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 "НАВЕСНАЯ ВЕШАЛКА ДЛЯ ВЕРХНЕЙ ОДЕЖДЫ" ДЛЯ САПР Компас 3D

Проект системы по лабораторному проекту

по дисциплине «ОСНОВЫ РАЗРАБОТКИ САПР»

Выполнил:

студент гр. 588-2

\_\_\_\_\_\_\_\_\_\_\_\_\_\_А.Д. Целицкий

«\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2022 г.

Руководитель:

к.т.н., доцент каф. КСУП

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

«\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2022 г.

# 1 Описание САПР

# Описание программы

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 [1].

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. [2]

# Описание API

API (англ. Application Programming Interface) – логический интерфейс для программного компонента, скрывающий внутренние детали, отвечающие за реализацию [3].

Взаимодействие внешнего приложения или подключаемого модуля с системой КОМПАС осуществляется посредством программных интерфейсов, называемых API. В КОМПАС на данный момент существуют API двух версий: API 5 и API 7 [4].

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 (CreateKompasObject( )). Методы этого интерфейса реализуют наиболее общие функции работы с документами системы, системными настройками, файлами, а также дают возможность получить указатели на другие интерфейсы. В таблице 1.1 приведены методы интерфейса KompasObject [4].

Таблица 1.1 – Некоторые методы и свойства интерфейса KompasObject

|  |  |  |  |
| --- | --- | --- | --- |
| Название | Параметры | Тип | Описание |
| Document3D() |  | ksDocument | Даёт возможность получить указатель на интерфейс трёхмерного документа (детали или сборки) |

Продолжение таблицы 1.1.

|  |  |  |  |
| --- | --- | --- | --- |
| Название | Параметры | Тип | Описание |
| 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 документа |
| 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) |

Графические документы имеют собственный интерфейс – ksDocument3D, со своими специфическими свойствами и методами. С помощью функций, присутствующих в ksDocument3D, создаются изображения в эскизах трехмерных операций. Свойства (члены данных) этого интерфейса позволяют динамически управлять настройками любого трехмерного документа системы из модуля. Наиболее используемые из них приведены в таблице 1.3.

Таблица 1.3 – Методы интерфейса ksDocument3D

|  |  |  |  |
| --- | --- | --- | --- |
| Название | Тип входных параметров | Тип возвращаемых данных | Описание |
| Create () | invisible-признак режима редактирования документа (TRUE-невидимый режим, FALSE – видимый режим), typeDoc – тип документа | TRUE – в случае успешного завершения | Дает возможность создать пустой документ (деталь или сборку) |

Продолжение таблицы 1.3.

|  |  |  |  |
| --- | --- | --- | --- |
| Название | Тип входных параметров | Тип возвращаемых данных | Описание |
| GetPart() | Type – тип компонента из перечисления | Указатель на интерфейс компонента ksPart или IPart | Позволяет получить указатель на интерфейс компонента в сборке |
| UpdateDocumentParam() |  | TRUE – в случае успешного завершения. | Позволяет обновить настройки документа |

Метод ksDocument3D::GetPart возвращает указатель на интерфейс детали или компонента сборки – ksPart. Свойства и методы этого интерфейса управляют состоянием компонентов сборки, они почти полностью дублируют команды контекстного меню и панели свойств, доступные пользователю при работе с тем или иным компонентом.

Таблица 1.2.4 – Методы интерфейса ksPart.

|  |  |  |  |
| --- | --- | --- | --- |
| Название | Тип входных параметров | Тип возвращаемых данных | Описание |
| EntityCollection() | objType – тип объектов | В случае успеха указатель на интерфейс ksEntityCollection или IEntityCollection, в случае неудачи – NULL. | Формирует динамический массив трехмерных объектов и возвращает указатель на его интерфейс |
| GetPart() | type – тип компонента из перечисления | Указатель на интерфейс компонента ksPart или IPart | Получить указатель на интерфейс компонента |

Продолжение таблицы 1.4

|  |  |  |  |
| --- | --- | --- | --- |
| NewEntity() | objType – тип объекта. | Указатель на интерфейс ksEntity или IEntity. | Создает интерфейс нового трехмерного объекта и возвращает указатель на него |
| GetDefaultEntity() | objType – тип объекта. | Указатель на интерфейс ksEntity или IEntity. | Возвращает указатель на интерфейс объекта, создаваемого системой в трехмерном документе по умолчанию |

В таблице 1.5 представлены типы объектов документа-модели, которые были использованы при разработке плагина.

