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)

Пояснительная записка по лабораторному проекту

«Разработка плагина моделирования комод для системы   
Компас - 3D»  
по дисциплине «Основы разработки САПР»

Выполнил:

Студент гр. 588-2

\_\_\_\_\_\_\_\_\_ Л.Т. Пан

« » 2022

Проверил:

к.т.н., доцент каф. КСУП

\_\_\_\_\_\_\_\_\_ А.А.Калентьев  
« » 2022

Томск 2022

**Реферат**

Пояснительная записка 29 с., 21 рис., 10 таблиц, 8 источников.

Ключевые слова: КОМПАС-3D, VISUAL STUDIO, C#, КОМПАС API, ПЛАГИН, КОМОД, САПР.

Целью данной работы является разработка плагина для создания трехмерной модели комода, согласно заданным параметрам, для системы автоматизированного проектирования «КОМПАС-3D v.20»

В результате разработан плагин, строящий трехмерную модель комода в КОМПАС-3D.

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

Оглавление

[1. Описание САПР 4](#_Toc97803446)

[1.1 Описание программы «Компас - 3D» 4](#_Toc97803447)

[1.2 Описание API САПР КОМПАС-3D 5](#_Toc97803448)

[1.3 Обзор аналогов 10](#_Toc97803449)

[2 Описание предмета проектирования 11](#_Toc97803450)

[3 Стек технологий и системные требования 12](#_Toc97803451)

[4. Описание плагина 13](#_Toc97803452)

[4.1 Диаграмма классов 13](#_Toc97803453)

[4.2 Макет пользовательского интерфейса 18](#_Toc97803454)

[5 Тестирование программы 22](#_Toc97803455)

[5.1 Функциональное тестирование 22](#_Toc97803456)

[5.2 Модульное тестирование 25](#_Toc97803457)

[5.3 Нагрузочное тестирование 26](#_Toc97803458)

[Заключение 29](#_Toc97803459)

[Список используемых источников 30](#_Toc97803460)

# 1. Описание САПР

## 1.1 Описание программы «Компас - 3D»

ПРОЕКТИРОВАНИЕ— это комплекс работ с целью получения описаний нового или модернизируемого технического объекта, достаточных для реализации или изготовления объекта в заданных условиях. В процессе проектирования возникает необходимость создания описания, необходимого для построения еще не существующего объекта. Получаемые при проектировании описания бывают окончательными или промежуточными. Окончательные описания представляют собой комплект конструкторско-технологической документации в виде чертежей, спецификаций, программ для ЭВМ и автоматизированных комплексов и т.д.[1]

САПР – Система автоматизированного проектирования автоматизированная система, реализующая информационную технологию выполнения функций проектирования, представляет собой организационно техническую систему, предназначенную для автоматизации процесса проектирования, состоящую из комплекса технических, программных и других средств автоматизации его деятельности [2].

КОМПАС-3D – система трехмерного проектирования, ставшая стандартом для тысяч предприятий, благодаря сочетанию простоты освоения и легкости работы с мощными функциональными возможностями твердотельного и поверхностного моделирования. Ключевой особенностью продукта является использование собственного математического ядра С3D и параметрических технологий, разработанных специалистами АСКОН. КОМПАС-3D обеспечивает поддержку наиболее распространенных форматов 3D-моделей (STEP, ACIS, IGES, DWG, DXF), что позволяет организовывать эффективный обмен данными со смежными организациями и заказчиками, использующими любые CAD / CAM / CAE-системы в работе [3].

## 1.2 Описание API САПР КОМПАС-3D

API (англ. Application Programming Interface) – описание способов, которыми одна компьютерная программа может взаимодействовать с другой программой.

В КОМПАС-3D существуют API двух версий: API 5 и API 7. Обе версии реализуют различные функции системы и дополняют друг друга. Обе версии программных интерфейсов в равной мере поддерживаются и развиваются с учетом самих изменений в системе. В основном, для создания полноценных подключаемых модулей достаточно методов и свойств интерфейсов API 5.

Главным интерфейсом API системы КОМПАС-3D является KompasObject. Получить указатель на этот интерфейс (на интерфейс приложения API 5) можно при работе под управлением внешнего приложения (контроллера) – 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.

Ниже в таблице 1.1 представлены основные свойства и методы интерфейса KompasObject.

Таблица 1.1 – Методы и свойства интерфейса KompasObject

|  |  |  |  |
| --- | --- | --- | --- |
| Название | Входные данные | Тип возвращаемых данных | Описание |
| Document3D() |  | ksDocument | Метод для получения указателя на интерфейс трехмерного графического документа (детали или сборки) |

Продолжение таблицы 1.1

|  |  |  |  |
| --- | --- | --- | --- |
| Название | Входные параметры | Тип возвращаемых данных | Описание |
| GetParamStruct  (short structType) | structType – тип интерфейса параметров | StructType2D | Метод для получения указателя на интерфейс графического документа (чертежа или фрагмента) |
| Visible |  | bool | Свойство видимости приложения |
| Quit() |  |  | Метод для закрытия активного окна приложения КОМПАС |

В таблице 1.2 представлены методы интерфейса ksEntity, необходимые для разработки плагина.

