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"

Студент гр. 589-2

Зайнулин В.С.

Принял:

Доцент кафедры КСУП

Калентьев А.А

Томск 2022

**СОДЕРЖАНИЕ**

[**1 Введение** 3](#_Toc124581479)

[**2 Постановка и анализ задачи** 4](#_Toc124581480)

[**2.1 Описание объекта проектирования** 4](#_Toc124581481)

[**2.2 Назначение программы** 6](#_Toc124581482)

[**2.3 Выбор инструментов и средств разработки** 6](#_Toc124581483)

[**3 Обзор Аналогов** 7](#_Toc124581484)

[**4 Описание реализации** 8](#_Toc124581485)

[**4.1 Диаграмма классов** 8](#_Toc124581486)

[**5 Описание программы для пользователя** 11](#_Toc124581487)

[**6 Тестирование плагина** 14](#_Toc124581488)

[**6.1 Функциональное тестирование** 14](#_Toc124581489)

[**6.2 Модульное тестирование** 16](#_Toc124581490)

[**6.3 Нагрузочное тестирование** 17](#_Toc124581491)

[**Заключение** 20](#_Toc124581492)

[**Список источников** 21](#_Toc124581493)

# **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 .

Таким образом, целью данной работы является разработка плагина, автоматизирующего построение модели «Гаечный ключ» для системы автоматизированного проектирования КОМПАС-3D с помощью интегрированной среды разработки Visual Studio 2022 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 КОМПАС-3D, строит модель «Гаечный ключ».

## **2.1 Описание объекта проектирования**

Изображение моделируемого объекта:

