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 "НАПРАВЛЯЮЩАЯ" ДЛЯ САПР КОМПАС 3D

**ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**  
на разработку плагина моделирования мебельной направляющей для системы «Компас 3D V20»

Выполнил:

студент гр. 588-2

\_\_\_\_\_\_\_\_\_\_\_\_\_\_Матинин А.С.

«\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2021 г.

Руководитель:

к.т.н., доцент каф. КСУП

\_\_\_\_\_\_\_\_\_\_\_\_\_\_Калентьев А. А.

«\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2021 г.

Содержание

[1 Описание САПР 2](#_Toc92980157)

[1.1 Описание программы 2](#_Toc92980158)

[1.2 Описание API 2](#_Toc92980159)

[1.3 Обзор аналогов 8](#_Toc92980160)

[2 Описание предмета проектирования 10](#_Toc92980161)

[3 Проект программы 11](#_Toc92980162)

[3.1 Описание технических и функциональных аспектов проекта 11](#_Toc92980163)

[3.2 Диаграмма классов 11](#_Toc92980164)

[3.3 Макет пользовательского интерфейса 12](#_Toc92980165)

[Список литературы 14](#_Toc92980166)

# 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 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 [2].

## Описание API

API (англ. Application Programming Interface) – описание способов, которыми одна компьютерная программа может взаимодействовать с другой программой[3].

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 параметры | Тип возвращаемых данных | Описание |
| Document3D() |  | ksDocument | Метод для получения указателя на интерфейс трехмерного графического документа (детали или сборки) |
| 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 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 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 – | bool | Создать документ-модель (деталь или сборку) | |
|  | видимый режим),  typeDoc – тип документа (true – деталь, false – сборка). |  | |  |
| 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 – комплексная система автоматизации проектирования, технологической подготовки производства и реализации корпусной мебели.

БАЗИС-Мебельщик - основной модуль системы БАЗИС. Он предназначен для создания изделий корпусной мебели любой сложности, с возможностью автоматического получения полного комплекта чертежей и спецификации. Применение модуля БАЗИС-Мебельщик позволяет сократить время проектирования и технологической подготовки производства изделий в 10-15 раз по сравнению с ручной работой при значительном сокращении количества субъективных ошибок. [4]

На рисунках 1.1 – 1.3 представлено окно программы «Базис-мебельщик».



Рисунок 1.1 – Главное окно программы



Рисунок 1.2 – Окно выбора фурнитуры



Рисунок 1.3 – Окно для редактирования некоторых параметров направляющей

# Описание предмета проектирования

Предметом проектирования является модель направляющей.

Параметры направляющей:

А. Длина направляющей (50мм – 150мм)

Б. Ширина направляющей (10мм – 30мм)

В. Толщина направляющей (5мм – 20мм

Г. Длина хода крепления (20мм – 100мм; больше чем 5 \* ширина хода крепления)

Д. Ширина хода крепления (2мм-20мм; меньше чем 0.7 \* Ширина направляющей и больше чем 0.9 \* Ширина направляющей)

Е. Диаметр отверстия для крепления к плоскости (2мм -20мм)

Ж. Угол наклона направляющей (65° - 270°)

На рисунке 2.1 представлен 3D модель с выноской размеров.



Рисунок 2.1 – 3D модель с выноской размеров

# Проект программы

## 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. [5]

При использовании UML были простроена диаграмма классов.

## Диаграмма классов

Диаграмма классов (class diagram) показывает набор классов, интерфейсов и коопераций, а также их связи. Диаграммы этого вида чаще всего используются для моделирования объектно-ориентированных систем. [5]

На рисунке 3.1 представлена первая версия диаграммы классов.

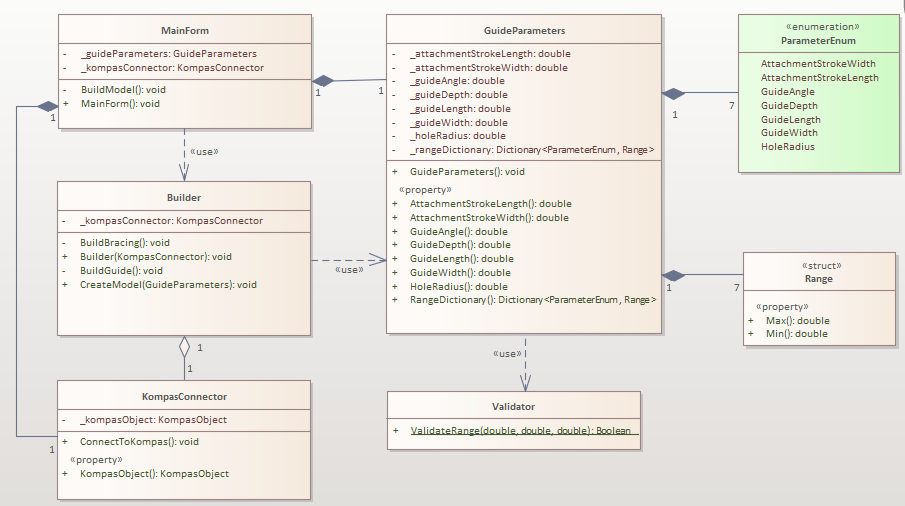


Рисунок 3.1 – Первая версия UML диаграмма классов

Финальная версия диаграммы классов представлена на рисунке 3.2.

