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)

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

По дисциплине

«Основы разработки САПР» (ОРСАПР)

Выполннил:

студент группы 588-1

\_\_\_\_\_\_\_\_\_\_\_Колесников А.М.

«\_\_\_»\_\_\_\_\_\_\_\_\_\_\_2022 г.

Руководитель:

к.т.н., доцент каф. КСУП

\_\_\_\_\_\_\_\_\_\_\_Калентьев А.А.

«\_\_\_»\_\_\_\_\_\_\_\_\_\_\_2022 г.

**Содержание**

[1 Введение 3](#_Toc92471620)

[2 Постановка задачи 4](#_Toc92471621)

[2.1 Описание предмета проектирования 4](#_Toc92471622)

[2.2 Выбор инструментов и средств реализации 6](#_Toc92471623)

[2.3 Назначение плагина 6](#_Toc92471624)

[3 Обзор аналогов 7](#_Toc92471625)

[4 Описание реализации 9](#_Toc92471626)

[4.1 Диаграмма классов 9](#_Toc92471627)

[5 Описание программы 13](#_Toc92471628)

[6 Тестирование программы 16](#_Toc92471629)

[6.1 Функциональное тестирование 16](#_Toc92471630)

[5.2 Модульное тестирование 18](#_Toc92471631)

[6.3 Нагрузочное тестирование 19](#_Toc92471632)

[7 Заключение 22](#_Toc92471633)

[Список литературы 23](#_Toc92471634)

1 Введение

Автоматизация моделирования играет огромную роль в упрощении работы на производстве и в науке. Она открывает новые возможности при достижении какой-либо цели (моделирование или проектирование объектов). Автоматизация позволяет повысить производительность и эффективность труда, при этом появляется возможность проектировать все более сложные объекты с отличной точностью. Только такой подход к моделированию способен значительно сократить затраты ресурсов на проект.

Целью данной работы является разработка плагина, который автоматизирует построение модели «Палец крепежный» для системы автоматизированного проектирования КОМПАС 3D с помощью интегрированной среды разработки Visual Studio 2019 Community.

2 Постановка задачи

Требуется разработать плагин, который с помощью использования API программы КОМПАС 3D, строит модель «Палец крепежный» на основе входных параметров. В плагине должны присутствовать следующие параметры (введенные по умолчанию или введенные пользователем):

- Длина стержня;

- Длина шляпки;

- Радиус стержня;

- Радиус шляпки;

- Расстояние до центра отверстия;

- Радиус шляпки;

- Углы фасок (на стержне и шляпке);

- Длина фасок (на стержне и шляпке).

Вырез для флажка строится автоматически на основе следующих параметров: радиус шляпки; расстояние до центра отверстия.

2.1 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 12 до 60 см;
2. B — длина шляпки: от 2.4 до 12 см;
3. C — радиус отверстия: от 1 до 3 см;
4. D — глубина фаски на стержне: от 0.2 до 2 см;
5. E — глубина фаски на шляпке: от 0.1 до 1.2 см;
6. F — расстояние от конца стержня до отверстия: от 4 до 9 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 20 см;
10. J — радиус шляпки: от 5 до 25 см.

Плагин имеет следующие зависимости:

1. Параметр A (длина стержня) не должен быть меньше пяти параметров B (длина шляпки);

2. Параметр F (расстояние от конца стержня до отверстия) не должен быть меньше суммы минимального значения параметра F и параметра C (радиус отверстия);

3. Параметр I (радиус стержня) не должен превышать 0.8 параметра J (радиус шляпки)

На рисунке 2.1 представлен чертеж модели «Палец крепежный».

