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 «ПАЛЕЦ КРЕПЕЖНЫЙ»   
ДЛЯ «КОМПАС-3D v.20»**

Проект системы по лабораторному проекту по дисциплине   
«Основы разработки САПР»

Выполнил:  
Студент гр. 588-1

\_\_\_\_\_\_\_\_\_\_\_/ Колесников А.М.

(подпись)

«\_\_\_» \_\_\_\_\_\_\_\_20\_\_г.

Руководитель:  
к.т.н, доцент каф. КСУП

\_\_\_\_\_\_\_\_\_\_\_\_ / Калентьев А.А.

(подпись)

«\_\_\_» \_\_\_\_\_\_\_\_20\_\_г.

Томск 2021

Оглавление

[1 Описание САПР 3](#_Toc87425420)

[1.1 Описание программы 3](#_Toc87425421)

[1.2 Описание API 3](#_Toc87425422)

[1.3 Обзор аналогов 9](#_Toc87425423)

[2 Описание предмета проектирования 11](#_Toc87425424)

[3 Проект программы 13](#_Toc87425425)

[3.1 Диаграмма классов 13](#_Toc87425426)

[3.2 Макеты пользовательского интерфейса 14](#_Toc87425427)

[Список использованных источников 17](#_Toc87425428)

1 Описание САПР

1.1 Описание программы

Система автоматизированного проектирования (САПР) или CAD (англ. Computer-Aided Design) – программный пакет, предназначенный для создания чертежей, конструкторской и/или технологической документации и 3D моделей. [1]

Система «КОМПАС-3D» широко используется для проектирования изделий в машиностроении и строительстве – от изделий народного потребления до авиастроения, судостроения и продукции военного назначения. [2]

Программы данного семейства автоматически генерируют ассоциативные виды трехмерных моделей (в том числе разрезы, сечения, местные разрезы, виды по стрелке, виды с разрывом), все они ассоциированы с моделью: изменения в модели приводят к изменению изображения на чертеже.

Система «КОМПАС-3D» отличается проектированием изделий любой сложности, простотой освоения, бесплатной технической поддержкой, автоматизацией отраслевых задач и многим другим.

1.2 Описание API

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

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 |

Продолжение таблицы 1.1

|  |  |  |  |
| --- | --- | --- | --- |
| Название | Входные параметры | Тип возвращаемых данных | Описание |
| 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 –  видимый режим),  typeDoc – тип документа (true – деталь, false – сборка). | bool | Создать документ-модель (деталь или сборку) |
| 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 |

1.3 Обзор аналогов плагина

Аналогов у разрабатываемого плагина много. Одним из таких является библиотека стандартных изделий. Библиотека стандартных изделий – библиотека трехмерных моделей стандартных изделий для вставки в сборку.

Ключевые возможности:

1. Удобный поиск элементов по наименованиям и значениям атрибутов;
2. Наглядное представление элементов при помощи трехмерных моделей;
3. Быстрый доступ к содержанию, размещенному на тематических вкладках;
4. Формирование индивидуальных списков избранных элементов;
5. Создание типовых крепежных соединений с автоматическим подбором размеров элементов;
6. Экспорт графических представлений в файлы различных графических форматов.

Модульная структура приложения позволяет пользователю самостоятельно определить требуемый для своих нужд перечень стандартных изделий.

Приложение способно построить крепёжные изделия по стандартам ГОСТ: болты, винты, гайки, шайбы, шпильки, заклепки и прочие крепежные изделия. [3]

На рисунке 1.1 изображено окно библиотеки «Стандартные изделия».



Рисунок 1.1 – Окно библиотеки «Стандартные Изделия»

2 Описание предмета проектирования

Предметом проектирования является палец крепежный.

Палец крепежный – не резьбовой крепеж в виде короткого цилиндрического стержня. В сборке несущие другие детали, сборочные единицы устанавливаются на нём шарнирно или неподвижно или опираются на один конец или оба конца. Простейший и классический палец - это палец, соединяющий две проушины или проушину и петлю троса.

