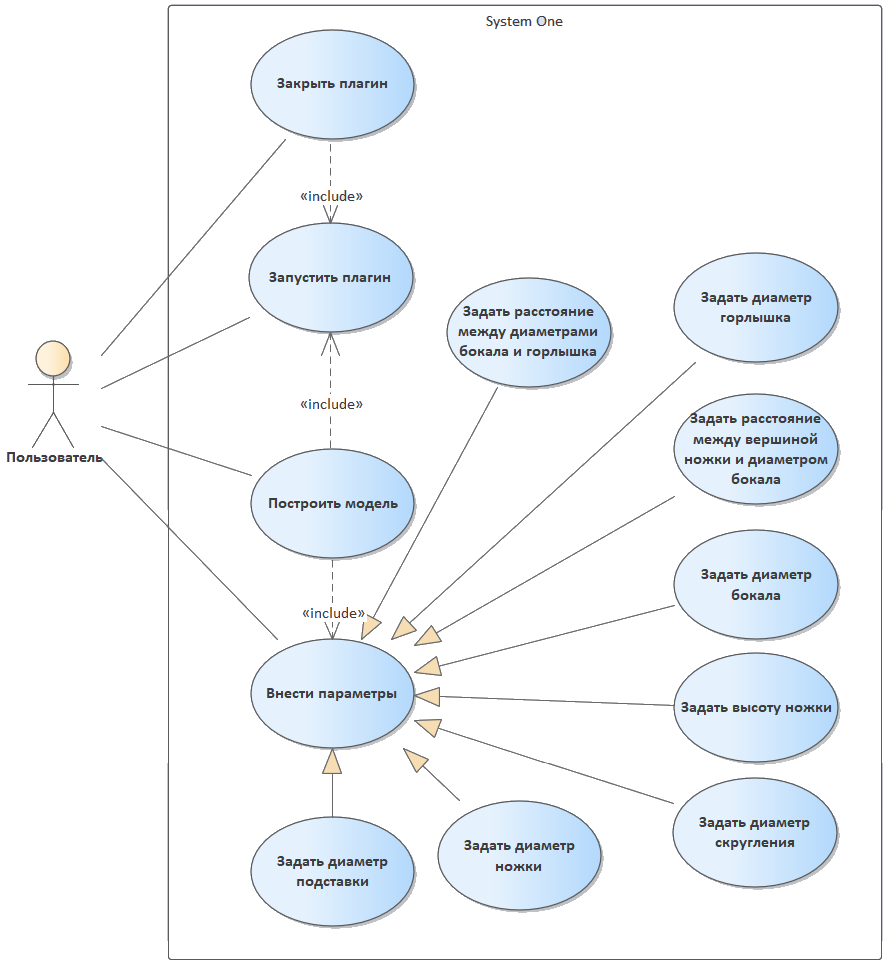


Рисунок 3.1 – Изначальная диаграмма прецедентов плагина

В ходе работы была добавлена дополнительная функциональность, в результате чего диаграмма прецедентов подверглась изменениям. Добавлена возможность выбора готовых параметров для конкретной рюмки.

