Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждения

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)

Студент гр. 586-2

\_\_\_\_\_\_\_\_\_\_\_\_\_В.В. Бельчиков

«\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2020г.

Руководитель

К.т.н., доцент каф. КСУП

\_\_\_\_\_\_\_\_\_\_А.А. Калентьев

«\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2020г.

Томск 2020

**РЕФЕРАТ**

Пояснительная записка 31 с., 19 рис., 16 источников.

SOLIDWORKS 2020/2014, ПЛАГИН, МОЛОТОК, АВТОМАТИЗИРОВАННОЕ ПРОЕКТИРОВАНИЕ, САПР.

Целью данной работы является разработка плагина, для построения модели молотка в системе автоматизированного проектирования SOLIDWORKS 2014/2020 [1] с помощью интегрированной среды разработки Visual Studio 2019 Сommunity [2].

В процессе работы изучены основные параметры молотка и основные функции API [3] для работы с системой автоматизированного проектирования SOLIDWORKS 2014/2020.

В результате работы был создан плагин, автоматизирующий построение модели молотка.

Отчет по пояснительной записке выполнен в текстовом редакторе Microsoft Word 2019 [4].

**Оглавление**

[1 Введение 4](#_Toc41431285)

[2 Постановка и анализ задачи 5](#_Toc41431286)

[2.1 Описание предмета проектирования 5](#_Toc41431287)

[2.2 Выбор инструментов и средств реализации 6](#_Toc41431288)

[2.3 Назначение плагина 7](#_Toc41431289)

[2.4 Описание аналогов разрабатываемого продукта 7](#_Toc41431290)

[2.4.1 Описание САПР КОМПАС-3D 7](#_Toc41431291)

[2.4.2 Плагин PDF для САПР КОМПАС-3D 8](#_Toc41431292)

[2.4.3 Экспорт из КОМПАС-3D в формат 3D PDF 8](#_Toc41431293)

[2.4.4 Описание аналога плагина для построения молотка 9](#_Toc41431294)

[3 Описание реализации 10](#_Toc41431295)

[3.1 Описание технических и функциональных аспектов проекта 10](#_Toc41431296)

[3.2 Диаграмма вариантов использования (Use Cases) 11](#_Toc41431297)

[3.3 Диаграмма классов 13](#_Toc41431298)

[4 Описание программы для пользователя 16](#_Toc41431299)

[5 Тестирование программы 18](#_Toc41431300)

[5.1 Функциональное тестирование 18](#_Toc41431301)

[5.2 Модульное тестирование 23](#_Toc41431302)

[5.3 Нагрузочное тестирование 24](#_Toc41431303)

[Заключение 27](#_Toc41431304)

[Список использованных источников 28](#_Toc41431305)

[Приложение А 30](#_Toc41431306)

# 1 Введение

Автоматизация моделирования имеет огромное значение для развития науки, техники и производства в современном обществе. В настоящее время автоматизация – основной способ повышения производительности и эффективности труда инженерно-технических работников, занимающихся моделированием сложных устройств. Использование автоматизации в проектировании позволяет создавать все более сложные технические объекты и гибко реагировать на появление новых решений и технологий в той или иной области техники. Она позволяет значительно повысить точность расчетов, выбрать наилучшие варианты для реализации на основе строгого математического анализа всех или большинства вариантов проекта с оценкой технических, технологических и экономических характеристик производства и эксплуатации проектируемого объекта, значительно повысить качество конструкторской документации, существенно сократить сроки проектирования и передачи конструкторской документации в производство, эффективнее использовать технологическое оборудование с программным управлением [5].

Таким образом, целью данной работы является разработка плагина, автоматизирующего построение модели молотка для системы автоматизированного проектирования SOLIDWORKS 2014/2020 с помощью интегрированной среды разработки Visual Studio 2019 Сommunity.

Интегрированная среда разработки Visual Studio — это стартовая площадка для написания, отладки и сборки кода, а также последующей публикации приложений. Интегрированная среда разработки (IDE) представляет собой многофункциональную программу, которую можно использовать для различных аспектов разработки программного обеспечения.

# 2 Постановка и анализ задачи

В рамках лабораторных работ в соответствии с технически заданием требовалось разработать плагин, который на основе входных параметров, интегрируя с системой SOLIDWORKS 2014/2020, строит модель молотка. Необходимо чтобы плагин позволял задавать такие параметры, как: длина бойка, высота бойка, ширина оголовья, длина носка, ширина наконечника, длина рукояти, диаметр рукояти.

# 2.1 Описание предмета проектирования

Молоток – небольшой ударный инструмент, применяемый для забивания гвоздей, разбивания предметов и других работ. В основном изготавливается из стали. Молоток — один из древнейших инструментов, используемых разумным человеком. [6]

Параметры моделируемого молотка:

* Head length – длина бойка, от 30мм до 80мм;
* Head width – ширина оголовья, от 20мм до 50мм;
* Head height – высота бойка, от 20мм до 50мм;
* Toe length – длина носка, от 30мм до 80мм;
* Tip width – ширина наконечника, от 1мм до 10мм;
* Handle diameter – диаметр рукояти, от 15мм до 40мм;
* Handle length – длина рукояти, от 130мм до 280мм.

Пример проектируемого изделия приведен ниже, на рисунке 2.1.

