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)

БИБЛИОТЕКА РУЧНОЙ СОКОВЫЖИМАЛКИ ДЛЯ СИСТЕМЫ АВТОМАТИЗИРОВАННОГО ПРОЕКТИРОВАНИЯ

«КОМПАС-3D»

Пояснительная записка по дисциплине

«Основы разработки САПР»

Студент гр. 589-1

\_\_\_\_\_\_\_\_\_ Михеева А.В.

\_\_\_\_\_\_\_\_\_

Руководитель

преподаватель каф., к.т.н.

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

(оценка) \_\_\_\_\_\_\_\_\_

Томск 2022

**Реферат**

Учебная работа 33 страница, 12 таблиц, 18 рисунков, 15 использованных источников.

Ключевые слова: КОМПАС-3D, РУЧНАЯ СОКОВЫЖИМАЛКА, СИСТЕМА АВТОМАТИЗИРОВАННОГО ПРОЕКТИРОВАНИЯ, САПР, БИБЛИОТЕКА.

Целью данной работы является разработка библиотеки для построения ручной соковыжималки для системы автоматизированного проектирования «КОМПАС-3D».

В процессе данной работы были пройдены все стадии разработки программного обеспечения:

* выбор темы;
* составление технического задания;
* составление проекта системы;
* разработка первой версии программы;
* тестирование;
* разработка готовой библиотеки.

Отчет по учебной работе выполнен в текстовом редакторе Microsoft Word 2016.

**Оглавление**

[Введение 4](#_Toc121260194)

[1 ПОСТАНОВКА И АНАЛИЗ ЗАДАЧИ 6](#_Toc121260195)

[1.1 Описание предмета проектирования 6](#_Toc121260196)

[1.2 Описание инструментов и средств реализации 8](#_Toc121260197)

[1.3 Назначение библиотеки 8](#_Toc121260198)

[2 ОБЗОР АНАЛОГОВ 10](#_Toc121260199)

[2.1 Плагин «Macro Honeycomb» для САПР FreeCAD 10](#_Toc121260200)

[3 ОПИСАНИЕ РЕАЛИЗАЦИИ 11](#_Toc121260201)

[4 ОПИСАНИЕ ПРОГРАММЫ ДЛЯ ПОЛЬЗОВАТЕЛЯ 19](#_Toc121260202)

[5 ТЕСТИРОВАНИЕ ПРОГРАММЫ 22](#_Toc121260203)

[5.1 Функциональное тестирование 22](#_Toc121260204)

[5.2 Модульное тестирование 26](#_Toc121260205)

[5.3 Нагрузочное тестирование 27](#_Toc121260206)

[Заключение 30](#_Toc121260207)

[Список использованных источников 31](#_Toc121260208)

# **Введение**

Концепция автоматизированного проектирования становится важнейшим фактором ускорения научно-технического прогресса. Только с использованием методов автоматизированного проектирования, возможно, получить требуемое качество разрабатываемого прибора базе большого количества проектных решений. Под автоматизацией проектирования понимается такой способ выполнения процесса разработки проекта, когда проектные процедуры операции осуществляются проектировщиком при тесном взаимодействии с компьютером. Автоматизация проектирования предполагает систематическое использование средств вычислительной техники при рациональном распределении функций между проектировщиком и компьютером и обоснованном выборе методов машинного решения задач. Рациональное распределение функций между человеком и компьютером подразумевает, что человек должен в основном решать задачи творческого характера, а компьютер – задачи, допускающие формализованное описание в виде алгоритма, что позволяет достичь большей эффективности по сравнению с традиционными ручными способами [1].

Целью данной работы является разработка библиотеки для построения ручной соковыжималки для системы автоматизированного проектирования «КОМПАС-3D».

«КОМПАС-3D» – это российская импортонезависимая система трехмерного 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 [2].

# **1 ПОСТАНОВКА И АНАЛИЗ ЗАДАЧИ**

В рамках учебной дисциплины «Основы разработки САПР» необходимо было разработать библиотеку в соответствии с техническим заданием. Библиотека должна позволять изменять входные параметры ручной соковыжималки. На основе введенных параметров библиотека, взаимодействуя с САПР «КОМПАС-3D», должна строить трёхмерную модель ручной соковыжималки.

Изменяемые параметры:

* диаметр тарелки;
* диаметр кола;
* высота кола;
* количество зубцов кола;
* количество отверстий в тарелке;
* длина отверстий в тарелке.

## **Описание предмета проектирования**

Для проектирования выбрана модель ручной соковыжималки. 3D-изображение моделируемого объекта представлено на рисунке 1.1, 2D-чертеж представлено на рисунке 1.2.

