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 «Шкаф-стол» ДЛЯ «КОМПАС-3D»**

Проект системы по лабораторному проекту

по дисциплине «ОСНОВЫ РАЗРАБОТКИ САПР»

«Построение шкаф-стола в системе КОМПАС 3D»

|  |
| --- |
| Выполнил:  студент гр. 588-1  \_\_\_\_\_\_\_\_\_\_\_\_\_\_ Малышев А.И.  «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2022г. |
| Руководитель:  к.т.н., доцент каф. КСУП:  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Калентьев А.А.  «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2022. |

Содержание

[1 Описание САПР 3](#_Toc67148369)

[1.1 Описание программы 3](#_Toc67148370)

[1.2 Описание API 5](#_Toc67148371)

[1.3 Обзор аналогов 3](#_Toc67148372)

[2 Описание предмета проектирования 9](#_Toc67148373)

[3 Описание технических и функциональных аспектов проекта 10](#_Toc67148374)

[3.1 Описание полей, свойств и методов, используемых в проекте 10](#_Toc67148375)

[3.3 Макет пользовательского интерфейса 11](#_Toc67148376)

[Список литературы 13](#_Toc67148377)

1. **Описание САПР**

**1.1 Описание программы**

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 C3D 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.[1]

* 1. **Обзор аналогов**

**PRO100**

PRO100 — программа для проектирования мебели, кухни, ванных комнат, интерьеров офисов и помещений.

PRO100 упрощает работу на каждом этапе производственного процесса. Проектирование мебели, разработка дизайна интерьера, расчёт стоимости, получение списка деталей для производства, оптимизация производства заказов. Кухни, спальни, библиотеки, ванные комнаты, офисы, гардеробные, шкафы-купе, помещения. Одна программа для решения множества задач. [2]

На рисунке 1.1 показан интерфейс программы PRO100.

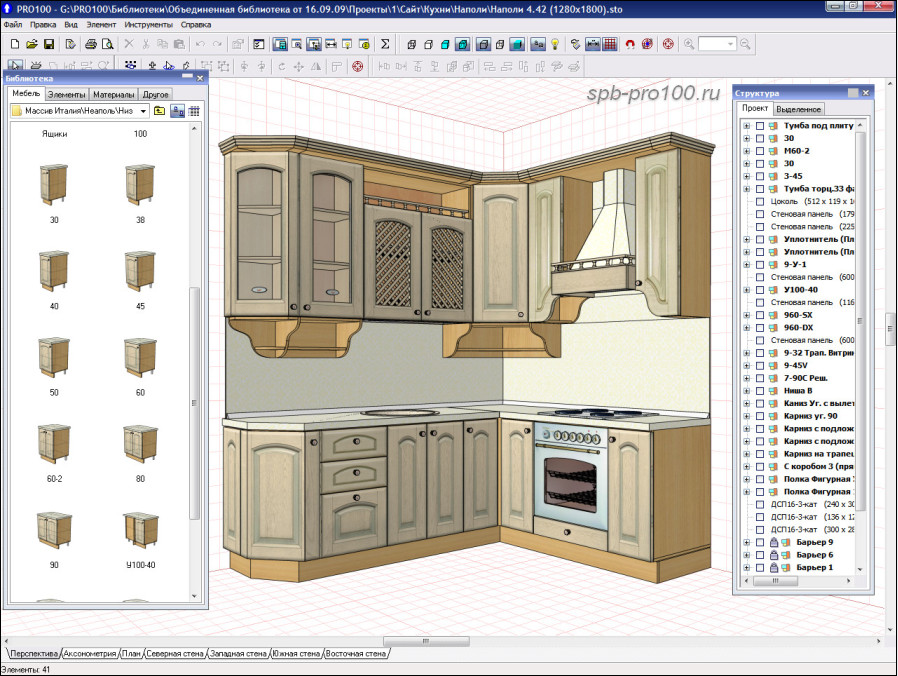


Рисунок 1.1 — Интерфейс PRO100

**К3-Мебель**

«К3-Мебель» – профессиональный комплекс для проектирования, производства и дизайна корпусной мебели. Мощный и одновременно простой инструмент, позволяющий создать изделие любой степени сложности, получить полный пакет документации в один клик и представить заказчику реалистичное изображение его будущего проекта. [3]

На рисунке 1.1 показан интерфейс программы К3-Мебель.

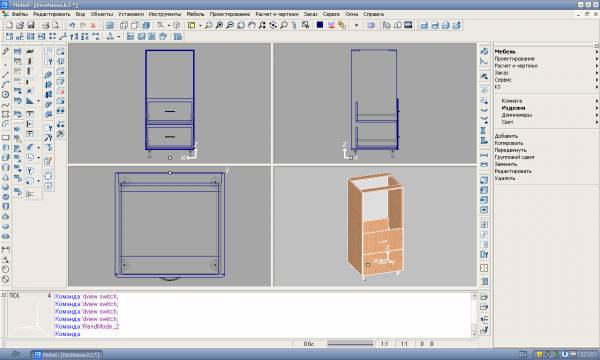


Рисунок 1.2 — Интерфейс К3-Мебель

**1.3 Описание 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. Отсюда очевидно, что 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 является KompasObject. Получить указатель на этот интерфейс (если быть точным, 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 (интерфейсы динамического массива, работы с математическими функциями, библиотек моделей или фрагментов и различных структур параметров определенного типа). [4]

Ниже в таблице 1.1 представлены свойства и методы интерфейса KompasObject, которые будут использованы при разработке плагина.