Таблица 1.2 – Методы интерфейса ksEntity

|  |  |  |
| --- | --- | --- |
| Название | Тип возвращаемых данных | Описание |
| Create() | bool | Создать объект модели |
| GetDefinition() | IUnkown | Получить указатель на интерфейс параметров объектов и элементов |
| Update() | bool | Изменить свойства объекта (используя ранее установленные свойства) |

В таблице 1.3 представлены свойства и методы интерфейса ksDocument2D, необходимые для разработки плагина

Таблица 1.3 – Методы интерфейса ksDocument2D

|  |  |  |  |
| --- | --- | --- | --- |
| Название | Входные параметры | Тип возвращаемых данных | Описание |
| ksRectangle  (ksRectangleParam param, int style) | param– параметры прямоугольника.  style–стиль линии. | int | Получить указатель на прямоугольник на двумерной плоскости либо 0 в случае ошибки |
| ksCircle  (double xc, double yc, double rad, int style) | xc, yc – координаты центра окружности.  rad – радиус окружности.  style – стиль линии. | int | Получить указатель на окружность на двумерной плоскости либо 0 в случае ошибки |

В таблице 1.4 представлены свойства и методы интерфейса ksDocument3D, необходимые для разработки плагина.

Таблица 1.4 – Методы интерфейса ksDocument3D

|  |  |  |  |
| --- | --- | --- | --- |
| Название | Входные параметры | Тип возвращаемых данных | Описание |
| Create (bool invisible, bool \_typeDoc) | invisible – признак режима редактирования документа (true – невидимый режим,  false – видимый режим),  typeDoc – тип документа (true – деталь, false – сборка). | bool | Создать документ-модель (деталь или сборку) |

Продолжение таблицы 1.4

|  |  |  |  |
| --- | --- | --- | --- |
| Название | Входные параметры | Тип возвращаемых данных | Описание |
| GetPart(int type) | type – тип компонента из перечисления Типы компонентов. | ksPart | Получить указатель на интерфейс компонента в соответствии с заданным типом |

В таблице 1.5 представлены методы интерфейса ksPart, необходимые для разработки плагина.

Таблица 1.5 – Свойства и методы интерфейса ksPart.

|  |  |  |  |
| --- | --- | --- | --- |
| Название | Входные параметры | Тип возвращаемых данных | Описание |
| EntityCollection  (short objType) | objType – тип объектов, содержащихся в массиве. | ksEnintyCollection | Формирует массив объектов и возвращает указатель на его интерфейс |
| GetDefaultEntity  (short objType) | objType – тип объекта | ksEntity | Получить указатель на интерфейс объекта, создаваемого системой по умолчанию |
| GetPart (int type) | Type – тип компонента | ksPart | Получить указатель на интерфейс компонента в соответствии с заданным типом |
| NewEntity (short objType) | objType – тип объекта | ksEntity | Создать новый интерфейс объекта и получить указатель на него |

В таблице 1.6 представлены типы объектов документа-модели, необходимые для разработки плагина.

Таблица 1.6 – Некоторые типы объектов документа-модели

|  |  |  |
| --- | --- | --- |
| Идентификатор объекта | Название объекта | Интерфейс параметров |
| o3d\_unknown | Неизвестный (включает все объекты) |  |
| o3d\_planeXOZ | Плоскость XOZ | ksPlaneParam |
| o3d\_planeYOZ | Плоскость YOZ | ksPlaneParam |
| o3d\_planeXOY | Плоскость XOY | ksPlaneParam |
| o3d\_sketch | Эскиз | ksSketchDefinition |
| o3d\_face | Грань | ksFaceDefinition |
| o3d\_cutExtrusion | Вырезать выдавливанием | ksCutExtrusionDefinition |
| o3d\_fillet | Операция «скругление» | ksFilletDefinition |

## 1.3 Обзор аналогов

Базис-Мебельщик

Базис-Мебельщик помогает проектировать сложную корпусную мебель массового производства. Благодаря мощному и быстрому движку вы можете в кратчайшие сроки рассчитать необходимые данные, сформулировать схемы и раскрой, создавать схемы и чертежи и запускать просмотр в режиме 3D. Точная панель управления позволяет создавать даже мелкие элементы с высочайшей деталировкой и подготавливать фотореалистичные изображения для презентации.



Рисунок 1.1 – Базис-Мебельщик.

Библиотека помогает в решении следующих задач:

– автоматизация сложных повторяющихся процессов;

– графический редактор профессионального уровня для планов высочайшей точности;

– запуск трехмерного и двухмерного виртуального просмотра для более наглядной работы;

– создание реалистичных изображений с солнечными бликами и тенями, направлением света и пр.[4].

# 2 Описание предмета проектирования

Комоды – это достойная альтернатива большим и непрактичным шкафам, они компактны и функциональны, а главное – отлично вписываются как в маленькие комнатки, так и в просторные помещения. Комод представляет собой шкафчик небольшого размера различных конструкций: с выдвижными ящиками, которые расположены друг над другом, иногда с распашными или раздвижными дверцами или открытыми полочками.