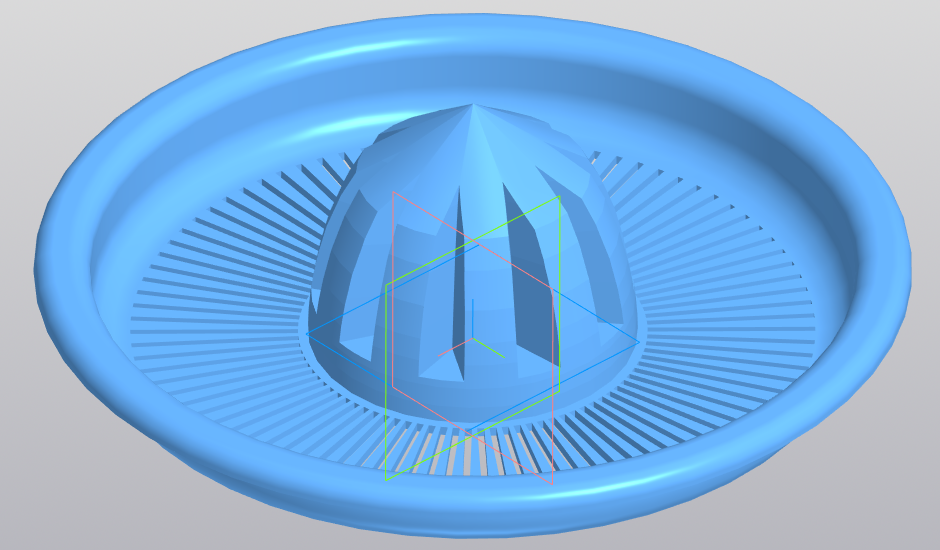


Рисунок 1.1 – 3D-модель ручной соковыжималки

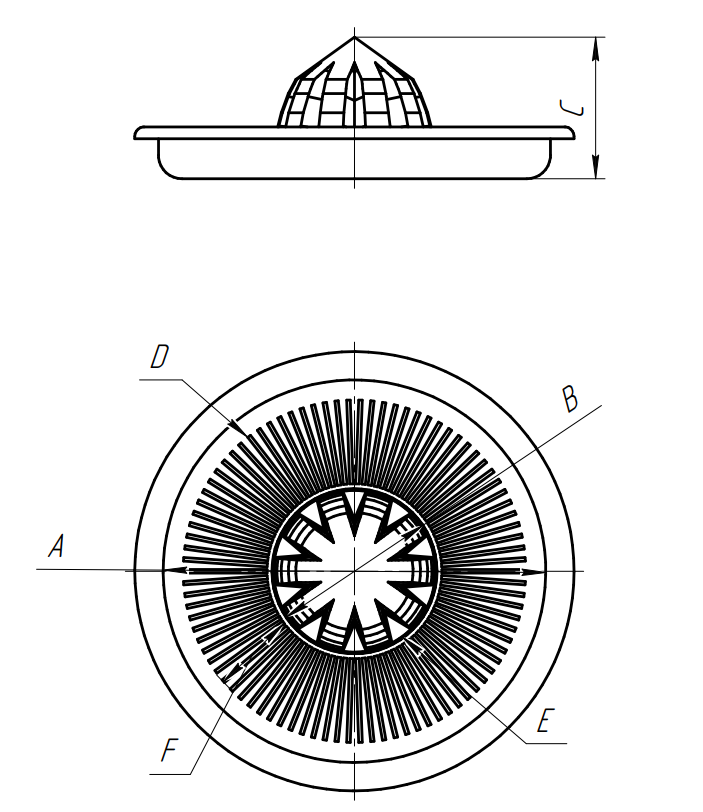


Рисунок 1.2 – 2D изображение моделируемого объекта

Параметры ручной соковыжималки:

* A – диаметр тарелки;
* B – диаметр кола;
* C – высота кола;
* D – количество отверстий в тарелке;
* E – количество зубцов в коле;
* F – длина отверстий в тарелке.

Взаимосвязь параметров:

* диаметр кола C должен быть не менее чем на 96 мм меньше диаметра тарелки A и не менее чем на 10 мм больше высоты кола C.

## **Описание инструментов и средств реализации**

Для создания библиотеки используется среда разработки Visual Studio 2019 [3]. Библиотека написана с пользовательским интерфейсом WindowsForm с использованием .NET Framework 4.7.2 [4]. Для тестирования используются библиотеки NUnit 3.13.3 [5] и NUnit3TestAdapter 4.2.1 [6]. В качестве САПР выбран отечественный «КОМПАС-3D» v.21 Учебная версия [7].

## **Назначение библиотеки**

Программа предназначена для автоматизации моделирования модели ручной соковыжималки.

Плагин позволяет пользователю ввести параметры через графический интерфейс, при этом диапазоны допустимых значений будут указаны рядом с полями для ввода. Также предусмотрена проверка корректности введенных данных. При вводе данных, выходящих за пределы указанных диапазонов, поля для ввода будут подсвечиваться красным цветом. При вводе значений параметров диапазоны зависимых параметров будут изменяться на экране. При попытке построения модели с неправильно введенными данными, пользователь увидит сообщение на экране о невозможности построения модели.

При правильно введенных значениях результатом работы программы будет созданная модель ручной соковыжималки.

# **ОБЗОР АНАЛОГОВ**

## **2.1 Плагин «Macro Honeycomb» для САПР FreeCAD**

Macro Honeycomb создает полностью параметрический объект Honeycomb, который совместим как внутри, так и вне верстака Part Design. Сотовая сетка включает необязательную границу и может иметь овальную (эллиптическую) или прямоугольную форму. Это обновление и возможная замена макроса FCHoneycombMaker, который не создает объект функции Python, а скорее полагается на электронную таблицу и некоторые массивы черновиков [8].



Рисунок 2.1 – Интерфейс «Macro Honeycomb»

# **3 ОПИСАНИЕ РЕАЛИЗАЦИИ**

UML (Unified Modeling Language) – язык графического описания для объектного моделирования в области разработки программного обеспечения, для моделирования бизнес-процессов, системного проектирования и отображения организационных структур [9].

Sparx Systems Enterprise Architect – это инструмент визуального моделирования и проектирования, основанный на OMG UML. Платформа поддерживает: проектирование и построение программных комплексов; моделирование бизнес-процессов; и моделирование отраслевых доменов [10].

Диаграмма классов – структурная диаграмма языка моделирования UML, демонстрирующая общую структуру иерархии классов системы, их коопераций, атрибутов, методов, интерфейсов и взаимосвязей между ними [9].

При использовании UML была построена диаграмма классов в программе Sparx Systems Enterprise Architect.

На рисунке 3.1 представлена начальная диаграмма классов.

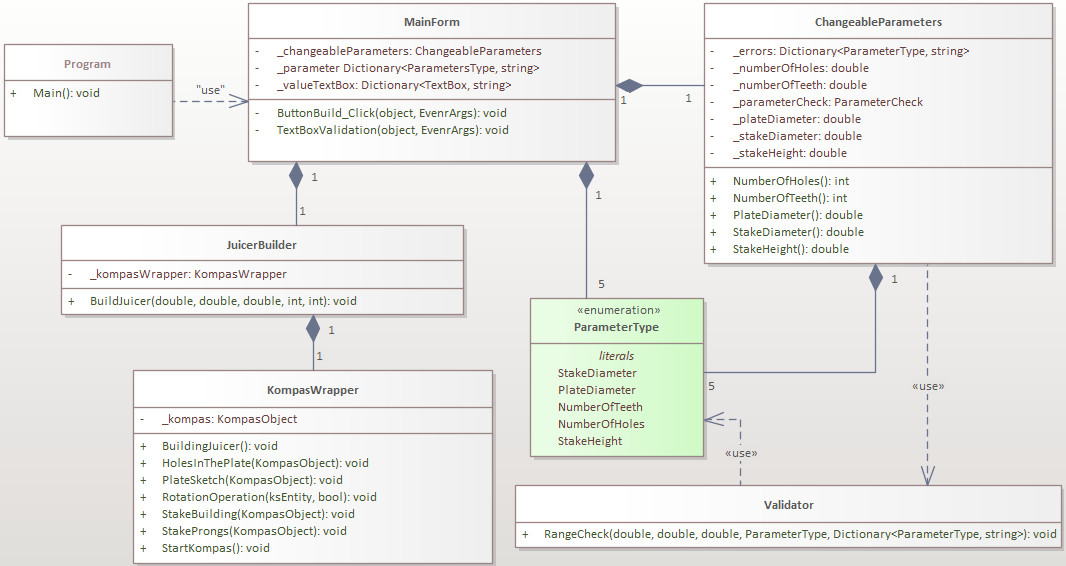


Рисунок 3.1 – Начальная диаграмма классов

Класс «KompasWrapper» предназначен для взаимодействия с «Компас-3D». Класс «JuicerBuilder» вызывает методы по построению модели из «KompasWrapper». Класс «CheangeableParameters» содержит в себе изменяемые параметры и использует класс «Validator», который проверяет введенные параметры. Класс «MainForm» содержит методы для взаимодействия с элементами формы, использует перечисление «ParameterType» и «CheangeableParameters» для устранения ошибки при вводе значений параметров, использует «KompasWrapper» и «JuicerBuilder» для построения 3D модели ручной соковыжималки в «Компас-3D».

