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 «Рэковая стойка для гитар» ДЛЯ**

**«КОМПАС 3D V20»**

Проект системы по лабораторному проекту

по дисциплине «Основы разработки САПР»

Выполнил:

студент гр. 588-3

\_\_\_\_\_\_\_\_\_Высоких С.А.

«\_\_\_» \_\_\_\_\_\_\_\_\_ 2022 г.

Руководитель:

к.т.н., доцент каф. КСУП:

\_\_\_\_\_\_\_\_ Калентьев А.А.

«\_\_\_» \_\_\_\_\_\_\_\_\_ 2022 г.

Томск 2022

**Оглавление**

[1. Описание САПР 3](#_Toc86356821)

[1.1. Описание программы 3](#_Toc86356822)

[1.2. Описание API 3](#_Toc86356823)

[1.3. Обзор аналогов 7](#_Toc86356824)

[2. Описание предмета проектирования 8](#_Toc86356825)

[3. Проект программы 9](#_Toc86356826)

[3.1 Макет пользовательского интерфейса 9](#_Toc86356827)

[Список литературы 12](#_Toc86356828)

# 1. **Описание САПР**

## **1.1. Описание программы**

КОМПАС-3D — система трехмерного проектирования, ставшая стандартом для тысяч предприятий и сотни тысяч профессиональных пользователей, благодаря сочетанию проектирования изделий любой сложности и простоты освоения. Особенностью продукта является использование российского геометрического ядра C3D и собственных программных технологий. КОМПАС-3D обеспечивает поддержку файлов наиболее распространённых CAD-систем (SolidWorks, Autodesk Inventor, Solid Edge, Creo, NX, Catia), что позволяет организовывать совместную работу со смежными организациями и заказчиками, использующими другие программные продукты.[1]

## 1.2. Описание API

В КОМПАС-3D 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.

Главным интерфейсом API системы КОМПАС является KompasObject. Получить указатель на этот интерфейс (если быть точным, на интерфейс приложения API 5) можно с помощью экспортной 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 (интерфейсы динамического массива, работы с математическими функциями, библиотек моделей или фрагментов и различных структур параметров определенного типа). [2]

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 | Тип | Описание |
| Document3D() | ksDocument | Метод для получения указателя на интерфейс трёхмерного документа (детали или сборки) |
| GetMathematic2D() | ksMathematic2D | Метод для получения указателя на интерфейс для работы с математическими функциями в графическом документе |
| GetDynamicArray(long type) | ksDynamicArray | Метод для получения указателя на интерфейс динамического массива ksDynamicArray |
| GetParamStruct (short structType) | StructType2D | Метод для получения указателя на интерфейс структуры параметров объекта нужного типа |
| visible | bool | Cвойство видимости приложения |

В таблице 1.2 представлены методы интерфейса ksDocument3D, которые будут использованы при разработке плагина

Таблица 1.2 – Некоторые методы интерфейса ksDocument3D

|  |  |  |
| --- | --- | --- |
| Название | Тип | Описание |
| Create (bool invisible, bool typeDoc) | bool | Метод для создания документа-модели (детали или сборки) |
| UpdateDocumentParam() | bool | Метод для активизации изменённых параметров продукта |
| GetPart(int type) | ksPart | Метод для получения указателя на интерфейс компонента в соответствии с заданным типом |

В таблице 1.3 представлены методы интерфейса ksPart, которые будут использованы при разработке плагина

Таблица 1.3 – Некоторые методы интерфейса ksPart

|  |  |  |
| --- | --- | --- |
| Название | Тип | Описание |
| EntityCollection (short objType) | ksEntityCollection | Формирует массив объектов и возвращает указатель на его интерфейс |
| GetDefaultEntity (short objType) | ksEntity | Получить указатель на интерфейс объекта, создаваемого системой по умолчанию |
| GetPart(int type) | ksPart | Получить указатель на интерфейс компонента в соответствии с заданным типом |
| NewEntity (short objType) | ksEntity | Создать новый интерфейс объекта и получить указатель на него |

В таблице 1.4 представлены методы интерфейса ksEntity, которые будут использованы при разработке плагина

Таблица 1.4 – Некоторые методы интерфейса ksEntity

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

В таблицах 1.5 и 1.6 представлены методы и свойства интерфейса ksPlaneOffsetDefinition, которые будут использованы при разработке плагина

Таблица 1.5 – Свойства интерфейса ksPlaneOffsetDefinition

|  |  |  |
| --- | --- | --- |
| Название | Тип | Описание |
| direction | bool | Направление смещения от базовой плоскости |
| offset | double | Смещение (расстояние) от базовой плоскости |

Таблица 1.6 – Методы интерфейса ksPlaneOffsetDefinition

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

## 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.[3].

# 2. Описание предмета проектирования

Предметом проектирования является модель рэковой стойки для гитар. Данная модель имеет 6 основных параметров

1. Высота стойки H (от 600 до 700 мм);
2. Ширина опоры для гитар s1 (от 200 до 300 мм);
3. Количество разделителей n (от 2 до 7);
4. Ширина разделителей ln (от 50 до 100 мм);
5. Ширина стойки L (от (2\*s1+100) до (n\*s1+150) мм);
6. Длина опоры стойки S (от 400 до (s1+250) мм);

Все размеры указаны ниже на рисунке 2.1.