На сегодняшний день существует большое количество изделий, которые различаются размерами и формами, имеют разное число ящиков и изготавливаются из самых разнообразных материалов. При этом их главное преимущество остается неизменным – возможность дотянуться рукой в самый дальний уголок любого из ящиков (что нехарактерно для шкафа).

Обычно материалом для изготовления комодов служат натуральная древесина, ДСП, МДФ или пластик, а для облицовки используется ламинат или шпон. В качестве отделки могут быть использованы: кожа, стекло или натуральный камень, глянец, краска, морилка, художественная роспись или рельефная фактура.[5]

Плагин должен уметь изменять такие параметры как:

* длина ящика А (от 1000 до 5000 мм);
* ширина ящика М (от 400 до 500 мм);
* высота комода Z (от 700 до 1524 мм);
* ширина комода S (от 450 до 550 мм);
* количество ящиков n (от 3 до 7 шт);
* форма комода.

На рисунке 2.1 представлен чертеж комода.



Рисунок 2.1 – Чертеж разработанной модели комода.

# 3 Стек технологий и системные требования

Язык программирования: C#;

Среда разработки: Visual Studio 2022 версия 17.1.0 с применением Microsoft.NET Framework 4.7.2;

Тестирование с помощью библиотек: NUnit версия 3.12.0, NUnit3TestAdapter версия 3.16.1.;

Система КОМПАС 3D v20.

Для реализации пользовательского интерфейса использовался WinForm.

Взаимодействие плагина с системой КОМПАС (с функциями моделирования, математическими функциями ядра системы и пр.) осуществляется посредством программных интерфейсов, называемых API. В КОМПАС на данный момент существуют API двух версий: API 5 и API 7. Обе версии реализуют различные функции системы и взаимно дополняют друг друга [6].

В основном, для создания полноценных подключаемых модулей, достаточно методов и свойств интерфейсов API 5.

# 4. Описание плагина

## 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]

Изначальная диаграмма классов плагина представлена на рисунке 4.1.

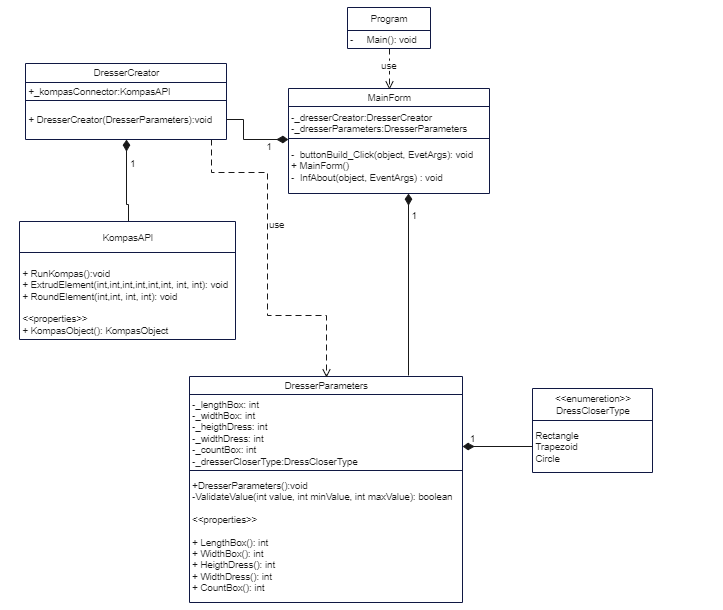


Рисунок 4.1 – Диаграмма классов плагина «Комод»

Класс «Program» использует «MainForm» для обработки действий в графическом интерфейсе. «DresserCreator» содержит в себе метод DresserCreator() для создания 3D модели в «Компас 3D», которая также подключается к САПР при помощи «KompasElement». В «KompasElement» содержатся методы выдавливания (ExtrudElement()) и скругления (RoundElement()) элементов. Класс «DresserParameters» содержит введенные значения в графическом интерфейсе и перечисление DressCloserType(). При передаче значений свойствам класса «DresserParameters» в сеттерах при помощи метода «ValidateValue» проверяется правильность диапазона значения. В случае выхода из диапазона вызывается исключение.

Таблица 4.1 – Класс DresserCreater.

|  |  |  |  |
| --- | --- | --- | --- |
| Название | Входные данные | Тип возвращаемых данных | Описание |
| \_kompasConnector | KompasAPI |  | Открытое поле, содержащее объект, API системы КОМПАС, реализующий основной функционал взаимодействия |
| DresserCreator() | DresserParameters – экземпляр объекта класса | void | Конструктор класса  DresserCreator |

Таблица 2.2 – Класс KompasAPI.

|  |  |  |  |
| --- | --- | --- | --- |
| Название | Входные данные | Тип возвращаемых данных | Описание |
| KompasObject() | KompasObject – главный интерфейс API системы КОМПАС, реализующий основной функционал взаимодействия |  | Свойство, содержащее объект, API системы КОМПАС, реализующий основной функционал взаимодействия |
| RunKompas() |  | void | Открывает программу Компас 3D о возвращает объект KompasObject |