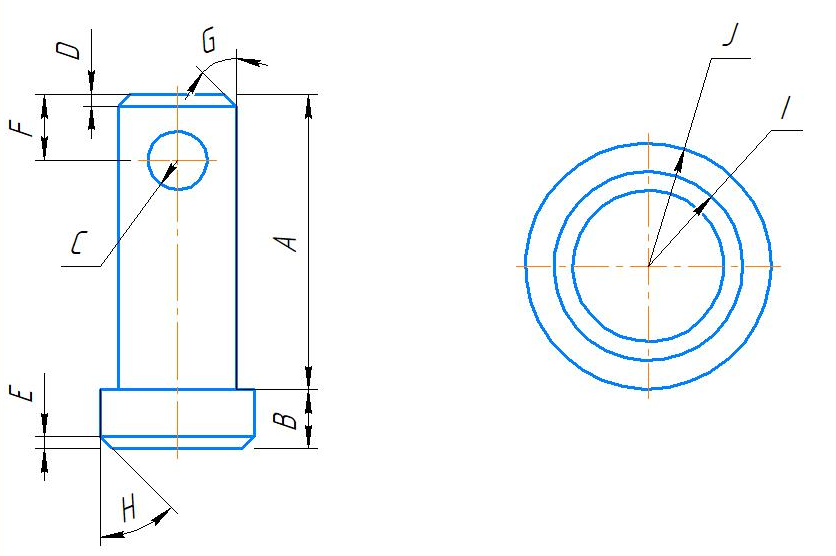


Рисунок 2.1 — Чертеж модели «Палец крепежный»

2.2 Выбор инструментов и средств реализации

Программа выполнена на языке программирования C# в среде Microsoft Visual Studio 2019 с использованием .NET Framework 4.7.2, библиотеки для Kompas 3D.

Инструментом тестирования и создания модульных тестов был выбран тестовый фреймворк NUnit версии 3.13.2.

2.3 Назначение плагина

Плагин предназначен для автоматического построения по введенным пользователем параметрам модели «Палец крепежный» в системе автоматизированного проектирования КОМПАС 3D.

3 Обзор аналогов

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.

На рисунке 1.1 изображено окно библиотеки «Стандартные изделия».



Рисунок 3.1 – Окно библиотеки «Стандартные Изделия»

4 Описание реализации

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.

При использовании UML были простроена диаграмма классов.

4.1 Диаграмма классов

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

Диаграмма классов плагина представлена на рисунке 4.1.

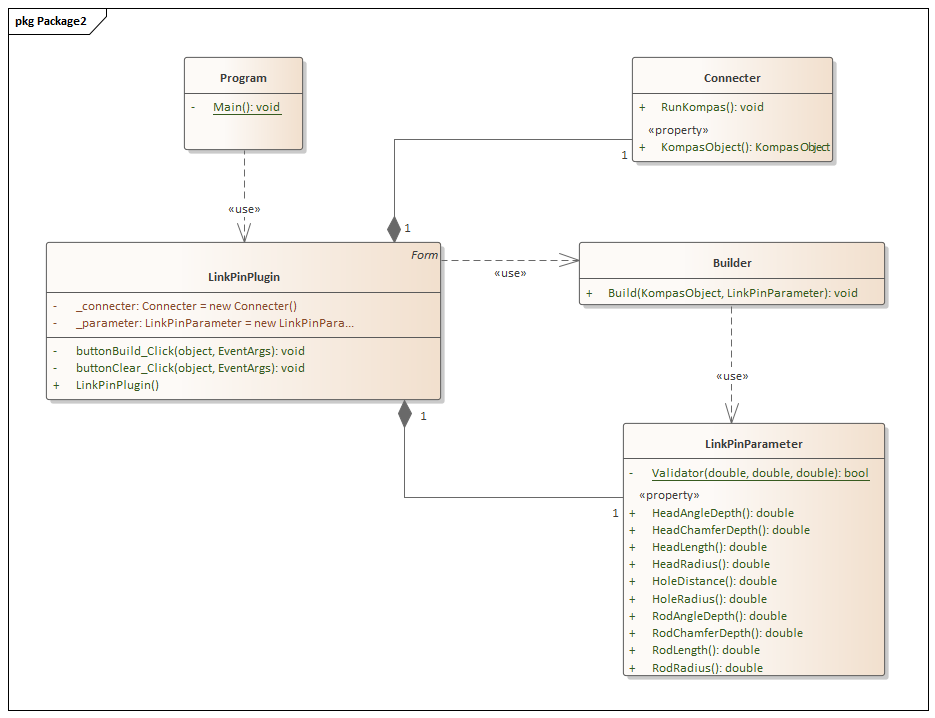


Рисунок 4.1 — Изначальная диаграмма классов

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.

В итоговой версии проекта добавлены новые методы, которые отражены в итоговой диаграмме классов (рисунок 4.2).