Таблица 1.5 – 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) |

# Обзор аналогов

**DS 3D**

**Система автоматизированного проектирования DS 3D** — интегрированная программная система автоматизации процессов проектирования, производства и продажи мебели. [5]

Данное программа позволит осуществлять автоматическое и ручное построение помещений заданного размера, балок, перегородок, пола и потолка. Вставка окон, дверей, проёмов (сквозных и глухих). Управление освещением помещений (точечный, прожектор и направленный источники света с рядом настраиваемых параметров).

САПР **DS 3D** предлагает динамичный выбор материалов из широкого списка различных видов металла, дерева, камня, пластика и других материалов с возможностью предварительного просмотра. С **DS 3D** вы сможете управлять такими параметрами материалов покрытия, как цвет поверхности, отражающая способность, зеркальность, прозрачность, текстура и другие различные настройки. Возможности формирования собственного каталога материалов.

Также позволяет создавать фотореалистичные изображения интерьеров и 3D моделей мебели с требуемым разрешением. Имеется возможность управления цветом заднего фона или использования изображений стандартных графических форматов. Сохранение и вывод на печать изображений.

Интерфейс САПР показан на рисунке 1.1.

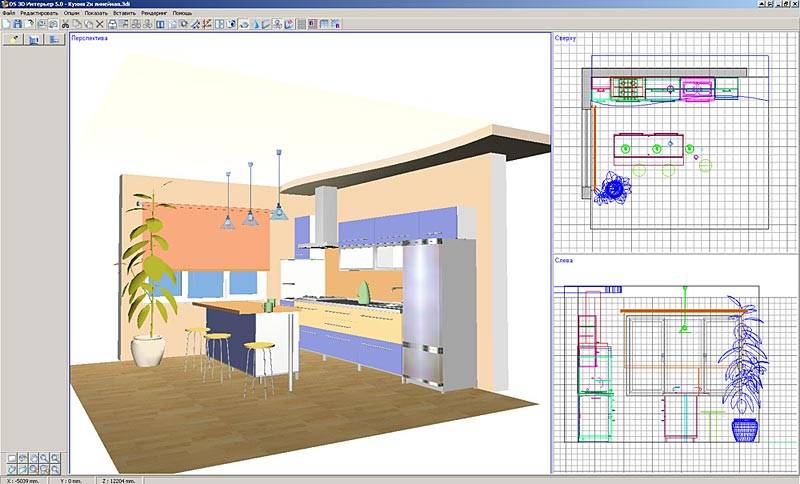


Рисунок 1.1 — Интерфейс САПР DS 3D

**PRO100**

PRO100 — программа для проектирования мебели, кухни, ванных комнат, интерьеров офисов и помещений. [6]

Данный САПР имеет в себе большое разнообразие мебели, что позволяет быстро проектировать интерьер помещения. А также множество других инструментов для решения самых разнообразных задач, связанных с интерьером и мебелью.

