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 «Гранёный стакан» ДЛЯ «КОМПАС-3D»**

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

по дисциплине

«Основы разработки САПР» (ОРСАПР)

Выполнил:

студент гр. 582-1

\_\_\_\_\_\_\_\_\_ Корнаков А.Е.

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

Проверил:

доцент кафедры КСУП, к.т.н

\_\_\_\_\_\_\_\_\_\_ Калентьев А. А.

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

# **1 ОПИСАНИЕ САПР**

**1.1 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.

**1.2 Описание API**

API(аббр. От англ. 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.

Используемые свойства класса (интерфейса) Application представлены в таблице 1.1:

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 |

Используемые методы класса (интерфейса) Application представлены в таблице 1.2:

Таблица 1.2 − Используемые методы класса (интерфейса) Application

|  |  |  |  |
| --- | --- | --- | --- |
| Название | Входные параметры | Тип возвращаемых данных | Описание |
| ExecuteCompasCommand | commandId: long (константа из перечисления ProcessTypeEnum или ksKompasCommandEnum),  post: bool | bool | Выполнение команды системы КОМПАС. |

Окончание таблицы 1.2

|  |  |  |  |
| --- | --- | --- | --- |
| MessageBoxEx | Text: BSTR (текст сообщения), Caption: BSTR (заголовок сообщения), Flags: long (флаги) | long | Выдача всплывающего сообщения |

Используемые поля класса (интерфейса) IDocuments представлены в таблице 1.3:

Таблица 1.3 − Используемые поля класса (интерфейса) IDocuments

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

Используемые методы класса (интерфейса) Idocuments представлены в таблице 1.4:

Таблица 1.4 − Используемые методы класса (интерфейса) Idocuments

|  |  |  |  |
| --- | --- | --- | --- |
| Название | Входные параметры | Тип возвращаемых данных | Описание |
| Add | Type: long (тип документа из перечисления DocumentTypeEnum), Visible: bool (видимость документа:  1. TRUE создавать документ в видимом режиме;  2. FALSE в слепом). | IKompasDocument | Создаёт новый документ и добавляет его в коллекцию |
| Open | PathName: LPCTSTR (полное имя файла документа, состоящее из пути, имени и расширения файла), Visible: bool (видимость документа:  1. TRUE создавать документ в видимом режиме;  2.FALSE в невидимом.), ReadOnly: bool (TRUE: открыть документ только для чтения). | IKompasDocument | Открывает документ (существующий) |

Используемые поля класса (интерфейса) IProcess2D представлены в таблице 1.5:

Таблица 1.5 − Используемые поля класса (интерфейса) IProcess2D

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

Окончание таблицы 1.5

|  |  |  |
| --- | --- | --- |
| X | double | Координата X |
| Y | double | Координата Y |

Используемые поля класса (интерфейса) IProcess3D представлены в таблице 1.6:

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 |

Используемые методы класса (интерфейса) IProcess3D представлены в таблице 1.7:

Таблица 1.7 − Используемые методы класса (интерфейса) IProcess3D

|  |  |  |  |
| --- | --- | --- | --- |
| Название | Входные параметры | Тип возвращаемых данных | Описание |
| RunTakeCreateObjectProcess | ProcessType: ProcessTypeEnum (тип подпроцесса), TakeObject: LPDISPATCH (указатель на интерфейс редактируемого в подпроцессе объекта), NeedCreateTakeObj: bool (необходимость создания объекта в подпроцессе), LostTakeObj: bool (фиксировать параметры создаваемого объекта) | bool | Запустить подчинённый режим создания объектов |

**1.3 Обзор аналогов плагина**

Первым аналогом является приложения «Teapot Plugin» [3] для Компас-3D, который предназначен для создания различных форм чайников. Он демонстрирует, как можно расширить функциональность КОМПАС-3D для создания 3D-моделей через пользовательский интерфейс плагина. Плагин может служить примером того, как организовать создание объектов с помощью параметров, передаваемых из пользовательского интерфейса в процесс моделирования. Данный аналог является прямым для разрабатываемого плагина «Гранёный стакан». Интерфейс взаимодействия представлен на рисунке 1.1:



Рисунок 1.1 − Интерфейс приложения «Teapot Plugin»