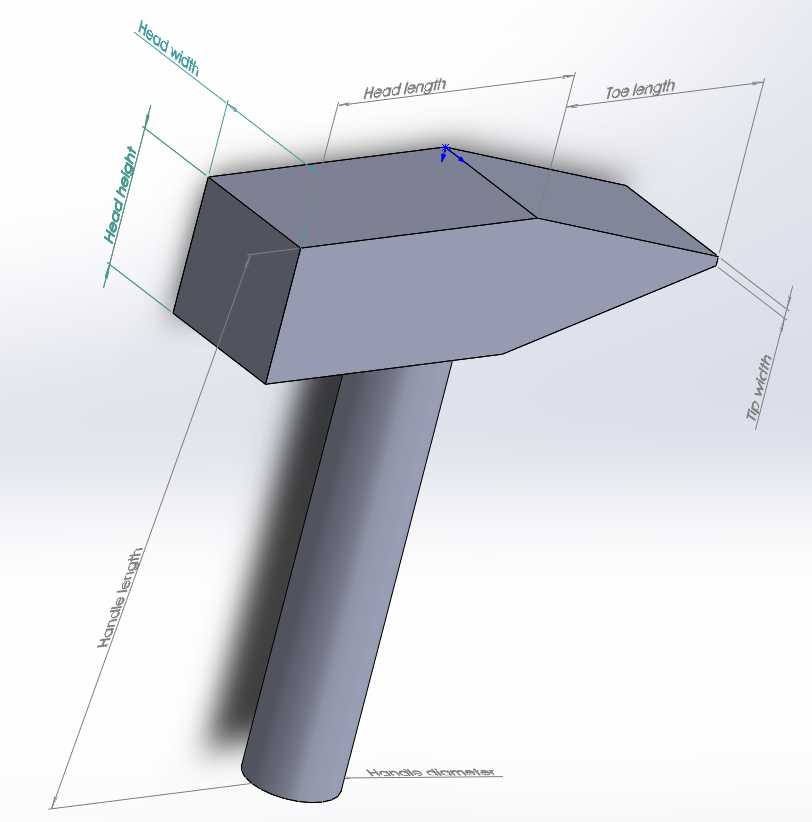


Рисунок 2.1 – Предмет изделия

# 2.2 Выбор инструментов и средств реализации

На основе требований к техническому заданию программа выполнена на языке программирования C# в среде Microsoft Visual Studio 2019 с использованием .NET Framework 4.7.2 [7], библиотеки «SolidWorks.Interop. sldworks» [8] версии 28 для основных операций в системе SOLIDWORKS 2020, а так же «SolidWorks.Interpop.sldworks» версии 22 для основных операция в системе SOLIDWORKS 2014.

Для реализации пользовательского интерфейса использовалась система для построения настольные и мобильные приложения Windows – Windows Forms (WinForms) [9].

# 2.3 Назначение плагина

Назначение разрабатываемого плагина обусловлено быстрым моделированием молотка с заданными параметрами. Благодаря данному расширению, проектировщики могут быстро задать параметры и получить спроектированную модель, при необходимости перестроить под необходимые им параметры.

# 2.4 Описание аналогов разрабатываемого продукта

# 2.4.1 Описание САПР КОМПАС-3D

КОМПАС-3D — система трехмерного проектирования, ставшая стандартом для тысяч предприятий. Ключевой особенностью продукта является использование собственного математического ядра С3D и параметрических технологий, разработанных специалистами АСКОН.

КОМПАС-3D обеспечивает поддержку наиболее распространенных форматов 3D-моделей (STEP, ACIS, IGES, DWG, DXF), что позволяет организовывать эффективный обмен данными со смежными организациями и заказчиками, использующими любые CAD / CAM / CAE-системы в работе.

Система «Компас-3D» предназначена для создания трёхмерных ассоциативных моделей отдельных деталей и сборочных единиц, содержащих как оригинальные, так и стандартизованные конструктивные элементы.

Система «Компас-3D» предоставляет следующие возможности:

* проектирование машиностроительных и приборостроительных изделий;
* классическое твердотельное моделирование;
* создание листовых деталей и обечаек;
* проектирование с применением сложных поверхностей;
* формирование электронной модели изделий;
* решение различных задач в архитектурно-строительном и технологическом проектировании;
* проектирование с помощью технологии MinD.

# 2.4.2 Плагин PDF для САПР КОМПАС-3D

Данный плагин позволяет производить экспорт моделей и сборок из КОМПАС-3D в формат PDF формат [10]. 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. 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:

* сохранение деталей и сборок в формате 3D PDF для интерактивного просмотра при помощи бесплатной программы Adobe Reader;
* создание анимации, 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.

# 2.4.3 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 [11].

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.

# 2.4.4 Описание аналога плагина для построения молотка

Один из аналогов плагина выполнен студентом группы 585-2. Плагин для построения молотка в САПР КОМПАС-3D. Принцип работы такой же, как и у разрабатываемого плагина. Плагин включает в себя изменение шести параметров молотка, обладает функцией отрытия/закрытия САПР КОМПАС-3D. На рисунках 2.3-2.4 представлены интерфейс плагина и построенная деталь.

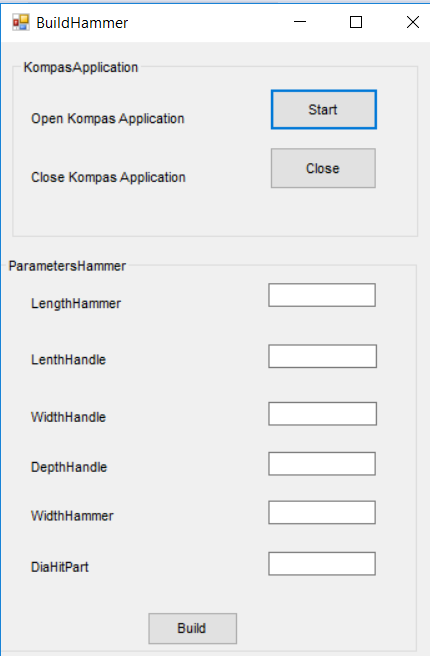
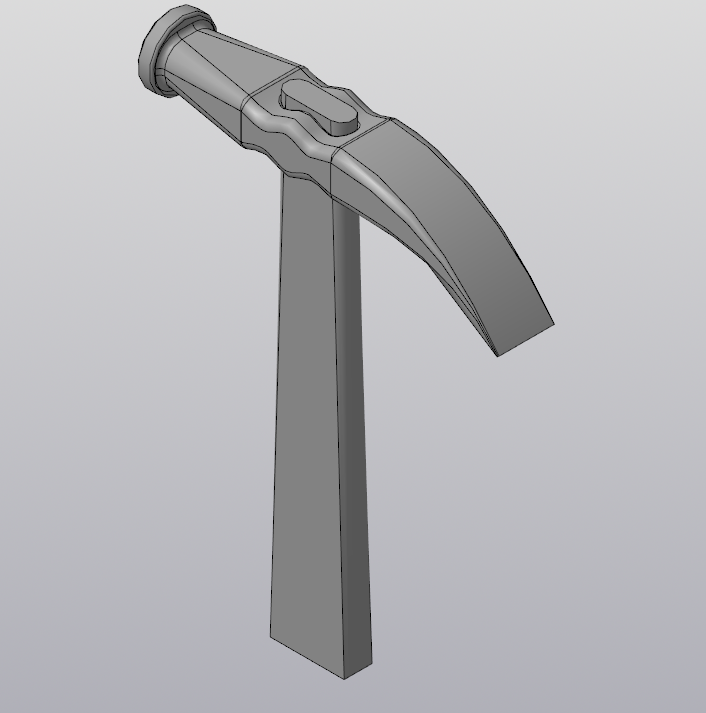


Рисунок 2.3 – Макет главного окна плагина

  
Рисунок 2.4 – Молоток, построенный с помощью плагина в САПР КОМПАС

# 3 Описание реализации

# 3.1 Описание технических и функциональных аспектов проекта

Для графического описания абстрактной модели проекта, а также пользовательского взаимодействия (сценарии действия) использован стандарт UML [12].