Измененная диаграмма прецедентов представлена на рисунке 3.2

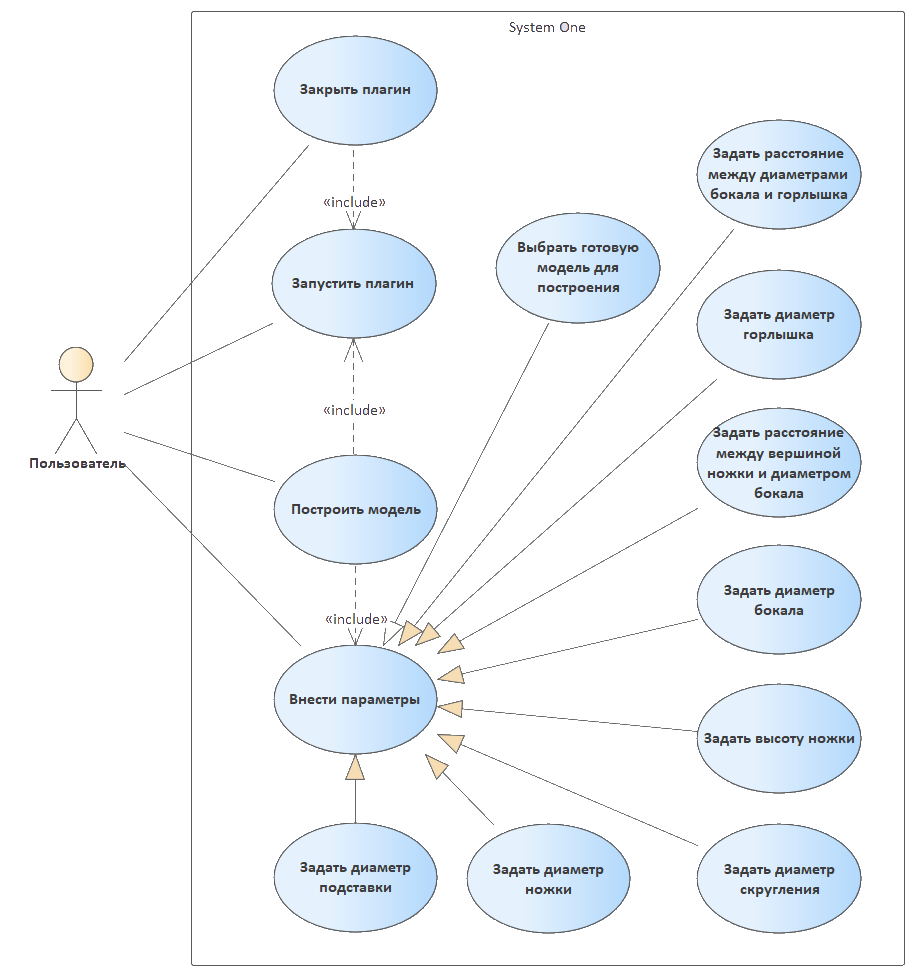


Рисунок 3.2 – Измененная диаграмма прецедентов

# 3.2 Диаграмма классов

Диаграмма классов описывает типы объектов системы и различного рада статические отношения, которые существуют между ними. На диаграммах классов отображаются также свойства классов, операции классов и ограничения, которые накладываются на связи между объектами. В UML термин функциональность применяется в качестве основного термина, описывающего и свойства, и операции класса. Целью создания диаграммы классов является графическое представление статической структуры декларативных элементов системы [8].

На рисунке 3.3 представлена изначальная диаграмма классов данного плагина

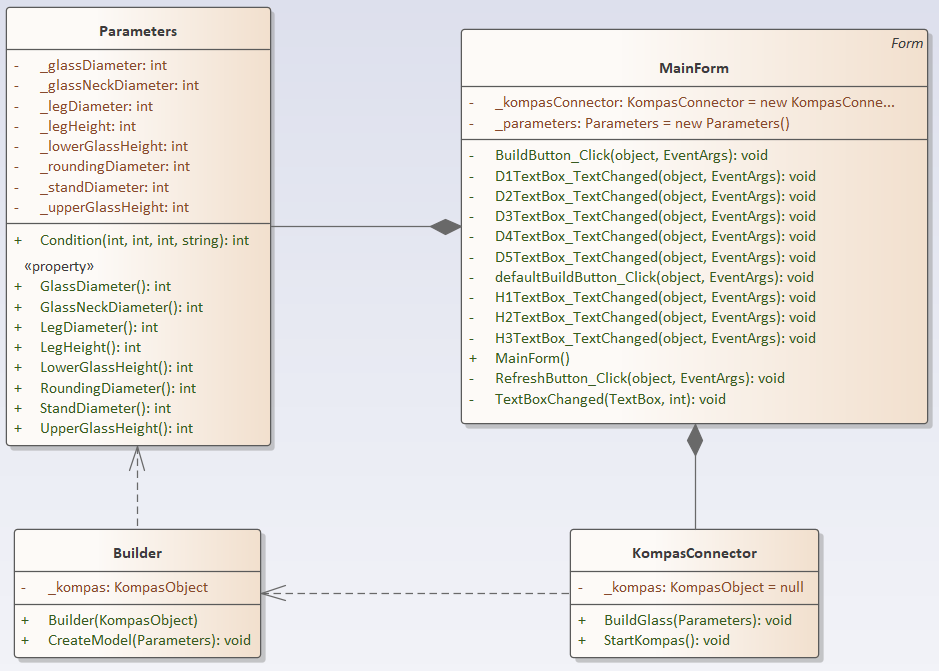


Рисунок 3.3 – Диаграмма классов плагина

Класс Parameters хранит данные о параметрах и их свойствах.

Класс Builder хранит параметры модели, введенные пользователем. Реализует метод построение модели на основании параметров.

Класс KompasConnector хранит объект класса построителя 3D модели (Builder), реализует методы для начала работы с САПР.

Класс MainForm является формой пользовательского интерфейса. Реализует методы, используемые для взаимодействия с пользователем.

В ходе работы была добавлена дополнительная функциональность, в результате чего диаграмма классов подверглась изменениям.

В MainForm были добавлены следующие элементы: функция Assignment, которая присваивает значения из введенных полей; обработчик GlassComboBox\_SelectedIndexChanged, который предоставляет возможность выбора заранее заданных параметров для конкретной рюмки.

На рисунке 3.4 изображена измененная диаграмма классов.