Рисунок 3.2 – Финальная версия UML диаграмма классов

## Макет пользовательского интерфейса

Макет пользовательского интерфейса представляет собой форму для ввода параметров направляющей (рисунок 3.3). Построение модели осуществляется путем нажатия на кнопку «Построить». Если в поле были введены некорректные данные, при потере фокуса, поле окрасится в красный (рисунок 3.4) и на экране высветиться сообщение с информацией об ошибке (рисунок 3.5). Пока все параметры не будут введены корректно, кнопка «Построить» будет неактивна. При нажатии на поле, в правой части окна будет выводится изображение, показывающее изменяемый параметр (рисунок 3.6).

Так же после корректного ввода, параметры сохраняются в json файл. При последующем запуске, параметры выгружаются в плагин.

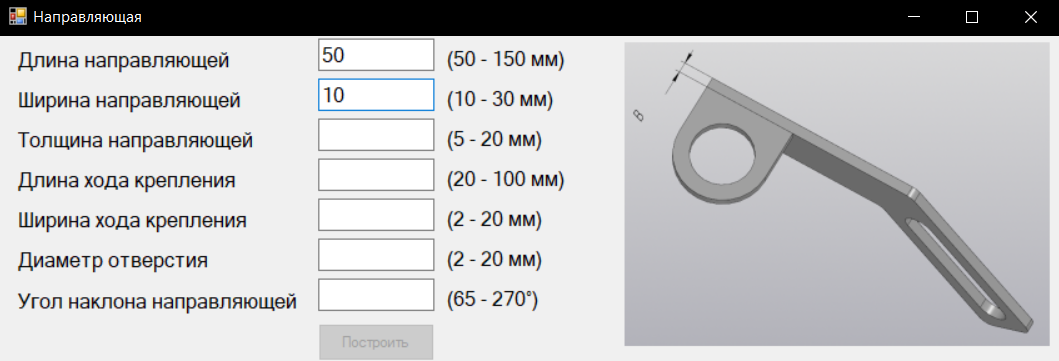


Рисунок 3.3 – Макет пользовательского интерфейса

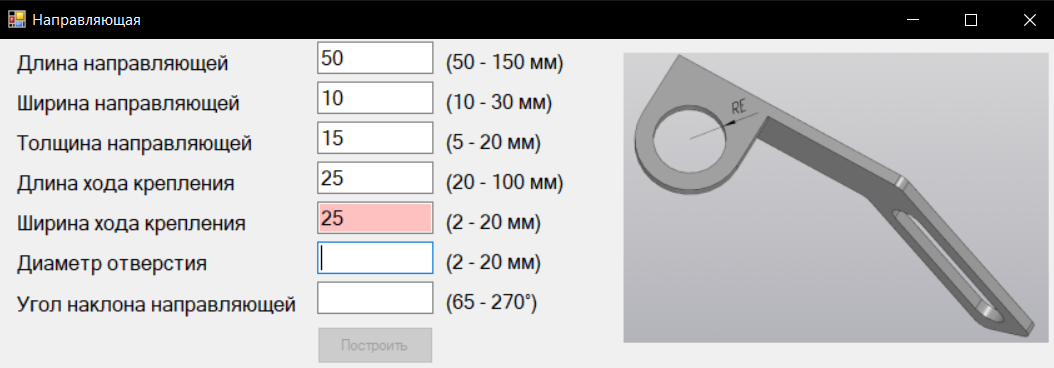


Рисунок 3.3 – Макет пользовательского интерфейса с введенными некорректными данными