Продолжение таблицы 2.2

|  |  |  |  |
| --- | --- | --- | --- |
| Название | Входные  данные | Тип возвращаемых данных | Описание |
| ExtrudElement() | int – основание ящика  int – стенка ящика по длине  int – стенка по ширине  int – дверца ящика  int – короткая стенка комода  int – длинная стенка комода  int – крышка комода  int – основание комода | void | Выдавливает эскиз фигур |
| RoundElement() | int – крышка  int – стенка комода по длине  int – стенка комода по ширине  int – основание | void | Скругляет элементы фигуры |

Таблица 2.3 – Класс DresserParameters

|  |  |  |
| --- | --- | --- |
| Название | Тип  возвращаемых данных | Описание |
| \_lengthBox | int – длина ящика | Закрытое поле, содержащее длину ящика |
| \_widthBox | int – ширина ящика | Закрытое поле, содержащее ширину ящика |
| \_heigthDress | int – высота комода | Закрытое поле, содержащее высоту комода |
| \_widthDress | int – ширина комода | Закрытое поле, содержащее ширину комода |

Продолжение таблицы 2.3

|  |  |  |
| --- | --- | --- |
| Название | Тип возвращаемых данных | Описание |
| \_countBox | int – количество ящиков | Закрытое поле, содержащее количество ящиков |
| \_dresserCloserType | DressCloserType – перечисление видов формы комода | Закрытое поле, содержащее перечисление форм комода |
| LengthBox() | int – длина ящика комода | Считывается значение длины ящика, Validator проверяет правильность ввода данных |
| WidthBox() | int – ширина ящика | Считывается значение ширина ящика, Validator проверяет правильность ввода данных |
| HeigthDress() | int – высота комода | Считывается значение высота комода, Validator проверяет правильность ввода данных |
| WidthDress() | int – ширина комода | Считывается значение ширина комода, Validator проверяет правильность ввода данных |
| CountDress() | int – количество ящиков в комоде | Считывается количество ящиков, Validator проверяет правильность введенного значения |
| DresserParameters() | void | Создается экземпляр класса DresserParameters |

Диаграмма классов после добавления дополнительной функциональности представлена на рисунке 4.2.

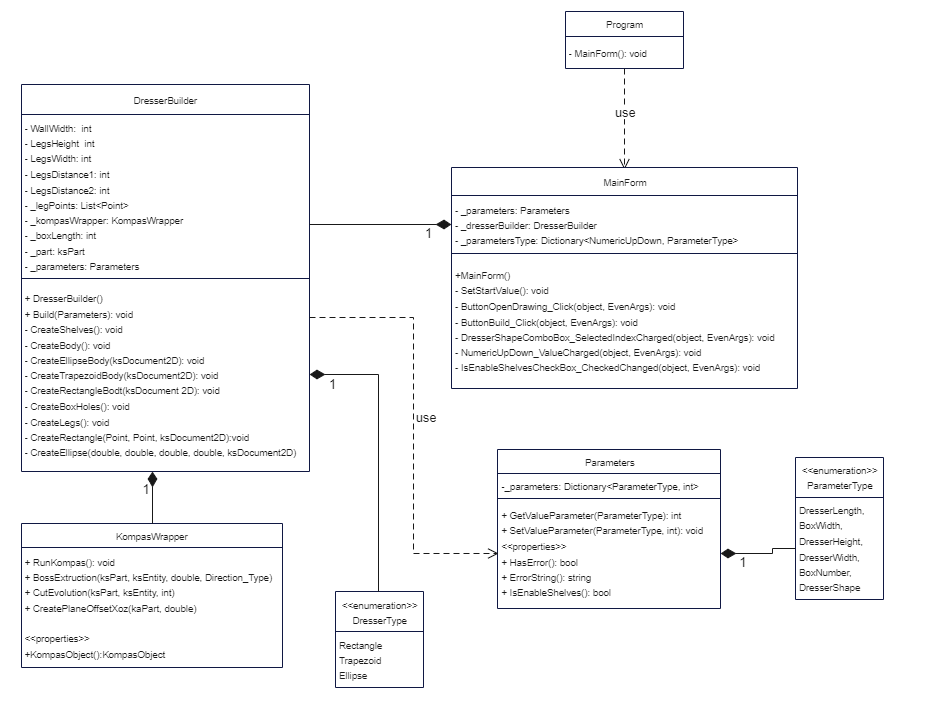


Рисунок 4.2 – Диаграмма классов с дополнительной функциональностью

В процессе работы инзачальная UML-диаграмма была изменена, и вышло следующее:

– Были переименованы классы и некоторые методы, для большего удобства в понимании и соответствия настоящему назначению этих компонентов.

– Добавлено перечисление ParameterType, где перечисляются параметры комода.

– Добавлены новые методы в DresserBuilder, которые строят части комода и составляют его форму.

– Также, в KompasWrapper добавлен метод CreatePlaneOffsetXoz(), который создает плоскость относительно XOZ.

– В MainForm добавлены обработчики событий и метод, устанавливающий начальные значения.

## 4.2 Макет пользовательского интерфейса

Макет пользовательского интерфейса создан с помощью Windows Form. На рисунке 4.3 представлен макет пользовательского интерфейса для ввода параметров модели. Так же на рисунке 4.3 представлено, как будет отображаться не правильный ввод параметров (поле окрашивается в красный цвет), если значения введены правильно, то окно не изменяет цвет.

