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

\_\_\_\_\_\_\_\_\_\_\_\_\_Р. А. Рогозин

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

Руководитель

К.т.н., доцент каф. КСУП

\_\_\_\_\_\_\_\_\_\_А.А. Калентьев

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

Томск 2020

**РЕФЕРАТ**

Пояснительная записка 25 с., 16 рис., 16 источника, 1 прил., 1 таблица.

SOLIDWORKS 2020, ПЛАГИН, ШКАТУЛКА, АВТОМАТИЗИРОВАННОЕ ПРОЕКТИРОВАНИЕ, САПР.

Целью данной работы является разработка плагина, автоматизирующего построение модели «Шкатулка» для системы автоматизированного проектирования SOLIDWORKS 2020 [1] с помощью интегрированной среды разработки Visual Studio 2019 Сommunity [2].

В процессе работы изучены основные параметры формы для выпечки и основные функции API [3] системы автоматизированного проектирования SOLIDWORKS 2020.

В результате работы был создан плагин, автоматизирующий построение модели «Шкатулка».

Отчет по пояснительной записке выполнен в текстовом редакторе Microsoft Word 2016 [4].

**Оглавление**

[1 Введение 4](#_Toc39421319)

[2 Постановка и анализ задачи 5](#_Toc39421320)

[2.1 Описание предмета проектирования 5](#_Toc39421321)

[2.2 Выбор инструментов и средств реализации 6](#_Toc39421322)

[2.3 Назначение плагина 7](#_Toc39421323)

[2.4 Описание аналогов разрабатываемого продукта 7](#_Toc39421324)

[3 Описание реализации 10](#_Toc39421325)

[3.1 Описание технических и функциональных аспектов проекта 10](#_Toc39421326)

[3.2 Диаграмма вариантов использования (Use Cases) 10](#_Toc39421327)

[3.3 Диаграмма классов 11](#_Toc39421328)

[4 Описание программы для пользователя 12](#_Toc39421329)

[5 Тестирование программы 15](#_Toc39421330)

[5.1 Функциональное тестирование 15](#_Toc39421331)

[5.2 Модульное тестирование 16](#_Toc39421332)

[5.3 Нагрузочное тестирование 18](#_Toc39421333)

[Заключение 21](#_Toc39421334)

[Список использованных источников 22](#_Toc39421335)

[Приложение А 24](#_Toc39421336)

# 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 «Шкатулка» для системы автоматизированного проектирования SOLIDWORKS 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 2020, строит модель «Шкатулка». Необходимо чтобы плагин позволял задавать параметры по умолчанию, а также изменять входные параметры формы такие как длина, ширина и высота шкатулки (внешняя и внутренняя часть коробки и крышка).

# 2.1 Описание предмета проектирования

Шкатулка — маленькая коробка или ящик обычно, но не всегда, в форме прямоугольного параллелепипеда, используемая для хранения драгоценностей, денег, бумаг и других мелких, но обычно ценных предметов.

Параметры формы для выпечки:

* ширина основания X (от 100 мм до 200 мм);
* длина основания Y (от 100 мм до 200 мм);
* высота основания Z (от 50 мм до 150 мм);
* ширина внутренней части основания SX (от 50 до X-5 мм);
* длина внутренней части основания SY (от 50 до Y-5 мм);
* высота внутренней части основания SZ (от 30 до Z-5 мм);
* ширина крышки СX (от SX+1 мм до 230 мм);
* длина крышки СY (от SY+1 мм до 230 мм);
* высота крышки СZ (от 10 мм до 30 мм).

Пример проектируемого изделия приведен ниже, на рисунке 2.1.