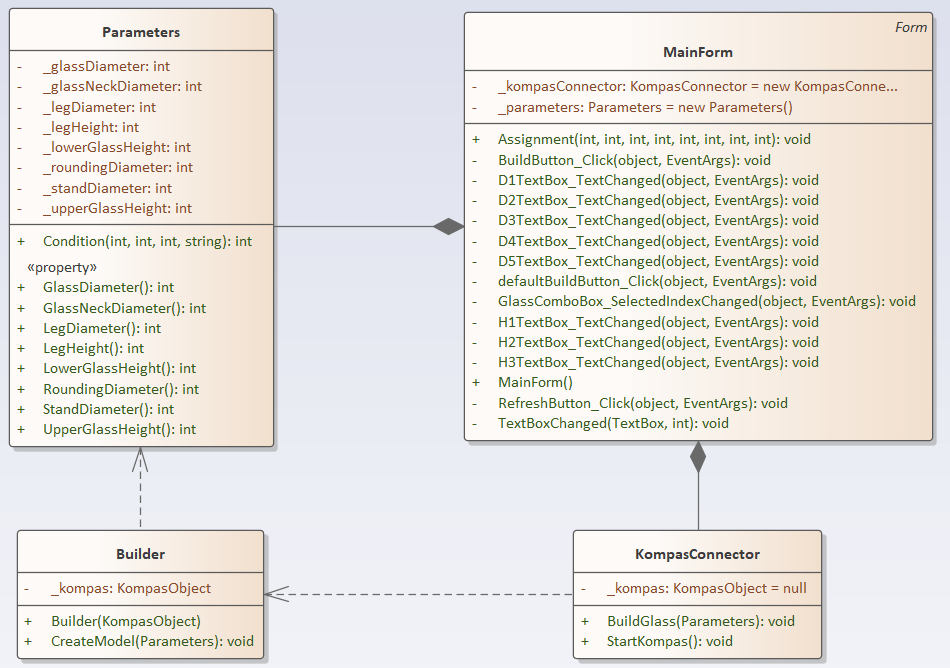


Рисунок 3.4 – Измененная диаграмма классов.

# 4 Описание программы для пользователя

Плагин состоит из одного диалогового окна с полями, их названиями и кнопок.

Начальный вид диалогового окна изображен на рисунке 4.1.

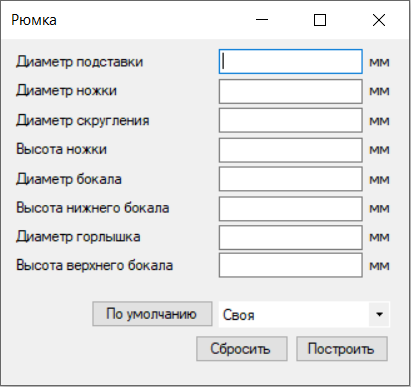


Рисунок 4.1 – Начальный вид диалогового окна

При введении значения, не входящего в установленной интервал, или посторонних символов, соответствующее поле окрасится в красный цвет и будет оставаться таковым до тех пор, пока не будет введено корректное значение.

Диалоговое окно с введенным неверными значениями показано на рисунке 4.2

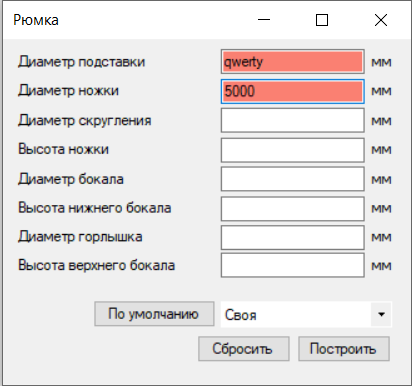


Рисунок 4.2 – Диалоговое окно с неверными значениями

В данном проекте нет зависимых параметров, однако модель нельзя будет построить, если хотя бы одно поле пустует.

Кнопка «По умолчанию» задает параметры, которые были заранее подобраны для быстрого построения модели (рисунок 4.3).

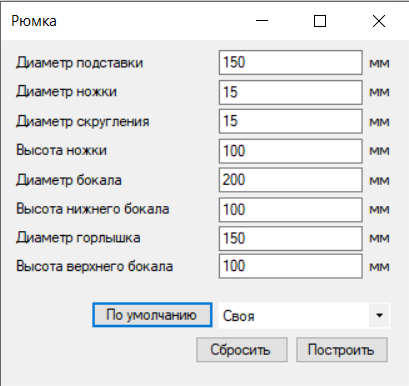


Рисунок 4.3 – Построение модели по умолчанию.

Поле с выбором, на котором по умолчанию установлена надпись «Своя», позволяет выбрать конкретную модель рюмки из предложенных. При этом заполняются поля с соответствующими значениями (рисунок 4.4).

