Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования

ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ СИСТЕМ УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ

Кафедра компьютерных систем в управлении и проектировании (КСУП)

**Техническое задание**

По дисциплине «Основы разработки САПР»

1. Выдано: студенту группы 589-2 Каминскому Виктору Михайловичу
2. Тема: разработка плагина “Стамеска” для САПР Компас 3D
3. Срок сдачи готовой работы: 31 декабря 2022 г.
4. Исходные данные для работы:

Требования к программному обеспечению:

* Microsoft Windows 10 (64-разрядная версия);
* язык программирования C# с использованием платформы .NET Framework 4.7.2;
* среда разработки Visual Studio 2019;
* плагин для программы Inventor 2022;
* библиотека для тестирования NUnit 3.13.2;
* система контроля версии Git.

Требование к аппаратному обеспечению:

* ЦП не менее 2.5 ГГц;
* 8 ГБ ОЗУ;
* место на диске — 40 ГБ;
* графический процессор объемом памяти 6 ГБ;
* экран расширением 1980 х 1240.

**1 Описание САПР**

**1.1 Описание Программы**

КОМПАС-3D — система трёхмерного твердотельного и поверхностного параметрического проектирования (САПР) компании Аскон, предназначенная для создания цифровых прототипов промышленных изделий. Инструменты Inventor обеспечивают полный цикл проектирования и создания конструкторской документации:

* 2D-/3D-моделирование;
* создание изделий из листового материала и получение их разверток;
* разработка электрических и трубопроводных систем;
* проектирование оснастки для литья пластмассовых изделий;
* динамическое моделирование;
* параметрический расчет напряженно-деформированного состояния деталей и сборок;
* визуализация изделий;
* автоматическое получение и обновление конструкторской документации (оформление по ЕСКД).

**1.2 Описание API**

**API(**от анг.**A**pplication **P**rogramming **I**nterface, дословно Интерфейс Прикладного Программирования**) –** описание способов взаимодействия одной программы с другой.

API упрощает процесс программирования при создании приложений, абстрагируя базовую реализацию и предоставляя только объекты или действия, необходимые разработчику. Если графический интерфейс для почтового клиента может предоставить пользователю кнопку, которая выполнит все шаги для выборки и выделения новых писем, то API для ввода/вывода файлов может дать разработчику функцию, которая копирует файл из одного места в другое, не требуя от разработчика понимания операций файловой системы, происходящих за кулисами

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.

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

Таблица 1.1 – Методы и свойства интерфейса KompasObject, используемые при разработке плагина

|  |  |  |
| --- | --- | --- |
| Название | Тип | Описание |
| Document3D() | ksDocument | Метод для получения указателя на интерфейс трехмерного графического документа (детали или сборки) |
| GetParamStruct(short structType) | [StructType2D](mk:@MSITStore:C:\Program%20Files\ASCON\KOMPAS-3D%20V16\SDK\SDK.chm::/StructType2D.htm) | 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, которые были использованы при разработке плагина.

Таблица 1.2 – Методы интерфейса ksEntity, используемые при разработке плагина

|  |  |  |
| --- | --- | --- |
| 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 ksDocument2D, которые были использованы при разработке плагина.

Таблица 1.3 – Методы интерфейса ksDocument2D, используемые при разработке плагина

|  |  |  |
| --- | --- | --- |
| Название | Тип | Описание |
| ksLineSeg(double x1, double y1, double x2, double y2, int style) | int | Получить указатель на отрезок на двумерной плоскости либо 0 в случае ошибки |
| ksRegularPolygon(  ksRegularPolygonParam param, int style) | int | Получить указатель на многоугольник на двумерной плоскости либо 0 в случае ошибки |
| ksRectangle(  ksRectangleParam param, int style) | int | Получить указатель на прямоугольник на двумерной плоскости либо 0 в случае ошибки |
| ksCircle(  double xc, double yc, double rad, int style) | int | Получить указатель на окружность на двумерной плоскости либо 0 в случае ошибки |

В таблице 1.4 представлены свойства и методы интерфейса ksDocument3D, которые были использованы при разработке плагина.

Таблица 1.4 – Методы интерфейса ksDocument3D, используемые при разработке плагина

|  |  |  |
| --- | --- | --- |
| Название | Тип | Описание |
| Create (bool invisible, bool \_typeDoc) | bool | Создать документ-модель (деталь или сборку) |
| GetPart(int type) | ksPart | Получить указатель на интерфейс компонента в соответствии с заданным типом |

В таблице 1.5 представлены методы интерфейса ksPart, которые были использованы при разработке плагина.