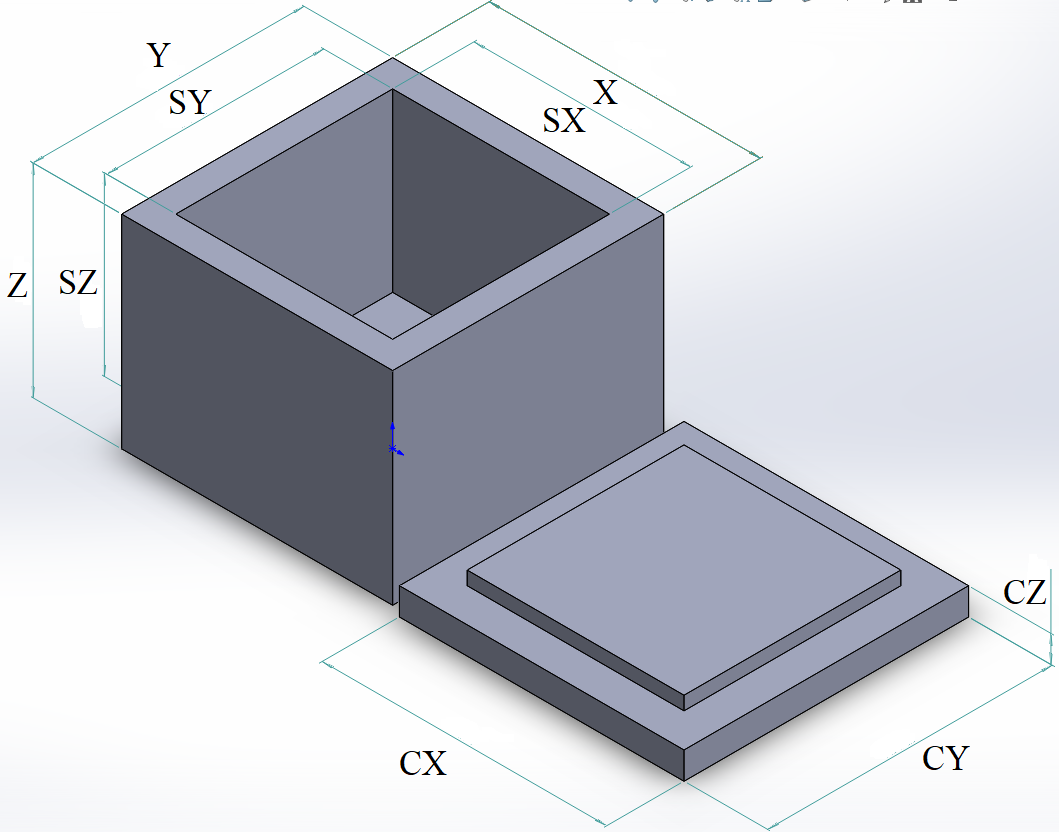


Рисунок 2.1 – Предмет изделия

# 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 4.8 [6], библиотеки «SolidWorks.Interop. sldworks» [7] версии 27.3 для основных операций в системе SOLIDWORKS 2020.

Инструментом тестирования и создания модульных тестов был выбран обозреватель тестов расширения для Visual Studio – ReSharper [8] с тестовым фреймворком NUnit [9] версии 3.12, процентное соотношение покрытия библиотеки тестами определено с помощью расширения для Visual Studio – dotCover [10].

Для реализации пользовательского интерфейса использовалась система для построения настольные и мобильные приложения Windows – Windows Forms (WinForms) [11].

# 2.3 Назначение плагина

Назначение разрабатываемого плагина обусловлено быстрым моделированием шкатулок разных типов. Благодаря данному расширению, столяры могут наглядно рассмотреть спроектированную модель, при необходимости перестроить под необходимые им параметры.

# 2.4 Описание аналогов разрабатываемого продукта

Макрос для Corel Draw «Шкатулка», предназначен для упрощения создания чертежей всевозможных изделий, изготавливаемых с помощью ручного выпиливания из листового материала (прежде всего фанеры) или с помощью станков ЧПУ (числового программного управления, на английском языке - CNC) фрезерной или лазерной резки. Под изделиями, для текущей версии макроса подразумевается создание простых ящиков, и всевозможных видом декоративных шкатулок, и ряда изделий не прямоугольной формы. На данный момент макрос работает с 32 и 64 битными версиями Corel Draw версии Х4 и выше. За основу разработки (в качестве идеи) лег макрос «Шипоящик». После внесения в него правок, было принято решение, что нужно писать новый макрос, так как исходный был написан не по модульной схеме, и его было очень трудно сопровождать и расширять его функционал. Так, в конце 2015 года была начата работа над макросом «Шкатулка». Спустя 4 месяца был готов базовый функционал, позволяющий создавать чертежи для простых ящиков, который потом был выделен в отдельный бесплатный макрос «BoxConctructor». В настоящий момент макрос существенно переработан и расширен в плане функционала и распространяется на платной основе [23].