UML язык графического описания для объектного моделирования в обрасти 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.

При использовании UML были простроены: диаграмма вариантов использования и диаграмма классов.

# 3.2 Диаграмма вариантов использования (Use Cases)

Диаграмма вариантов использования (сценариев поведения, прецедентов) является исходным концептуальным представлением системы в процессе ее проектирования и разработки. Данная диаграмма состоит из актеров, вариантов использования и отношений между ними. При построении диаграммы могут использоваться также общие элементы нотации: примечания и механизмы расширения.

Суть данной диаграммы состоит в следующем: проектируемая система представляется в виде множества актеров, взаимодействующих с системой с помощью так называемых вариантов использования. При этом актером (действующим лицом, актантом, актором) называется любой объект, субъект или система, взаимодействующая с моделируемой системой извне. В свою очередь вариант использования – это спецификация сервисов (функций), которые система предоставляет актеру [13].

Ниже, на рисунке 3.1 представлена диаграмма вариантов использования, что демонстрирует, как пользователь может взаимодействовать с плагином.

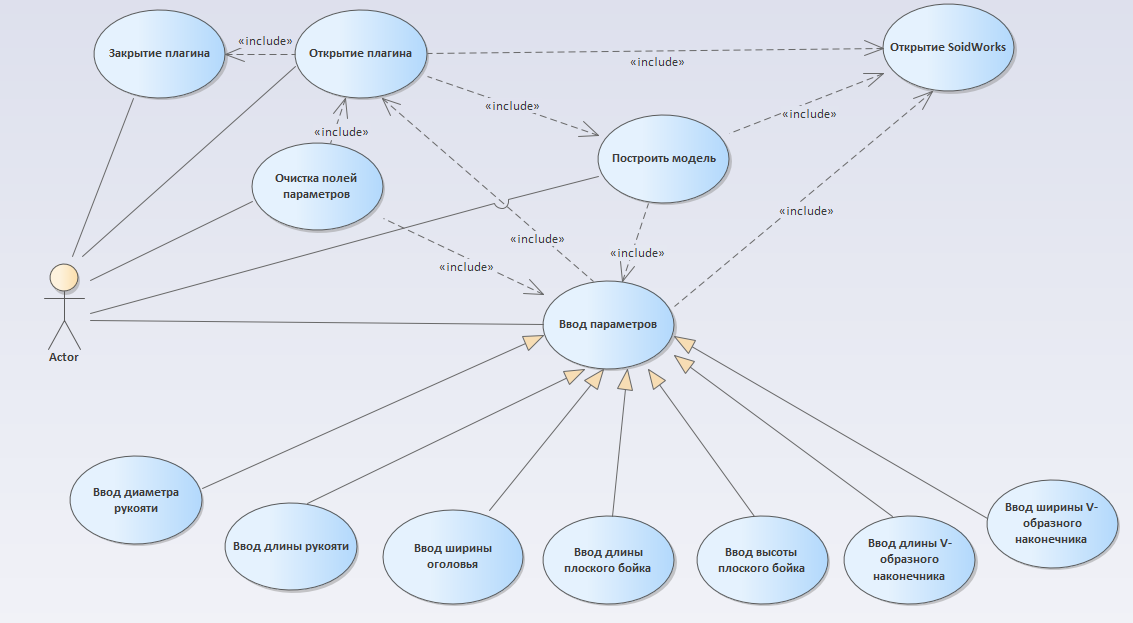


Рисунок 3.1– Начальная диаграмма вариантов использования

На рисунке 3.2 представлена финальная версия диаграммы вариантов использования, в которой появилась возможность выбора оголовья молотка и кнопка для ввода параметров по умолчанию, в отличие от начальной версии диаграммы вариантов использования.

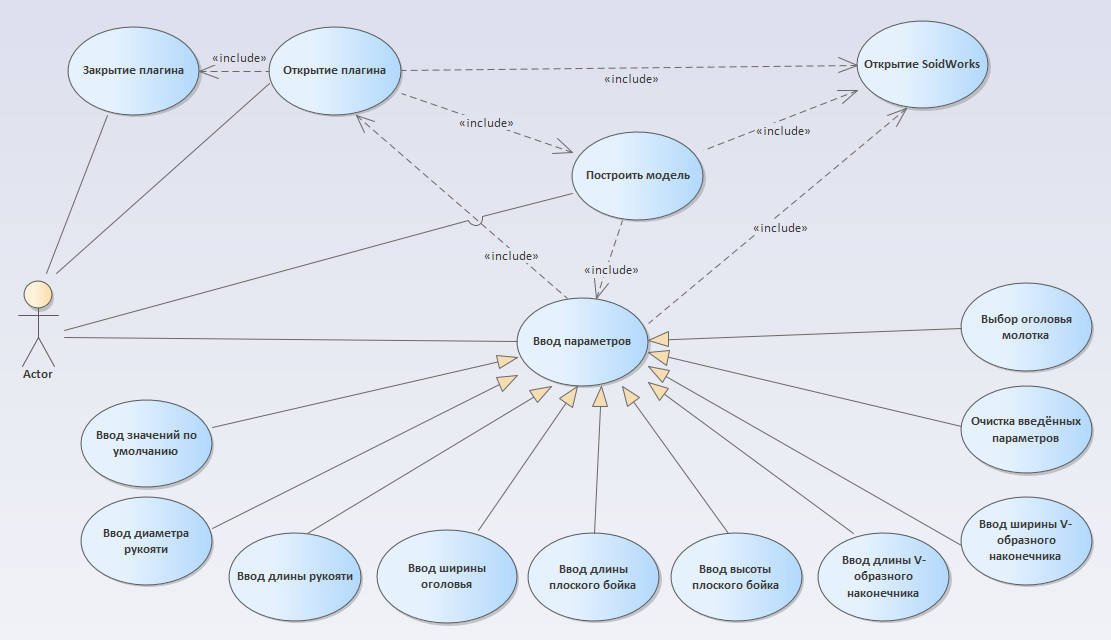


Рисунок 3.2 – Финальная диаграмма вариантов использования

# 3.3 Диаграмма классов

На рисунке 3.3 представлена начальная диаграмма классов, описывающая архитектуру приложения.