Для изготовления пальца используется углеродистая сталь марок 15, 20 или 45.

Параметры пальца возможно менять по следующим параметрам:

1. A — длина стержня: от 11.6 до 60 см;
2. B — длина шляпки: от 1 до 8 см;
3. C — радиус отверстия: от 0.4 до 4 см;
4. D — глубина фаски на стержне: от 0.2 до 5 см;
5. E — глубина фаски на шляпке: от 0.1 до 2.5 см;
6. F — расстояние от конца стержня до отверстия: от 1.6 до 10 см;
7. G — угол фаски на стержне: не превышает 30˚;
8. H — угол фаски на шляпке: не превышает 45˚;
9. I — радиус стержня: от 4 до 40 см;
10. J — радиус шляпки: от 5 до 50 см.

Плагин имеет следующие зависимости:

1. Параметр E (глубина фаски на шляпке) не должен превышать параметра B (длина шляпки);

2. Параметр D (глубина фаски на стержне) не должен превышать параметра C (радиус отверстия);

3. Параметр I (радиус стержня) не должен превышать параметра J (радиус шляпки);

4. Параметр F (расстояние от конца стержня до отверстия) должен входить в следующий интервал: ;

5. Параметр C (радиус отверстия) не должен превышать параметра I (радиус стержня);

6. Параметр B (длина шляпки) не должен превышать параметра A (длина стержня).

На рисунке 2.1 представлен чертеж модели «Палец крепежный».

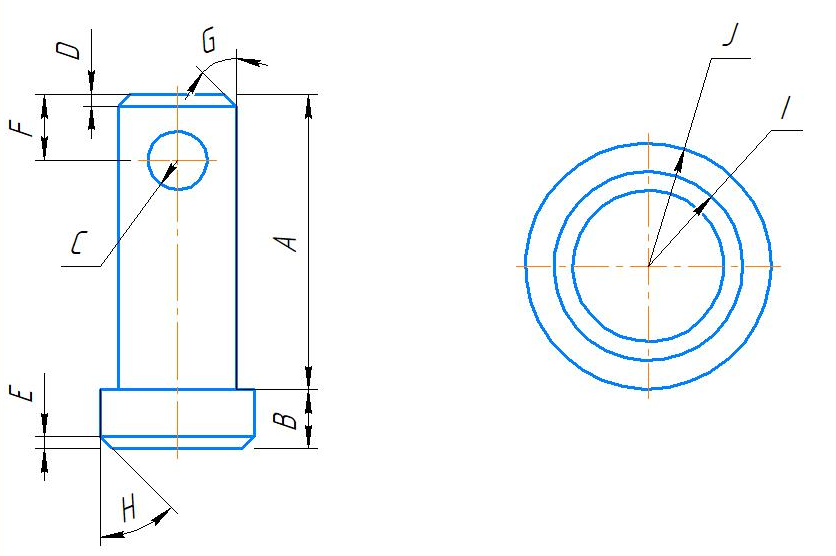


Рисунок 2.1 — Чертеж модели «Палец крепежный»

3 Проект программы

3.1 Диаграмма классов

Диаграмма классов определяет типы классов системы и различного рода статические связи, которые существуют между ними. [4]

Диаграмма классов плагина представлена на рисунке 3.1.

