Министерство науки и высшего образования Российской Федерации Федеральное государственное автономное образовательное учреждение высшего образования

ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ СИСТЕМ

УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ (ТУСУР)

Кафедра компьютерных систем в управлении и проектировании (КСУП)

**РАЗРАБОТКА ПЛАГИНА «Гриф» ДЛЯ «КОМПАС-3D»**

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

по дисциплине

«Основы разработки САПР» (ОРСАПР)

Выполнил:

студент гр. 582-1

\_\_\_\_\_\_\_\_\_ Рязанов А.Д.

«\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2025 г.

Проверил:

к.т.н., доцент каф. КСУП \_\_\_\_\_\_\_\_\_\_ Калентьев А.А.

«\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2025 г.

**1 Описание САПР**

* 1. Описание программы

КОМПАС-3D – это российская импортонезависимая система трёхмерного проектирования, ставшая стандартом для тысяч предприятий и сотен тысяч профессиональных пользователей.

КОМПАС-3D широко используется для проектирования изделий основного и вспомогательного производств в таких отраслях промышленности, как машиностроение (транспортное, сельскохозяйственное, энергетическое, нефтегазовое, химическое и т.д.), приборостроение, авиастроение, судостроение, станкостроение, вагоностроение, металлургия, промышленное и гражданское строительство, товары народного потребления и т. д.[1]

Данная САПР позволяет проектировать модели и сборки разного уровня сложности, благодаря разнообразному функционалу, включающего в себя работу как с 2-мерными эскизами, так и с 3D-моделями. В САПР есть возможность работать со всеми основными примитивами необходимыми для создания эскизов и моделей, а также существует достаточное количество инструментов для работы с 3D-моделями (вытягивание, вращение, вырезание и др.).

Компас 3D имеет множество прямых аналогов на рынке, среди них встречаются Autodesk Inventor, SOLIDWORKS и др.

В рамках дисциплины выбор данной САПР объясняется наличием описания API на русском языке, доступность учебной версии САПР без необходимости получать одобрения от компании, а также большим количеством информации на сторонних ресурсах на русском языке, позволяющим детальнее узнать о возможностях работы с САПР.

* 1. Описание API

API([аббр.](https://ru.wikipedia.org/wiki/%D0%90%D0%B1%D0%B1%D1%80%D0%B5%D0%B2%D0%B8%D0%B0%D1%82%D1%83%D1%80%D0%B0" \o "Аббревиатура) от [англ.](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) application programming interface,дословно интерфейс программирования приложения) —программный интерфейс, то есть описание способов взаимодействия одной компьютерной программы с другими.[2]

Для подключения и работы с API на C# потребуется выполнить ряд следующих действий:

1. Включить в свойствах проекта функцию Register for COM Interop;
2. Создать DLL-обёртку для TLB Компас API с помощью Tlblmp.exe;
3. Подключить созданный DLL к проекту;
4. Зарегистрировать библиотеку в системе КОМПАС (а именно реализовать статический метод типа .htmSample с рядом настроек)
5. Зарегистрировать библиотеку на компьютере пользователя, воспользовавшись утилитой RegAsm.exe

Таблица 1.1 − Используемые свойства класса (интерфейса) Application

|  |  |  |
| --- | --- | --- |
| Название | Тип данных | Описание |
| ActiveDocument | IKompasDocument | Свойство, содержащее текущий активный документ |
| Documents | IDocuments | Коллекция всех открытых документов в приложении |
| Math2D | IMath2D | Интерфейс 2D математики |

Таблица 1.2 − Методы класса (интерфейса) Application

|  |  |  |  |
| --- | --- | --- | --- |
| Название | Входные параметры | Тип возвращаемых данных | Описание |
| ExecuteCompasCommand | commandId: int - идентификатор команды КОМПАС  post: int - способ отправки | bool | Выполнение команды системы КОМПАС |
| MessageBoxEx | Text: string - текст сообщения в окне  caption: string - заголовок окна  flags: int[] - комбинация кнопок и иконок | long | Выдача всплывающего сообщения |

Таблица 1.3 − Поля класса (интерфейса) IDocuments

|  |  |  |
| --- | --- | --- |
| Название | Тип данных | Описание |
| Item | IKompasDocument | Документ, заданный по имени, ссылке или индексу |