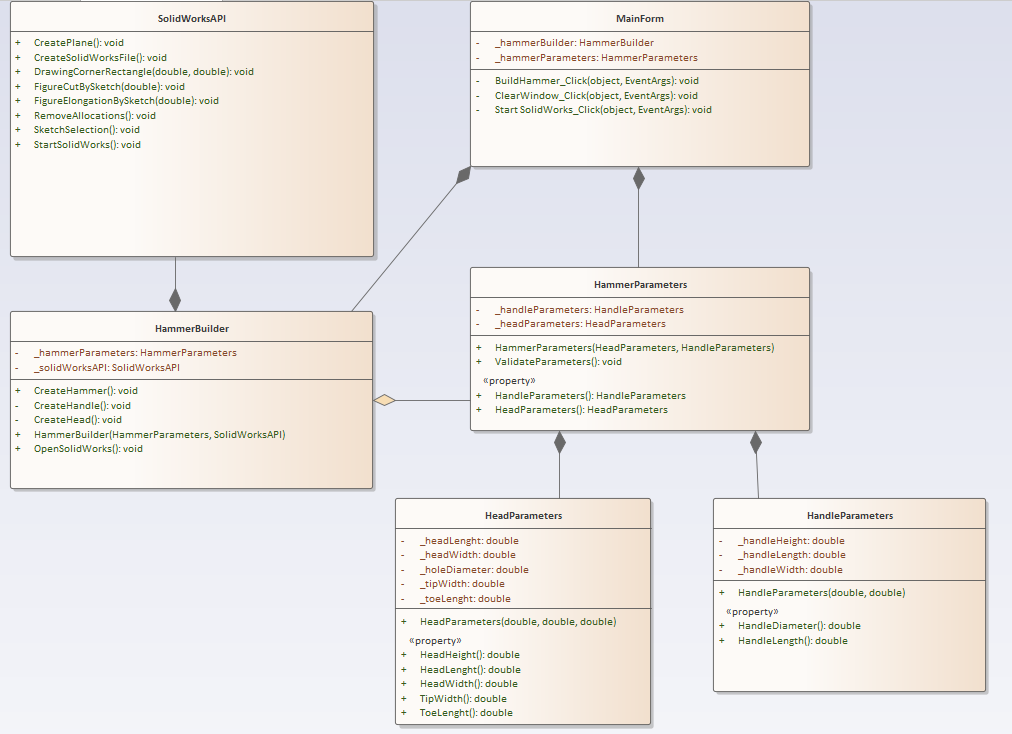


Рисунок 3.3 – Начальная диаграмма классов

Для реализации приложения были спроектированы следующие классы:

* MainForm – класс диалогового окна обеспечивающий взаимодействие между пользователем и программой;
* HammerParameters – класс, хранящий в себе все параметры модели, а также обеспечивающий проверку введенных параметров;
* HeadParameters – класс, хранящий в себе параметры оголовья молотка;
* HandleParameters – класс, хранящий в себе параметры рукояти молотка;
* HammerBuilder – класс, отвечающий за вызов методов Solidworks API необходимых для постройки объекта проектирования;
* SolidworksAPI – класс, отвечающий за работу с API Solidworks.

В итоговом проекте были созданы следующие классы и методы, которые отображены на финальной диаграмме классов (рисунок 3.4).

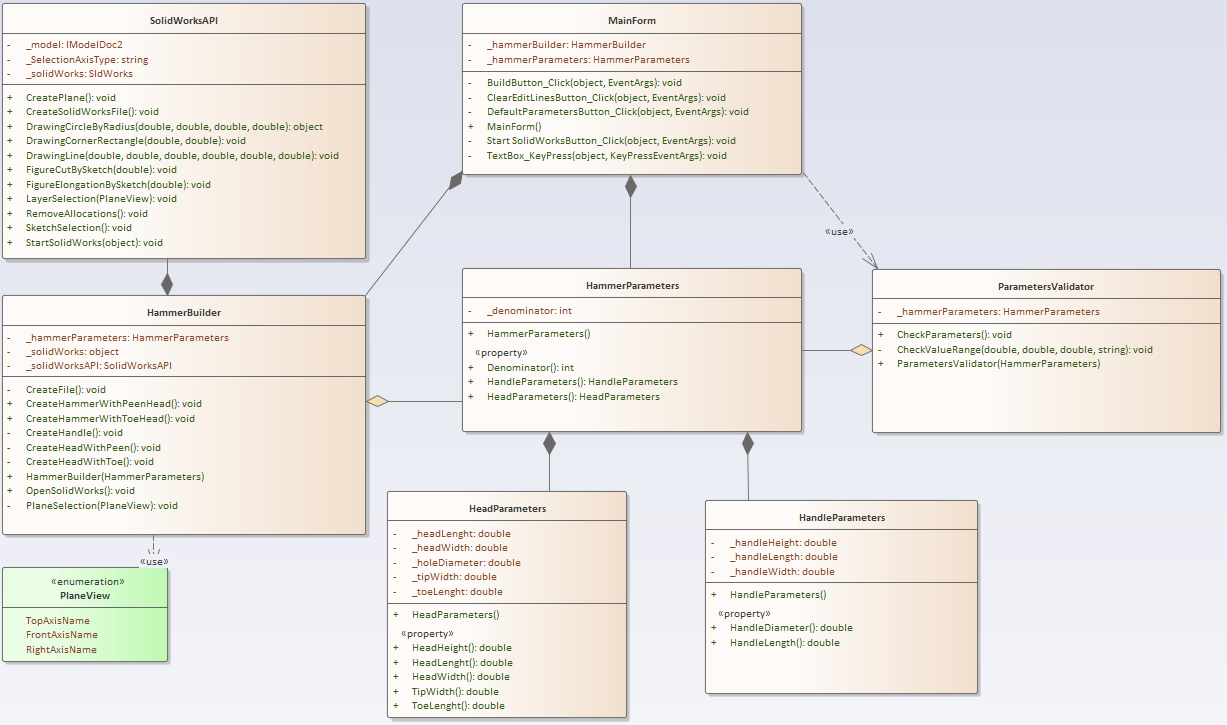


Рисунок 3.4 – Финальная диаграмма классов

Список основных изменений:

* Полное обновление методов класса SolidWorksAPI;
* В класс MainForm были добавлены методы, без которых использование программы было бы менее комфортным для конечного пользователя;
* В класс HammerBuilder добавлены методы для построения молотка с двумя видами оголовья;
* Добавлен класс ParametersValidator, отвечающий за валидацию вводимых пользователем параметров;
* Добавлен класс PlaneView, содержащий перечисления используемых видов на плоскость.

В классе SolidWorksAPI были реализованы следующие методы:

* IsThereSolidWorks – метод, проверяющий наличие SolidWorks 2020 на персональном компьютере пользователя;
* SketchSelection – метод, для выбора эскиза;
* StartSolidWorks – метод, запускающий SolidWorks 2014/2020;
* RemoveAllocations – метод, убирающий выделения с объекта моделирования;
* LayerSelection – метод, необходимый для выбора вида и плоскости;
* FigureCutBySketch – метод, для вырезания по эскизу;
* FigureElongationBySketch – метод, вытягивающий объект по эскизу;
* CreateSolidWorksFile – метод, создающий файл в котором моделируется объект;
* CreatePlane – метод, создающий плоскость;
* DrawingLine – метод, рисующий линию по координатам;
* DrawingCornerRectangle – метод, рисующий эскиз прямоугольника на плоскости;
* DrawingCircleByRadius – метод, рисующий эскиз круга на плоскости.