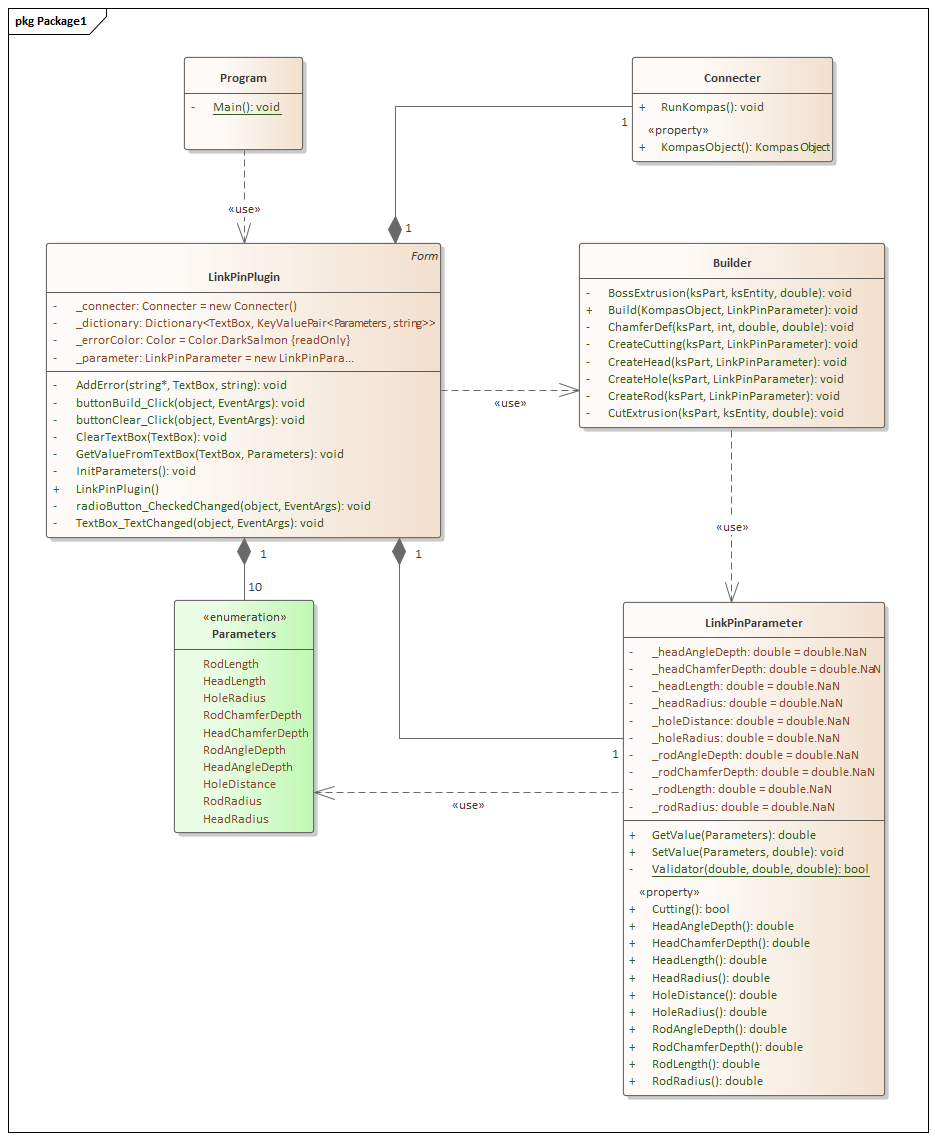


Рисунок 4.2 – Итоговая диаграмма классов

Добавлено перечисление Parameters для получения или установления значения определенного параметра в LinkPinParameter.

Для избавления я дублей в класс LinkPinParameter добавлено два метода: SetValue устанавливает значение параметра; GetValue возвращает значение параметра. Также добавлено свойство Cutting, которое извещает построитель об необходимости строить вырез.

Для автоматического построения детали добавлены приватные методы в класс Builder: BossExtrusion – выдавливание объекта; ChamferDef – построение фасок; CutExtrusion – вырез; CreateRod – создание стержня; CreateHole – создание отверстия; CreateHead – создание шляпки; CreateCutting – создание выреза.

В класс LinkPinPlugin добавлен словарь \_dictionary для сокращения дублей. В константу вынесен цвет ошибки. Добавлен обработчик событий TextBox\_TextChanged общий для каждого текстового поля. Добавлены следующие методы: AddError для обработки ошибки; ClearTextBox для отчистки текстовых полей; GetValueFromTextBox для занесения значения в параметр; InitParameters для инициализации всех параметров.