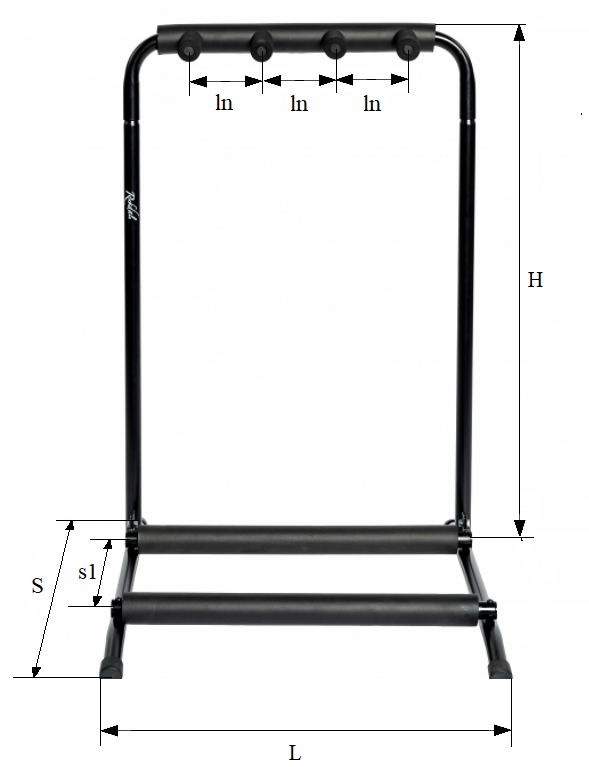


Рисунок 2.1 – Схема стойки с параметрами.

# 3. Проект программы

## 3.1 Диаграмма классов

Диаграмма классов определяет типы классов системы и различного рода статические связи, которые существуют между ними. На диаграммах классов изображаются также атрибуты классов, операции классов и ограничения, которые накладываются на связи между классами.[4] Целью создания диаграммы классов является графическое представление статической структуры элементов системы. Диаграмма классов плагина представлена на рисунке 3.1.