В класс MainForm были добавлены следующие методы:

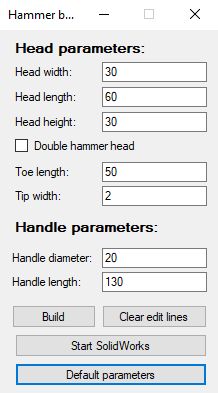
* DefaultParametersButton\_Click – метод, выставляющий параметры по умолчанию в поля ввода параметров;
* TextBox\_KeyPress – метод, ограничивающий ввод любых символов, кроме цифр, клавиши Backspace и запятой;
* HeadCheckBox\_CheckedChanged – метод, активирующий и деактивирующий поля ToeLength и TipWidth.

В класс HammerBuilder были добавлены следующие методы:

* CreateHeadWithToe – метод, строящий оголовье с наконечником;
* CreateHeadWithPeen – метод, строящий оголовье с двойным бойком;
* CreateHammerWithPeenHead – метод, строящий молоток, у которого оголовье с двойным бойком;
* CreateHammerWithToeHead – метод, строящий молоток, у которого оголовье с наконечником.

# 4 Описание программы для пользователя

Плагин состоит из диалогового окна, который имеет кнопки: «Build», «Clear edit lines», «Start SolidWorks», «Default Parameters; блоки: «Head parameters» и «Handle parameters», внутри которого находятся параметры модели и опциональный выбор вида оголовья модели. При запуске имеет следующий вид (рисунок 4.1).

  
Рисунок 4.1 – Главное окно пользовательского интерфейса

При старте плагина пользователь должен ввести необходимые параметры и, опционально, выбрать вид оголовья. Если ввести неверный параметры, после нажатия кнопки «Build», высветится окно с описание неправильного заполненного поля (рисунок 4.2).

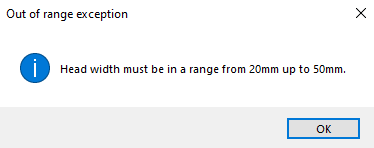
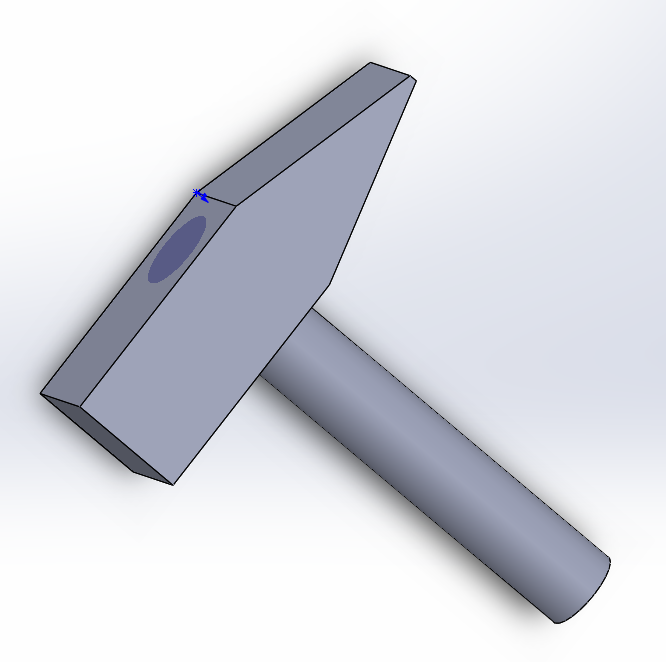


Рисунок 4.2 – Окно сообщения об ошибке ввода параметра

После ввода необходимых параметров, построить деталь в САПР SOLIDWORKS 2014/2020 можно с помощью кнопки «Build». Например, фигура, построенная по заданным параметрам по умолчанию, выглядит следующим образом (рисунки 4.3-4.4).

  
Рисунок 4.3 – Модель с v-образным наконечником

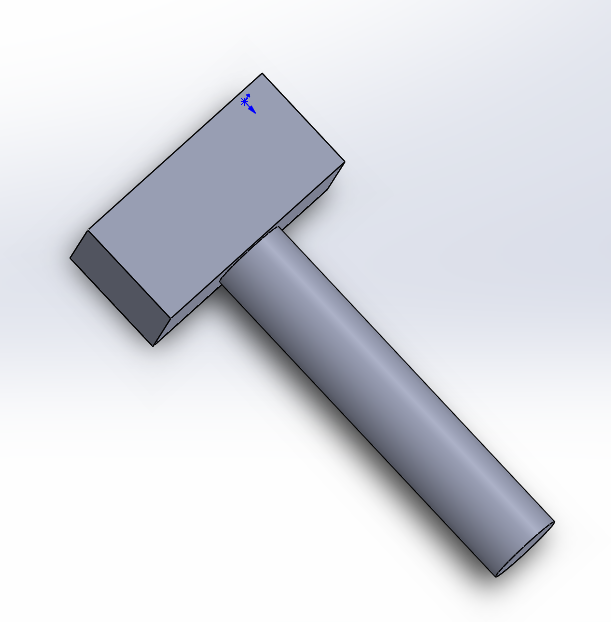


Рисунок 4.4 – Модель с двойным бойком

# 5 Тестирование программы

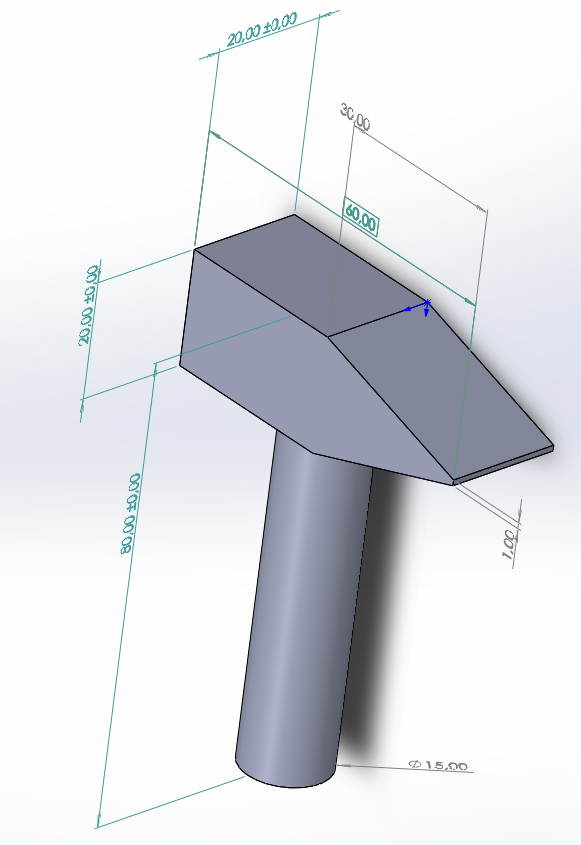
Тестирование позволяет убедиться в работоспособности программы, выявлять ошибки при изменении какой-либо функциональности.