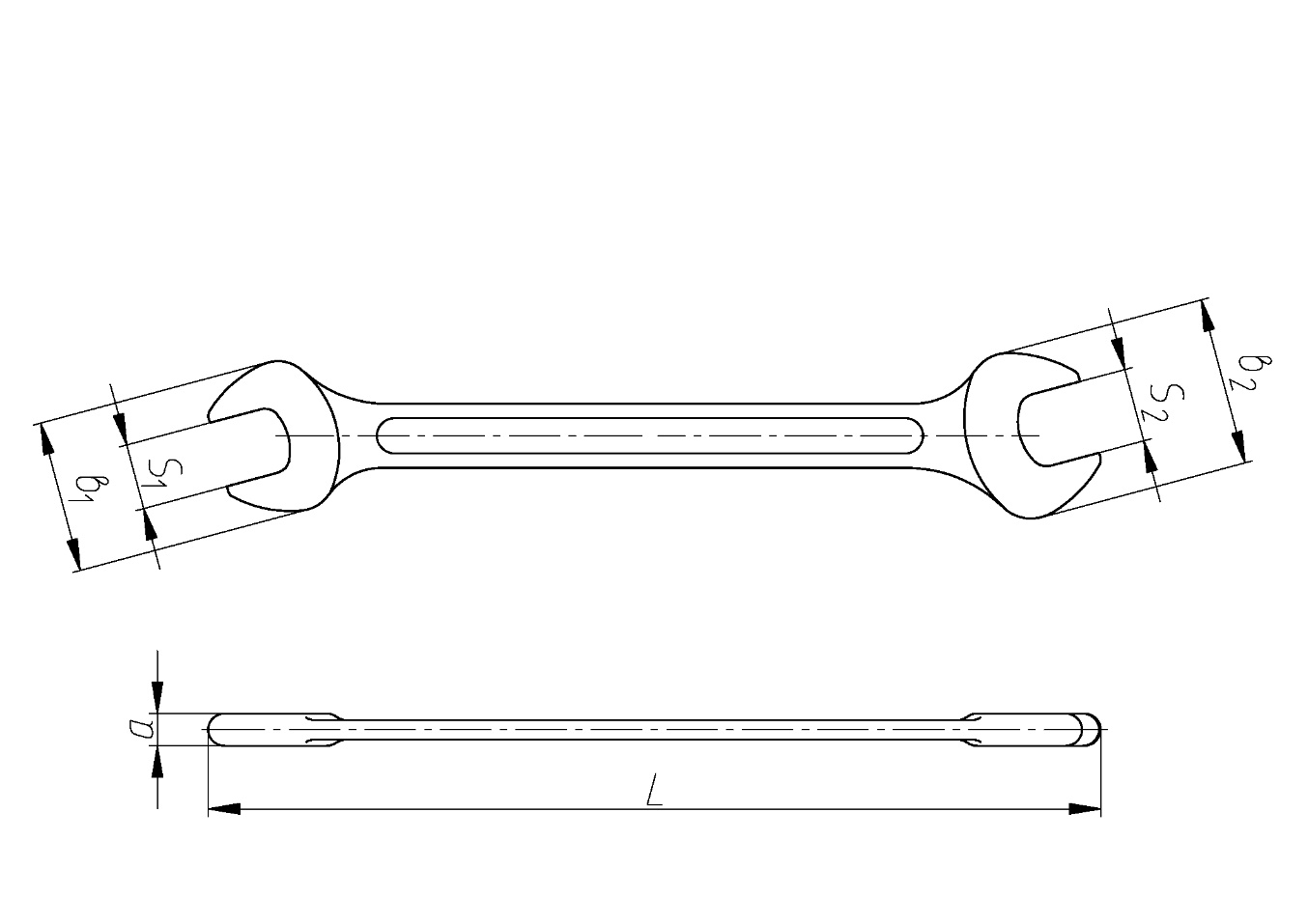


Рисунок 2.1 – модель гаечного ключа

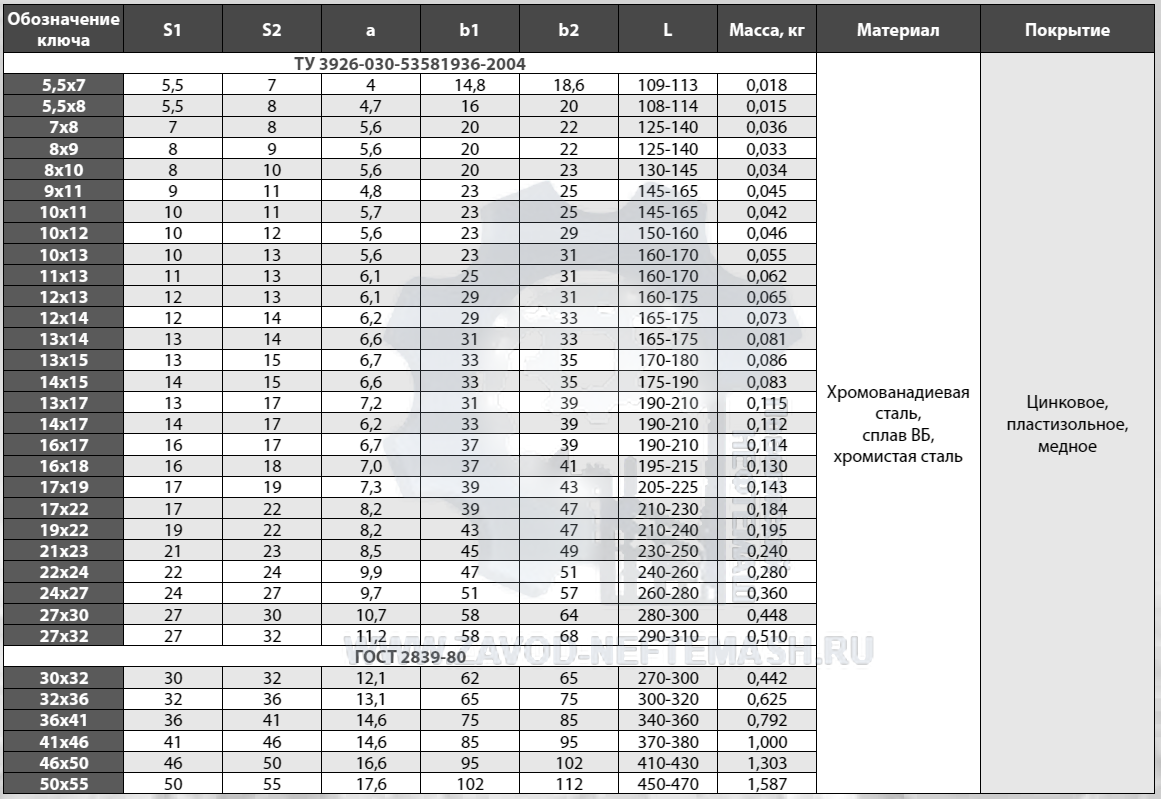


Рисунок 2.2 – Измеряемые параметры

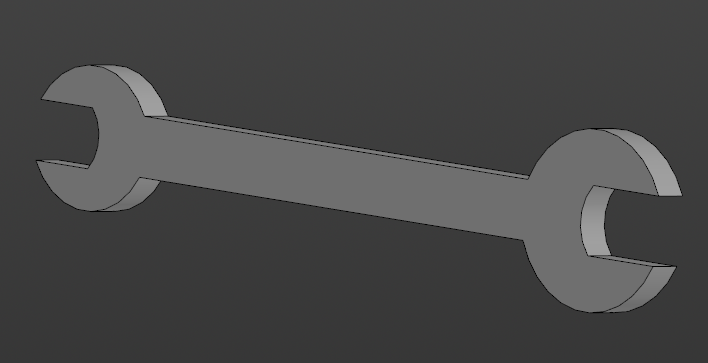


Рисунок 2.3 – 3D модель

Измеряемые параметры для плагина:

* S1 – размер зёва №1;
* S2 – размер зёва №2;
* b1 – ширина гаечного ключа №1;
* b2 – ширина гаечного ключа №2;
* a – толщина гаечного ключа;
* L – длинна гаечного ключа;
* Размер накидного ключа(7 – 55);

B1 => S1\*2;

B2 => S2\*2;

L => S1\*4;

## **2.2 Назначение программы**

Программа предназначена для автоматизации моделирования детали «Гаечный ключ». Плагин позволяет пользователю ввести вышеперечисленные значения через графический интерфейс. В программе предусмотрена проверка корректности введенных данных и сообщение пользователю о неправильно заполненных полях с помощью цветового выделения и всплывающих подсказок.

При запуске моделирования с некорректными значениями программа выводит сообщение об ошибке и отменяет построение модели.

При правильно введенных значениях результатом работы программы будет созданная по ним модель гаечного ключа.

## **2.3 Выбор инструментов и средств разработки**

На основе требований к техническому заданию программа выполнена на языке программирования C# в среде Microsoft Visual Studio 2022 с использованием .NET Framework 4.7.2 , библиотека «Kompas6API5» для основных операций в САПР КОМПАС-3D.

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

Технология разработки графического интерфейса: Windows Forms.

# **3 Обзор Аналогов**

PTC Creo — это масштабируемый, функционально совместимый пакет программного обеспечения для конструирования изделий. Он позволяет группам конструкторов создавать, анализировать, просматривать и максимально использовать проекты изделий при дальнейшем конструировании, используя 2 - и 3-мерное моделирование CAD, параметрическое и прямое моделирование. Классический интерфейс PTC Creo представлен на рисунке 3.1.