Таблица 1.5 – Свойства и методы интерфейса ksPart, используемые при разработке плагина

|  |  |  |
| --- | --- | --- |
| Название | Тип | Описание |
| EntityCollection(short objType) | ksEnintyCollection | 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 1.6 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\_unknown | Неизвестный (включает все объекты) |  |
| o3d\_planeXOZ | Плоскость XOZ | ksPlaneParam |
| o3d\_planeYOZ | Плоскость YOZ | ksPlaneParam |
| o3d\_planeXOY | Плоскость XOY | ksPlaneParam |
| o3d\_sketch | Эскиз | [ksSketchDefinition](mk:@MSITStore:C:\Program%20Files\ASCON\KOMPAS-3D%20V16\SDK\SDK.chm::/ksSketchDefinition.htm) |
| o3d\_face | Грань | [ksFaceDefinition](mk:@MSITStore:C:\Program%20Files\ASCON\KOMPAS-3D%20V16\SDK\SDK.chm::/ksEdgeDefinition.htm) |
| o3d\_baseExtrusion | Базовая операция выдавливания | [ksBaseExtrusionDefinition](mk:@MSITStore:C:\Program%20Files\ASCON\KOMPAS-3D%20V16\SDK\SDK.chm::/ksBaseExtrusionDefinition.htm) |
| o3d\_cutExtrusion | Вырезать выдавливанием | [ksCutExtrusionDefinition](mk:@MSITStore:C:\Program%20Files\ASCON\KOMPAS-3D%20V16\SDK\SDK.chm::/ksCutExtrusionDefinition.htm) |
| o3d\_baseLoft | Создание элемента по сечениям | [ksBaseLoftDefinition](mk:@MSITStore:C:\Program%20Files\ASCON\KOMPAS-3D%20V16\SDK\SDK.chm::/ksBaseExtrusionDefinition.htm) |
| o3d\_baseEvolution | Создание кинематического элемента | [ksBaseEvolutionDefinition](mk:@MSITStore:C:\Program%20Files\ASCON\KOMPAS-3D%20V16\SDK\SDK.chm::/ksBaseExtrusionDefinition.htm) |
| o3d\_cutEvolution | Вырезать кинематический элемент | [ksCutEvolutionDefinition](mk:@MSITStore:C:\Program%20Files\ASCON\KOMPAS-3D%20V16\SDK\SDK.chm::/ksCutExtrusionDefinition.htm) |

**1.3 Обзор Аналогов**

**1.3.1 Плагин PDF**

Данный плагин позволяет производить экспорт моделей и сборок из КОМПАС-3D в формат PDF формат [6]. Основной особенностью является возможность интерактивного взаимодействия пользователя с сохраненной 3D сценой внутри PDF файла. Например, пользователь может вращать, масштабировать, передвигать детали и сборки внутри 3D PDF файла. Также доступно создание анимации сборки и разборки изделий. Это полезно для подготовки интерактивных сборочных инструкций, создания маркетинговых материалов, презентаций, а также для налаживания взаимодействия между проектировщиками и заказчиками. В подобных ситуациях традиционным подходом являлся экспорт сборки или детали КОМПАС-3D в промежуточный формат и дальнейшее сохранение в формат 3D PDF. Используемый подход в плагине исключает использование промежуточных файлов для осуществления 3D преобразования, что существенно повышает качество выходной 3D модели в формате PDF.

Ключевые возможности плагина:

* сохранение деталей и сборок в формате 3D PDF для интерактивного просмотра при помощи бесплатной программы Adobe Reader;
* создание анимаций, имитирующих естественный порядок сборки и разборки создание имитации анимации гибки листовых тел;
* вставка в существующие PDF документы, содержащие основной текст, фоновые картинки, таблицы спецификаций, эмблемы, логотипы;
* пакетный режим для поочередной конвертации всех файлов.

**1.3.2 Экспорт из КОМПАС-3D в формат 3D PDF**

Пользователям «КОМПАС-3D» стала доступна функция экспорта созданных трехмерных моделей и дальнейшего их использования в создании технической документации. Экспорт происходит в формате 3D PDF [7].