Перед пользователем представлены 7 полей, предназначенный для ввода параметров (в мм) детали. Помимо этого, присутствует кнопка для построения модели, в дальнейшем при нажатии на которую будет загружаться Компас-3D и начинаться построение модели комода, и кнопка для открытия чертежа, чтобы пользователь видел наглядно, как выглядит модель комода.

Модель пользовательского интерфейса представлена на рисунке 4.3

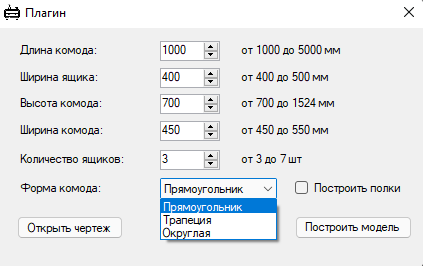


Рисунок 4.3 – Макет пользовательского интерфейса

После нажатия кнопки «Построить модель» появляется предупреждение, чтобы пользователь убедился в правильности вводимых значений. (Рисунок 4.4).

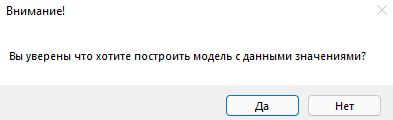


Рисунок 4.4 – Пример сообщения об ошибке

Если пользователь ввел неверное значение, появляется ошибка (Рисунок 4.5 – 4.6)

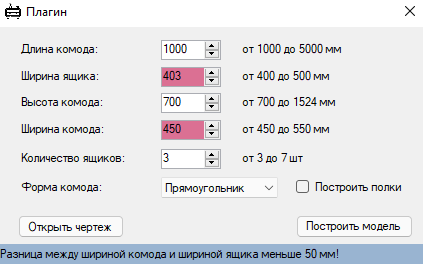
****

Рисунок 4.5 – Отображение ошибки в пользовательском интерфейсе

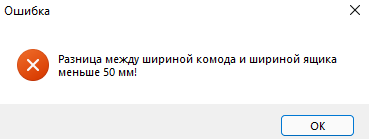


Рисунок 4.6 – Сообщение об ошибке

На рисунке 4.7 изображено диалоговое окно документа САПР «КОМПАС – 3D» после нажатия кнопки «Построить модель» и построения модели.