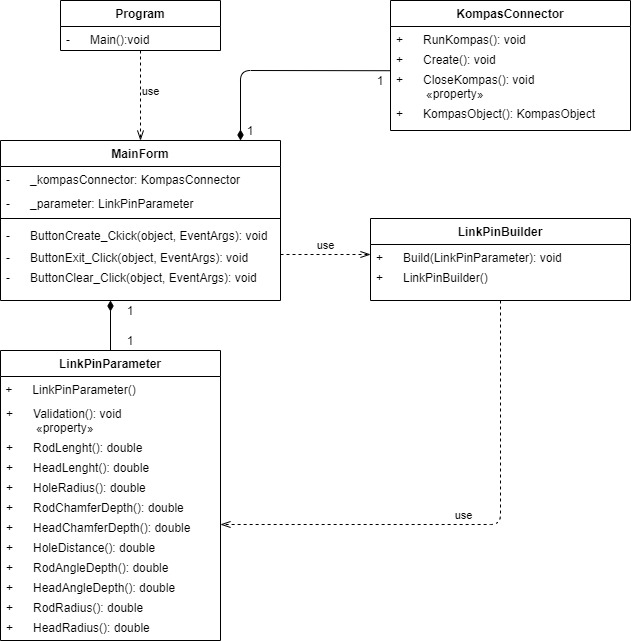


Рисунок 3.1 — Диаграмма классов плагина «Палец крепежный»

Для реализации выбран следующий набор классов и структур:

Класс «Program», использует «MainForm» для обработки действий в графическом интерфейсе. «LincPinBuilder» содержит в себе методы создания 3D модели в «Компас 3D», класс «LincPinParameter» введенные значения в графическом интерфейсе. «KompasConnector» запускает «КОМПАС-3D» и переносит объект в программу.

3.2 Макет пользовательского интерфейса

Пользовательский интерфейс представляет собой форму для ввода параметров. На форме расположены картинка, которая изображает обозначения параметров модели на чертеже, наименования параметров, поля для ввода параметров, и кнопка «Построить», при нажатии на которую строится 3D-модель «Палец крепежный».

На рисунке 3.2 представлен макет пользовательского интерфейса.

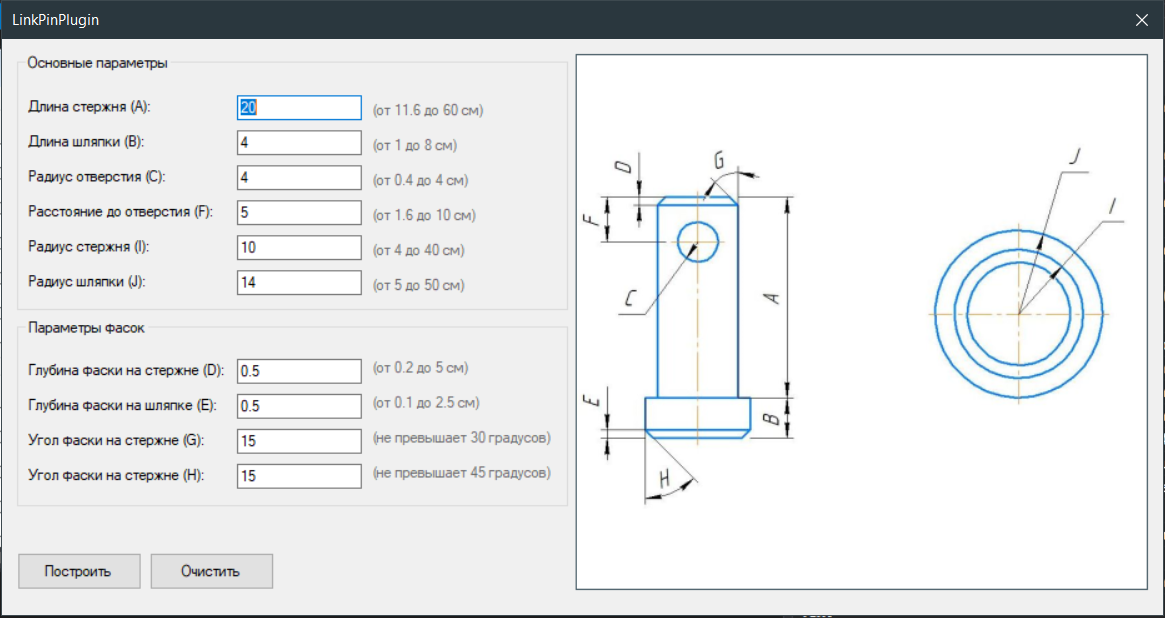


Рисунок 3.2 — Макет пользовательского интерфейса

Проверка правильности ввода значений проводится по ходу заполнения полей. Если поле заполнено неправильно, то оно подсвечивается красным цветом (рисунок 3.3).