На текущий момент, макрос поддерживает следующие разновидности

автоматического проектирования:

* Простые ящики;
* Простые шкатулки;
* Прорезные шкатулки;
* Прорезная шкатулка с накладками;
* Многогранные шкатулки;
* Овальные шкатулки;
* Наборные шкатулки;
* Спиральные корзинки;
* Шкатулки и прочие виды изделий с «гибкой» стенкой;
* Изделия произвольной формы;
* Чайный домик;
* Штучные изделия;

На рисунке 2.2 и 2.3 представлены примеры интерфейса плагина «Шкатулка»:



Рисунок 2.2 – Выбор типа изделия

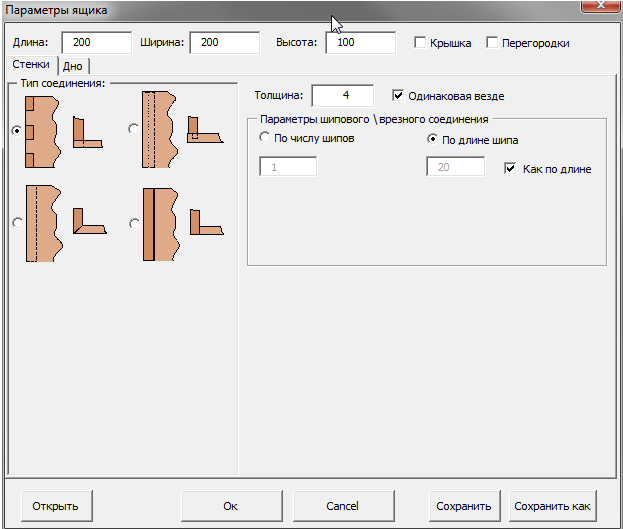


Рисунок 2.3 – Параметры ящика

# 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 [15].

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 использования и диаграмма классов.

# 3.2 Диаграмма вариантов использования (Use Cases)

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

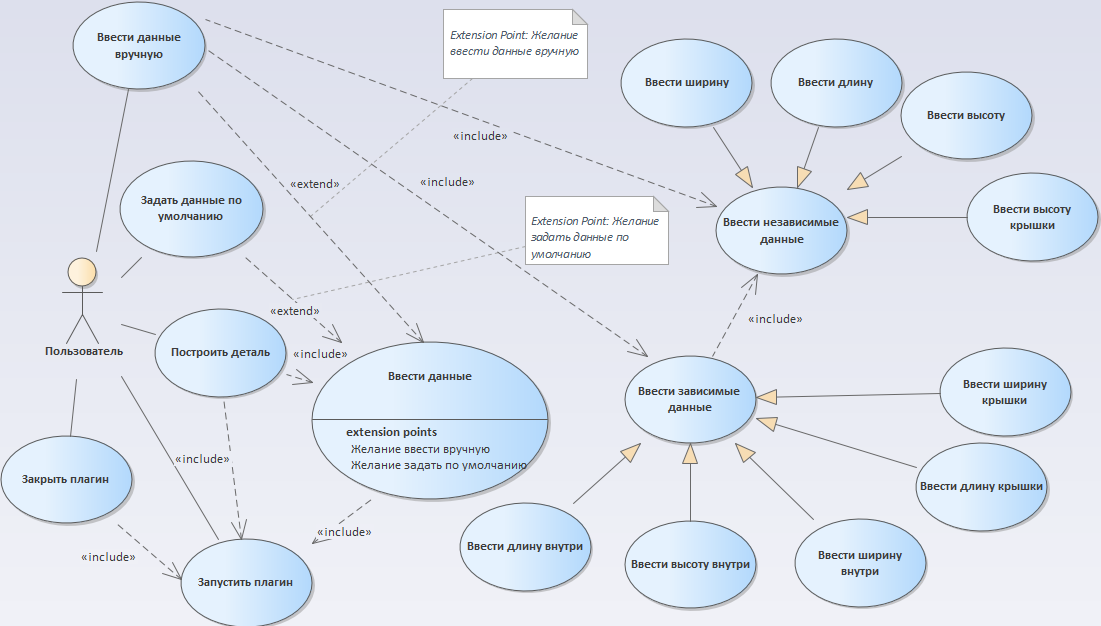


Рисунок 3.1– Диаграмма вариантов использования

# 3.3 Диаграмма классов

В данном проекте спроектированы следующие классы.