Таблица 1.1 – Некоторые методы и свойства интерфейса KompasObject

|  |  |  |  |
| --- | --- | --- | --- |
| Название | Параметры | Тип | Описание |
| Document3D() |  | ksDocument | Метод для получения указателя на интерфейс трехмерного графического документа (детали или сборки) |
| GetMathematic2D() |  | ksMathematic2D | Метод для получения указателя на интерфейс для работы с математическими функциями |
| GetParamStruct(short structType) | structType  - тип интерфейса параметров | StructType2D | Метод для получения указателя на интерфейс графического объекта |
| GetDynamicArray(long type) | type  - тип динамического массива. | ksDynamicArray | Метод для получения указателя на интерфейс динамического массива ksDynamicArray |
| Visible |  | bool | Свойство видимости приложения |

В таблице 1.2 представлены методы интерфейса ksEntity, которые были использованы при разработке плагина.

Таблица 1.2 – Некоторые методы интерфейса ksEntity

|  |  |  |
| --- | --- | --- |
| Название | Тип | Описание |
| Create() | bool | Создать объект в 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 ksDocument3D, которые были использованы при разработке плагина.

Таблица 1.3 – Некоторые свойства и методы интерфейса ksDocument3D

|  |  |  |  |
| --- | --- | --- | --- |
| Название | Параметры | Тип | Описание |
| Create () |  | bool | Создать документ-модель (деталь или сборку) |
| UpdateDocumentParam() |  | bool | Активизировать измененные параметры документа |
| GetPart(int type) | type  - тип компонента из перечисления | ksPart | Получить указатель на интерфейс компонента в соответствии с заданным типом |

В таблице 1.4 представлены методы интерфейса ksPart, которые были использованы при разработке плагина.

Таблица 1.4 – Некоторые свойства и методы интерфейса ksPart

|  |  |  |  |
| --- | --- | --- | --- |
| Название | Параметры | Тип | Описание |
| EntityCollection(short objType) | objType  - тип объектов (грани, ребра, вершины) | 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 из перечисления | 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 |

Таблица 1.5 – Свойства и методы интерфейса ksPlaneOffsetDefinition.

|  |  |  |  |
| --- | --- | --- | --- |
| Название | Тип входных параметров | Тип возвращаемых данных | Описание |
| SetPlane | plane - указатель на интерфейс плоскости ksEntity или IEntity. | TRUE - в случае успешного завершения. | Изменить указатель на интерфейс базовой плоскости |

В таблице 1.6 представлены типы объектов документа-модели, которые были использованы при разработке плагина.

Таблица 1.6 – Некоторые типы объектов документа-модели

|  |  |  |
| --- | --- | --- |
| Идентификатор объекта | Название объекта | Интерфейс параметров |
| o3d\_planeXOZ | Плоскость XOZ | ksPlaneParam |
| o3d\_sketch | эскиз | [ksSketchDefinition](about:blank) |
| o3d\_planeOffset | смещённая плоскость | [ksPlaneOffsetDefinition](about:blank) |
| o3d\_edge | ребро | [ksEdgeDefinition](about:blank) |
| o3d\_chamfer | операция "фаска" | [ksChamferDefinition](about:blank) |
| o3d\_cutExtrusion | вырезать выдавливанием | [ksCutExtrusionDefinition](about:blank) |
| o3d\_axisOY | Ось OY | ksPlaneParam |
| o3d\_circularCopy | операция копирования по концентрической сетке | [ksCircularCopyDefinition](about:blank) |
| o3d\_baseExtrusion | базовая операция выдавливания | [ksBaseExtrusionDefinition](about:blank) |

1. **Описание предмета проектирования**

Предметом проектирования является модель шкаф-стола. Данная модель имеет 6 основных параметров:

* W – ширина (мм, диапазон значений: 600-1000 мм);
* L – длина (мм, не меньше чем 75% от ширины, диапазон значений: 450-750 мм);
* H1 – высота ножек (мм, меньше H2 не менее чем в 7.5 раз, но не меньше 100 мм);
* H2 – высота стола (мм, диапазон значений: 600-1050 мм);
* R1 – закругление ребер стола (градусы, диапазон значений: 0-180);
* R2 – закругление углов стола (градусы, диапазон значений: 0-180).

На рисунке 2.1 представлен чертеж модели.

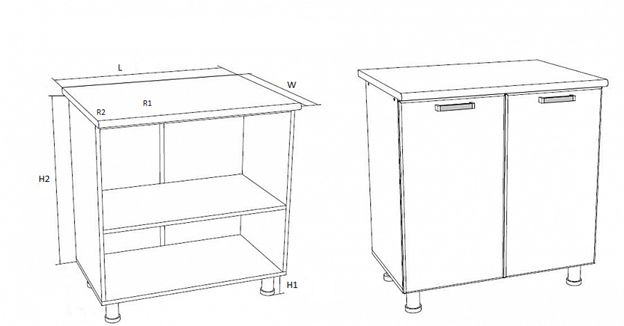


Рисунок 2.1 – Чертеж модели.