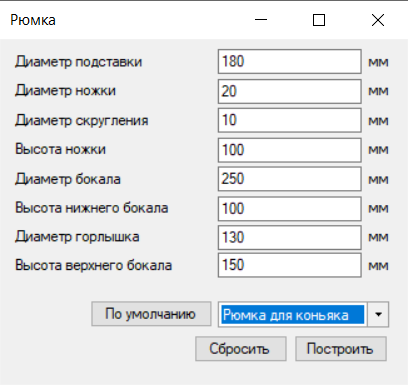


Рисунок 4.4 – Выбор модели рюмки для построения

Нажатие кнопки «Сбросить» возвращает параметры модели к начальным значениям. Нажатие кнопки доступно с момента запуска программы.

Нажатие кнопки «Построить» открывает систему автоматизированного проектирования Компас – 3D и создает новый пустой документ типа «Деталь» в котором производиться построение модели, если САПР Компас – 3D уже открыта, то произойдет создание только нового документа типа «Деталь». Нажатие кнопки возможно в любой момент в течении времени заполнения параметров, начиная с момента запуска программы.

Выделенная фокусом кнопка «Построить» на диалоговом окне изображена на рисунке 4.5.

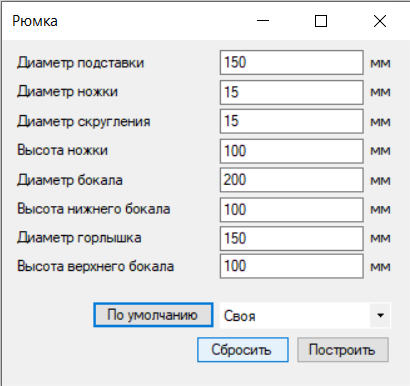


Рисунок 4.5 – Диалоговое окно с выделенной кнопкой «Построить»

Диалоговое окно документа САПР КОМПАС 3D после построения модели изображено на рисунке 4.6.

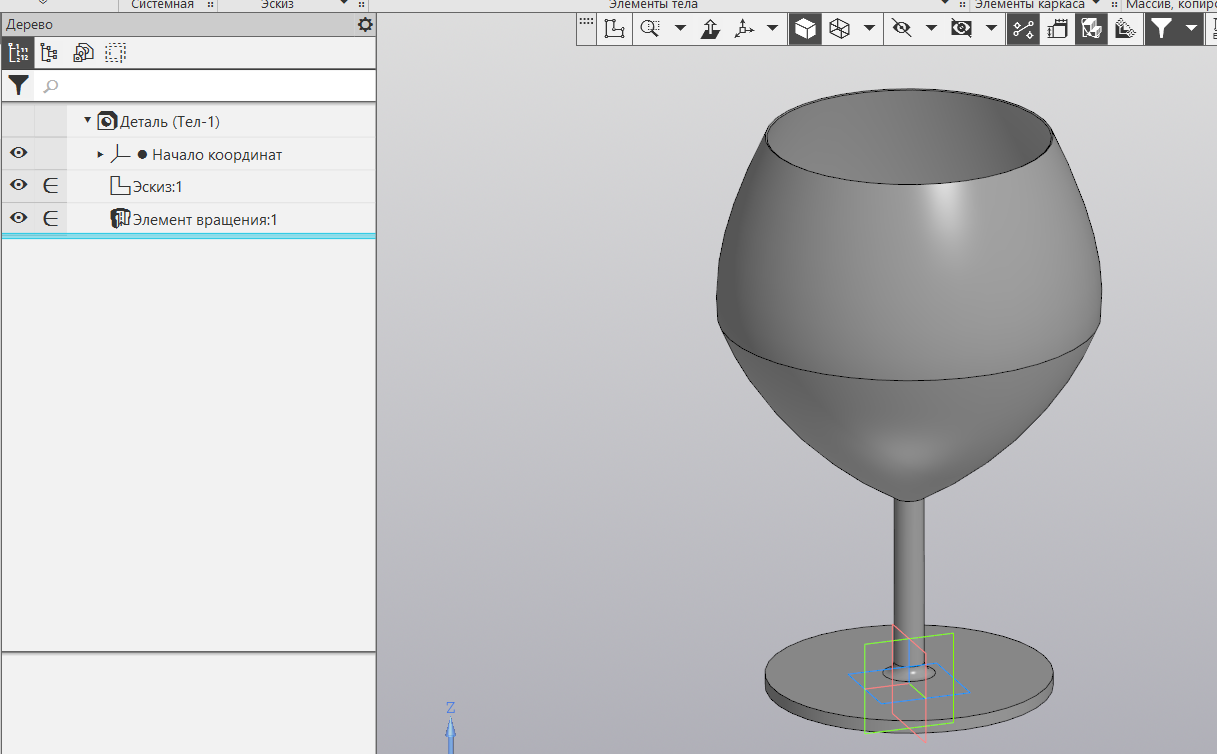


Рисунок 4.6 – Диалоговое окно документа САПР КОМПАС 3D после построения модели

# 5 Тестирование программы

Тестирование позволяет выявлять ошибки в программе в процессе разработки и при выпуске промежуточных и финальных версий приложения.