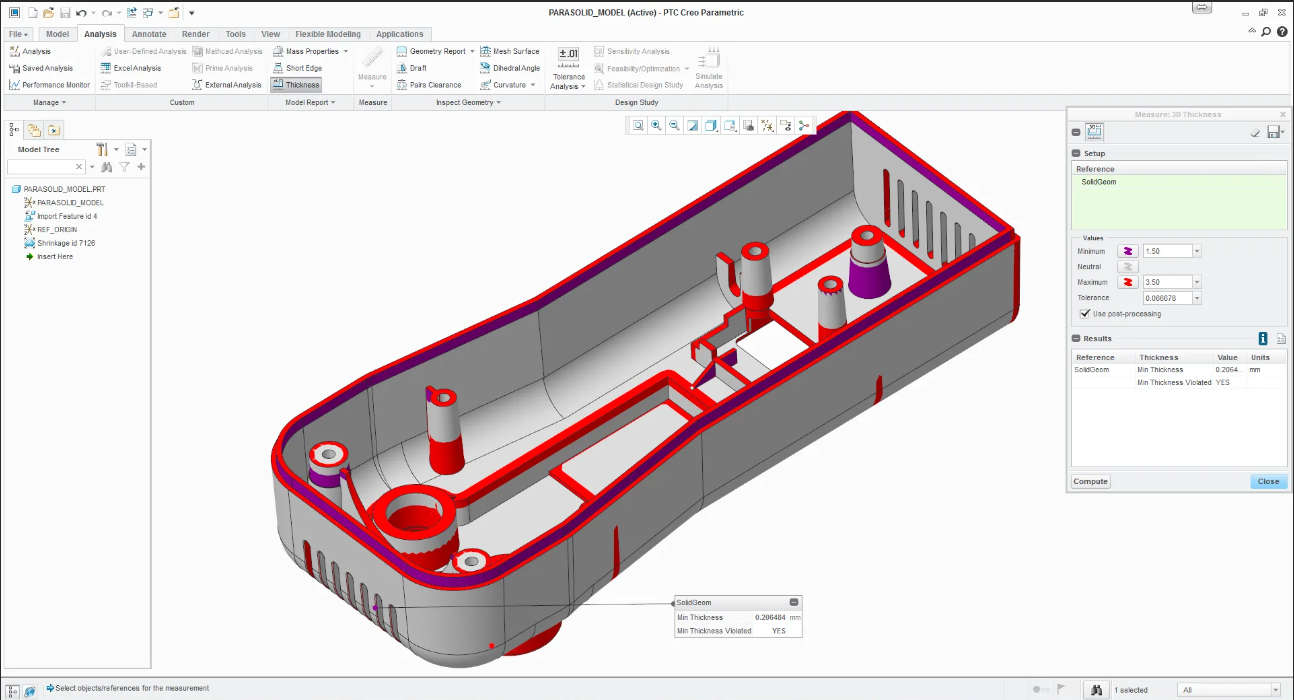


Рисунок 3.1 – Классический интерфейс PTC Creo

# **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**

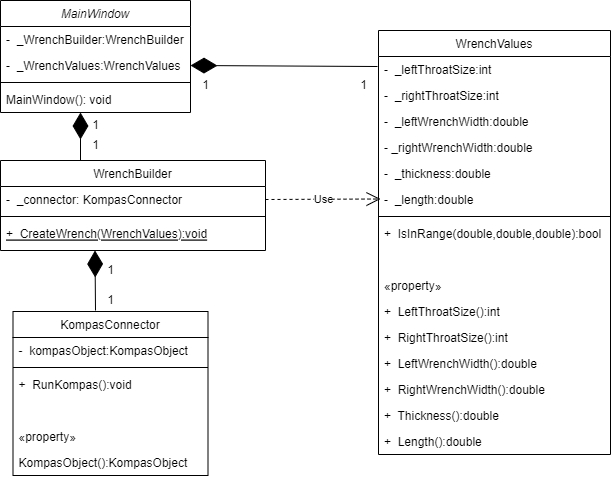


Рисунок 4.1 Диаграмма классов

1)MainForm – класс диалогового окна, который обеспечивает взаимодействие между пользователем и программой;

2)WrenchValues − класс, хранящий в себе все параметры 3D-модели;

3)WrenchBuilder – класс, осуществляющий вызов методов API, необходимых для постройки 3D-модели.

4)KompasConnector – класс для работы с API КОМПАС 3D.

В итоговом проекте созданы следующие классы и методы, которые отображены на итоговой диаграмме классов.