5 Описание программы

Макет пользовательского интерфейса представляет из себя окно для ввода параметров модели «Палец крепежный». Дефолтные значения уже введены в текстовые поля, чтобы их очистить, нужно нажать кнопку «Очистить». Также можно добавить вырез для флажка, предварительно нажав переключатель «Выполнить вырез». Для построения модели нужно нажать кнопку «Построить».

На окне расположено 10 полей для ввода параметров, 2 кнопки и картинка для визуализации вводимых параметров (рисунок 5.1).

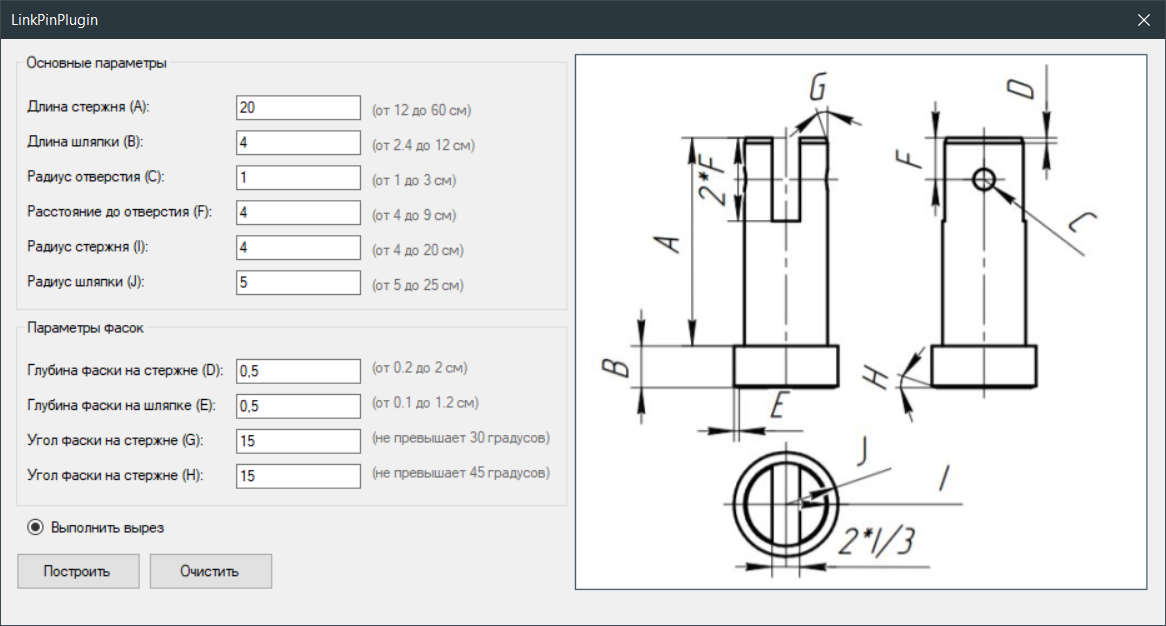


Рисунок 5.1 – Главное окно плагина

При попытки ввести некорректные значения поле, в котором происходил ввод, подсветится красным цветом (рисунок 5.2).