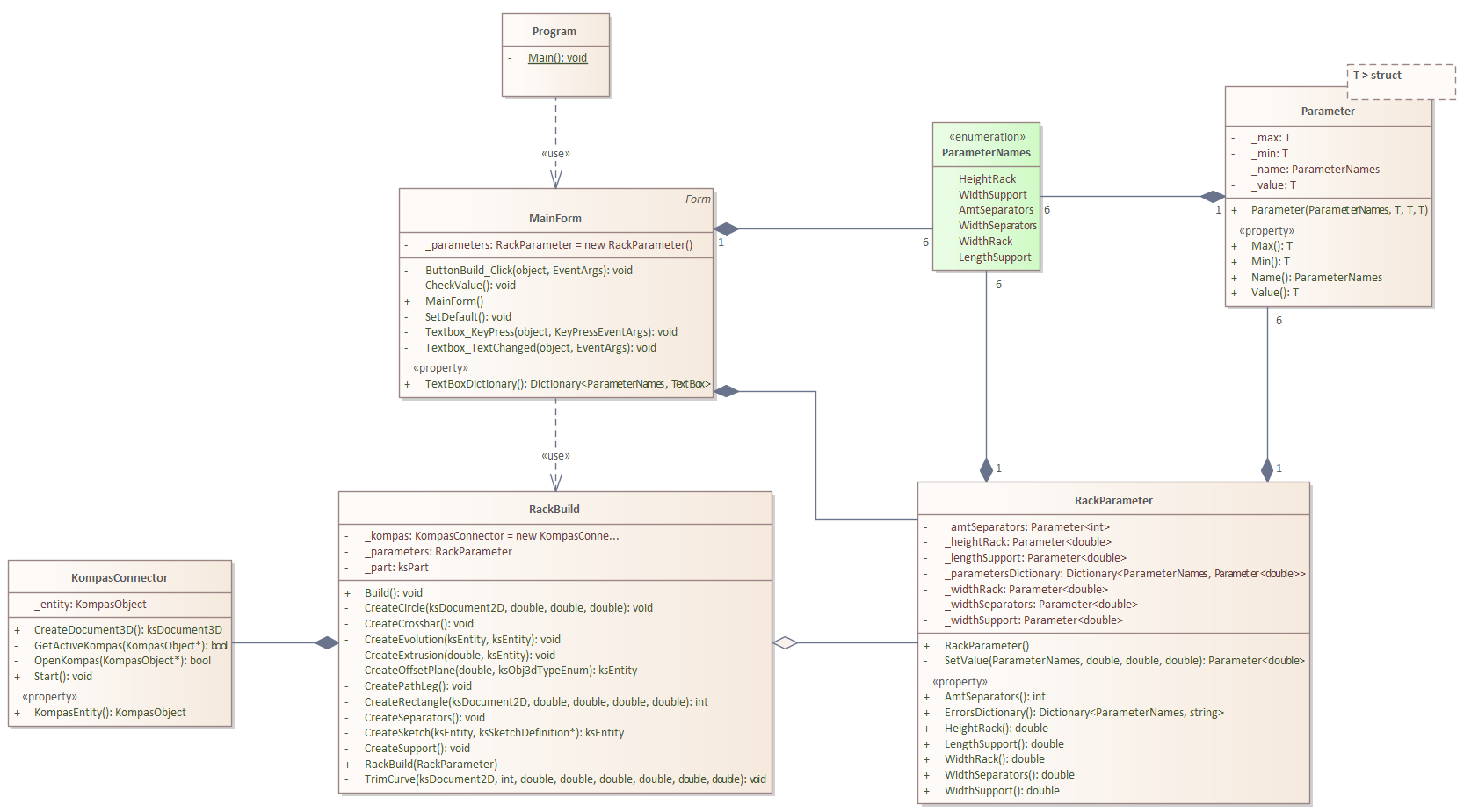


Рисунок 3.1 – Диаграмма классов плагина «Рэковая стойка для гитар»

Описание полей, свойств и методов представлено в приложении А (таблицы А.1 – А.4).

## 3.1 Макет пользовательского интерфейса

Плагин представляет собой пользовательскую форму с полями для ввода 6 геометрических параметров: «Высота стойки», «Ширина опоры для гитар», «Количество разделителей», «Ширина разделителей», «Ширина стойки», «Длина опоры стойки». Ниже располагается кнопка для запуска построения. Справа находится изображение, демонстрирующее размеры.

Макет пользовательского интерфейса представлен на рисунке 3.1.

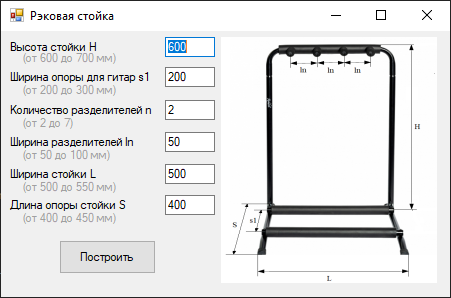


Рисунок 3.1 – Макет пользовательского интерфейса.

При вводе параметров, выходящих за диапазон, соответствующее поле подсвечивается красным цветом (рис. 3.2). При нажатии на кнопку «Построить запускается построение модели. При попытке запустить построение с неверными параметрами будет выведено соответствующее сообщение. Пример сообщения представлен на рисунке 3.3.

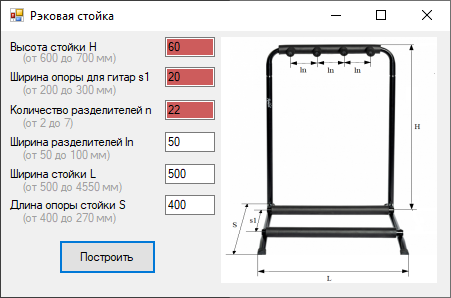


Рисунок 3.2 – Пример ввода неверных данных.

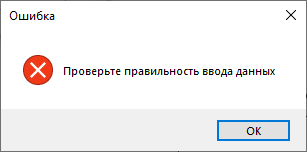


Рисунок 3.3 – Пример сообщения о некорректности данных.

# Список литературы

1. Система трёхмерного моделирования КОМПАС-3D [электронный ресурс]. – URL: <https://ascon.ru/products/7/review/> (дата обращения 11.12.2021).

2. Базовые интерфейсы API системы КОМПАС [электронный ресурс]. – URL: <https://it.wikireading.ru/23741> (дата обращения 11.12.2021).

3. Конструкторы мебели для самостоятельного проектирования: обзор лучшего софта. [Электронный ресурс]. – URL: <https://amssoft.ru/repair/programmy-dlya-proektirovaniya-mebeli.php> (дата обращения 13.12.2021).

4. Фаулер М. UML. Основы. 3-е издание / М.Фаулер. – 3-е изд., пер. с англ. – СПб.: Символ-Плюс, 2019. – 192 с. (дата обращения 14.12.2021).