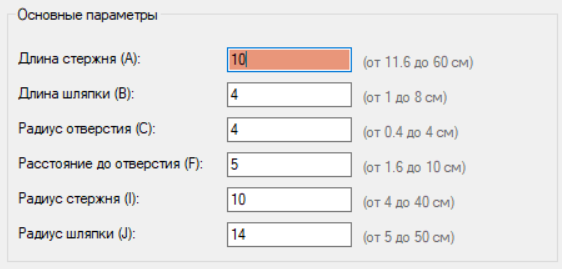


Рисунок 3.3 — Пример обработки ошибки при вводе параметров

Если пользователь при неправильных значениях нажмет кнопку «Построить», то появится сообщение об ошибке, сигнализирующее об ошибке (рисунок 3.4).

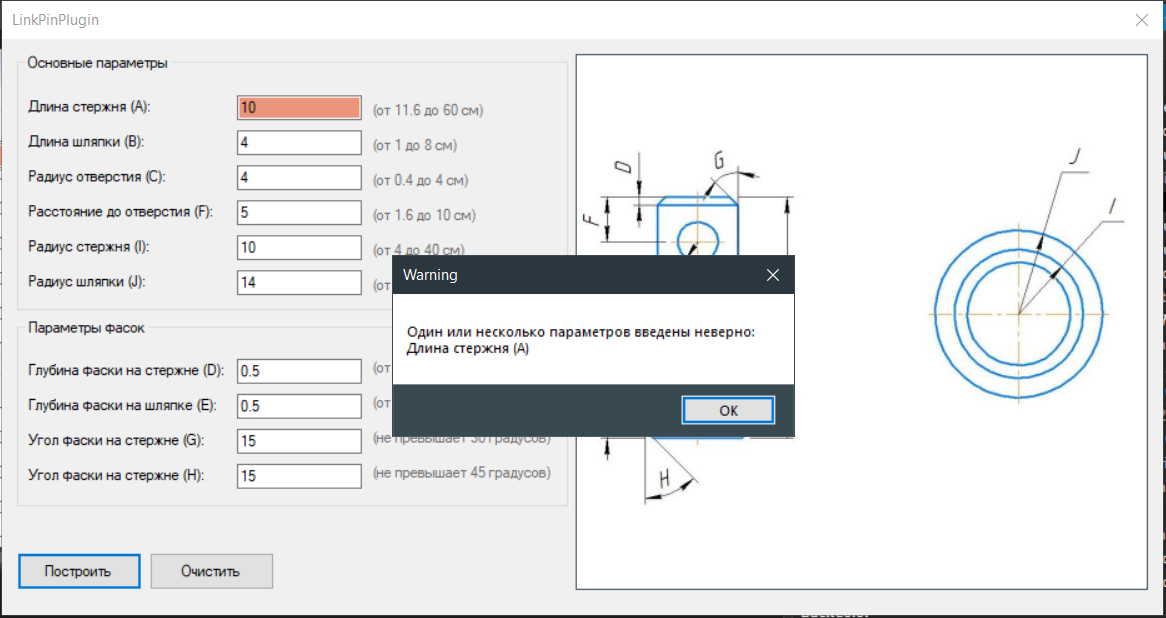


Рисунок 3.4 — Пример обработки ошибок при построении модели

Также на форме реализованы подсказки, в которых описаны зависимости между параметрами (рисунок 3.5).