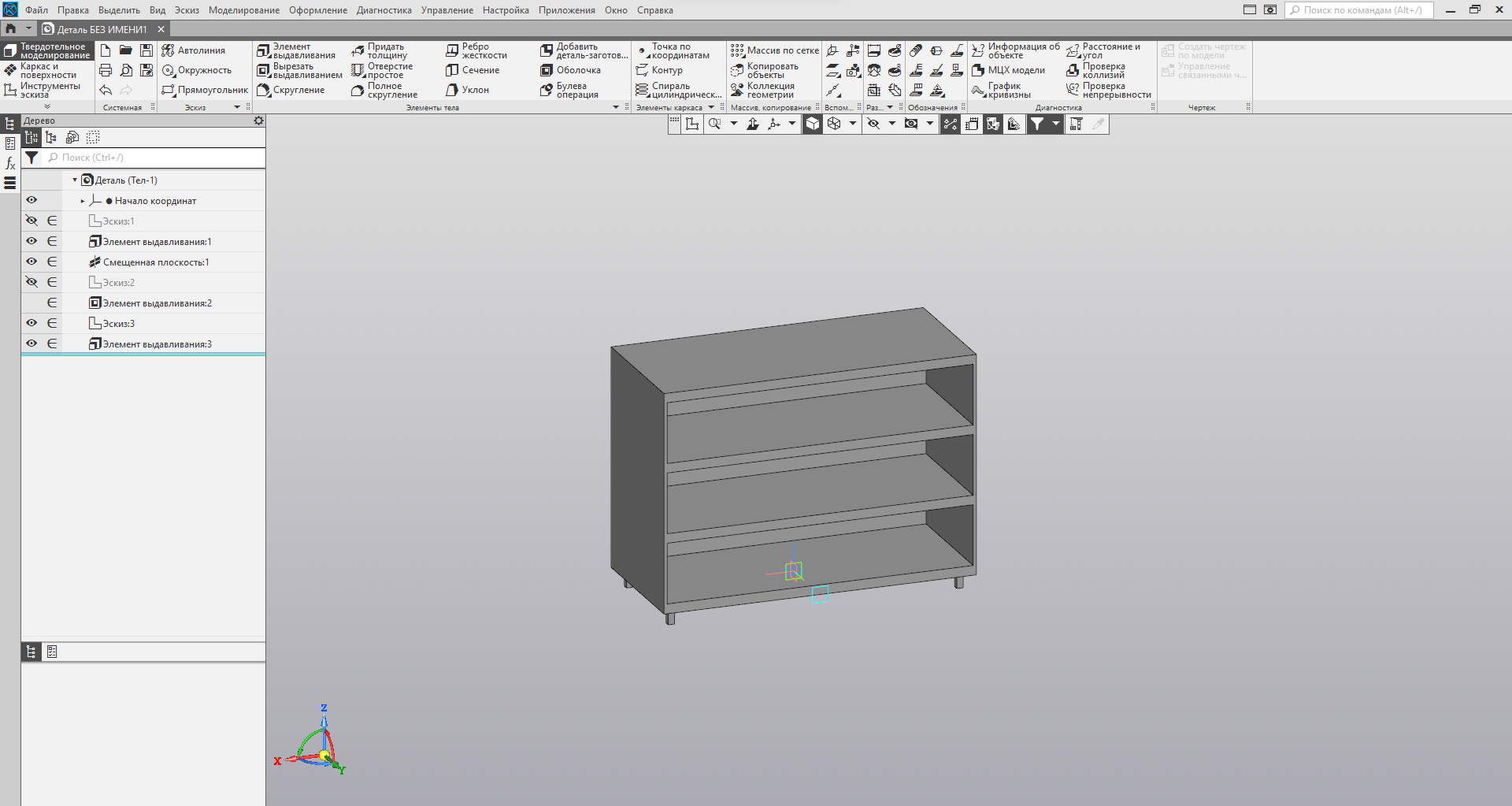


Рисунок 4.7 – Диалоговое окно документа САПР «КОМПАС – 3D»

Макет пользовательского интерфейса после добавления дополнительной функциональности представлен на рисунке 4.8.

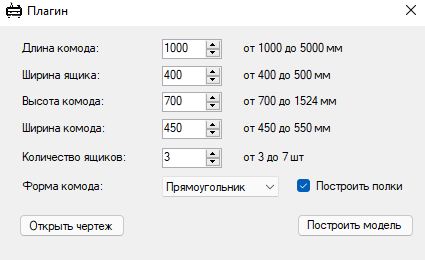


Рисунок 4.8 – Макет пользовательского интерфейса после добавления дополнительной функциональности.

На рисунке 4.9 изображено диалоговое окно документа САПР «КОМПАС – 3D» после нажатия кнопки «Построить модель» и построения модели с дополнительной функциональностью.

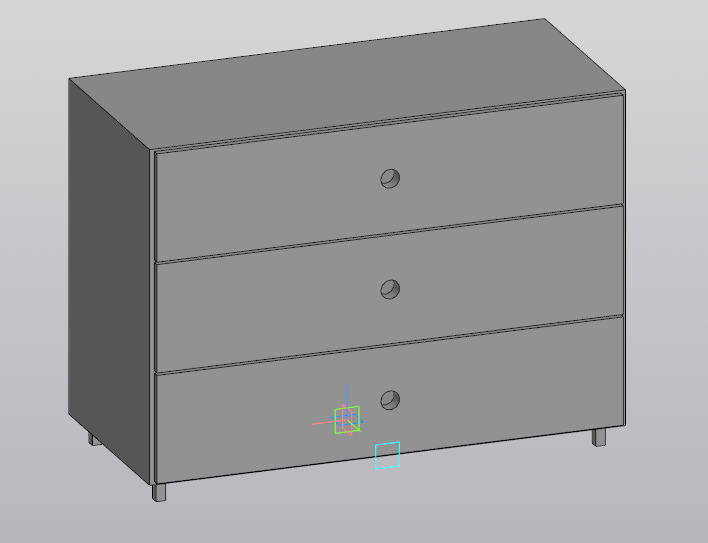


Рисунок 4.9 – Диалоговое окно документа САПР «КОМПАС – 3D»

# 5 Тестирование программы

Тестирование — это проверка соответствия объекта желаемым и требуемым критериям [8]. Несоответствие критериям называется ошибкой.

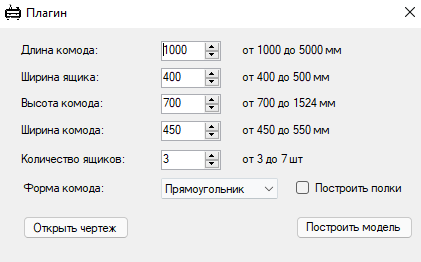
В проекте будет применятся три вида тестирования: функциональное тестирование, модульное тестирование, нагрузочное тестирование.

## 5.1 Функциональное тестирование

При функциональном тестировании комода проверялось соответствие полученной трёхмерной модели с введенными параметрами.

Проведено тестирование с максимальными, минимальными данными.

На рисунках 5.1 (а,б) представлены пользовательский интерфейс и модель с минимальными данными.

 а)

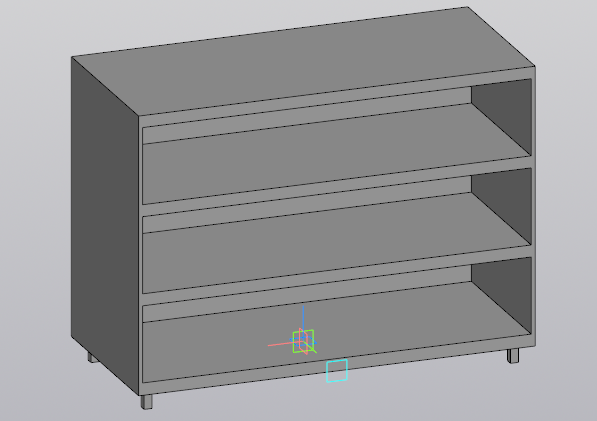
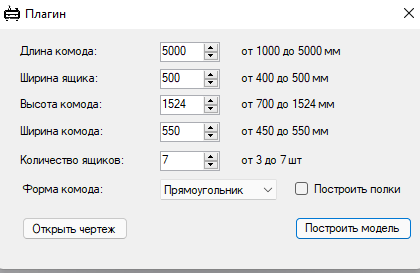
 б)

Рисунок 5.1(а,б) – Пользовательский интерфейс и модель комода с минимальными данными.