«Builder» содержит в себе методы рисования 3D модели в программе «SolidWorks 2020», принимая класс «Casket», содержащий пользовательские настройки фигуры.

«NumverExtensions» переводит входные данные в миллиметры.

«RangedValue» описывает поля класса «Casket».

Интерфейс «ISolidWorksCommander» включает команды для управления программой «SolidWorks 2020», его наследует и реализует «SolidWorksCommander».

«ValueOutOfRangeException» наследует класс исключений «Exception», отслеживает нарушение границ полей входных параметров.

Классы «Walls» описывает высоту стен и точки вдоль вектора перечисления «Vector», «SizeRange» содержит диапазон размеров векторов.

«Main» является графическим интерфейсом.

Ниже, на рисунке 3.2 представлена диаграмма классов.

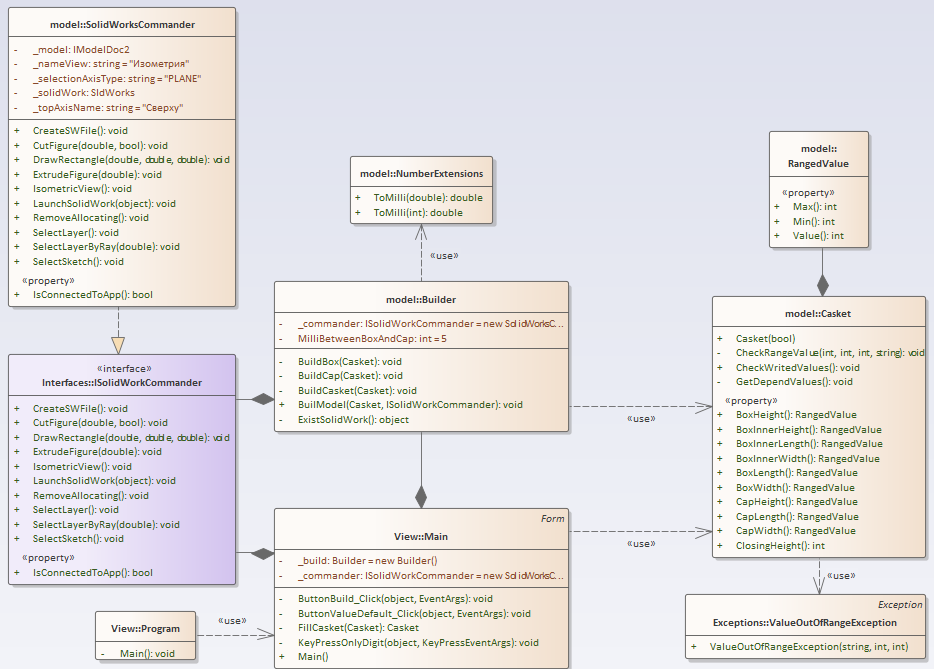
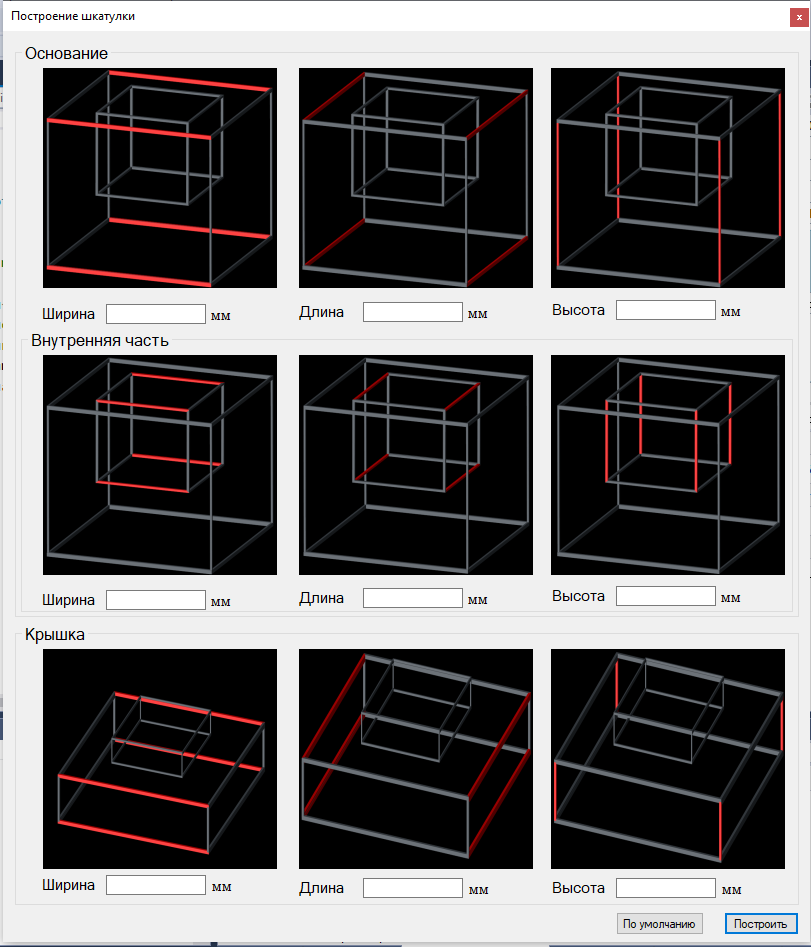