Таблица 1.4 − Используемые методы класса (интерфейса) IDocuments

|  |  |  |  |
| --- | --- | --- | --- |
| Название | Входные параметры | Тип возвращаемых данных | Описание |
| Add | Type: int - тип создаваемого документа  Visible: bool - открывать документ видимо в UI (true) или в скрытом режиме (false) | IKompasDocument | Создаёт новый документ |
| Open | PathName: string - полный путь к существующему документу КОМПАС  Visible: bool - открыть окно документа на экране  ReadOnly: bool - открыть в режиме «только чтение»  LoadCombinationIndex: int - индекс набора/комбинации загрузки/конфигурации при открытии | IKompasDocument | Открывает документ (существующий) |

Таблица 1.5 − Используемые поля класса (интерфейса) IProcess2D

|  |  |  |
| --- | --- | --- |
| Название | Тип данных | Описание |
| Angle | double | Угол отклонения в градусах |
| X | double | Координата X |
| Y | double | Координата Y |

Таблица 1.6 − Используемые поля класса (интерфейса) IProcess3D

|  |  |  |
| --- | --- | --- |
| Название | Тип данных | Описание |
| MateConstraintsObjects | Variant | Выбранные объекты для сопряжения |
| Placement | IPlacement3D | Положение объекта |
| TakeProcessObject | IModelObject | Объект, создаваемый в подпроцессе |

Таблица 1.7 − Используемые методы класса (интерфейса) IProcess3D

|  |  |  |  |
| --- | --- | --- | --- |
| Название | Входные параметры | Тип возвращаемых данных | Описание |
| RunTakeCreateObjectProcess | ProcessType: int - тип подпроцесса  TakeObject: object - существующий объект, который нужно взять как вход  NeedCreateTakeObj: bool - если TakeObject не найден, разрешить создать новый объект автоматически  LostTakeObj: bool - пометка, что ранее связанный объект утерян/невалиден | bool | Запустить подчинённый режим создания объектов |

1.3 Обзор аналогов плагина

Первым аналогом является приложение «Build a Glass» [3] для КОМПАС-3D, которое предназначено для создания различных типов стаканов. Оно демонстрирует, как можно расширить функциональность КОМПАС-3D для параметрического создания 3D-моделей посуды через пользовательский интерфейс плагина. Плагин служит примером организации процесса моделирования на основе параметров, передаваемых из пользовательского интерфейса в ядро системы. Данный аналог является релевантным для разрабатываемого плагина «Гриф», поскольку также реализует концепцию параметрического проектирования сложных поверхностей. Интерфейс взаимодействия представлен на рисунке 1.1.

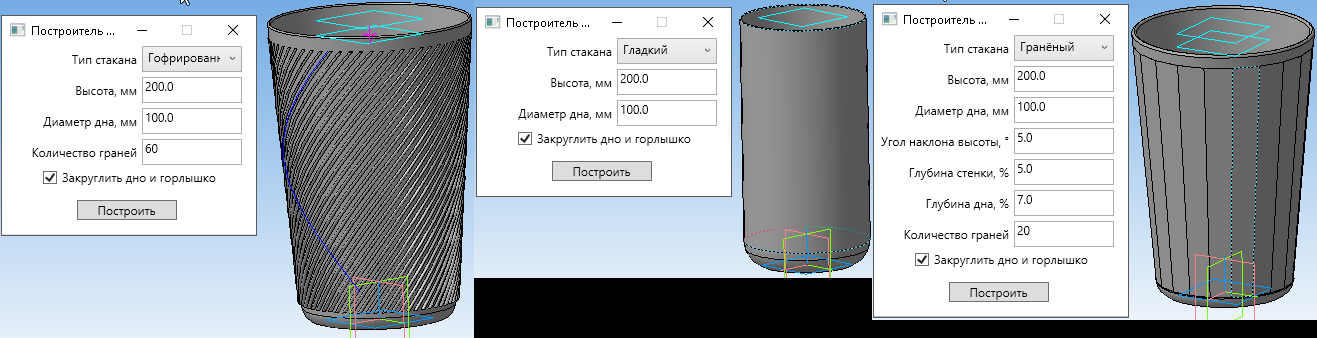


Рисунок 1.1 − Интерфейс приложения «Build a Glass»

Вторым аналогом является плагин для генерации модели клинка в КОМПАС-3D. Проект написан на C# и ориентирован на моделирование лезвий.[4] Плагин "Клинок" для КОМПАС-3D предназначен для автоматизированного создания модели клинка. Он разработан на языке C# и интегрируется в среду КОМПАС-3D, предоставляя пользователю инструменты для параметрического проектирования лезвий. Проект находится в открытом доступе на GitHub, что позволяет изучить его и проанализировать любому желающему.