В таблицах 3.1 – 3.6 Представлено описание классов начальной диаграммы.

Таблица 3.1 – Описание класса CheangeableParameters

|  |  |  |
| --- | --- | --- |
| **Название** | **Тип** | **Описание** |
| Поля | | |
| \_plateDiameter | double | Параметр диаметра тарелки |
| \_stakeDiameter | double | Параметр диаметра кола |
| \_stakeHeight | double | Параметр высоты кола |
| \_numberOfTeeth | double | Параметр количества зубцов |
| \_numberOfHoles | double | Параметр количества отверстий в тарелке |
| \_errors | Dictionary<ParameterType, string> | Словарь, связывающий параметр с ошибкой |
| \_parameterCheck | ParameterChecker | Объект класса ParameterChecker |
| Свойства | | |
| PlateDiameter | double | Возвращает и задает значение параметра тарелки |
| StakeDiameter | double | Возвращает и устанавливает значение диаметра кола |
| StakeHeight | double | Возвращает и устанавливает значение высоту кола |
| NumberOfTeeth | int | Возвращает и устанавливает значение количества зубцов кола |

Окончание таблицы 3.1.

|  |  |  |
| --- | --- | --- |
| **Название** | **Тип** | **Описание** |
| Свойства | | |
| NumberOfHoles | int | Возвращает и устанавливает значение количества отверстий в тарелке |

Таблица 3.2 – Описание класса Validator

|  |  |  |
| --- | --- | --- |
| **Название** | **Тип** | **Описание** |
| RangeCheck | void | Проверяет значение параметра на диапазон |

Таблица 3.3 – Описание перечисления ParameterType

|  |  |
| --- | --- |
| **Название** | **Описание** |
| PlateDiameter | Диаметр тарелки |
| StakeDiameter | Диаметр кола |
| StakeHeight | Высота кола |
| NumberOfTeeth | Количество зубцов кола |
| NumberOfHoles | Количество отверстий в тарелке |

Таблица 3.4 – Описание класса KompasWrapper

|  |  |  |
| --- | --- | --- |
| **Название** | **Тип** | **Описание** |
| Поля | | |
| \_kompas | KompasObject | Объект Компас API |
| Методы | | |
| StartKompas | void | Запускает «КОМПАС-3D» |
| BuildingJuicer | void | Создает файл в «КОМПАС-3D» |
| PlateSketch | void | Строит тарелку соковыжималки |
| RotationOperation | void | Выдавливает вращением эскиз |
| StakeBuilding | void | Строит кол |
| BuildStakeTeeth | void | Строит зубцы |
| HolesInThePlate | void | Строит отверстия в тарелке |

Таблица 3.5 – Описание класса JuicerBuilder

|  |  |  |
| --- | --- | --- |
| **Название** | **Тип** | **Описание** |
| Поля | | |
| \_kompasWrapper | KompasWrapper | Объект класса KompasWrapper |
| Методы | | |
| BuildJuicer | void | Строит модель соковыжималки |

Таблица 3.6 – Описание класса MainForm

|  |  |  |
| --- | --- | --- |
| **Название** | **Тип** | **Описание** |
| Поля | | |
| \_changeableParameters | CheangeableParameters | Объект класса CheangeableParameters |
| \_parameter | Dictionary<ParameterType, string> | Словарь, связывающий параметр с ошибкой |
| \_valueTextBox | Dictionary<TextBox, string> | Словарь, связывающий параметр с полем для ввода |
| Методы | | |
| BuildJuicer | void | Строит модель соковыжималки |
| TextBoxValidation | void | Подсвечивает поля нужным цветом |

В таблицах 3.7 – 3.12 представлено описание итоговых классов.

Таблица 3.7 – Описание класса ChangeableParameters

|  |  |  |
| --- | --- | --- |
| **Название** | **Тип** | **Описание** |
| Поля | | |
| \_plateDiameter | double | Параметр диаметра тарелки |
| \_stakeDiameter | double | Параметр диаметра кола |
| \_stakeHeight | double | Параметр высоты кола |
| \_numberOfTeeth | double | Параметр количества зубцов |
| \_numberOfHoles | double | Параметр количества отверстий в тарелке |
| \_lengthOfHoles | double | Параметр длины отверстия в тарелке |
| Parameters | Dictionary<ParameterType, string> | Словарь, связывающий параметр с ошибкой |

Окончание таблицы 3.7.

|  |  |  |
| --- | --- | --- |
| **Название** | **Тип** | **Описание** |
| Поля | | |
| \_parameterCheck | ParameterChecker | Объект класса ParameterChecker |
| Свойства | | |
| PlateDiameter | double | Возвращает и задает значение параметра тарелки |
| StakeDiameter | double | Возвращает и устанавливает значение диаметра кола |
| StakeHeight | double | Возвращает и устанавливает значение высоту кола |
| NumberOfTeeth | double | Возвращает и устанавливает значение количества зубцов кола |
| NumberOfHoles | double | Возвращает и устанавливает значение количества отверстий в тарелке |
| LengthOfHoles | double | Возвращает и устанавливает значение длины отверстий в тарелке |

Таблица 3.8 – Описание класса ParameterChecker

|  |  |  |
| --- | --- | --- |
| **Название** | **Тип** | **Описание** |
| RangeCheck | void | Проверяет значение параметра на диапазон |

Таблица 3.9 – Описание перечисления ParameterType

|  |  |
| --- | --- |
| **Название** | **Описание** |
| PlateDiameter | Диаметр тарелки |
| StakeDiameter | Диаметр кола |
| StakeHeight | Высота кола |
| NumberOfTeeth | Количество зубцов кола |
| NumberOfHoles | Количество отверстий в тарелке |
| LengthOfHoles | Длина отверстий в тарелке |