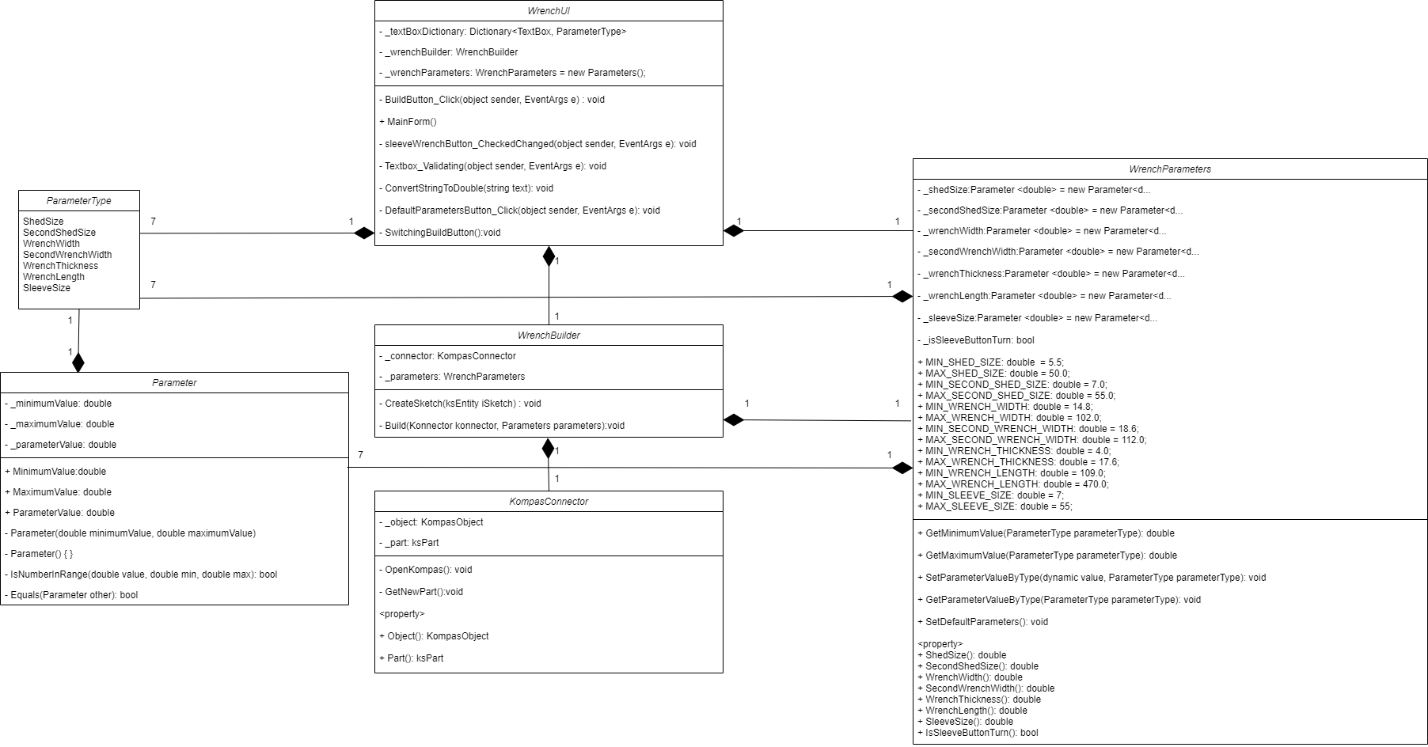


Рисунок 4.2 Итоговая диаграмма классов

В процессе разработки была переосмыслена изначальная диаграмма классов, в результате чего:

1)Было решено отказаться от задания имени в строковом формате, вместо этого было создано перечисление ParameterType, что предотвратит ошибок при поиске и занесения значений в параметры;

2)Константы минимальных и максимальных значений параметров были занесены в поля класса WrenchParameters для удобства и доступности их значений в пользовательском коде, так же в нескольких классах был использован Ассоциативный массив для уменьшения дублирования кода и упрощения передачи значений параметров.

# **5 Описание программы для пользователя**

Пользовательский интерфейс состоит из окна, в котором вводятся данные для построения гаечного ключа. Если все данные были введены корректно, то при нажатии кнопки «Построить» происходит построение модели.

Так же под списком параметров расположен переключатель выбора второй части ключа между обычным и накидным ключами.

На рисунке 5.1 представлен макет интерфейса программы.

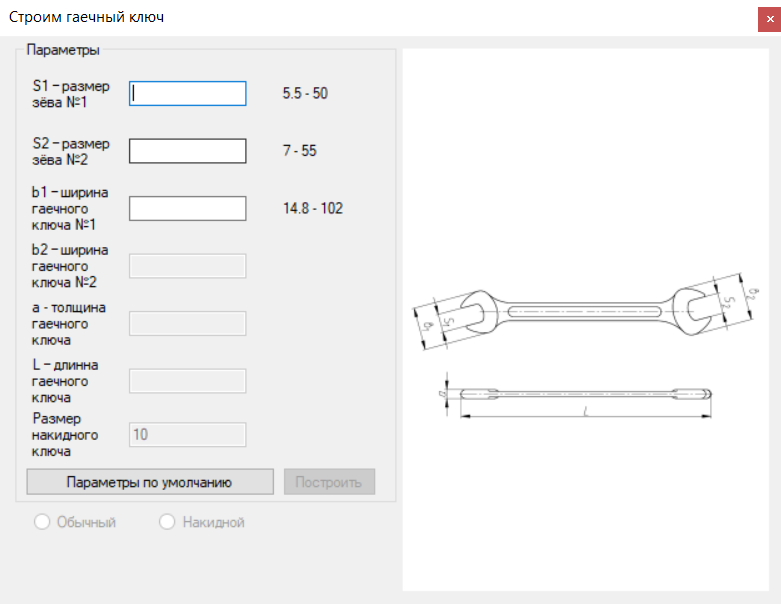


Рисунок 5.1 – Пользовательский интерфейс

Если ввести неверные параметры, то кнопка «Построить» будет заблокирована для нажатия, а поля с неправильными данными будут выделены красным цветом. На рисунке 5.2 представлен макет интерфейса с некорректно введенными данными.