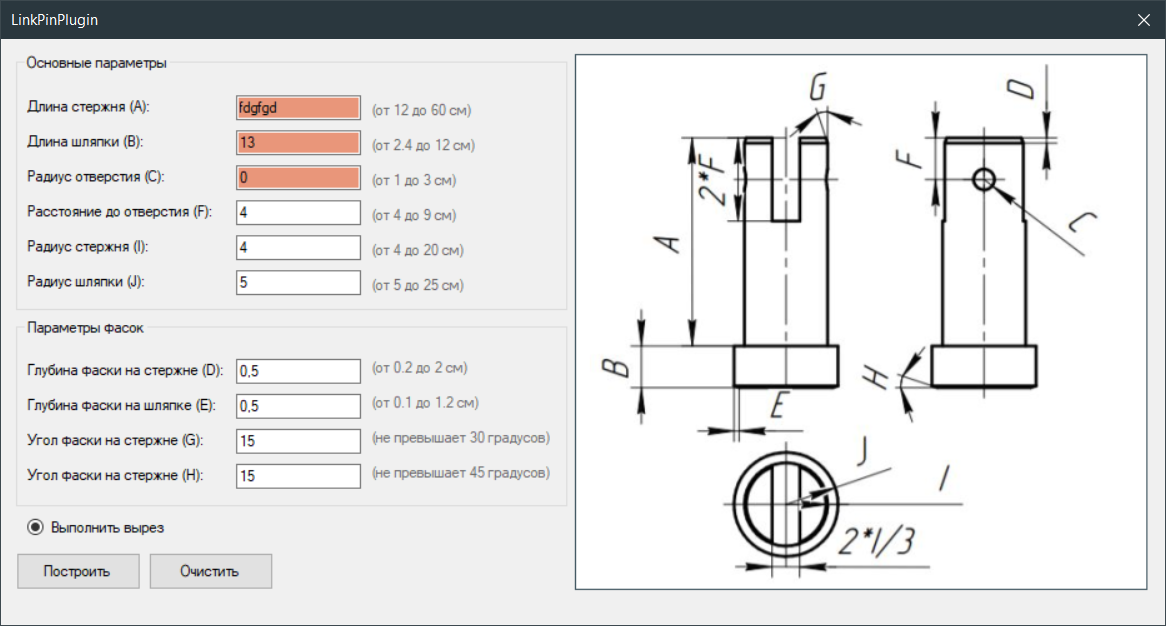


Рисунок 5.2 – Обработка некорректного ввода

При попытке построить модель с некорректным вводом программа выдаст сообщение со списком неправильно введенных полей (рисунок 5.3).

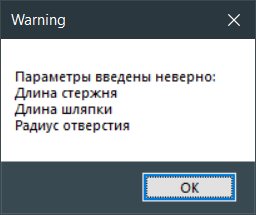


Рисунок 5.3 – Окно с ошибкой

После ввода корректных значений можно построить модель «Палец крепежный». Для этого требуется нажать кнопку «Построить», после чего откроется программа КОМПАС 3D, в которой создастся новый файл с построенной моделью (рисунок 5.4).

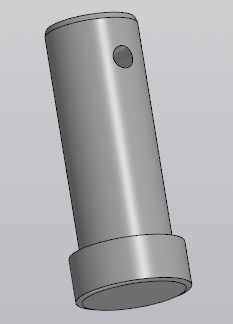


Рисунок 5.4 – Деталь с дефолтными параметрами

Та же деталь, но с вырезом под флажок представлена на рисунке 5.5.

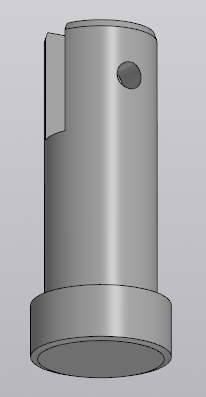


Рисунок 5.5 – Деталь с дефолтными параметрами и вырезом

6 Тестирование программы

6.1 Функциональное тестирование

При функциональном тестировании проверяется корректность работы плагина, то есть правильность построения модели по введенным параметрам.

Проводилось тестирование работы плагина при вводе максимальных и минимальных значений.

Минимальные значения:

1. Длина стержня = 12 см;
2. Длина шляпки = 2.4 см;
3. Расстояние от конца стержня до середины отверстия = 4 см;
4. Длина фаски на стержне = 0.2 см;
5. Длина фаски на шляпке = 0.1 см;
6. Угол фаски на стержне = 1;
7. Угол фаски на шляпке = 1;
8. Радиус отверстия = 1 см;
9. Радиус стержня = 4 см;
10. Радиус шляпки = 5 см.

Максимальные значения:

1. Длина стержня = 60 см;
2. Длина шляпки = 12 см;
3. Расстояние от конца стержня до середины отверстия = 9 см;
4. Длина фаски на стержне = 2 см;
5. Длина фаски на шляпке = 1.2 см;
6. Угол фаски на стержне = 30;
7. Угол фаски на шляпке = 45;
8. Радиус отверстия = 3 см;
9. Радиус стержня = 20 см;
10. Радиус шляпки = 25 см.

Модель с минимальными параметрами представлена на рисунке 6.1.

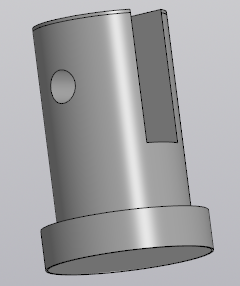


Рисунок 6.1 – Модель с минимальными параметрами

Модель с максимальными параметрами представлена на рисунке 6.2.