Рисунок 1.2 − Интерфейс приложения «Плагин для Компас 3D “Клинок”»

**2 Описание предмета проектирования**

Гриф – это спортивный снаряд в виде металлического стержня, на который надеваются блины. Он используется для силовых упражнений.

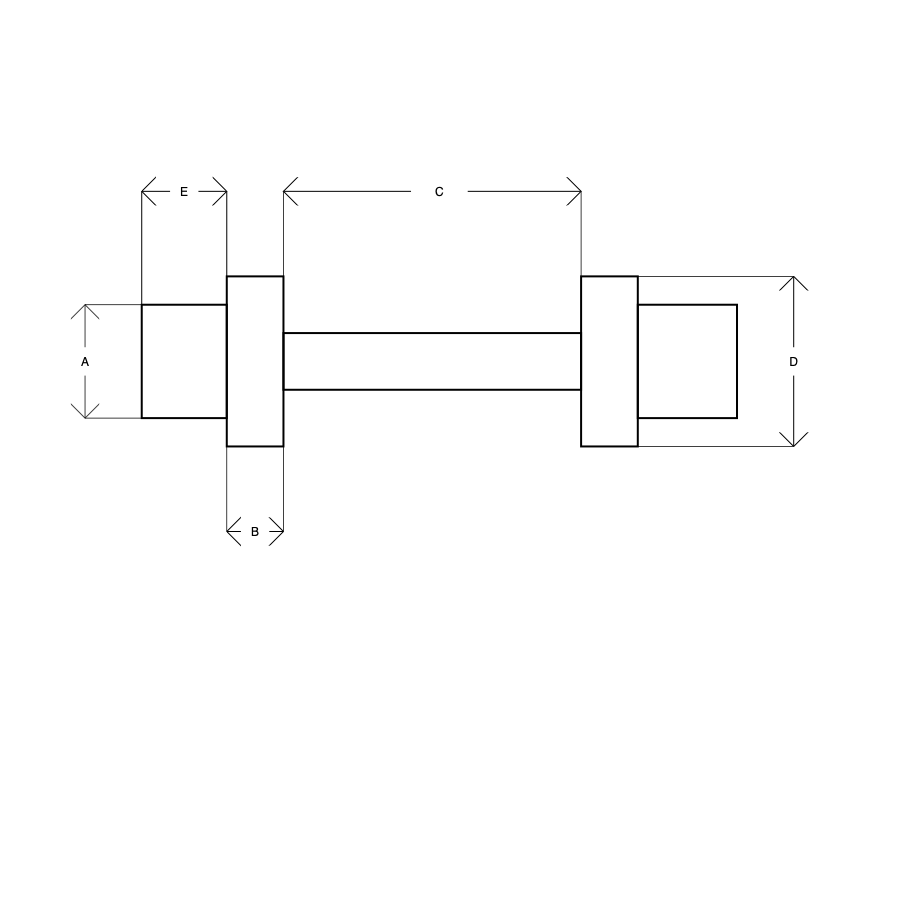


Рисунок 2.1 − Модель грифа.

***Изменяемые параметры для предмета проектирования*** (также все обозначения показаны на рисунке 2.1):

− Диаметр посадочной части А (25мм – 40мм);

− Длинна разделителя В (40мм – 60мм);

− Длинна ручки С (1200мм – 1310мм);

− Диаметр разделителя D (35мм – 50мм, должен быть больше A на 10мм);

− Длинна посадочной части E (320мм – 420мм).

**3 Проект системы**

3.1 UML диаграмма классов

UML — это стандартный язык визуального моделирования, предназначенный для следующего использования:

− моделирование бизнеса и подобных процессов;

− анализ, проектирование и внедрения программных систем.

UML — это общий язык для бизнес-аналитиков, архитекторов и разработчиков программного обеспечения, используемый для описания, спецификации, проектирования и документирования существующих или новых бизнес-процессов, структуры и поведения артефактов программных систем.[5].

UML диаграмма классов для плагина «Гриф» представлена на рисунке 3.1.