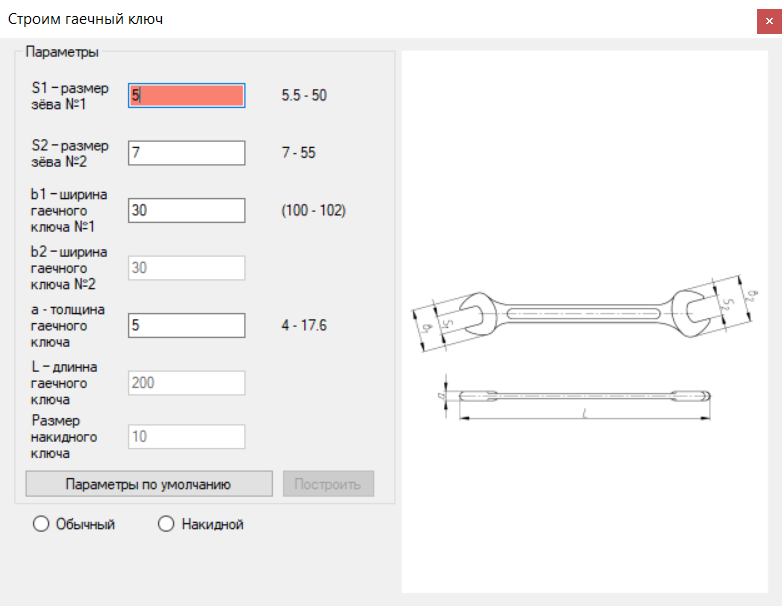


Рисунок 5.2 – Макет интерфейса с неверно введенными данными

После ввода корректных параметров кнопка «Построить» будет снова доступна и при ее нажатии будет построена модель ключа.

Ключ, построенный по заданным параметрам по умолчанию представлен на рисунке 5.3

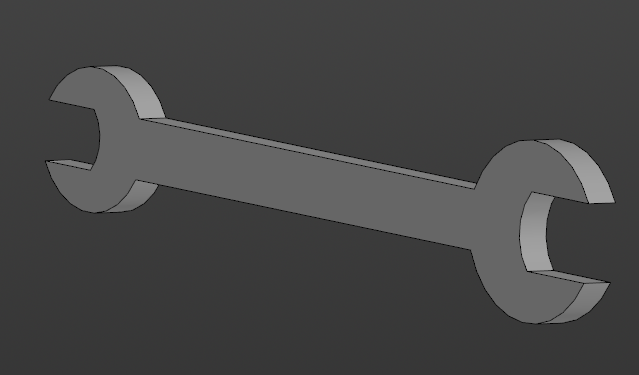


Рисунок 5.3 – Модель ключа, построенная

по параметрам по умолчанию

Модель с накидным ключом представлена на рисунке 5.4.

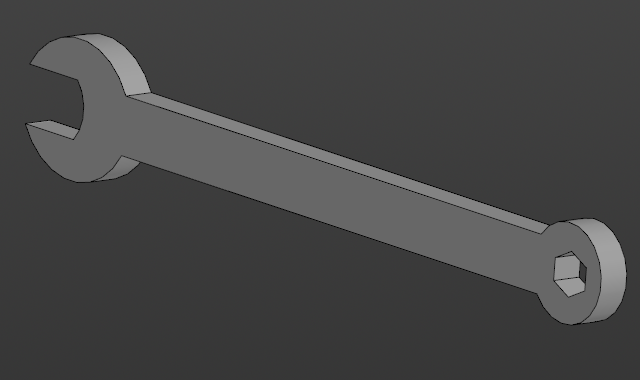


Рисунок 5.4 – Модель с накидным ключом

# **6 Тестирование плагина**

Тестирование позволяет убедиться в работоспособности программы, выявлять ошибки при изменении какого-либо функционала.

## **6.1 Функциональное тестирование**

При функциональном тестировании проверялось корректность работы плагина, а именно, соответствие полученного результата в виде трехмерной модели, с входными параметрами.

Проведено тестирование максимальных и минимальных параметров модели.

Минимальные значения:

* S1 – размер зёва №1 = 6;
* S2 – размер зёва №2 = 7;
* b1 – ширина гаечного ключа №1 = 15;
* b2 – ширина гаечного ключа №2 = 19;
* a – толщина гаечного ключа = 4;
* L – длинна гаечного ключа = 133;
* Размер накидного ключа = 7;

Максимальные значения:

* S1 – размер зёва №1 = 50;
* S2 – размер зёва №2 = 55;
* b1 – ширина гаечного ключа №1 = 102;
* b2 – ширина гаечного ключа №2 = 112;
* a – толщина гаечного ключа = 17;
* L – длинна гаечного ключа = 470;
* Размер накидного ключа = 55;

На рисунке 6.1 представлена модель ключа с минимально введенными параметрами.

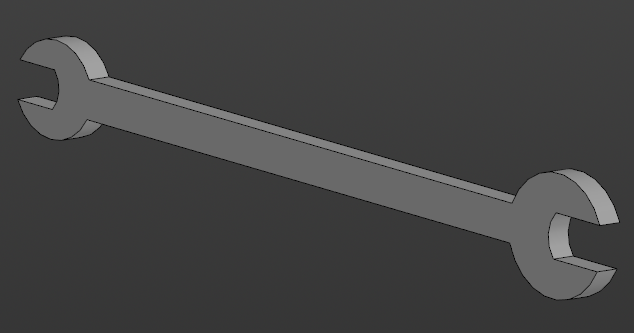


Рисунок 6.1 – Модель ключа с минимальными введенными параметрами

На рисунке 6.2 представлена модель ключа с максимальными введенными параметрами.