# 5.1 Функциональное тестирование

При функциональном тестировании проверялось корректность работы плагина «Hammer», а именно, соответствие полученного результата в виде трехмерной модели, с входными параметрами. [14]

Проведено тестирование максимальных и минимальных параметров модели.

Ниже на рисунке 5.1 представлена проверка размеров модели с минимальным введенными параметрами (длина оголовья 30 мм, ширина оголовья 20 мм, высота оголовья 20 мм, длина носка 30 мм, ширина носка 1 мм, диаметр рукояти 15мм, длина рукояти 80мм).

  
Рисунок 5.1 – Модель с минимальными веденными параметрами

Ниже на рисунке 5.2 представлена проверка размеров модели с максимальными введенными параметрами (длина оголовья 80 мм, ширина оголовья 50 мм, высота оголовья 50 мм, длина носка 80 мм, ширина носка 10 мм, диаметр рукояти 40мм, длина рукояти 200мм).

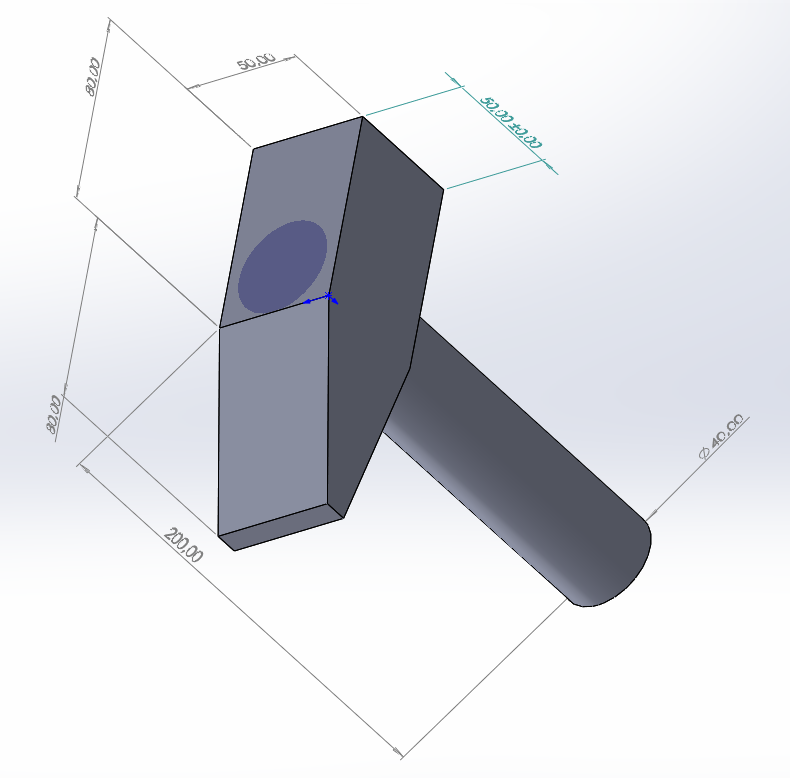


Рисунок 5.2 - Модель с максимально веденными параметрами

Ниже на рисунке 5.3 представлена проверка размеров модели с минимально введенными параметрами и выбранным параметром «Double hammer head» (длина оголовья 30 мм, ширина оголовья 20 мм, высота оголовья 20 мм, диаметр рукояти 15мм, длина рукояти 80мм).

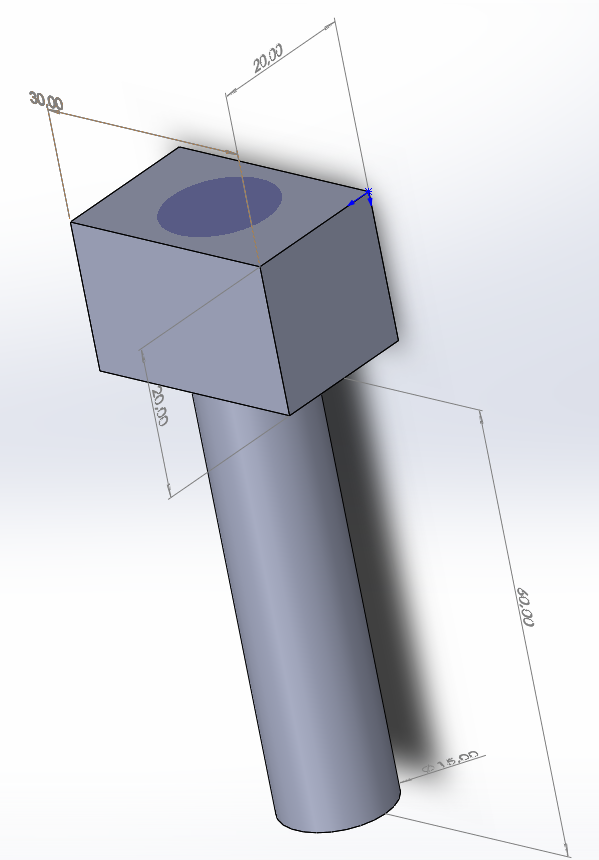


Рисунок 5.3 – Модель с минимальными значениями и двусторонним бойком

Ниже на рисунке 5.4 представлена проверка размеров модели с максимально введенными параметрами и выбранным параметром «Double hammer head» (длина оголовья 30 мм, ширина оголовья 20 мм, высота оголовья 20 мм, диаметр рукояти 15мм, длина рукояти 80мм).