На рисунке 1.2 показан интерфейс программы PRO100

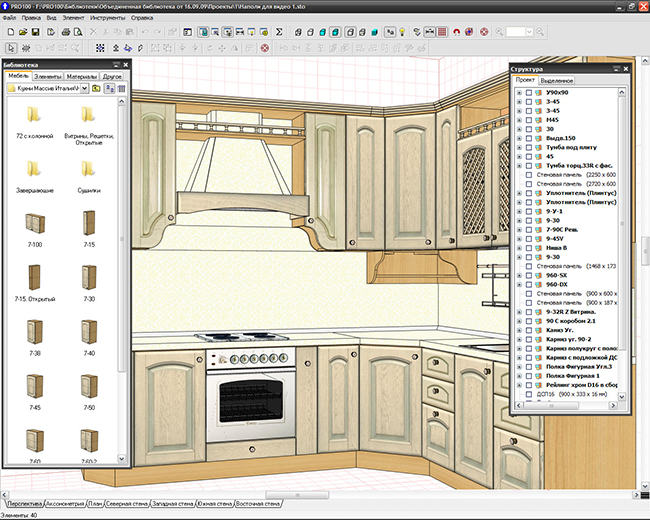


Рисунок 1.2 – Интерфейс САПР PRO100

# 2 Описание предмета проектирования

**Вешалка** — приспособление для хранения одежды в подвешенном состоянии. Одежда на ней вешается на крючки. Также есть место для головных уборов. [7]

На рисунке 2.1 представлен чертеж вешалки

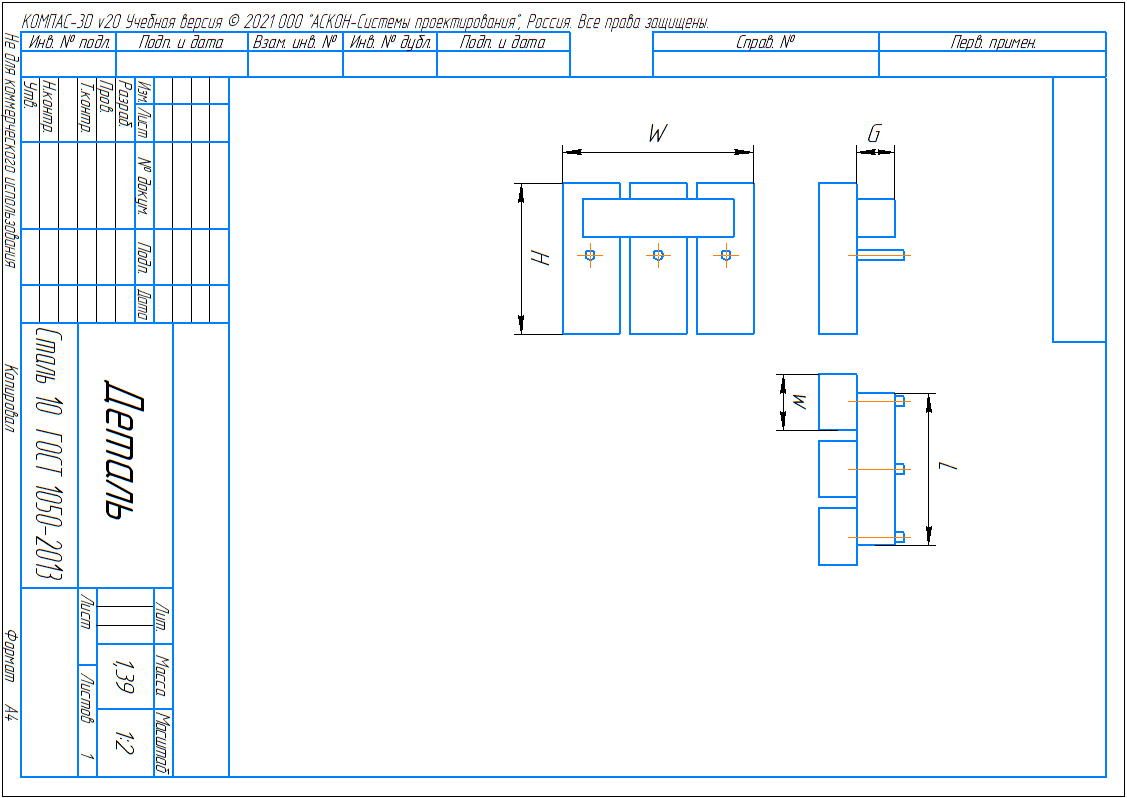


Рисунок 2.1 – Чертеж вешалки

Параметры забора:

1. общая ширина вешалки ***W*** (60см — 100см);
2. общая высота вешалки ***H*** (80см — 150см);
3. ширина одной секции ***w*** (20см — 1/3 от общей ширины вешалки);
4. ширина полки ***L***(50см — полной ширины вешалки);
5. глубина полки ***G*** (10см — 20см);

# 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. [8]

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. [8]

На рисунке 3.2 представлена диаграмма классов.