На рисунках 5.2 (а,б) представлены пользовательский интерфейс и модель с максимальными значениями параметров.

 а)

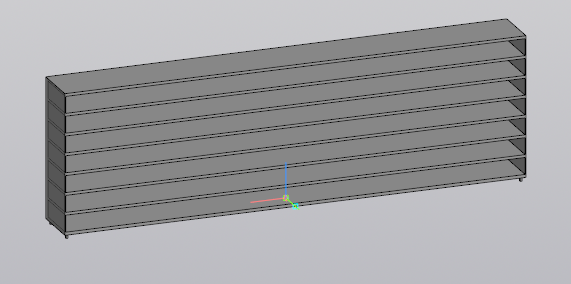
 б)

Рисунок 5.2 (а,б) – Пользовательский интерфейс и модель комода с максимальными значениями.

На рисунке 5.3 представлена модель в форме трапеции.

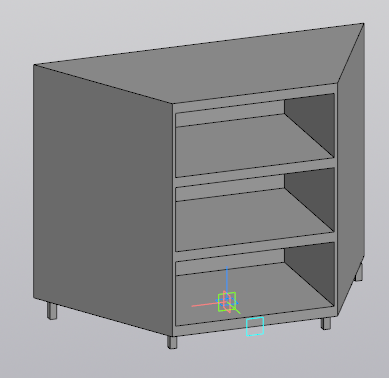


Рисунок 5.3 – Модель комода в форме трапеции.

На рисунке 5.4 представлена модель в округлой форме.

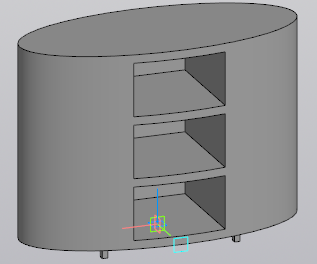


Рисунок 5.4 – Модель комода в округлой форме.

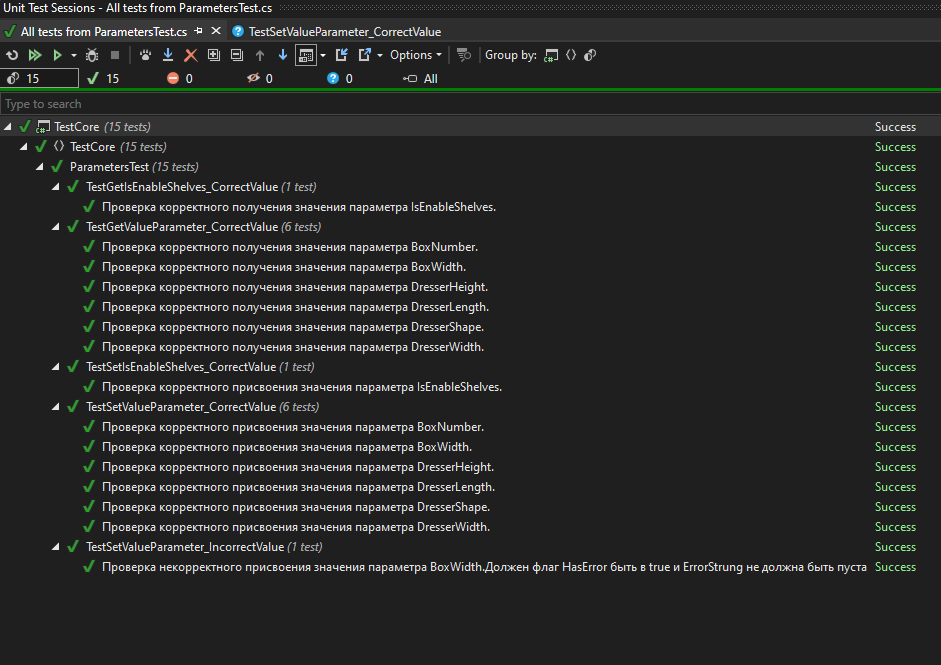
## 5.2 Модульное тестирование

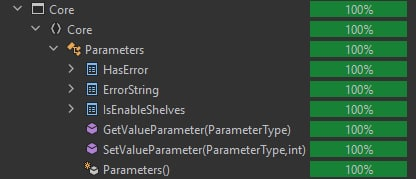
Юнит-тестирование (модульное тестирование) — тестирование отдельного элемента изолированно от остальной системы [8]. На основе тестовых сценариев проводилось тестирование работы методов и свойств классов при помощи обозревателя тестов Visual Studio. Проверялись открытые поля и методы, для этого были созданы тестовые классы:

В качестве проверки корректности работы методов и свойств классов при помощи тестового фреймворка NUnit версия 3.13.2 выполнено модульное тестирование, проверялись открытые поля, свойства и методы.