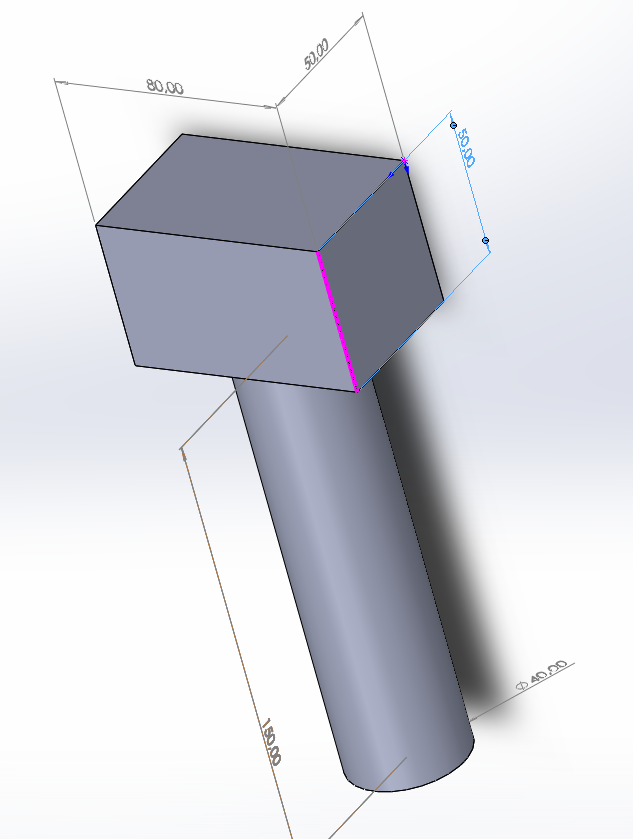


Рисунок 5.4 - Модель с максимальными значениями и двусторонним бойком

# 5.2 Модульное тестирование

В целях проверки корректности работы методов и свойств классов при помощи обозревателя тестов расширения для Visual Studio – ReSharper с тестовым Фреймворком NUnit версии 3.12 проведено модульное тестирование [15], проверялись открытые поля и методы. На рисунке 5.5 представлено тестирование класса «HammerParameters. Классы «SolidWorkAPI», «HammerBuilder», «MainForm» не подвергаются тестированию (описание тестовых случаев находится в приложении А.1).

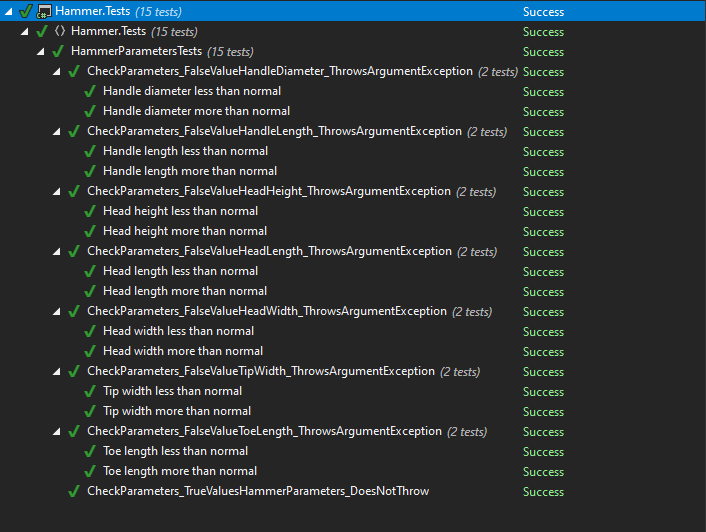


Рисунок 5.5 – Тестирование приложения

# 5.3 Нагрузочное тестирование

В целях проверки производительности работы плагина, было проведено нагрузочное тестирование [16]. Тестирование производилось на ПК со следующей конфигурацией:

* Процессор: Intel Core i5-7300HQ
* Графическая система: Intel HD Graphics 630
* ОЗУ: 8 GB DDR4 1200 MHz

На рисунке 5.6 для проведения нагрузочного тестирования был добавлен секундомер («Stopwatch»), который засекал время от начала построения, с каждым успешным построением фигуры производилась запись результатов в текстовый файл «Logs.txt».

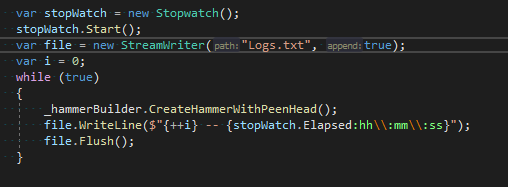


Рисунок 5.6 – Зацикливание построения фигуры

На графике, изображенном на рисунке 5.7 в текущей главе, ось «X» – время в секундах, ось «Y» – количество построенных деталей. На графике, изображенном на рисунке 5.8 ось «Y» - количество построенных деталей, ось «X» - количество потребляемой оперативной памяти. На протяжении всех тестов (продолжительностью 213 секунд, до сбоя SolidWorks) общая загруженность процессора была в пределах 30% процентов, потребление ОЗУ плагином прямолинейное от 22МБ до 40МБ.

На рисунке 5.6 представлено тестирование зацикленного перестроения фигуры с параметрами по умолчанию:

* Head width 30 мм;
* Head length 60 мм;
* Head height 30 мм;
* Handle diameter 20 мм;
* Handle length 130 мм;

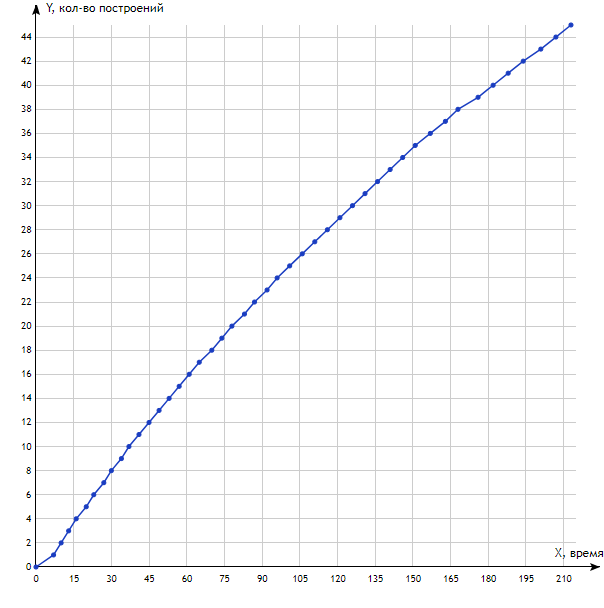


Рисунок 5.8 – График зависимости времени на построение детали от количества построенных деталей с параметрами по умолчанию

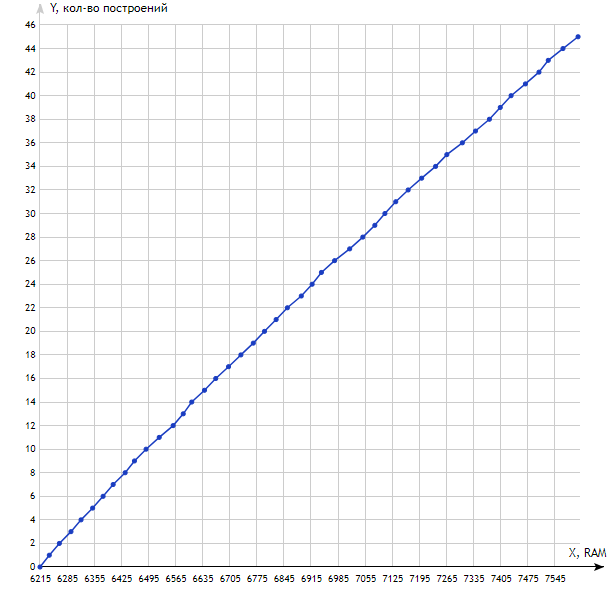


Рисунок 5.9 – График зависимости загруженности памяти от количества деталей с параметрами по умолчанию