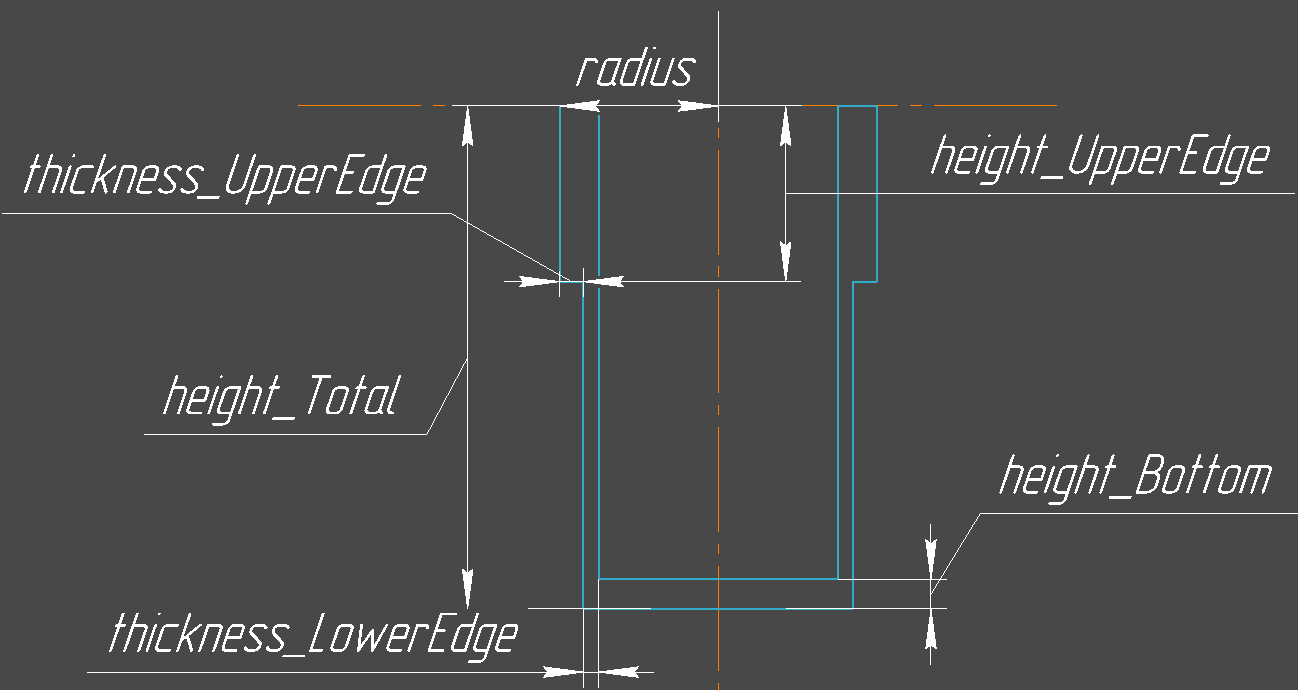
Вторым аналогом является плагин для генерации модели клинка в КОМПАС-3D. Проект написан на C# и ориентирован на моделирование лезвий.[4] Плагин "Клинок" для КОМПАС-3D предназначен для автоматизированного создания модели клинка. Он разработан на языке C# и интегрируется в среду КОМПАС-3D, предоставляя пользователю инструменты для параметрического проектирования лезвий. Интерфейс плагина продемонстрирован на рисунке 1.2:



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

**2 ОПИСАНИЕ ПРЕДМЕТА ПРОЕКТИРОВАНИЯ**

Гранёный стакан — тип стеклянного стакана, который отличается использованием прочного гранёного стекла. На рисунке 2.1 представлена модель гранёного стакана:

Рисунок 2.1 – Модель гранёного стакана с размерами

Размер грани состоит из двух значений: высота и ширина. Ширина каждой такой грани будет зафиксирована и будет составлять 20 мм. Высота такой грани(height\_OuterEdge) является исчисляемой и составляет результат вычитания переменных height\_Bottom и height\_UpperEdge из переменной height\_Total. Чертёж грани представлен на рисунке 2.2:

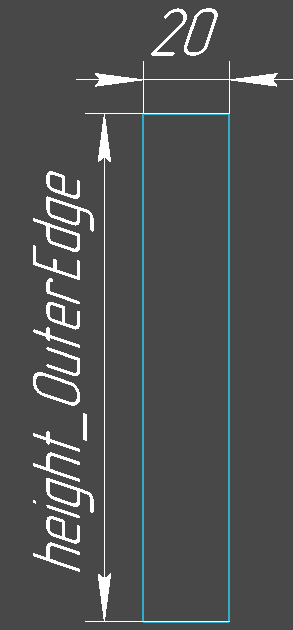


Рисунок 2.2 – Модель грани гранёного стакана с размерами

Изменяемые параметры для плагина (также все обозначения показаны на рис. 2.1):

1. Высота стакана «height\_Total» (100 — 150мм);

2. Радиус внешней окружности стакана «radius» (45 — 60мм);

3. Высота дна «height\_Bottom» (10 — 25мм);

4. Толщина нижней грани «thickness\_LowerEdge» (2 — 5мм);

5. Толщина верхней грани «thickness\_UpperEdge» (не может быть меньше толщины нижней грани(«thickness\_LowerEdge») и должна быть больше толщины нижней грани как минимум на 40%);

6. Высота верхней грани стакана «height\_UpperEdge» (должна составлять не менее 20% от общей высоты стакана(«height\_Total»));

7. Количество граней (сумма каждой ширины всех граней должна быть меньше длины нижней окружности).