Таблица 3.10 – Описание класса KompasWrapper

|  |  |  |
| --- | --- | --- |
| **Название** | **Тип** | **Описание** |
| Поля | | |
| \_kompas | KompasObject | Объект Компас API |
| origin | int | Начало координат |
| styleLine | int[] | Стиль линий |
| degreeOfRotation | int | Угол поворота |
| Методы | | |
| StartKompas | void | Запускает «КОМПАС-3D» |
| CreateFile | void | Создает файл в «КОМПАС-3D» |
| PlateSketch | void | Строит тарелку соковыжималки |
| RotateExtrusion | void | Выдавливает вращением эскиз |
| BuildStake | void | Строит кол |
| BuildStakeTeeth | void | Строит зубцы |
| BuildHolesInThePlate | void | Строит отверстия в тарелке |

Таблица 3.11 – Описание класса JuicerBuilder

|  |  |  |
| --- | --- | --- |
| **Название** | **Тип** | **Описание** |
| Поля | | |
| \_kompasWrapper | KompasWrapper | Объект класса KompasWrapper |
| Методы | | |
| BuildJuicer | void | Построение модели соковыжималки |

Таблица 3.12 – Описание класса MainForm

|  |  |  |
| --- | --- | --- |
| **Название** | **Тип** | **Описание** |
| Поля | | |
| \_kompasWrapper | KompasWrapper | Объект класса KompasWrapper |
| Методы | | |
| BuildJuicer | void | Построение модели соковыжималки |

На рисунке 3.2 представлена итоговая диаграмма классов.

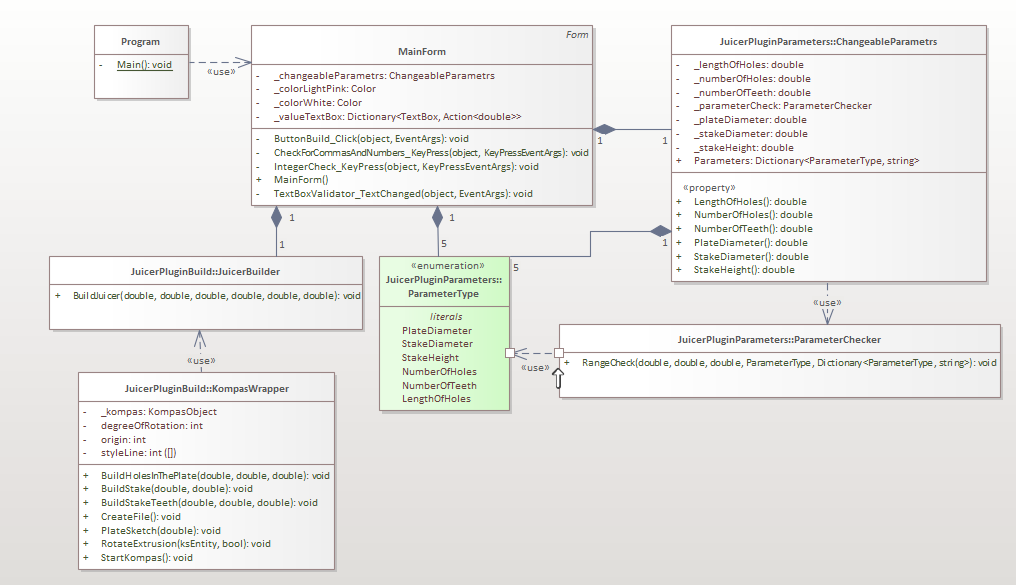


Рисунок 3.2 – Итоговая диаграмма классов

В MainForm были убраны объекты коллекции \_parameter и \_valueTextBox, вместо них была добавлена одна новая \_valueTextBox для связи между параметрами и полями для ввода значений параметров. Также добавлены две переменные типа Color для закрашивания полей ввода в белый и красный цвет и методы IntegerCheck\_KeyPressдля для ввода только цифр и CheckForCommasAndNumbers\_KeyPress для ввода только цифр и одной запятой.

В классе CheangeableParameters изменилось только название словаря для связи перечисления параметров и ошибки и было добавлено поле \_ lengthOfHoles и свойство LengthOfHoles дополнительного параметра.

Название класса Validator было изменено на ParameterChecker.

В классе JuicerBuilder был убран объект класса KompasWrapper.

В классе KompasWrapper были вынесены из методов переменные, которые встречались в нескольких методах в одном методе.

В перечисление ParameterType был добавлен дополнительный параметр LengthOfHoles.

# **4 ОПИСАНИЕ ПРОГРАММЫ ДЛЯ ПОЛЬЗОВАТЕЛЯ**

Интерфейс пользователя, он же пользовательский интерфейс – интерфейс, обеспечивающий передачу информации между пользователем-человеком и программно-аппаратными компонентами компьютерной системы [11].

Пользовательский интерфейс представляет собой диалоговое окно с 6 полями для ввода и кнопкой «Построить». По нажатию на кнопку запускается САПР «КОМПАС-3D», строится модель ручной соковыжималки по введенным пользователем параметрам. Если нажатие на кнопку «Построить» происходит, когда САПР «КОМПАС-3D» запущена, то создается новый файл и в нем строится модель с новыми параметрами. Рядом с каждым полем ввода указан допустимый диапазон параметра. Если введенное значение не входит в диапазон, то поле ввода красится в красный цвет. Если изменяется параметр, который зависит от другого или других параметров, то диапазон рядом с полем ввода меняется. При нажатии на кнопку «Построить», когда введены недопустимые значения параметров, выводится сообщении о невозможности построить модель и САПР не запускается.

На рисунке 4.1 пользовательского интерфейса. На рисунке 4.2 представлен макет пользовательского интерфейса с неправильно введенными значениями параметров, который также показывает изменение диапазонов зависимых параметров на макете. На рисунке 4.3 представлен вывод сообщения на экран при попытке построения модели с недопустимыми значениями.