Исходя из вышеуказанных графиков на рисунках 5.8 и 5.9, построение первой детали самое долгое – 7 секунд. После первого построения детали, построение последующих деталей занимало от 3-и до 7-и секунд. На рисунке 5.8 видно, что до запуска плагина, было занято 6200 МБ оперативной памяти системой и сторонними процессами, которые к самому плагину отношения не имеют. Также при запуске плагином процесса SolidWorks, оперативная память при постройке всех деталей потребляется почти равномерно.

# Заключение

В ходе выполнения лабораторных работ были изучены предметная область проектирования, предмет проектирования, аналоги предмета проектирования, API и на основании полученных данных были спроектированы UML диаграммы классов, разработан плагин для создания 3D модели «Молоток» в САПР SOLIDWORKS 2014/2020, а также проведено функциональное и нагрузочное тестирование приложения.

# Список использованных источников

1. SOLIDWORKS [Электронный ресурс]. – URL: <https://www.solidworks.com/> (дата обращения: 20.04.2020);
2. Visual Studio [Электронный ресурс]. – URL: <https://visualstudio.microsoft.com/ru/> (дата обращения: 20.04.2020);
3. Что такое API? Хабр [Электронный ресурс]. – URL: <https://habr.com/ru/post/464261/> (дата обращения: 20.04.2020);
4. Microsoft Word 2019 [Электронный ресурс]. – URL: <https://officeproduct.info/> (дата обращения: 20.04.2020);
5. Автоматизация вычислительных процедур в прикладных задачах инженерного проектирования [Электронный ресурс]. – URL: https://scienceforum.ru/2014/article/2014000201 (дата обращения: 20.04.2020);
6. Молоток – Википедия. [Электронный ресурс]. – Режим доступа: <https://ru.wikipedia.org/wiki/Молоток> (дата обращения 20.04.2020)
7. .NET Framework – Microsoft. [Электронный ресурс]. – Режим доступа: <https://docs.microsoft.com/ru-ru/dotnet/framework/get-started/overview> (дата обращения 20.04.2020)
8. SolidWorks.Interop.sldworks Namespace. SOLIDWORKS API Help. [Электронный ресурс]. – URL: https://help.solidworks .com/2019/English/api/sldworksapi/SolidWorks.Interop.sldworks~SolidWorks.Interop.sldworks\_namespace.html (дата обращения: 20.04.2020);
9. Начало работы с WinForms [Электронный ресурс]. – URL: <https://docs.microsoft.com/ru-ru/dotnet/framework/winforms/> (дата обращения: 25.04.2020);
10. Плагин PDF [Электронный ресурс]. – URL: <http://gkmsoft.ru/allcatalog/pdf2dkompas_plugin/> (дата обращения: 20.04.2020);
11. 3D PDF [Электронный ресурс]. – URL: <http://sapr-journal.ru/novosti/eksport-iz-kompas-3d-v-formate-3d-pdf/> (дата обращения: 20.04.2020);
12. UML [Электронный ресурс]. – URL: <http://www.uml.org/> (дата обращения: 13.04.2020);
13. Диаграммы вариантов использования – Учебная и научная деятельность Анисимова В.В. [Электронный ресурс]. – Режим доступа: <https://sites.google.com/site/anisimovkhv/learning/pris/lecture/tema12/tema12_2> (дата обращения 13.04.2020)
14. Функциональное тестирование [Электронный ресурс]. – URL: <https://daglab.ru/funkcionalnoe-testirovanie-programmnogo-obespechenija/> (дата обращения: 15.04.2020);
15. Юнит-тестирование для чайников [Электронный ресурс]. – URL: <https://habr.com/ru/post/169381/> (дата обращения: 15.04.2020);
16. Нагрузочное тестирование: с чего начать и куда смотреть [Электронный ресурс]. – URL: <https://habr.com/ru/company/jugru/blog/329174/> (дата обращения: 17.05.2020);

# Приложение А

(Справочное)

Описание полей и методов используемых для проверки тестовых случаев класса «ParametersTests» представлено в таблице А.1

Таблица А.1 – тестовые случаи

|  |  |
| --- | --- |
| Наименование | Описание |
| CheckParameters  \_TrueValuesHammerParameters  \_DoesNotThrow() | Тест всех корректно введённых значений. |
| CheckParameters  \_FalseValueHeadWidth  \_ThrowsArgumentException  (double headParametersWidth)  (10, 60) | Тест на некорректный ввод значений в поле HeadWidth (Значения выходят за минимальный и максимальный пределы) |
| CheckParameters  \_FalseValueHeadLength  \_ThrowsArgumentException  (double headParametersLength)  (20,90) | Тест на некорректный ввод значений в поле HeadLength (Значения выходят за минимальный и максимальный пределы) |
| CheckParameters  \_FalseValueHeadHeight  \_ThrowsArgumentException  (double headParametersHeight)  (10, 60) | Тест на некорректный ввод значений в поле HeadHeight (Значения выходят за минимальный и максимальный пределы) |
| CheckParameters  \_FalseValueTipWidth  \_ThrowsArgumentException  (double headParametersTipWidth)  (0, 15) | Тест на некорректный ввод значений в поле TipWidth (Значения выходят за минимальный и максимальный пределы) |

Окончание таблицы А.1

|  |  |
| --- | --- |
| Наименование | Описание |
| CheckParameters  \_FalseValueToeLength  \_ThrowsArgumentException  (double headParametersToeLength)  (20, 90) | Тест на некорректный ввод значений в поле ToeLength (Значения выходят за минимальный и максимальный пределы) |
| CheckParameters  \_FalseValueHandleDiameter  \_ThrowsArgumentException  (double handleParametersDiameter)  (10, 50) | Тест на некорректный ввод значений в поле HandleDiameter (Значения выходят за минимальный и максимальный пределы) |
| CheckParameters  \_FalseValueHandleLength  \_ThrowsArgumentException  (double HandleParametersLength)  (70, 210) | Тест на некорректный ввод значений в поле HandleLength (Значения выходят за минимальный и максимальный пределы) |