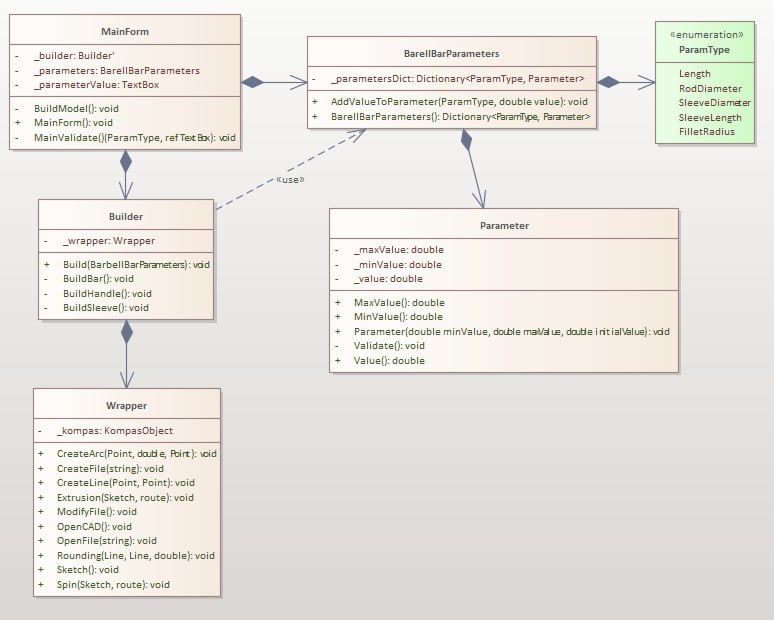


Рисунок 3.1 – UML диаграмма классов для плагина «Гриф»

В таблицах ниже представлена информация о свойствах и методах каждого из классов.

Таблица 3.1 – Поля класса MainForm

|  |  |  |
| --- | --- | --- |
| Название | Тип данных | Описание |
| \_builder | Builder | Хранит в себе объект построения |
| \_parameters | BarbellBarParameters | Хранит в себе параметры для объекта построения |

Таблица 3.2 – Методы класса MainForm

|  |  |  |
| --- | --- | --- |
| Название | Входные параметры | Описание |
| BuildModel | − | Построение модели по заданным параметрам |
| DiameterSleeveTextBox | object sender, EventArgs e | Ввод диаметра посадочной части (A) |
| LengthSleeveTextBox | object sender, EventArgs e | Ввод длины посадочной части € |
| DiameterSeparatorTextBox | object sender, EventArgs e | Ввод диаметра разделителя (D) |
| LengthSeparatorTextBox | object sender, EventArgs e | Ввод длины разделителя (B) |
| LengthHandleTextBox | object sender, EventArgs e | Ввод длины рукояти © |
|  | | |
| Продолжение таблицы 3.2 − Методы класса MainForm | | |
| ButtonBuild | object sender, EventArgs e | Запуск  построения модели по заданным параметрам |
| FormHandleComboBox | object sender, EventArgs e | Выбор формы ручки |
| MainValidate | ParamType, ref textBox | Второй уровень валидации |

Таблица 3.3 − Поля класса BarbellBarParameters

|  |  |  |
| --- | --- | --- |
| Название | Тип данных | Описание |
| \_parametersDict | Dictionary<ParamType, Parameter> | Хранит в себе словарь с параметрами объекта построения |

Таблица 3.4 − Методы класса BarbellBarParameters

|  |  |  |  |
| --- | --- | --- | --- |
| Название | Входные параметры | Тип данных | Описание |
| BarbellBarParameters | - | Dictionary<ParamType, Parameter> | Конструктор класса |
| AddValueToParameter | ParamType type, double value | void | Добавление параметров |

Таблица 3.5 − Поля класса Builder

|  |  |  |
| --- | --- | --- |
| Название | Тип данных | Описание |
| \_wrapper | Wrapper | Хранит в себе объект обёртки API |

Таблица 3.6 − Методы класса Builder

|  |  |  |
| --- | --- | --- |
| Название | Входные параметры | Описание |
| Build | BarbellBarParameters | Построение модели по заданным параметрам |
| BuildSleeve | − | Построение посадочной части грифа |
| BuildSeparator | − | Построение разделителей |
| BuildHandle | − | Построение рукояти грифа |
| BuildBar | − | Сборка всех частей грифа в единую модель |

Таблица 3.7 − Поля класса Parameter

|  |  |  |
| --- | --- | --- |
| Название | Тип данных | Описание |
| \_maxValue | double | Максимально допустимое значение параметра |
| \_minValue | double | Мин. допустимое значение параметра |
| \_value | double | Значение параметра |

Таблица 3.7 − Методы класса Parameter

|  |  |  |
| --- | --- | --- |
| Название | Входные параметры | Описание |
| MaxValue | double | Геттер, Сеттер |
| MinValue | double | Геттер, Сеттер |
| Value | double | Геттер, Сеттер |
| Parameter | double minValue, double maxValue, double initialValue | Конструктор |
| Validate | − | Валидация |