# Приложение А

(справочное)

Таблица А.1 – Описание полей, свойств и методов класса Parameter

|  |  |  |
| --- | --- | --- |
| Название | Тип данных | Описание |
| \_min | T | Закрытое поле. Минимальное значение параметра |
| \_max | T | Закрытое поле. Максимальное значение параметра |
| \_value | T | Закрытое поле. Значение параметра |
| \_name | ParameterNames | Закрытое поле. Имя параметра |
| Name | ParameterNames | Свойство. Устанавливает или возвращает имя |
| Value | T | Свойство. Устанавливает или возвращает значение |
| Min | T | Свойство. Устанавливает или возвращает минимальное значение |
| Max | T | Свойство. Устанавливает или возвращает максимальное значение |
| Parameter  (ParameterNames,  T, T, T) |  | Конструктор класса |

Таблица А.2 – Описание полей, свойств и методов класса KompasConnector

|  |  |  |
| --- | --- | --- |
| Название | Тип данных | Описание |
| \_entity | KompasObject | Закрытое поле. Экземпляр API КОМПАС-3D |
| KompasEntity | KompasObject | Свойство. Возвращает экземпляр API КОМПАС-3D |
| Start() | void | Устанавливает связь с КОМПАС-3D |
| GetActiveKompas  (KompasObject\*) | bool | Подключается к открытому КОМПАС-3D |
| OpenKompas  (KompasObject\*) | bool | Открывает КОМПАС-3D |
| CreateDocument3D() | ksDocument3D | Создаёт документ в КОМПАС-3D |

Таблица А.3 – Описание полей, свойств и методов класса RackParameter

|  |  |  |
| --- | --- | --- |
| Название | Тип данных | Описание |
| \_heightRack | Parameter<double> | Закрытое поле. Высота стойки |
| \_widthSupport | Parameter<double> | Закрытое поле. Ширина опоры для гитар |
| \_amtSeparators | Parameter<int> | Закрытое поле. Количество разделителей |
| \_widthSeparators | Parameter<double> | Закрытое поле. Ширина разделителей |
| \_widthRack | Parameter<double> | Закрытое поле. Ширина стойки |
| \_lengthSupport | Parameter<double> | Закрытое поле. Длина опоры стойки |
| \_parametersDictionary | Dictionary  <ParameterNames, Parameter<double>> | Закрытое поле. Словарь типа «Имя параметра – поле параметра» |
| ErrorsDictionary | Dictionary  <ParameterNames, string> | Словарь типа «Имя параметра – текст ошибки» |
| HeightRack | double | Свойство. Устанавливает или возвращает значение высоты стойки |
| WidthSupport | double | Свойство. Устанавливает или возвращает значение ширины опоры для гитар |
| AmtSeparators | double | Свойство. Устанавливает или возвращает значение количество разделителей |
| WidthSeparators | double | Свойство. Устанавливает или возвращает значение ширину разделителей |
| WidthRack | double | Свойство. Устанавливает или возвращает значение ширины стойки |
| LengthSupport | double | Свойство. Устанавливает или возвращает значение длины опоры стойки |
| SetValue  (ParameterNames, double, double, double) | Parameter<double> | Устанавливает значения параметров |
| RackParameter() |  | Конструктор класса |

Таблица А.4 – Описание полей, свойств и методов класса RackBuild

|  |  |  |
| --- | --- | --- |
| Название | Тип данных | Описание |
| \_parameters | RackParameter | Закрытое поле. Содержит размеры модели |
| \_part | ksPart | Закрытое поле. Объект детали |
| \_kompas | KompasConnector | Закрытое поле. Устанавливает связь с КОМПАС-3D |
| Build() | void | Строит деталь |
| CreatePathLeg() | void | Строит основание стойки |
| CreateCrossbar() | void | Строит верхнюю перекладину |
| CreateSeparators() | void | Строит разделители |
| CreateSupport() | void | Строит опоры для гитар |
| CreateSketch(ksEntity, ksSketchDefinition\*) | ksEntity | Создаёт эскиз |
| CreateOffsetPlane(double, ksObj3dTypeEnum) | ksEntity | Создаёт смещённую плоскость |
| CreateRectangle(ksDocument2D, double, double, double, double) | int | Создаёт прямоугольник |
| CreateCircle(ksDocument2D, double, double, double) | void | Создаёт окружность |
| TrimCurve(ksDocument2D, int, double, double, double, double, double, double) | void | Усекает указанную кривую |
| CreateEvolution(ksEntity, ksEntity) | void | Операция «Элемент по траектории» |
| CreateExtrusion(double, ksEntity) | void | Операция «Выдавливание» |
| RackBuild(RackParameter) |  | Конструктор класса |