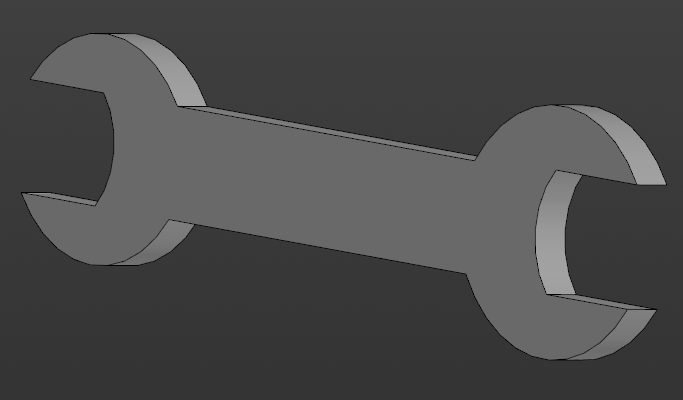


Рисунок 6.2 – Модель ключа с максимальными параметрами

На рисунке 6.3 представлена модель накидного ключа с максимальными параметрами.

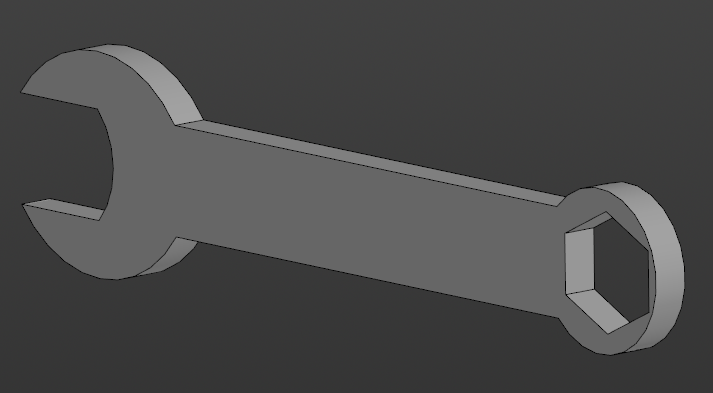


Рисунок 6.3 – Модель накидного ключа с максимальными значениями

На рисунке 6.4 представлена модель накидного ключа с минимальными параметрами.

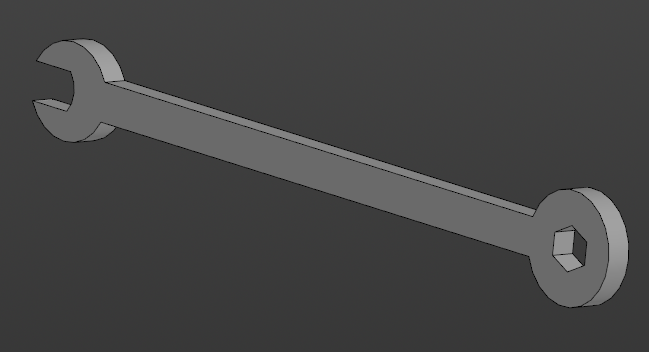


Рисунок 6.4 – Модель накидного ключа с минимальными значениями

## **6.2 Модульное тестирование**

В целях проверки корректности работы методов и свойств классов при помощи тестового фреймворка NUnit проведено модульное тестирование, проверялись открытые поля, свойства и методы. Были протестированы классы модели: Parameter, Parameters. На рисунке 6.5 представлена информация о модульном тестировании программы.