Главной особенностью является то, что пользователь по-прежнему имеет возможность интерактивно взаимодействовать с 3D сценой, находясь внутри файла 3D PDF. То есть пользователь может передвигать детали, вращать их, масштабировать, передвигать сборки внутри самого файла. Пользователь также может создать анимацию сборки и разборки изделия. Этот функционал очень удобен. Он используется при создании презентаций, маркетинговых материалов, при подготовке интерактивных сборочных конструкций. Он значительно упрощает взаимодействие между заказчиками и проектировщиками.

Компания Visual Technology Services Ltd. из Великобритании разработала плагин PDF3D, предоставляющий доступ к описанным выше возможностям.

Основной функционал плагина:

* сохранение сборок и деталей для интерактивного просмотра в формате 3D PDF с помощью программы Adobe Reader;
* создание анимации, имитирующей естественный порядок разборки и сборки;
* создание анимации, имитирующей гибки листового тела;
* возможность вставки в PDF документ, содержащий основной текст, фоновых картинок, таблиц, логотипов, эмблем, спецификаций и т.д.

**2 Описание объекта проектирования**

Изображение моделируемого объекта:

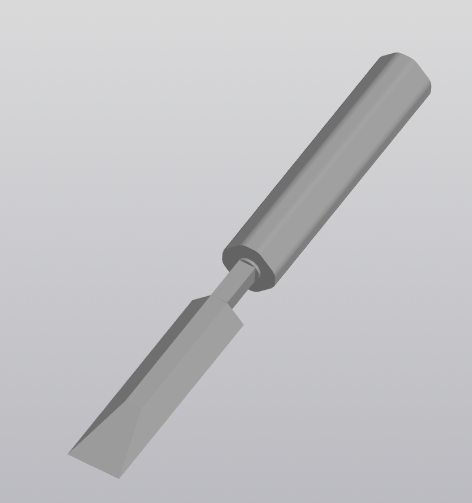


Рисунок 1 Изображение моделируемого объекта в 3Д



Рисунок 2 Изображение фронтальной проекции моделируемого объекта



Рисунок 3 Изображение боковой проекции моделируемого объекта

Измеряемые параметры для плагина:

* Lr – Длина рукояти (120-200 мм);
* L – Длиина всей стамески ( 240-400 мм);
* R – Диаметр кольца (10 – 20 мм);
* Hl –Высота лезвия( 8-18 мм);
* Lzm – Расстояние между кольцом и лезвием (15-25 мм);
* Lzb – Расстояние между рукоятью и лезвием (17 – 27 мм);
* W – Ширина всей стамески (25-40 мм).

Длина лезвия L-Lr всегда больше чем длина Lr рукояти максимум на 20%, либо равна ей (Lr=<1.2(L-Lr)). Диаметр кольца R больше высоты лезвия Hl на 10%, но меньше ширины стамески W ((R=Hl\*1.1)||(R<W)). Длина зазора Lzm меньше длины зазора Lbm на 6% (1.06\*Lzm=Lzb).

Назначение программы:

Программа предназначена для автоматизации моделирования детали «Стамеска»

Плагин позволяет пользователю ввести вышеперечисленные значения через графический интерфейс. В программе предусмотрена проверка корректности введенных данных и сообщение пользователю о неправильно заполненных полях с помощью цветового выделения и всплывающих подсказок.

При запуске моделирования с некорректными значениями программа выводит сообщение об ошибке и отменяет построение модели.

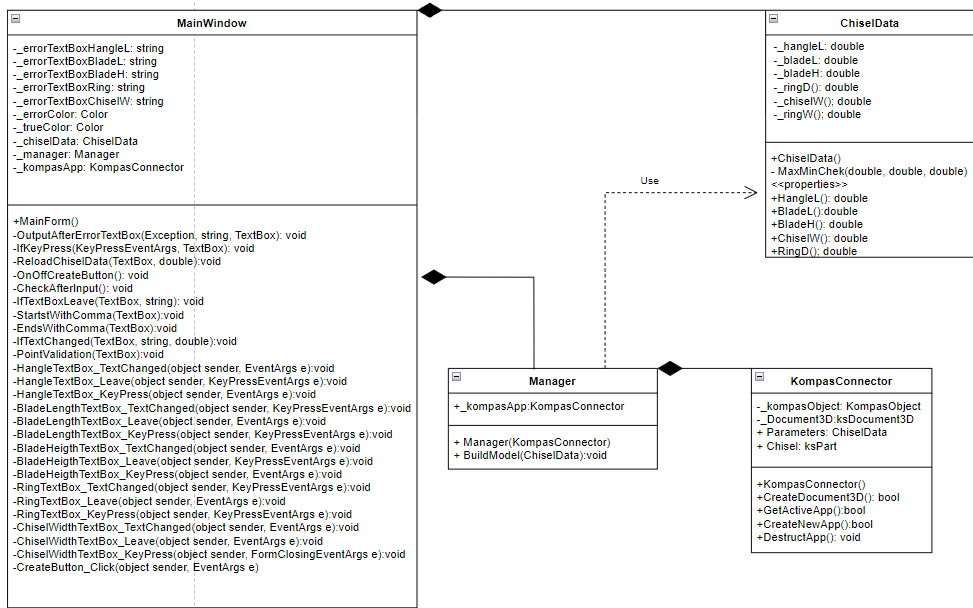
При правильно введенных значениях результатом работы программы будет созданная по ним модель стамески.