На рисунках 5.5 (а,б) представлено модульное тестирование.

Описание тестовых случаев указано в приложении А.

 а)



б)

Рисунок 5.5 – Прохождение модульного тестирования.

Описание тестов классов ParametersTest и их параметров представлено в таблице 5.1.

Таблица 5.1 – Описание тестов ParametersTest.

|  |  |
| --- | --- |
| Название теста | Описание |
| GetValueParameter\_CorrectValue  (ParameterType, int) | Тест геттера свойства ValueParameter (проверка правильного значения) |
| SetValueParameter\_CorrectValue  (ParameterType, int) | Тест сеттера свойства ValueParameter (проверка правильного значения) |
| SetValueParameter\_IncorrectValue  (ParameterType, int) | Тест сеттера свойства ValueParameter (проверка неверного значения) |
| SetIsEnableShelves\_CorrectValue() | Тест сеттера свойства IsEnableShelves (включить полки) |
| GetIsEnableShelves\_CorrectValue() | Тест геттера свойства IsEnableShelves (включить полки) |

## 5.3 Нагрузочное тестирование

Нагрузочное тестирование — тестирования производительности, сбор показателей и определение производительности и времени отклика программно-технической системы или устройства в ответ на внешний запрос с целью установления соответствия требованиям, предъявляемым к данной системе (устройству) [7].

1) Тестирование проводилось на Windows 10 x64.

2) Процессор Intel(R) Core(TM) i5-9300H CPU @ 2.40GHz 2.40 GHz.

3) 8,00 ГБ (доступно: 7,84 ГБ).

Для нагрузочного тестирования был задан бесконечный цикл построения детали. Для измерения времени был использован класс Stopwatch. Тестирование заключалось в построении ладьи со стандартными параметрами. На рисунке 5.6 представлен график зависимости загруженности памяти от количества построенных деталей.

Стандартные параметры модели:

– Длина комода – 1000 мм;

– ширина ящика – 400 мм;

– высота комода – 700 мм;

– ширина комода – 450 мм;

– количество ящиков – 3 шт;

– форма комода – прямоугольник.

Рисунок 5.6 – График зависимости количества построенных моделей кости от времени построения.

График потребляемой оперативной памяти относительно числа созданных деталей изображен на рисунке 5.7.

Рисунок 5.7 – График зависимости количества построенных моделей кости от потребляемой памяти

Тестирование длилось почти 6,5 минут, за которые было построено 127 моделей комода.

По графику затраченного времени не наблюдается больших скачков по времени, и все модели были построены примерно за один промежуток времени.

По графику затрачиваемой памяти видно, что примерно на 127 модели постепенно начинается освобождение памяти, чтобы было возможно продолжать работу. Можно предположить, что в этот момент выделялась виртуальная память.

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 выполнения лабораторных работ были изучены основные этапы проектирования программного продукта, изучена предметная область предмета проектирования, также было изучено API системы автоматизированного проектирования КОМПАС-3D. Было составлено техническое задание, разработан проект системы, составлены UML диаграммы классов, разработан макет пользовательского интерфейса.

В результате работы был разработан и реализован плагин для САПР «КОМПАС-3D», выполняющий построение 3D-модели ладьи по заданным параметрам.

Над реализованным плагином были проведены функциональное, модульное и нагрузочное тестирование на платформе Windows 10.

# Список используемых источников

1. Общие сведения о САПР [Электронный ресурс]. – Режим доступа: http://www.hi-edu.ru/e-books/xbook116/01/part-002.htm, свободный (дата обращения: 16.10.2021).
2. САПР – Википедия. [Электронный ресурс]. – Режим доступа: https://ru.wikipedia.org/wiki/Система\_автоматизированного\_проектирования (дата обращения 24.11.2021).
3. КОМПАС-3D: О программе. [Электронный ресурс]. – Режим доступа: http://kompas.ru/kompas-3d/about/ (дата обращения 24.11.2021)
4. Конструкторы мебели для самостоятельного проектирования: обзор лучшего софта. [Электронный ресурс]. – Режим доступа: <https://amssoft.ru/repair/programmy-dlya-proektirovaniya-mebeli.php> (дата обращения 13.12.2021)
5. Формула мебели: Виды комодов: как сделать правильный выбор. [Электронный ресурс]. – Режим доступа: <https://formulamebeli.com/shopping-tips/vidy-komodov/> (дата обращения 27.11.2021)
6. Расширение возможностей системы КОМПАС [электронный ресурс]. – режим доступа: [http://www.k2x2.info/kompas\_3d\_v10\_na\_100/p9.php](http://www.k2x2.info/kompas_3d_v10_na_100/p9.php%20%20)  (дата обращения 21.10.2021).
7. UML. [Электронный ресурс]. – Режим доступа: http://www.uml.org/ (дата обращения 27.11.2021)
8. Виды тестирования [электронный ресурс]. – режим доступа: <https://qa-academy.by/qaacademy/news/klassifikaciya-vidov-testirovaniya/> (дата обращения: 17.12.2021).