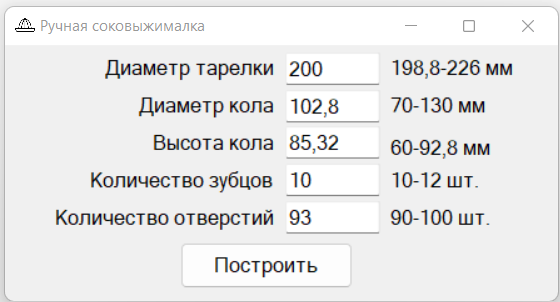


Рисунок 4.1 – Макет пользовательского интерфейса

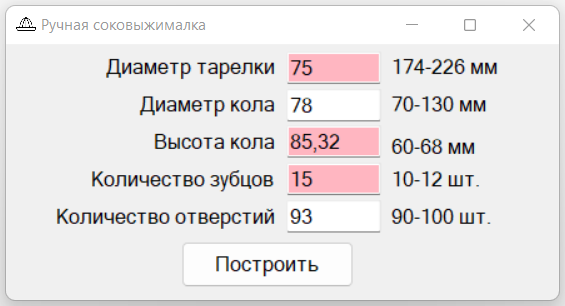


Рисунок 4.2 – Макет пользовательского интерфейса с введенными значениями вне диапазона и изменением диапазона у зависимых параметров

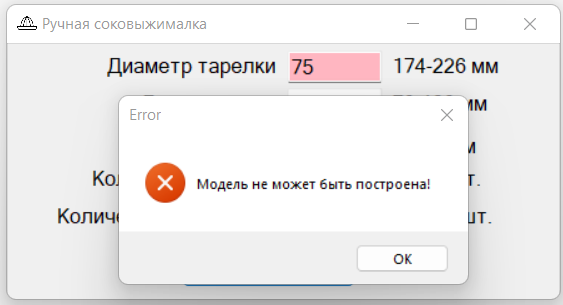


Рисунок 4.3 – Макет пользовательского интерфейса с попыткой построения модели с неправильно введенными значениями

При правильно введенных значениях программа запустит «КОМПАС-3D» и построит модель соковыжималки, представленной на рисунке 4.4.

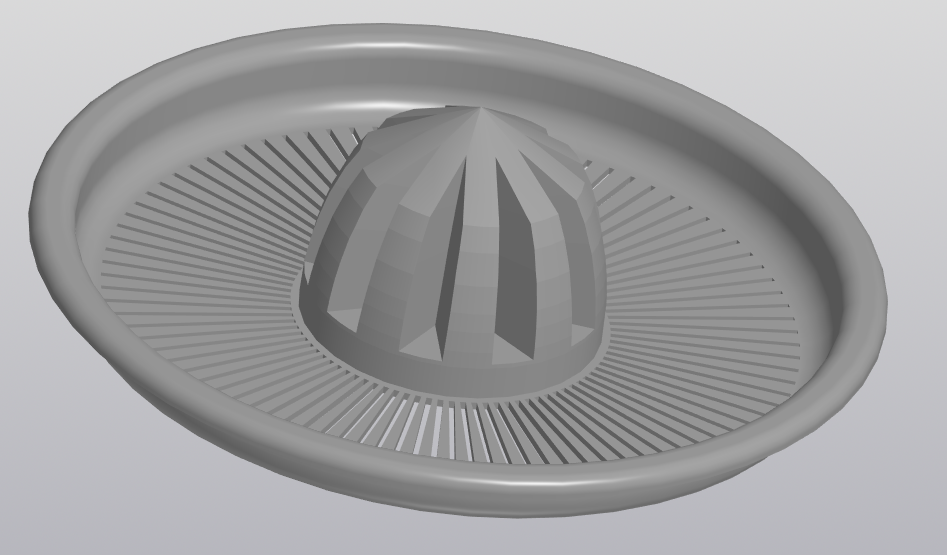


Рисунок 4.4 – Построенная модель ручной соковыжималки

# **5 ТЕСТИРОВАНИЕ ПРОГРАММЫ**

## **5.1 Функциональное тестирование**

Функциональное тестирование – это тестирование ПО в целях проверки реализуемости функциональных требований, то есть способности ПО в определённых условиях решать задачи, нужные пользователям. Функциональные требования определяют, что именно делает ПО, какие задачи оно решает [12].

Функциональное тестирование будет проведено при минимальных, средних и максимальных значениях параметров.

Минимальные значения параметров:

* диаметр тарелки равный 166 мм;
* диаметр кола равный 70 мм;
* высота кола равная 60 мм;
* количество зубцов кола равное 10 шт.;
* количество отверстий в тарелке равное 90 шт.
* длина отверстий в тарелке равная 16 мм.

Результаты представлены на рисунках 5.1 – 5.2.