Таблица 3.8 − Методы класса Wrapper

|  |  |  |
| --- | --- | --- |
| Название | Входные параметры | Описание |
| CreateArc | Point, double, Point | Создание дуги по двум точкам |
| Rounding | Line, Line, double | Скругление |
| CreateLine | Point, Point | Создание линии |
| Sketch | − | Создание эскиза |
| Spin | Sketch, route | Вращение эскиза |
| Extrusion | Sketch, route | Выдавливание эскиза |
| CreateFie | string | Создание файла |
| OpenFile | string | Открытие файла |
| ModifyFile | − | Редактирование файла |
| OpenCAD | − | Открытие Компас3D |

Таблица 3.9 − Поля класса Wrapper

|  |  |  |
| --- | --- | --- |
| Название | Тип данных | Описание |
| -\_kompas | KompasObject | Хранит в себе ссылку на компас |

На рисунке 3.2 представлен макет пользовательского интерфейса.

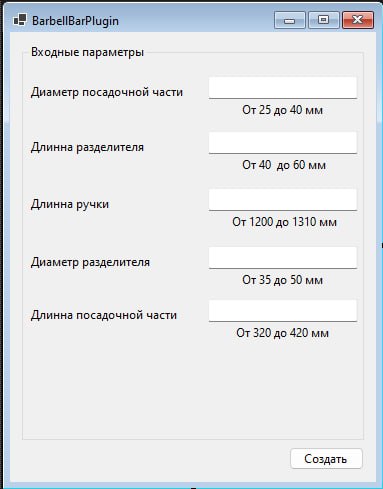


Рисунок 3.2 − Макет пользовательского интерфейса

Плагин состоит из четырех основных функциональных блоков. **MainForm** отвечает за взаимодействие с пользователем, ввод и валидацию параметров через графические элементы. **BarbellBarParameters** централизованно хранит и проверяет корректность всех геометрических характеристик грифа. **Builder** координирует процесс создания 3D-геометрии, последовательно формируя отдельные компоненты грифа и объединяя их в конечную модель. **Wrapper** обеспечивает низкоуровневое взаимодействие с системой проектирования, инкапсулируя вызовы геометрических операций создания эскизов, тел вращения и выдавливания.

При задании параметра значение введенное в TextBox сначала проверяется на верный тип данных, а именно double, при помощи метода MainValidate() в MainForm. Далее MainValidate вызывает метод AddValueToParameter() в переменной Parameters \_parameter для добавления значения в словарь \_parametersDict. Метод AddValueToParameter() вызывает Set для свойства Value(), задает новое значение. Метод Validate() класса Parameter вызывается в сеттере свойства Value().

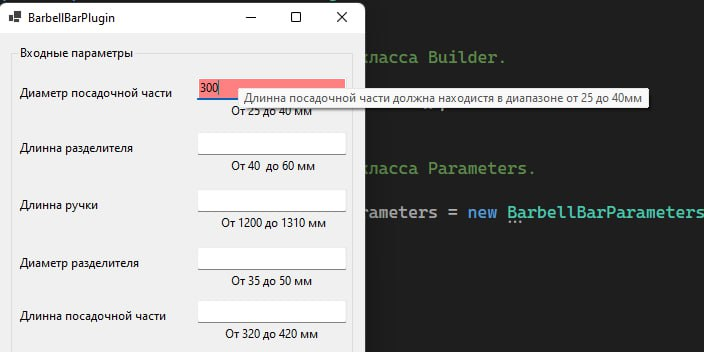


Рисунок 3.3 – Реакция системы на ошибки в введенных параметрах

**4 Список источников**

1. КОМПАС-3D [Электронный ресурс]. − Режим доступа <https://kompas.ru/kompas-3d/about/> (дата обращения 28.09.2025)
2. API [Электронный ресурс]. − Режим доступа <https://habr.com/ru/articles/464261/> (дата обращения 28.09.2025)
3. Plugin glass [Электронный ресурс]. − Режим доступа <https://github.com/GregoryGhost/plugin-glass-for-compass3d> (дата обращения 29.09.2025)
4. Плагин для компас 3D “Клинок”. [Электронный ресурс]. − Режим доступа https://github.com/eskorpado/kompas-blade-plugin (дата обращения 05.10.2025)
5. UML [Электронный ресурс]. − Режим доступа <https://www.uml> diagrams.org/ (дата обращения 07.10.2025)