**3** **Описание технических и функциональных аспектов проекта**

**3.1 Описание полей, свойств и методов, используемых в проекте**

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.[5]

Диаграмма классов – структурная диаграмма языка моделирования UML, демонстрирующая общую структуру иерархии классов системы, их коопераций, атрибутов (полей), методов, интерфейсов и взаимосвязей между ними.

При использовании UML были простроена диаграмма классов.

При использовании UML была построена диаграмма классов. Данная диаграмма представлена на рисунке 3.3.

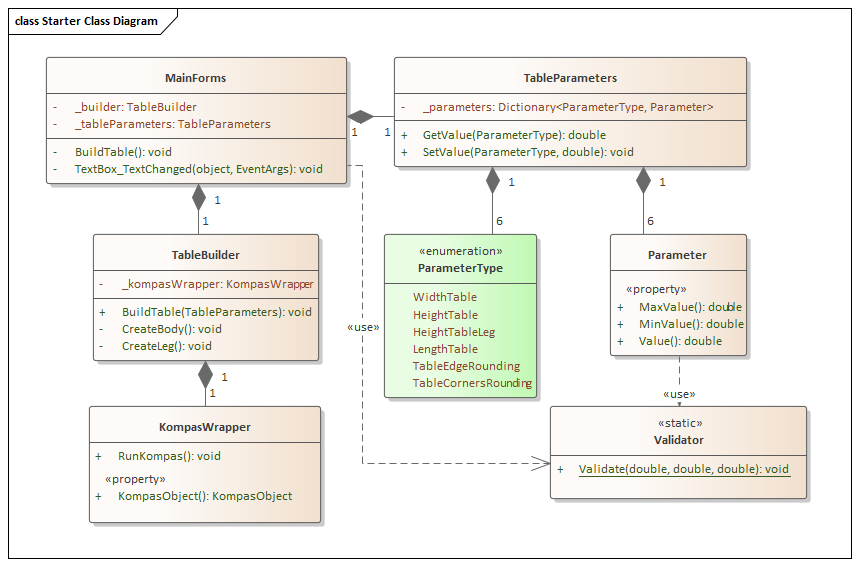


Рисунок 3.1 – UML-диаграмма классов.

Parameter — класс, содержащий в себе максимальное и минимальное значение числа, а также текущее значение параметра. Проверка изменения значения происходит в классе Validator. Взаимная проверка параметров будет происходить при изменении текста в TextBox. Все элементы TextBox будут подписаны на обработчик события TextBox\_TextChanged. Хранить параметры в форме нужно для изменения текущих значений, а также значений минимумов и максимумов параметров.

ParameterType — перечисление, хранящая в себе все названия параметров.

TableParameters — хранит в себе все параметры стола, а также через него происходит получение и установка значений параметров.

TableBuilder — класс построения объекта, содержащий в себе класс KompasWrapper (объединяющий API Компаса и программу).

MainForm — главное окно, в котором пользователь сможет изменять значения параметров.

**3.2 Макет пользовательского интерфейса**

Плагин представляет собой меню и пользовательскую форму с полями для ввода для ввода параметров. Ниже находятся форма для заполнения: «Ширина», «Длина», «Высота стола», «Высота ножек», «Радиус закругления углов» и «Радиус закругления рёбер». Так же есть подсказки для пользователя.

Макет пользовательского интерфейса представлен на рисунке 3.2.

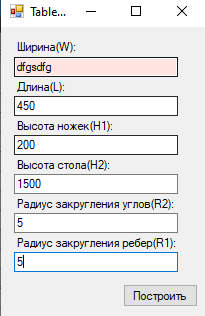


Рисунок 3.2 – Макет пользовательского интерфейса программы.

После нажатия на кнопку «Построить» при введенных некорректных значениях, появится окно, приведенный на рисунке 3.3.

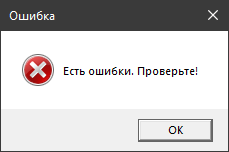


Рисунок 3.3 — Окно ошибки

**Список литературы**

1. Компас 3D – Функционал, описание продукта. [Электронный ресурс]. – Режим доступа: https://kompas.ru/ (дата обращения 10.03.2022).
2. Сайт PRO100 [Электронный ресурс]. – <https://www.ecru.pl/ru> (дата обращения 10.03.2022).
3. Сайт К3-Мебель [Электронный ресурс]. – <https://k3-mebel.ru/> (дата обращения 10.03.2022).
4. КОМПАС-3D для разработчиков [Электронный ресурс]. – Режим доступа: <https://kompas.ru/solutions/developers/> (дата обращения 10.03.2022).
5. UML. [Электронный ресурс]. – Режим доступа: http://www.uml.org/ (дата обращения 10.03.2022).