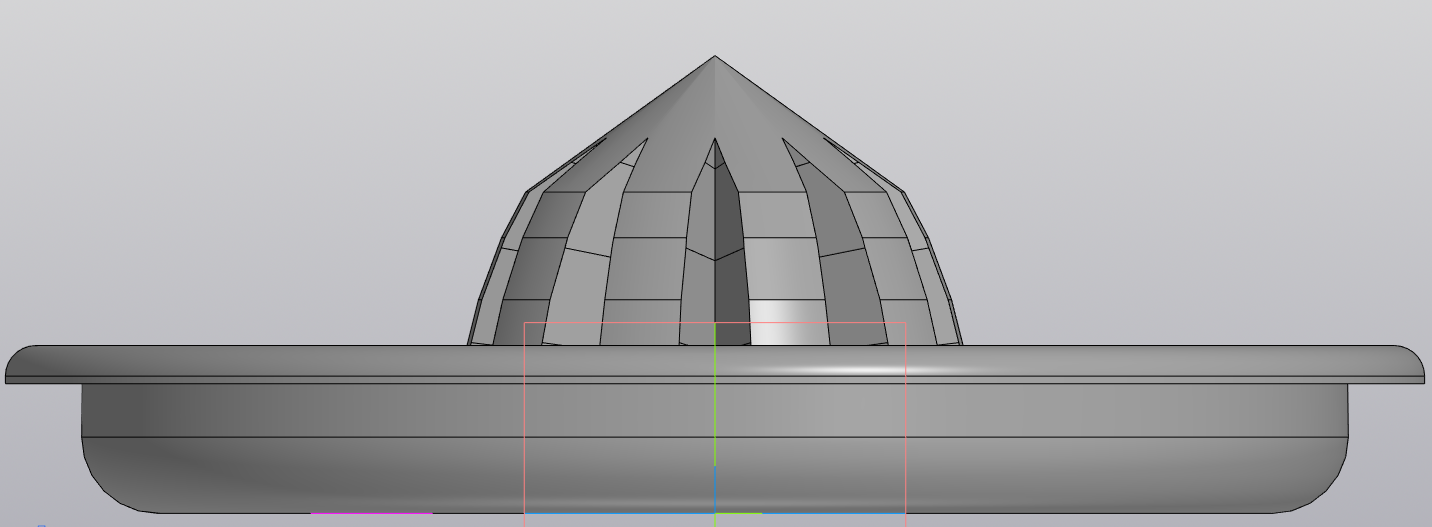


Рисунок 5.1 – Результат программы при минимальных значениях в плоскости ZY

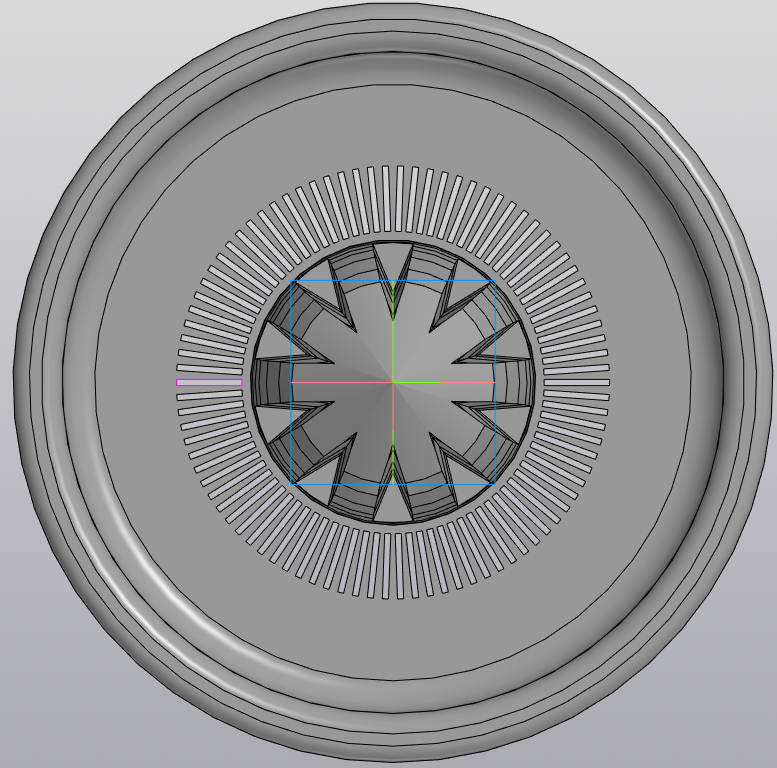


Рисунок 5.2 – Результат программы при минимальных значениях в плоскости XY

Средние значения параметров:

* диаметр тарелки равный 196 мм;
* диаметр кола равный 100 мм;
* высота кола равная 90 мм;
* количество зубцов кола равное 11 шт.;
* количество отверстий в тарелке равное 95 шт.
* длина отверстий в тарелке равная 25,75 мм.

Результаты представлены на рисунках 5.3 – 5.4.

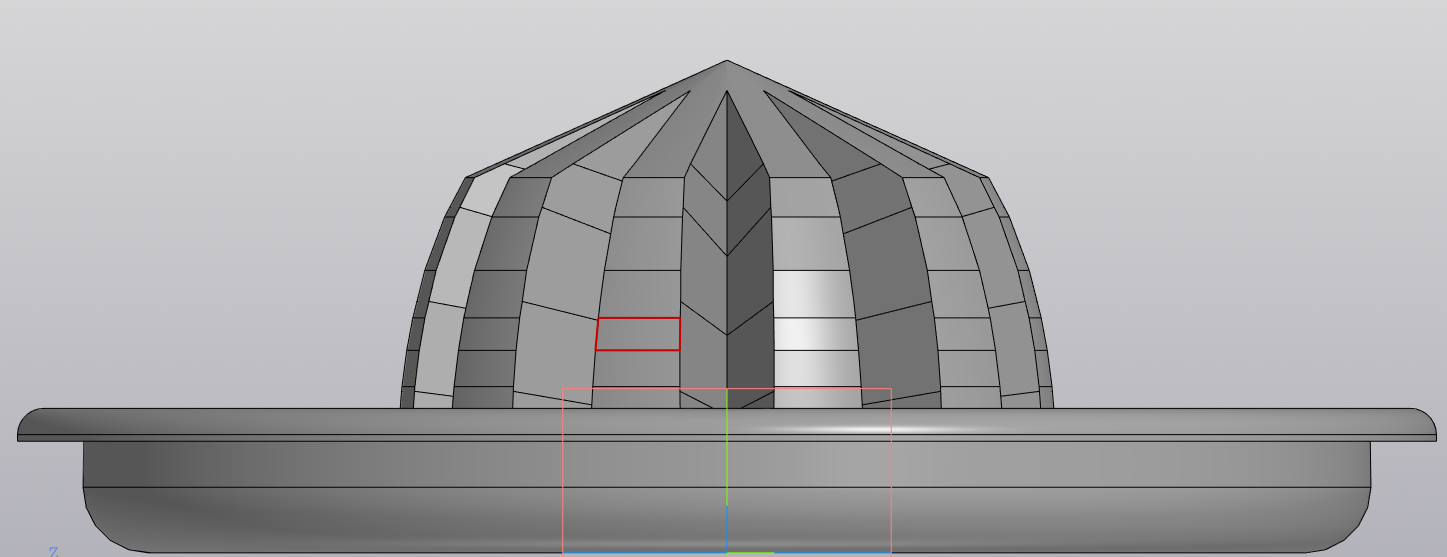


Рисунок 5.3 – Результат программы при средних значениях в плоскости ZY

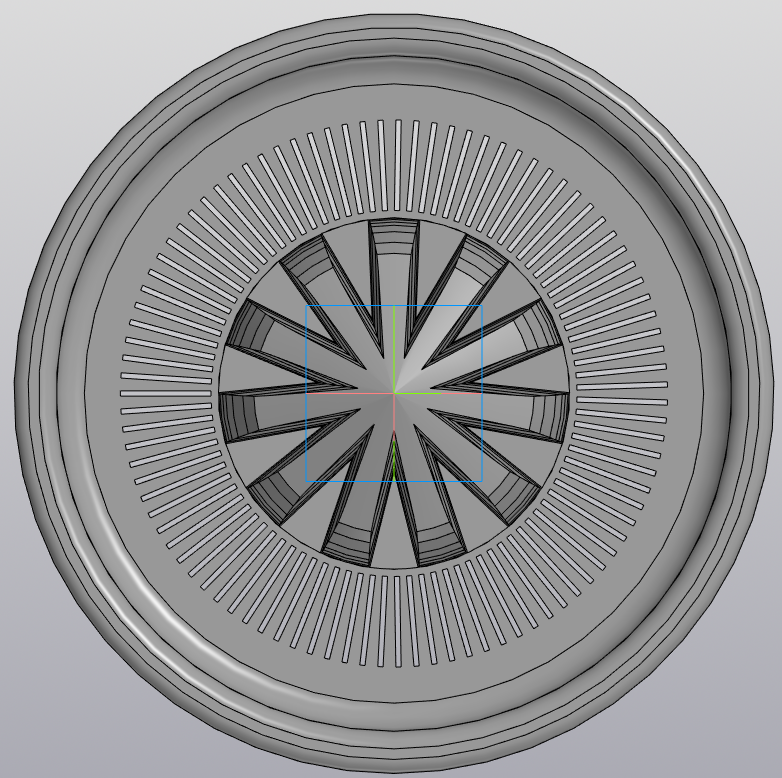


Рисунок 5.4 – Результат программы при средних значениях в плоскости XY

Максимальные значения параметров:

* диаметр тарелки равный 226 мм;
* диаметр кола равный 130 мм;
* высота кола равная 120 мм;
* количество зубцов кола равное 12 шт.;
* количество отверстий в тарелке равное 100 шт.
* длина отверстий в тарелке равная 35,5 мм.