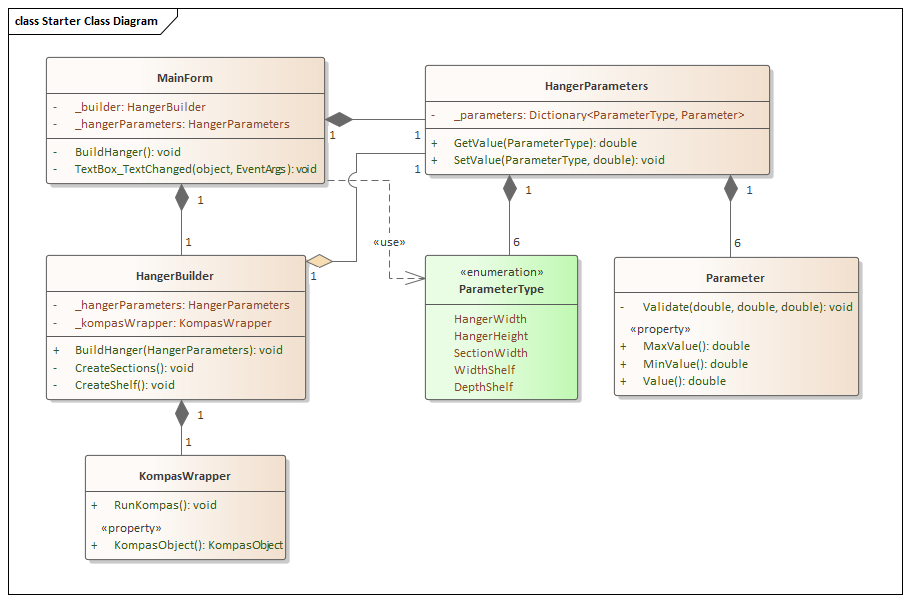


Рисунок 3.1 – Диаграмма классов

Класс MainForms является главным окном, содержит экземпляр класса HangerParameter и экземпляр класса HangerBuilder (таблицы 4.1 – 4.2).

Класс HangerParameters хранит в себе словарь параметров. Ключом является ParameterType (хранит в себе все изменяемые параметры для построения), а значениями являются экземпляры классов Parameter (таблицы 4.3 – 4.4).

ParameterType является перечислением типов параметров для построение вешалки (таблица 4.5).

Класс Parameter содержит значение самого параметра, а также допустимые максимальное и минимальное значения параметра. Проверка параметров осуществляется в методеValidate (таблица 4.6).

Класс HangerBuilder выступает в роли строителя вешалки. Данный класс содержит в себе экземпляр класса KompasWrapper, который связывает программу с Компас API, а также экземпляр класса параметров вешалки (таблицы 4.7 – 4.8).

Таблица 4.1 – Описание полей класса «MainForm»

|  |  |  |
| --- | --- | --- |
| Название | Тип данных | Описание |
| \_builder | HangerBuilder | Экземпляр класса строителя вешалки |
| \_hangerParameters | HangerParameters | Экземпляр класса параметров |

Таблица 4.2 – Описание методов класса «MainForm»

|  |  |  |
| --- | --- | --- |
| Название | Входные параметры | Описание |
| BuildHanger |  | Метод вызова построения фигуры |
| TextBox\_TextChanged | object,  EventArgs args | Обработчик событий при изменении текста TextBox |

Таблица 4.3 – Описание полей класса «HangerParameters»

|  |  |  |
| --- | --- | --- |
| Название | Тип данных | Описание |
| \_parameters | Dictionary<ParameterType, Parameter> | Словарь параметров |

Таблица 4.4 – Описание методов класса и свойств «HangerParameters»

|  |  |  |  |
| --- | --- | --- | --- |
| Название | Входные параметры | Возвращаемое значение | Описание |
| GetValue | ParameterType | double | Метод получения значения по типу параметра |
| SetValue | ParameterType, double | void | Метод записи значения по типу параметра |