3 Проект системы

3.1 Диаграмма классов

Диаграмма классов описывает типы объектов системы и различного рода статические отношения, которые существуют между ними. На диаграммах классов отображаются также свойства классов, операции классов и ограничения, которые накладываются на связи между объектами [4].

Диаграмма классов для данного проекта представлена на рисунке 3.1.



*Рисунок 3.1 – UML-диаграмма классов*

1) MainWindow – класс диалогового окна, который обеспечивает взаимодействие между пользователем и программой.

2) ChiselData – Класс, хранящий в себе данные о стамеске

3) Manager – Класс, осуществляющий инициализацию, создание модели и валидацию данных, а также вызов методов API.

4) KompasConnector – класс для работы с API КОМПАС 3D;

Класс MainWindow композирует классы ChiselData и Manager. Класс Manager использует класс ChiselData и композирует класс KompasConnector.

3.2 Макеты пользовательского интерфейса

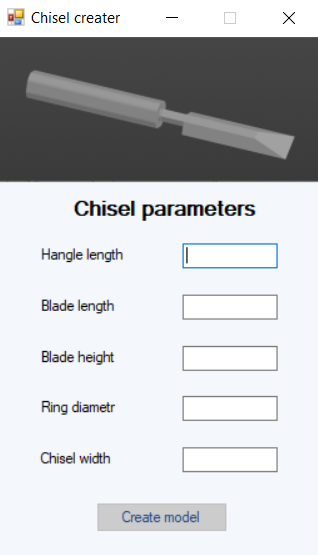


Рисунок 3.2 – Окно плагина без запущенной программы «КОМПАС-3D»

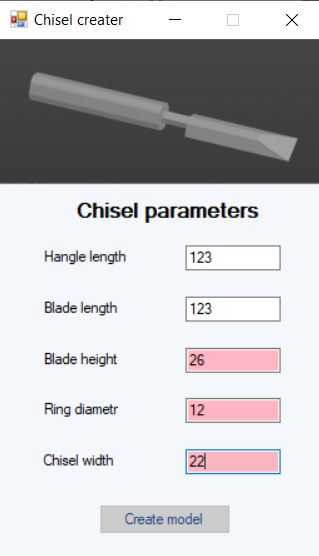


Рисунок 3.3 – Окно плагина с запущенной программой «КОМПАС-3D

Далее, при попытке ввода неправильных символов в ячейки ввода параметров, а также при пустой ячейке ввода параметров программа выдает ошибку проверки правильности ввода данных (рисунок 3.4)

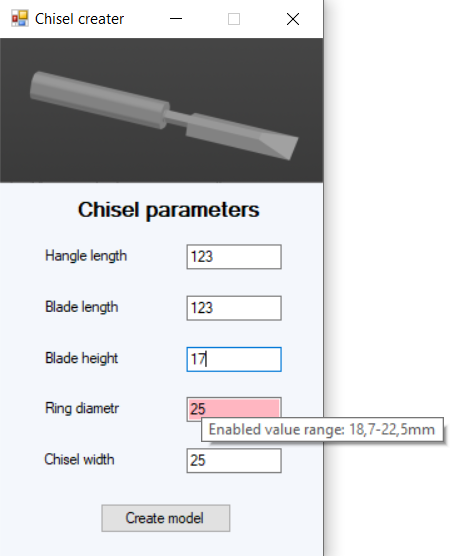


Рисунок 3.4 – некорректный ввод данных. Отображение на основном окне