Результаты представлены на рисунках 5.5 – 5.6.

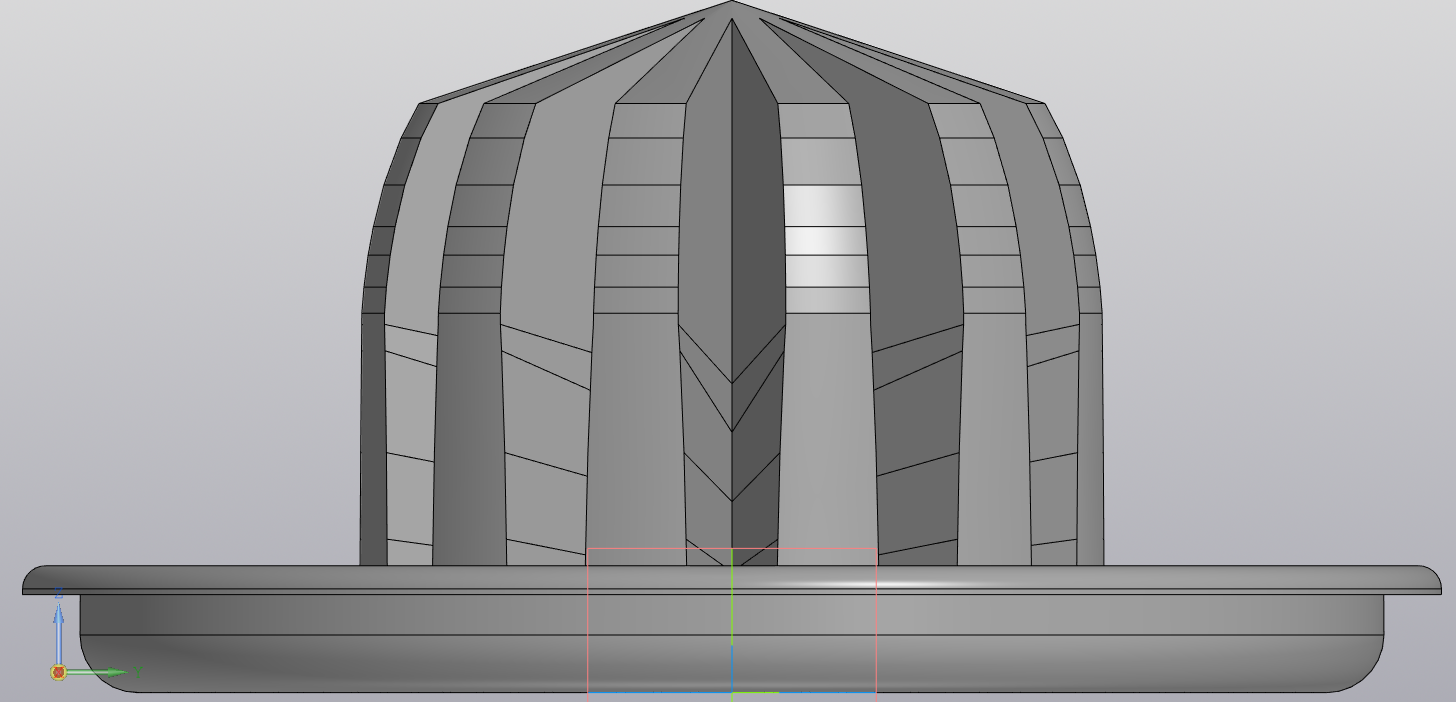


Рисунок 5.5 – Результат программы при максимальных значениях в плоскости ZY

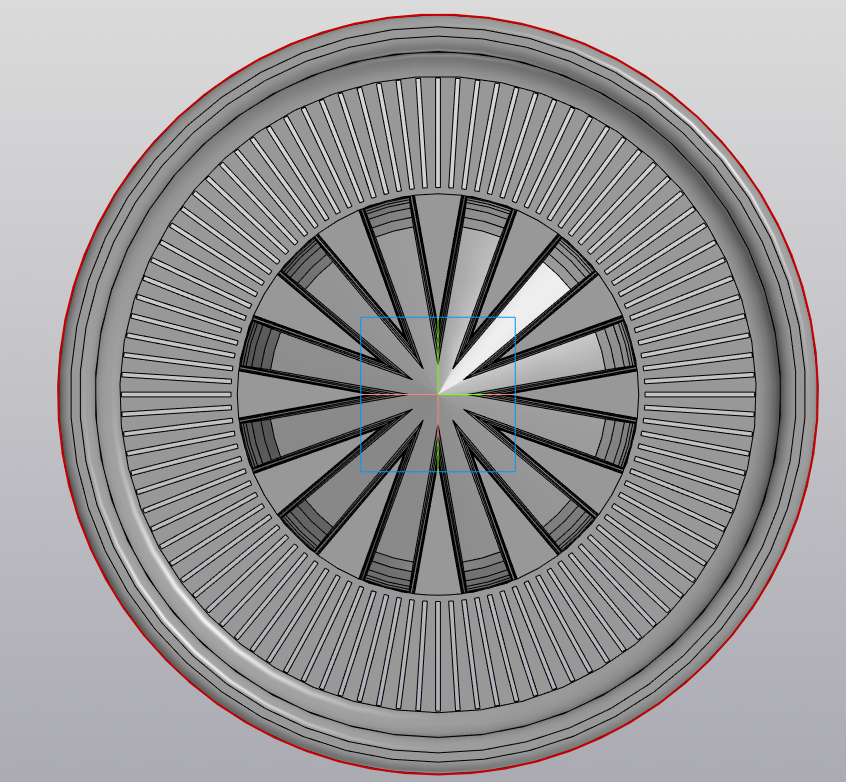


Рисунок 5.4 – Результат программы при максимальных значениях в плоскости XY

## **5.2 Модульное тестирование**

Модульное тестирование, иногда блочное тестирование или юнит-тестирование – процесс в программировании, позволяющий проверить на корректность отдельные модули исходного кода программы, наборы из одного или более программных модулей вместе с соответствующими управляющими данными, процедурами использования и обработки [12].