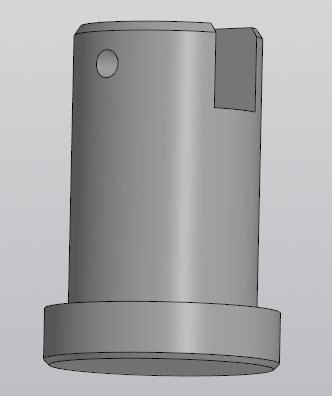


Рисунок 6.2 – Модель с максимальными параметрами

5.2 Модульное тестирование

Для проверки корректности работы модуля Core был создан класс UnitTest, который с помощью библиотеки NUnit получает возможность создавать тестовые сценарии, а расширение ReSharper позволит запустить все тестовые сценарии.

Была проверена работа публичных методов класса LinkPinRarameter, результаты работы тестов представлены на рисунке 6.3.

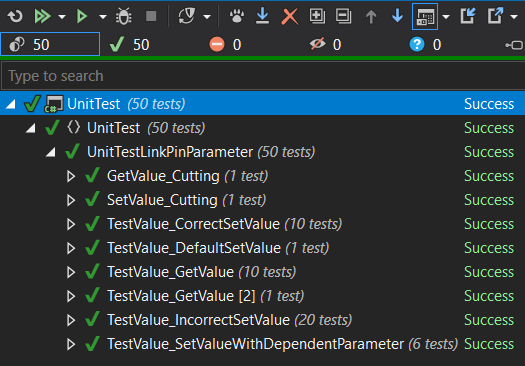


Рисунок 6.3 – Окно состояний запущенных тестов

Также через расширение ReSharper проверено покрытие кода тестами (рисунок 6.4). Оно показывает, что покрытие тестами составило 100% для модуля Core.

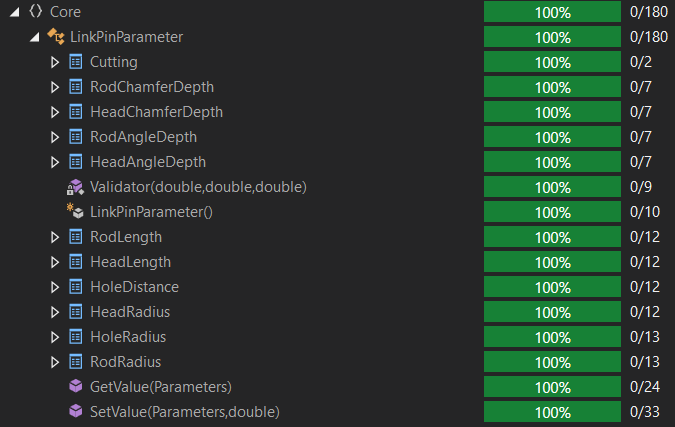


Рисунок 6.4 – Покрытие кода тестами

6.3 Нагрузочное тестирование

В целях проверки производительности работы плагина, было проведено нагрузочное тестирование. Для проведения нагрузочного тестирования был добавлен секундомер, который засекал время от начала построения, с каждым успешным построением модели производилась запись результатов в текстовый файл «log.txt».

Конфигурация ПК, на котором проводилось тестирование:

1. Центральный процессор: intel Core i5-4210H 2.90 GHz;
2. ОЗУ: 16 ГБ;
3. Графический процессор объемом памяти 2 ГБ.

Тестирование проводилось в течение 5 минут, в течение которого происходило зацикленное построение 67 моделей со стандартными значениями параметров.

На рисунке 6.5 представлен график, где ось «X» – время, ось «Y» – количество построенных моделей.

Рисунок 6.5 – График зависимости времени от количество построенных моделей

Такой скачек на графике обусловлен тем, что на компьютере установлено два разных модуля оперативной памяти, которые отличаются частотой и таймингом работы.

На рисунке 6.6 представлен график, где ось «X» – количество построенных моделей, ось «Y» – количество потребляемой оперативной памяти.

Рисунок 6.6 – График зависимости, потребляемой ОЗУ от количества построенных деталей

Исходя из графика, можно сделать вывод, что использование оперативной памяти, затрачиваемое программой, линейно увеличивается до окончания свободного места (график доходит до 12 Гб из-за того, что 3 Гб памяти зарезервированы системой).

7 Заключение

В ходе выполнения лабораторных работ были изучены предметная область проектирования, предмет проектирования, аналоги предмета проектирования, использование API программы КОМПАС 3D. На основании полученных данных были спроектированы архитектура и макет системы, создан плагин «Палец крепежный», проведены функциональные, модульные и нагрузочные тесты.

Список литературы