# 5.1 Функциональное тестирование

При функциональном тестировании [9] проверялась корректность работы плагина «Рюмка», а именно, соответствие полученного результата в виде трехмерной модели, с входными параметрами.

Было проведено тестирование минимальных и максимальных параметров модели.

Модель с минимальными параметрами (D1=100, D2=10, D3=10, H1=100, D4=100, H2=100, D5=100, H3=100) представлена на рисунке 5.1.

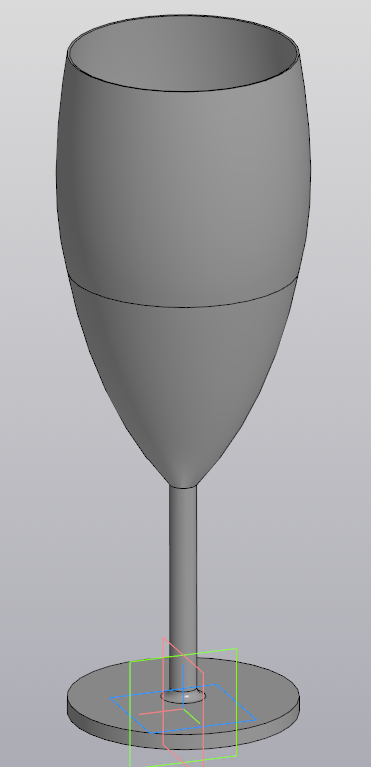


Рисунок 5.1 – Модель рюмки с минимальными возможными параметрами

Модель с максимальными параметрами (D1=180, D2=20, D3=30, H1=200, D4=250, H2=200, D5=180, H3=200) представлена на рисунке 5.2.

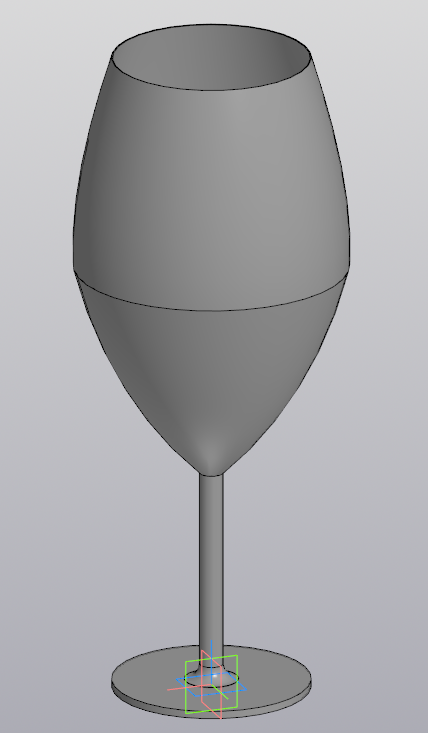


Рисунок 5.2 – Модель рюмки с максимальными возможными параметрами

Далее было проведено тестирование дополнительной функциональности с выбором моделей. Модель «Рюмка для вина» представлена на рисунке 5.3

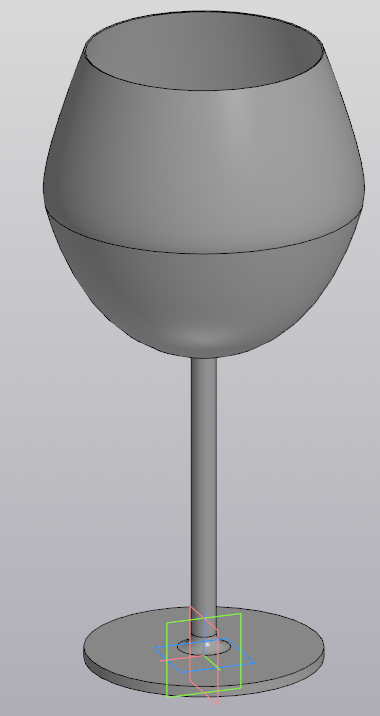


Рисунок 5.3 – Модель «Рюмка для вина»

Модель «Рюмка для шампанского» представлена на рисунке 5.4

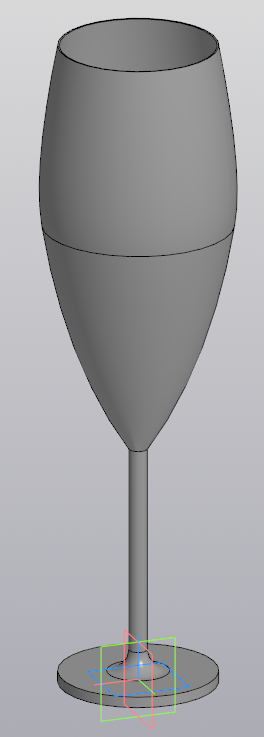


Рисунок 5.4 – Модель «Рюмка для шампанского»