Рисунок 3.2 – Диаграмма классов

# 4 Описание программы для пользователя

Плагин состоит из диалогового окна, который имеет меню, блоки «Основание», внутри которого находится «Внутренняя часть» и «Крышка». При запуске имеет следующий вид (рисунок 4.1).

  
Рисунок 4.1 – Меню плагина

При старте плагина над полями видны графические элементы, показывающие за какую часть шкатулки, отвечает параметр. Если ввести неверный параметры, после нажатия кнопки «Построить», высветится окно с описание неправильного заполненного поля (рисунок 4.2).

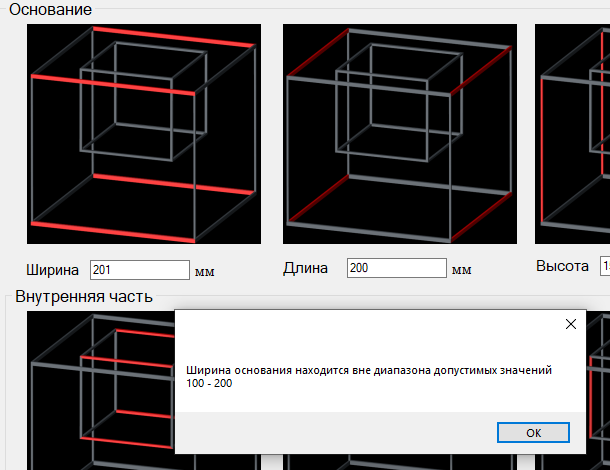
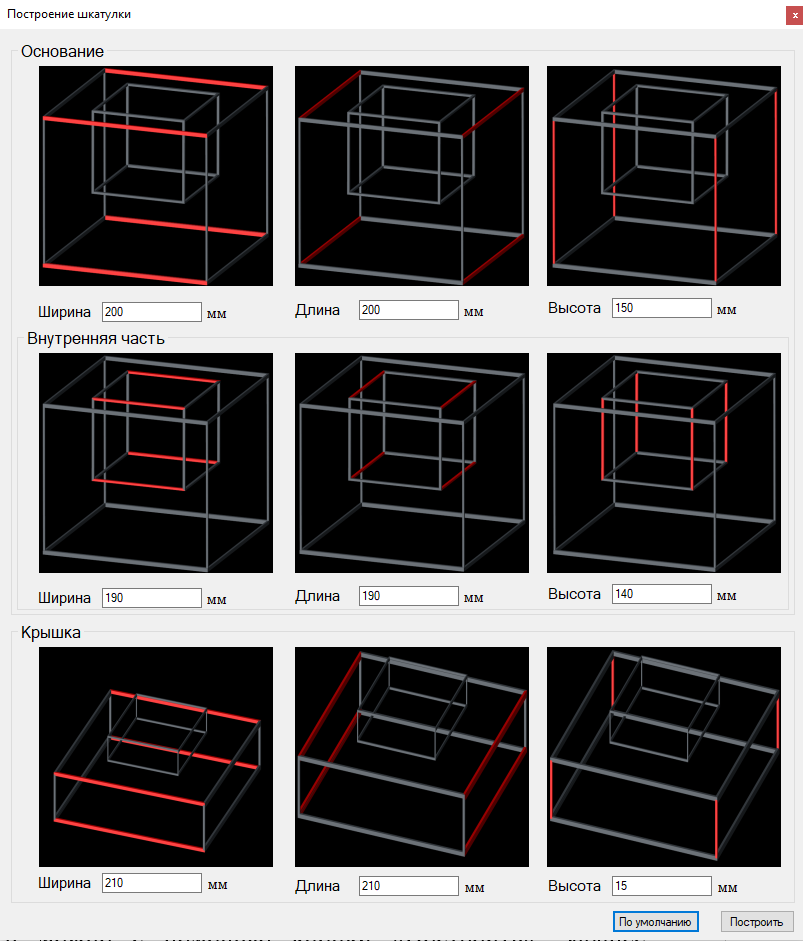
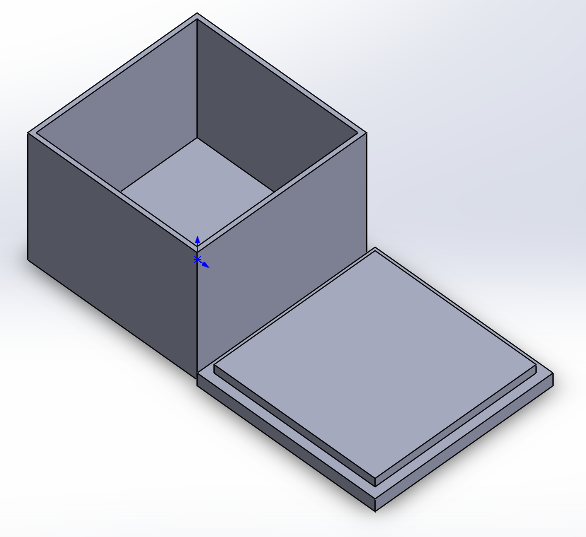