**3 ПРОЕКТ СИСТЕМЫ**

**3.1 UML диаграмма классов**

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

1. Моделирование бизнеса и подобных процессов;

2. 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 «Гранёный стакан» представлена на рисунке 3.1:

В таблицах ниже представлена информация о свойствах и методах каждого из классов. В таблице 3.1 продемонстрированы поля класса MainForm:

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 |

В таблице 3.2 продемонстрированы методы класса MainForm:

Таблица 3.2 – Методы класса MainForm

|  |  |  |
| --- | --- | --- |
| Название | Входные параметры | Описание |
| BuildModel | − | Построение модели по заданным параметрам |
| height\_TotalTextBox | object sender, EventArgs e | Ввод высоты стакана |
| radiusTextBox | object sender, EventArgs e | Ввод радиуса внешней окружности стакана |

|  |  |  |
| --- | --- | --- |
| height\_BottomTextBox | object sender, EventArgs e | Ввод высоты дна стакана |
| thickness\_LowerEdgeTextBox | object sender, EventArgs e | Ввод толщины нижней грани стакана |
| thickness\_UpperEdgeTextBox | object sender, EventArgs e | Ввод толщины верхней грани стакана |
| height\_UpperEdgeTextBox | object sender, EventArgs e | Ввод высоты верхней грани стакана |
| height\_OuterEdgeTextBox | object sender, EventArgs e | Ввод высоты внешней грани стакана |
| ButtonBuild | object sender, EventArgs e | Запуск  построения модели по заданным параметрам |
| MainValidate | ParamType, ref textBox | Второй уровень валидации |

Таблица 3.3 − Поля класса faceted\_GlassParameters

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

Таблица 3.4 − Методы класса faceted\_GlassParameters

|  |  |  |  |
| --- | --- | --- | --- |
| Название | Входные параметры | Тип данных | Описание |
| faceted\_GlassParameters | - | 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 | faceted\_GlassParameters | Построение модели по заданным параметрам |
| BuildGlass | − | Построение основы стакана |
| BuildOuterEdge | − | Построение граней на стакане |

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:

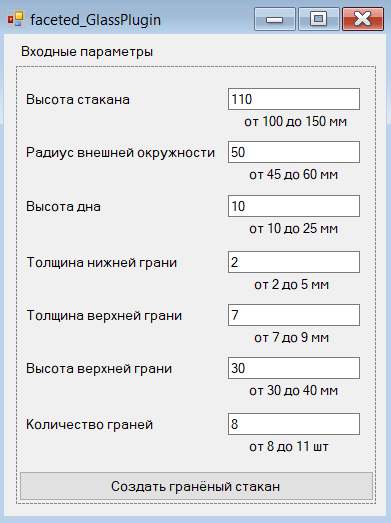


Рисунок 3.2 — Макет пользовательского интерфейса

Для валидации введённых значений добавлена статусная строка, которая до выявления ошибок остаётся невидимой для пользователя. На рисунке 3.3 представлена валидация введённых значений, а именно реакция программы на ввод ошибочного значения:

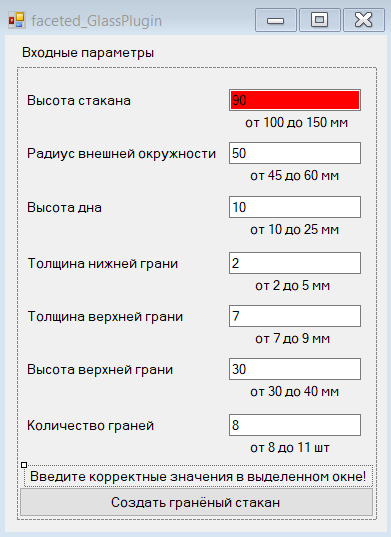


Рисунок 3.3 — Реакция программы на ошибку в введённом значении

**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. Teapot Plugin [Электронный ресурс]. − Режим доступа https://github.com/kurocha/teapot (дата обращения 29.09.2025)

4. Плагин для компас 3D “Клинок”. [Электронный ресурс]. − Режим доступа https://github.com/eskorpado/kompas-blade-plugin (дата обращения 10.10.2025)

5. UML [Электронный ресурс]. − Режим доступа https://www.uml-diagrams.org/ (дата обращения 13.10.2025)