Для тестирования библиотеки используется NUnit 3.13.3 [5] и NUnit3TestAdapter 4.2.1 [6]. Вся логика программы покрыта тестами. На рисунке 5.5 представлен результат тестирования логики.

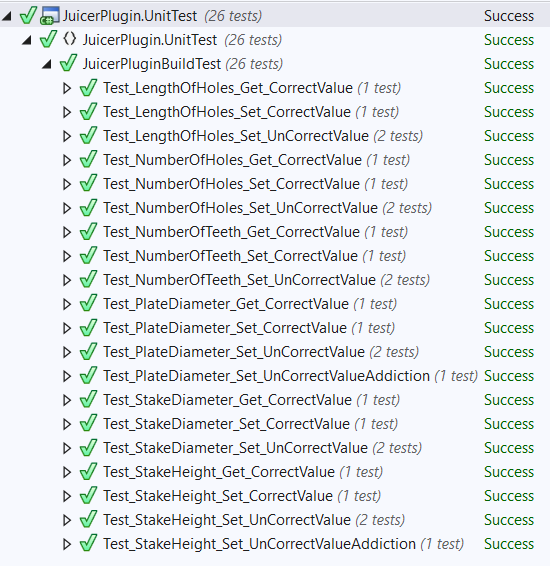


Рисунок 5.5 – Результат модульного тестирования

## **5.3 Нагрузочное тестирование**

Нагрузочное тестирование – подвид тестирования производительности, сбор показателей и определение производительности и времени отклика программно-технической системы или устройства в ответ на внешний запрос с целью установления соответствия требованиям, предъявляемым к данной системе [12].

Тестирование проводилось при следующих конфигурациях:

* процессор Intel(R) Pentium(R) CPU 6405U @ 2.40GHz 2.40 GHz;
* оперативная память 8 ГБ;
* графический процессор объемом памяти 4 ГБ.

Для тестирования будет зациклено построение модели со следующими параметрами:

* диаметр тарелки равный 166 мм;
* диаметр кола равный 70 мм;
* высота кола равная 60 мм;
* количество зубцов кола равное 10 шт.;
* количество отверстий в тарелке равное 90 шт.
* длина отверстий в тарелке равная 16 мм.

Было произведено 5 тестов. В каждом было построено 24 модели. В каждом построении выбраны средние значения времени и оперативной памяти.

На рисунке 5.6 представлен график зависимости памяти ОЗУ от построения модели, а на рисунке 5.7 представлен график зависимости времени от построения модели.

Рисунок 5.6 – График зависимости памяти ОЗУ от построения модели

Из графика 5.6 видно, что память изменяется линейно, но с некоторыми скачками вниз. Это связано с тем, что для регенерации памяти происходит остановка обмена данными, что понижает нагрузку на оперативную память. В конце график не сильно идет вверх, это связано с тем, что, когда на оперативную память идет сильная нагрузка, ОС начинает использовать файл подкачки для разгрузки оперативной памяти.

Рисунок 5.7 – График зависимости времени от построения модели

Из графика 5.7 видно, что время построения моделей изменяется линейно, но ближе к концу графика идет как экспонента. Это связано с увеличением нагрузки на оперативную память и центральный процессор с каждым построением модели. За счет этого время построения модели увеличивается.

# **Заключение**

В результате выполнения данной учебной работы были пройдены все стадии разработки программного обеспечения: выбор темы, составление технического задания, составление проекта системы, создание первой реализации и её тестирование, создание конечной библиотеки. Был изучен API приложения Компас-3D. Также были найдены аналоги разрабатываемого плагина. Были спроектированы UML диаграммы классов и было проведено функциональное, модульное и нагрузочное тестирования созданной библиотеки.

# **Список использованных источников**

1. Портал ТПУ. Учеба. НИОКР [Электронный ресурс] – Режим доступа: <https://portal.tpu.ru/SHARED/d/DMITRIEV/Ucheba/NIOKR/Tab/04_NIOKR_Avtomatizacija_proektirovanija.pptx> (дата обращения: 01.12.2022)
2. КОМПАС-3D [Электронный ресурс] – Режим доступа: <https://kompas.ru/kompas-3d/about/> (дата обращения: 05.10.2022)
3. Microsoft. Visual Studio [Электронный ресурс] – Режим доступа: <https://visualstudio.microsoft.com/ru/> (дата обращения: 01.12.2022)
4. Microsoft. Автономный установщик Microsoft платформа .NET Framework 4.7.2 для Windows [Электронный ресурс] – Режим доступа: <https://support.microsoft.com/ru-ru/topic/%D0%B0%D0%B2%D1%82%D0%BE%D0%BD%D0%BE%D0%BC%D0%BD%D1%8B%D0%B9-%D1%83%D1%81%D1%82%D0%B0%D0%BD%D0%BE%D0%B2%D1%89%D0%B8%D0%BA-microsoft-%D0%BF%D0%BB%D0%B0%D1%82%D1%84%D0%BE%D1%80%D0%BC%D0%B0-net-framework-4-7-2-%D0%B4%D0%BB%D1%8F-windows-05a72734-2127-a15d-50cf-daf56d5faec2> (дата обращения: 30.11.2022)
5. Nuget. NUnit [Электронный ресурс] – Режим доступа: <https://www.nuget.org/packages/NUnit/> (дата обращения: 05.12.2022)
6. Nuget. NUnit3TestAdapter 4.3.1 [Электронный ресурс] – Режим доступа: <https://www.nuget.org/packages/nunit3testadapter/> (дата обращения: 05.11.2022)
7. Компас-3D [Электронный ресурс] – Режим доступа: <https://kompas.ru/kompas-educational/about/> (дата обращения: 05.11.2022)
8. Macro Honeycomb [Электронный ресурс] – Режим доступа: <https://wiki.freecadweb.org/Macro_Honeycomb> (дата обращения: 5.10.2022)
9. Фаулер М. UML. Основы. Краткое руководство по стандартному языку объективного моделирования. М.: Символ-Плюс, 2011, 192 с.
10. Enterprise Architect [Электронный ресурс] – Режим доступа: [https://sparxsystems.com/](https://sparxsystems.com/%20) (дата обращения: 05.11.2022)
11. Горяинов А.Е., Калентьев А.А. ТЕХНОЛОГИЯ РАЗРАБОТКИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ // Методические указания к самостоятельной и лабораторным работам. 2018. 184 с.
12. Куликов С. Тестирование программного обеспечения. Базовый курс. 3-е издание, 301 с.