Таблица 4.5 – Описание перечисления «ParameterType»

|  |  |
| --- | --- |
| Название | Описание |
| HangerWidth | Общая ширина вешалки |
| HangerHeight | Общая высота вешалки |
| SectionWidth | Ширина секции |
| WidthShelf | Ширина полки |
| DepthShelf | Глубина полки |

Таблица 4.6 – Описание методов и свойств класса «Parameter»

|  |  |  |  |
| --- | --- | --- | --- |
| Название | Входные параметры | Возвращаемое значение | Описание |
| Validate | double, double, double | void | Метод проверки корректности значения |
| MaxValue |  | double | Свойство максимального допустимого значения параметра |
| MinValue |  | double | Свойство минимального допустимого значения параметра |
| Value |  | double | Свойство текущего значения параметра |

Таблица 4.7 – Описание полей класса «HangerBuilder»

|  |  |  |
| --- | --- | --- |
| Название | Тип данных | Описание |
| \_hangerParameters | HangerParameters | Экземпляр класса параметров |
| \_kompasWrapper | KompasWrapper | Экземпляр класса оболочки SDK Компаса |

Таблица 4.8 – Описание методов класса «HangerBuilder»

|  |  |  |  |
| --- | --- | --- | --- |
| Название | Входные параметры | Возвращаемое значение | Описание |
| BuildHanger | HangerParameters | void | Метод построения вешалки |
| CreateSections |  | void | Метод создания секции вешалки |
| CreateShelf |  | void | Метод создания полки вешалки |

Таблица 4.9 – Описание методов и свойств класса «KompasWrapper»

|  |  |  |  |
| --- | --- | --- | --- |
| Название | Входные параметры | Возвращаемое значение | Описание |
| RunKompas |  | void | Метод запуска САПР Компас 3D |
| KompasObject |  | KompasObject | Свойство интерфейсов API Компас |

# 3.3 Макет пользовательского интерфейса

Макет пользовательского интерфейса представляет собой форму для ввода параметров вешалки. Построение модели осуществляется путем нажатия на кнопку «Построить». При попытке ввода некорректных символов они будут подсвечиваться красным цветом и будет высвечиваться сообщение об ошибке, а кнопка «Построить» будет не активна.

На рисунке 3.2 представлен макет пользовательского интерфейса.

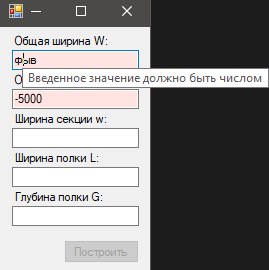


Рисунок 3.2 – Макет пользовательского интерфейса

# Список литературы

1. САПР — Википедия. [Электронный ресурс]. — Режим доступа: https://ru.wikipedia.org/wiki/Система\_автоматизированного\_проектирования (дата обращения 15.03.2022).

2. Компас 3D – Функционал, описание продукта. [Электронный ресурс]. – Режим доступа: https://kompas.ru/ (дата обращения 15.03.2022)

3. API — Википедия. [Электронный ресурс]. — Режим доступа: https://ru.wikipedia.org/wiki/API (дата обращения 15.03.2022).

4. Компас 3D для разработчиков. [Электронный ресурс]. – Режим доступа: https://kompas.ru/solutions/developers/ (дата обращения 15.03.2022)

5. САПР DS 3D. [Электронный ресурс]. — Режим доступа: https://allsoft.ru/software/vendors/pra-estetika/ds-3d-elektronnaya-versiya/#product\_description (дата обращения 15.03.2022).

6. САПР PRO100. [Электронный ресурс]. — Режим доступа: https://www.ecru.pl/ru (дата обращения 15.03.2022).

7. Настенная вешалка. [Электронный ресурс]. – Режим доступа: https://ru.wikipedia.org/wiki/Вешалка/ (дата обращения 15.03.2022).

8. UML. [Электронный ресурс]. – Режим доступа: http://www.uml.org/ (дата обращения 15.03.2022).