Модель «Рюмка для коньяка» представлена на рисунке 5.5

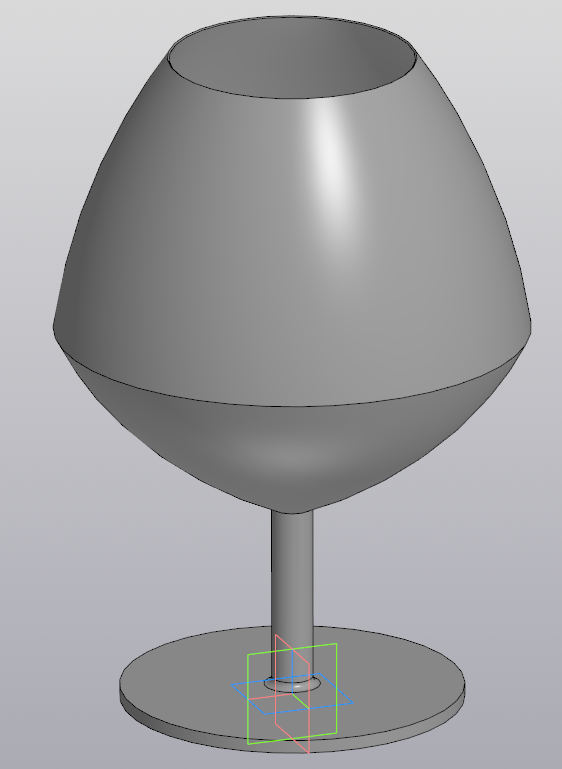


Рисунок 5.5 – Модель «Рюмка для коньяка»

# 5.2 Модульное тестирование

В целях проверки корректности работы методов и свойств классов при помощи обозревателя тестов Visual Studio было проведено модульное тестирование [10], проверялись открытые поля и методы, для этого был создан тестовый класс «ParametersTests», который тестирует свойства класса «Parameters». Описание класса «ParametersTests» представлено в Приложении А (Таблица А.1)

Диалоговое окно состояний запущенных тестов для класса «ParametersTests» изображено на рисунке 5.6

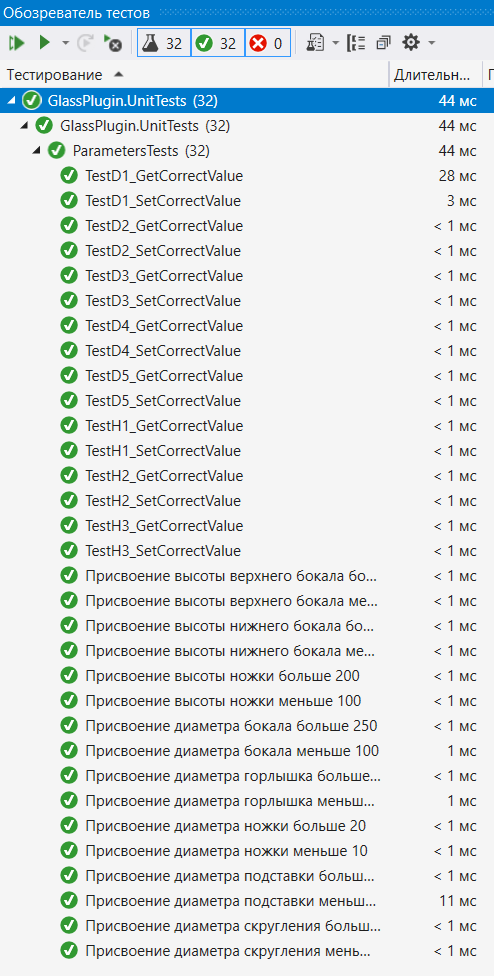


Рисунок 5.6 – Диалоговое окно состояний запущенных тестов для класса «ParametersTests»

# 5.3 Нагрузочное тестирование

В целях проверки производительности работы плагина, было проведено нагрузочное тестирование [11]. Для проведения нагрузочного тестирования был добавлен таймер, который засекал время от начала до конца построения. Измерялись потребляемые ресурсы процессора и потребляемая оперативная память.

Аппаратная конфигурация ПК, на котором проводилось нагрузочное тестирование:

* процессор: Intel Core i7-8750H 2.2 ГГц;
* графический контроллер: GeForce GTX 1060 3 ГБ;
* оперативная память: 8 ГБ.

Сперва было проведено тестирование для Windows 10 x64.

После построения 97 деталей программа КОМПАС-3D полностью перестала отвечать

Начиная с построения 65 детали, общая загрузка центрального процессора достигла максимального значения и затраты оперативной памяти начали уменьшаться, за счет увеличения времени построения детали. Время построения детали при построении первого экземпляра составляло 10,49 секунд, время построения 97 детали 27,5 секунды.

Затем было проведено тестирование для Windows 10 x32.