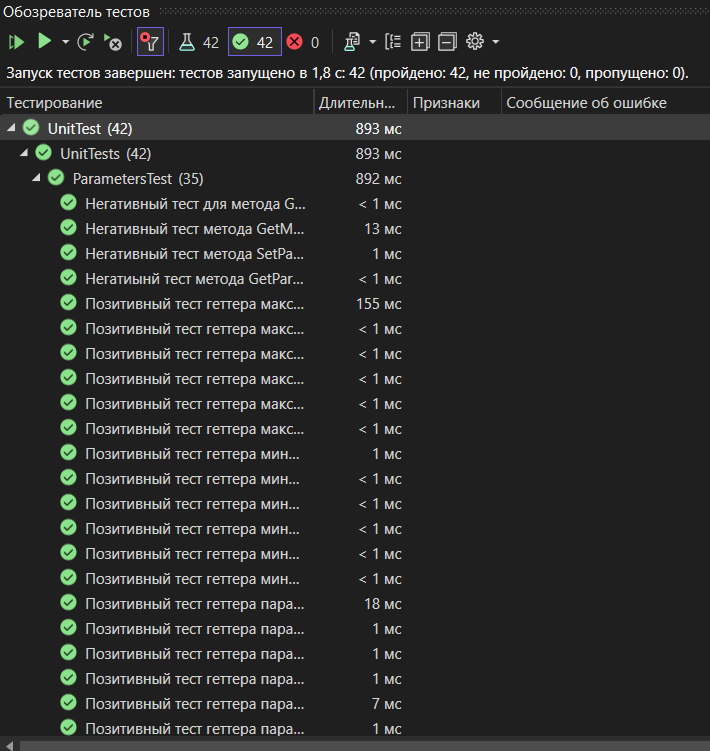


Рисунок 6.5 – Модульное тестирование плагина

## **6.3 Нагрузочное тестирование**

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

– ЦП Intel(R) Core(TM) i7-8750H CPU @ 2.20GHz 2.21 GHz;

– 16 Гб ОЗУ;

– Объем графической памяти 6 Гб.

Для нагрузочного тестирования был задан бесконечный цикл построения детали. Для измерения времени был использован класс Stopwatch. Тестирование заключалось в построении гаечного ключа со стандартными параметрами:

* S1 – размер зёва №1 = 13;
* S2 – размер зёва №2 = 17;
* b1 – ширина гаечного ключа №1 = 31;
* b2 – ширина гаечного ключа №2 = 39;
* a – толщина гаечного ключа = 7.2;
* L – длинна гаечного ключа = 200;

На графике, изображенном на рисунке 6.6 в текущей главе, ось «X» –количество построенных деталей, ось «Y» – время в секундах. На графике, изображенном на рисунке 6.7 ось «Х» - количество построенных деталей, а ось «Y» - количество потребляемой оперативной памяти.

Рисунок 6.6 Зависимость затраченного времени от количества построенных деталей

Рисунок 6.7 Зависимость используемой памяти от количества построенных деталей

По графику затраченного времени видно, что начале на построение застрачивалось в среднем 1 - 2 секундs, к концу время увеличилось до 8 - 10 секунд.

По графику затрачиваемой памяти видно, что примерно на 150 модели постепенно начинается освобождение небольших объемов данных для того, чтобы было возможно продолжать работу. Можно предположить, что в этот момент произошел переход в виртуальную память, то есть использование файла подкачки.

# **Заключение**

При выполнения лабораторных работ были изучены основные этапы проектирования программного продукта и его реализации, предметная область объекта проектирования, SDK «КОМПАС-3D». Было составлено техническое задание, разработан проект системы, составлены UML диаграммы классов, разработан макет пользовательского интерфейса.

В результате работы был разработан и реализован плагин для САПР «КОМПАС-3D», выполняющий построение 3D-модели гаечного ключа по заданным параметрам.

Над реализованным плагином были проведены функциональное, модульное и нагрузочное тестирование на платформе Windows 10.

# **Список источников**

1)Компас 3D [Электронный ресурс]. – Режим доступа: https://kompas.ru/kompas-3d/about/ Дата обращения (15.10.2022)

2)Кидрук Максим. КОМПАС-3D V10 на 100% / М. Кидрук. – СПб.: Питер, 2009 – 560 с.

3)Компас (САПР) [Электронный ресурс] – Режим доступа: https://ru.wikipedia.org/wiki/Компас\_(САПР) (дата обращения: 15.10.2020)

4)Программа автоматического построения 3D моделей и разверток по заданным значениям в AutoCAD «Лекало». Расчет и построение механических передач [Электронный ресурс] – Режим доступа: https://www.2d-3d.ru/3d-galereia/autocad/811-programma-dlya-autocad-lekalo.html (дата обращения: 15.10.2020)

5)NUnit [Электронный ресурс]. – URL: https://nunit.org/ (дата обращения: 20.10.2022).

6)Введение в Windows Forms [Электронный ресурс]. – URL: https://metanit.com/sharp/windowsforms/1.1.php (дата обращения: 20.10.2022).

7)Введение в UML от создателей языка. / Г. Буч, Д. Рамбо, И. Якобсон, 2-е изд. – ДМК Пресс, 2015 – 496 с.

8)Базовые интерфейсы API системы КОМПАС [Электронный ресурс]: PROGRAMMING-LANG.COM – режим доступа к статье: <http://programming-lang.com/ru/comp_soft/kidruk/1/j196.html>