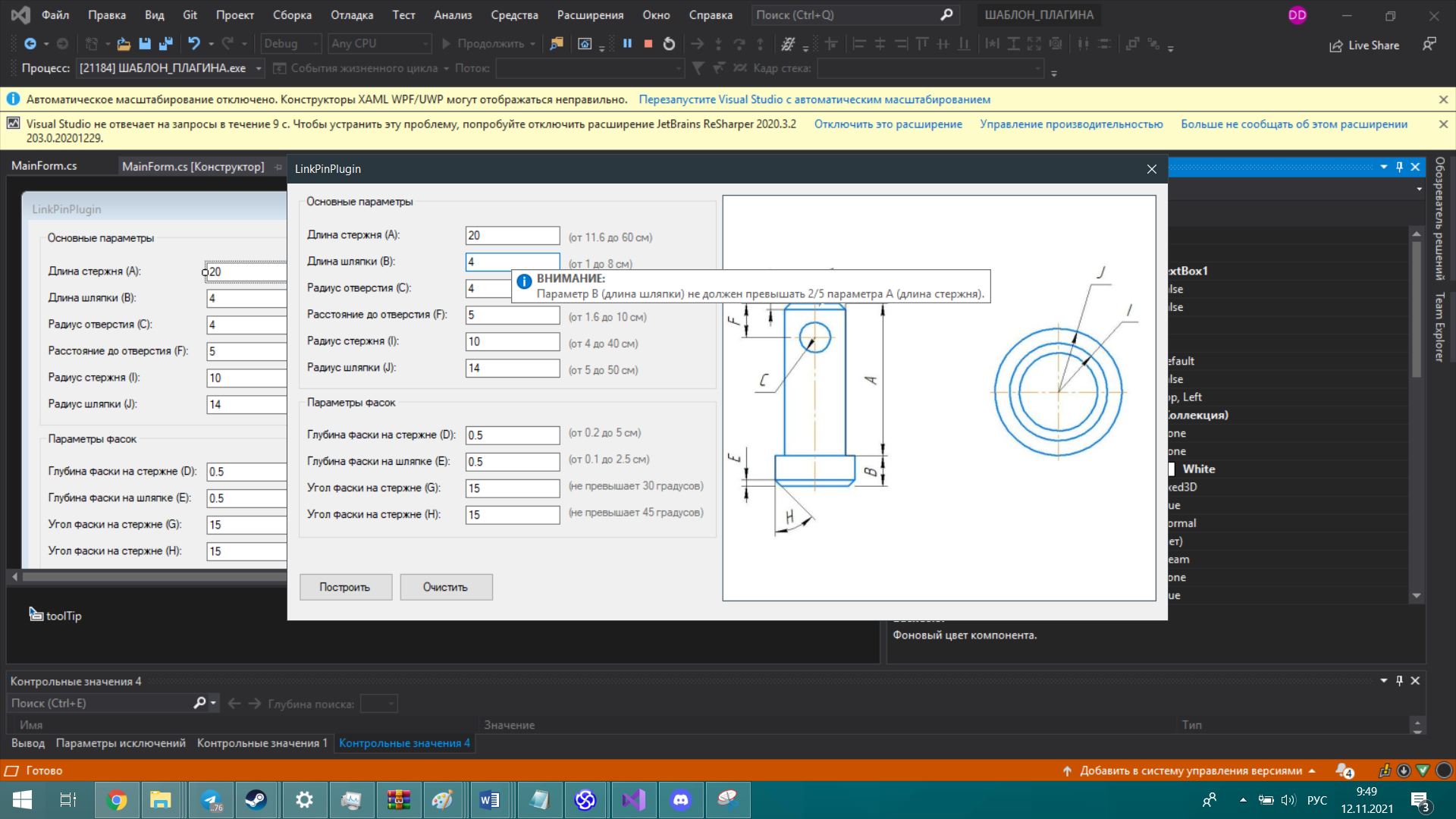


Рисунок 3.5 – Подсказки с описанием зависимостей между параметрами

Список использованных источников

1. Общие сведения о САПР [Электронный ресурс]. – Режим доступа: https://dic.academic.ru/dic.nsf/ruwiki/1132874, свободный (дата обращения: 20.10.2021).

2. Официальный сайт САПР КОМПАС-3D [Электронный ресурс]. – Режим доступа: <https://kompas.ru/kompas-3d/about/>, свободный (дата обращения: 20.10.2021).

3. Стандартные изделия для КОМПАС [Электронный ресурс]. – Режим доступа: https://ascon.ru/products/17/review/, свободный (дата обращения: 12.11.2021).

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