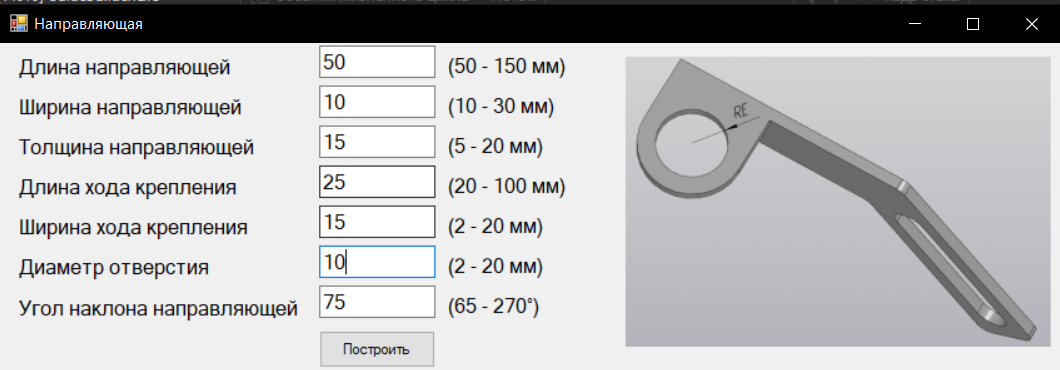


Рисунок 3.2 – Макет пользовательского интерфейса с активной кнопкой «Построить»

# Тестирование программы

## Функциональное тестирование

Во время функционального тестирования выполнялась проверка работы плагина, а именно, соответствие построенной модели с веденными параметрами.

На рисунке 4.1 представлена модель направляюще с минимальными параметрами (ПАРАМЕТРЫ)

На рисунке 4.2 представлена модель направляюще с максимальными параметрами (ПАРАМЕТРЫ)

## Модульное тестирование

В целях проверки бизнес логики плагина было проведено модульное тестирование. Тестирование проводилось при помощи тестового фреймворка NUnit. Степень покрытия бизнес логики составило сто процентов (рисунок 4.1).

Рисунок 4.1

## Нагрузочное тестирование

В целях проверки производительности работы плагина, было проведено нагрузочное тестирование [14]. Тестирование производилось на ПК со следующей конфигурацией:

* ЦП ЦП AMD Ryzen 5 4600H with Radeon Graphics 3.00 ГГц;
* 16 ГБ ОЗУ;

Для нагрузочного тестирования создан метод с бесконечным циклом построения детали, представленный на рисунке 4.2. Для измерения времени использовался класс Stopwatch.

Рисунок 5.5 – Реализация зацикленного перестроения модели

Первое тестирование заключалось в построении детали с минимальными параметрами (диаметр 101.6 мм, диаметр ЦО 10.16 мм, посадочная ширина 30 мм, вылет -0.75 мм, диаметр сверловки 4.376 мм, сверловка 4 шт, количество спиц 4 шт). Тестирование проводилось 49 минут, построено 45 моделей автомобильного диска. На протяжении тестирования загруженность процессора составляла около 40 процентов. Результаты тестирования представлены на рисунках 5.6 и 5.7.

Рисунок 5.6 – График зависимости количества потребляемой оперативной памяти от количества деталей для модели с базовыми параметрами

Рисунок 5.7 – График зависимости времени построения одной детали

от количества деталей для модели с базовыми параметрами

Второе тестирование заключалось в построении детали с максимальными параметрами (диаметр 1447.8 мм, диаметр ЦО 260.604 мм, посадочная ширина 593.598 мм, вылет 296.799 мм, диаметр сверловки 70.638 мм, сверловка 15 шт, количество спиц 18 шт). Тестирование проводилось 49 минут, построено 45 моделей автомобильного диска. На протяжении тестирования загруженность процессора составляла около 40 процентов. Результаты тестирования представлены на рисунках 5.8 и 5.9.

Рисунок 5.8 – График зависимости количества потребляемой оперативной памяти от количества деталей для модели с максимальными параметрами

Рисунок 5.9 – График зависимости времени построения одной детали

от количества деталей для модели с максимальными параметрами

Исходя из графиков на рисунках 5.7 и 5.8 можно сделать вывод что загруженность оперативной памяти уменьшалась с течением времени построения моделей, при этом построение модели с максимальными параметрами потребовало больше оперативной памяти, чем построение модели с минимальными параметрами.

Исходя из графиков, представленных на рисунках 5.8 и 5.9 можно утверждать, что время, затрачиваемое на построение модели с минимальными параметрами меньше, чем время, затрачиваемое на построение модели с максимальными параметрами.

# Список литературы

1. САПР — Википедия. [Электронный ресурс]. — Режим доступа: https://ru.wikipedia.org/wiki/Система\_автоматизированного\_проектирования (дата обращения 22.10.2021).

2. КОМПАС-3D. Официальный сайт САПР КОМПАС. [Электронный ресурс]. Режим доступа: https://kompas.ru (дата обращения 22.10.2021).

3. API – Википедия. [Электронный ресурс]. — Режим доступа: https://ru.wikipedia.org/wiki/API (дата обращения 22.10.2021).

4. Базис-центр – Комплекс программ для проектирования и продажи мебели. — Режим доступа: https://www.bazissoft.ru (дата обращения 03.11.2021).

5. Буч Г., Рамбо Д., Якобсон И. Язык UML. Руководство пользователя. 2-е изд.: ДМК Пресс, 2006. – 496 с.