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 «КОМОД»   
ДЛЯ «КОМПАС-3D v.20»**

Проект системы по лабораторному проекту по дисциплине   
«Основы разработки САПР»

«Построение комода в системе КОМПАС-3D v.20»

Выполнила:

Студент гр. 588-2

\_\_\_\_\_\_\_\_\_\_\_\_Пан Л.Т.

«\_\_\_» \_\_\_\_\_\_\_\_2021г.

Руководитель:

к.т.н, доцент каф. КСУП

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

«\_\_\_» \_\_\_\_\_\_\_\_2021г.

Томск 2021 г.

# Описание САПР

# Описание программы

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].

# Описание API

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\_baseExtrusion | Базовая операция выдавливания | ksBaseExtrusionDefinition |

# 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 800 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 Проект программы

# 3.1 Описание технических и функциональных аспектов проекта

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.[6]

При использовании UML были простроена диаграмма классов.

# 3.2 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.[6]

На рисунке 3.1 представлена диаграмма классов.

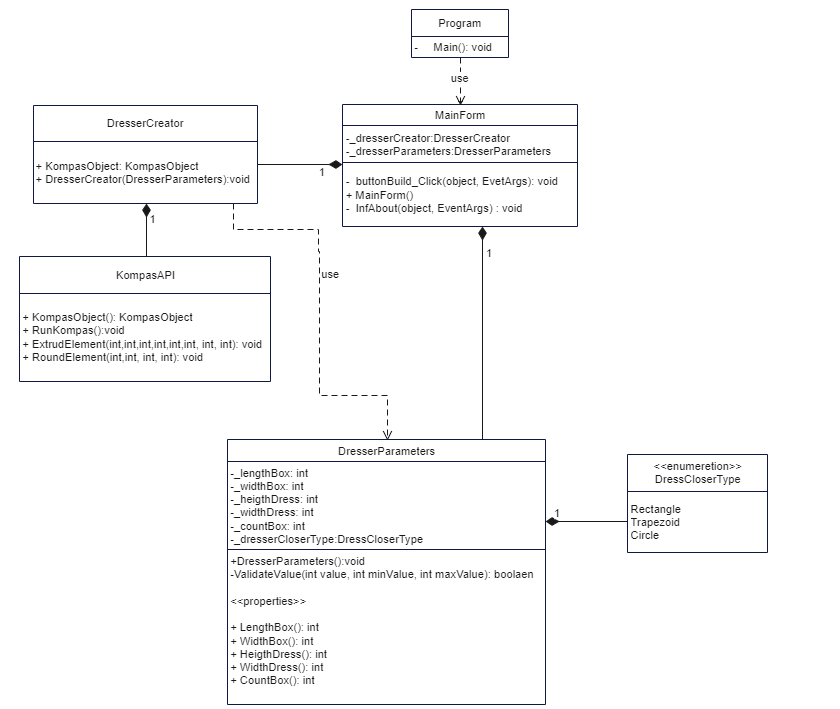


Рисунок 3.1 – Диаграмма классов.

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 2.1 – Класс DresserCreater.

|  |  |  |  |
| --- | --- | --- | --- |
| Название | Входные данные | Тип возвращаемых данных | Описание |
| KompasObject | KompasObject |  | Считывание сведений об изменении фигуры |
| DresserCreator() | DresserParameters – экземпляр объекта класса | void | Конструктор класса  DresserCreater |

Таблица 2.2 – Класс KompasAPI.

|  |  |  |  |
| --- | --- | --- | --- |
| Название | Входные данные | Тип возвращаемых данных | Описание |
| KompasObject() | KompasObject – главный интерфейс API системы КОМПАС, реализующий основной функционал взаимодействия |  | Открытое поле, содержащее объект, API системы КОМПАС, реализующий основной функционал взаимодействия |
| RunKompas() |  | void | Открывает программу Компас 3D о возвращает объект KompasObject |
| ExtrudElement() | int – основание ящика  int – стенка ящика по длине  int – стенка по ширине  int – дверца ящика  int – короткая стенка комода  int – длинная стенка комода | void | Выдавливает эскиз фигур |

Продолжение таблицы 2.2

|  |  |  |  |
| --- | --- | --- | --- |
| Название | Входные  данные | Тип возвращаемых данных | Описание |
|  | int – крышка комода  int – основание комода |  |  |
| 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 |
| \_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 |

Продолжение таблицы 2.3

|  |  |  |
| --- | --- | --- |
| Название | Тип возвращаемых данных | Описание |
| 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 |

# 3.3 Макет пользовательского интерфейса

Макет пользовательского интерфейса представляет собой форму для ввода параметров электрического чайника. Построение модели осуществляется путем нажатия на кнопку «Построить». При попытке ввода недопустимых символов, они не будут вводиться в строку (если необходимо ввести цифры, то невозможно будет ввести другие символы).

На рисунке 3.2 представлен макет пользовательского интерфейса.

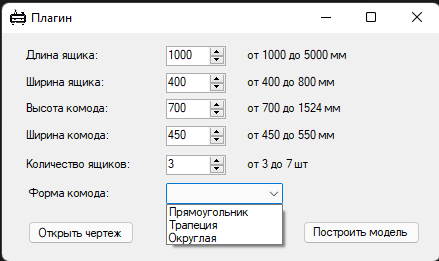


Рисунок 3.2 – Главное окно плагина.

Построение модели осуществляется путем нажатия на кнопку «Построить модель». При попытке ввода недопустимых символов, они не будут вводиться в строку (если необходимо ввести цифры, то невозможно будет ввести другие символы). Также при вводе в строку цифр из недопустимого диапазона, который указан справа от каждой области ввода, при попытке построить модель значения будут сбрасываться до минимально или максимально допустимых.

При нажатии кнопки «Открыть чертеж» открывается окно с изображением чертежа комода (рисунок 3.4)



Рисунок 3.4– Окно с чертежом.

# Список литературы

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. UML. [Электронный ресурс]. – Режим доступа: http://www.uml.org/ (дата обращения 27.11.2021)