После построения 17 детали программа перестала отвечать и не возобновила работу.

Начиная с построения 17 детали, общая загрузка центрального процессора достигла максимального значения и затраты оперативной памяти начали уменьшаться, за счет увеличения времени построения детали. Начиная с 17 детали общая загрузка диска достигла максимального значения. Время построения детали при построении первого экземпляра составляло 68,07 секунд, время построения 17 детали 7,43 секунды.

Графики потребляемой оперативной памяти относительно числа созданных деталей изображены на рисунке 5.7. Где y – количество используемой оперативной памяти, в мегабайтах, а x – число построенных деталей. Оранжевый график – x64, синий – x32.

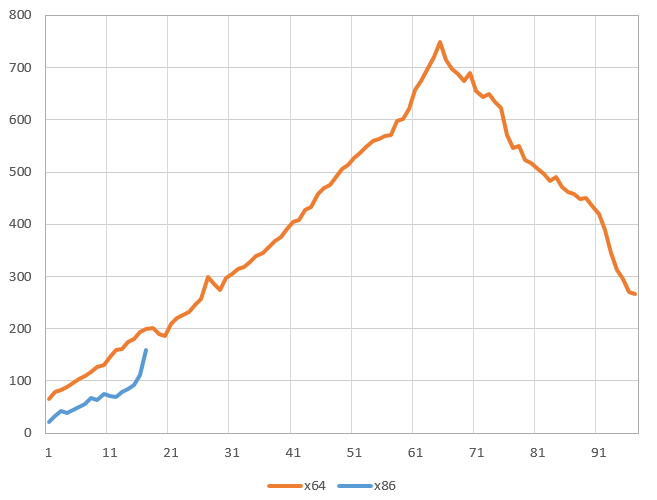


Рисунок 5.7 – Графики использования оперативной памяти относительно числа построенных деталей

Графики зависимости загрузки программой центрального процессора от количества построенных деталей изображены на рисунке 5.8. Где y – загрузка центрального процессора в %, а x – число построенных деталей. Оранжевый график – x64, синий – x32.

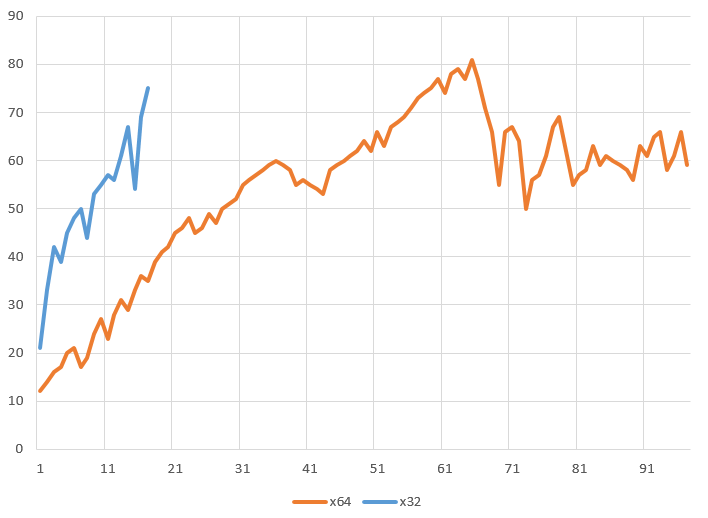


Рисунок 5.9 – Графики использования ресурсов центрального процессора относительно числа построенных деталей

При наличии свободных ресурсов оперативной памяти и процессора, зависимость между количеством построенных деталей и нагрузкой на ресурсы линейна.

При израсходовании свободных ресурсов нагрузка на процессор остается постоянной, но затраты оперативной памяти уменьшаются, так как время построения увеличивается.

# Заключение

В ходе выполнения лабораторных работ были изучены предметная область проектирования, предмет проектирования, аналоги предмета проектирования, API и на основании полученных данных были спроектированы архитектура и макет системы, создан плагин «Рюмка», проведены модульные, функциональные и нагрузочные тесты.

# Список использованных источников

1. КОМПАС – 3D [Электронный ресурс]. – URL: https://kompas.ru/ (дата обращения: 27.04.2020);
2. Microsoft Visual Studio 2019 [Электронный ресурс]. – URL: https://visualstudio.microsoft.com/ru/vs/ (дата обращения: 13.05.2020);
3. Иомдин Б. Л. Терминология быта. (2009) [Электронный ресурс]. – URL: http://www.dialog-21.ru/digests/dialog2009/materials/html/22.htm (дата обращения: 27.04.2020);
4. ГОСТ 5.121-69 Изделия из бесцветного и цветного хрусталя. [Электронный ресурс]. – URL: https://standartgost.ru/g/ГОСТ\_5.121-69 (дата обращения: 27.04.2020);
5. Базовые интерфейсы API системы КОМПАС [Электронный ресурс]. – URL: https://it.wikireading.ru/23741 (дата обращения: 13.05.2020);
6. Приложение «Оборудование: Трубопроводы» [Электронный ресурс]. – URL: https://kompas.ru/kompas-3d/application/machinery/pipelines-3d/ (дата обращения: 27.04.2020);
7. Приложение «Механика: Пружины» [Электронный ресурс]. – URL: https://kompas.ru/kompas-3d/application/machinery/springs/ (дата обращения: 27.04.2020);
8. Мартин Фаулер UML Основы. [Электронный ресурс]. – URL: https://martinfowler.com/books/uml.html (дата обращения: 16.05.2020);
9. Функциональное тестирование [Электронный ресурс]. – URL: https://daglab.ru/funkcionalnoe-testirovanie-programmnogo-obespechenija/ (дата обращения: 13.05.2020);
10. Модульное тестирование [Электронный ресурс]. – URL: http://espressocode.top/unit-testing-software-testing/ (дата обращения: 13.05.2020);
11. Нагрузочное тестирование [Электронный ресурс]. – URL: https://daglab.ru/nagruzochnoe-testirovanie-programmnogo-obespechenija/ (дата обращения: 13.04.2020);

# Приложение А

(Справочное)

В таблицах приложения для обозначения модификаторов доступа полей приняты следующие условные знаки:

* “ – ” обозначение private (закрытого) поля;
* “ + ” обозначение public (открытого) поля;

Описание полей и методов используемых для проверки тестовых случаев класса Parameters представлено в таблице А.1

Таблица А.1 – Описание полей и методов класса «ParametersTests»

|  |  |
| --- | --- |
| Наименование | Описание |
| – \_parameters: Parameters | Поле хранит элементы модели |
| + InitParameters() | Метод заполняет поле \_parameters тестовыми данными |
| + TestD1\_SetArgumentException(int wrongValue, string message) | Должно возникать исключение, если значение параметра «диаметр подставки» лежит вне диапазона |
| + TestD1\_SetCorrectValue() | Позитивный тест сеттера «StandDiameter» |
| + TestD1\_GetCorrectValue() | Позитивный тест геттера «StandDiameter» |
| + TestD2\_SetArgumentException(int wrongValue, string message) | Должно возникать исключение, если значение параметра «диаметр ножки» лежит вне диапазона |
| + TestD2\_SetCorrectValue() | Позитивный тест сеттера «LegDiameter» |
| + TestD2\_GetCorrectValue() | Позитивный тест геттера «LegDiameter» |
| + TestD3\_SetArgumentException(int wrongValue, string message) | Должно возникать исключение, если значение параметра «диаметр |

Продолжение таблицы А.1

|  |  |
| --- | --- |
|  | скругления» лежит вне диапазона |
| + TestD3\_SetCorrectValue() | Позитивный тест сеттера «RoundingDiameter» |
| + TestD3\_GetCorrectValue() | Позитивный тест геттера «RoundingDiameter» |
| + TestH1\_SetArgumentException(int wrongValue, string message) | Должно возникать исключение, если значение параметра «высота ножки» лежит вне диапазона |
| + TestH1\_SetCorrectValue() | Позитивный тест сеттера «LegHeight» |
| + TestH1\_GetCorrectValue() | Позитивный тест геттера «LegHeight» |
| + TestD4\_SetArgumentException(int wrongValue, string message) | Должно возникать исключение, если значение параметра «диаметр бокала» лежит вне диапазона |
| + TestD4\_SetCorrectValue() | Позитивный тест сеттера «GlassDiameter» |
| + TestD4\_GetCorrectValue() | Позитивный тест геттера «GlassDiameter» |
| + TestH2\_SetArgumentException(int wrongValue, string message) | Должно возникать исключение, если значение параметра «высота нижнего бокала» лежит вне диапазона |
| + TestH2\_SetCorrectValue() | Позитивный тест сеттера «LowerGlassDiameter» |
| + TestH2\_GetCorrectValue() | Позитивный тест геттера «LowerGlassDiameter» |
| + TestD5\_SetArgumentException(int wrongValue, string message) | Должно возникать исключение, если значение параметра «диаметр |

Продолжение таблицы А.1

|  |  |
| --- | --- |
|  | горлышка» лежит вне диапазона |
| + TestD5\_SetCorrectValue() | Позитивный тест сеттера «GlassNeckDiameter» |
| + TestD5\_GetCorrectValue() | Позитивный тест геттера «GlassNeckDiameter» |
| + TestH3\_SetArgumentException(int wrongValue, string message) | Должно возникать исключение, если значение параметра «высота верхнего бокала» лежит вне диапазона |
| + TestH3\_SetCorrectValue() | Позитивный тест сеттера «UpperGlassHeight» |
| + TestH3\_GetCorrectValue() | Позитивный тест геттера «UpperGlassHeight» |