Рисунок 4.2 – Ошибка ввода параметра

Нажав, на «По умолчанию», форма с полями для ввода данных примет следующий вид (рисунок 4.3).

  
Рисунок 4.3 – Окно плагина с параметрами по умолчанию

После ввода необходимых параметров, построить деталь в САПР SOLIDWORKS 2020 можно с помощью кнопки «Построить». Например, фигура, построенная по заданным параметрам по умолчанию, выглядит следующим образом (рисунок 4.4).

  
Рисунок 4.4 – Модель, построенная по заданным параметрам по умолчанию

# 5 Тестирование программы

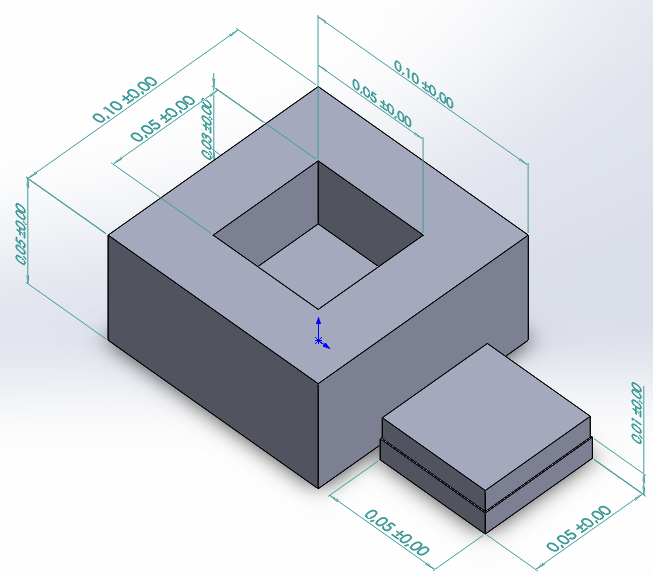
Тестирование позволяет убедиться в работоспособности программы, выявлять ошибки при изменении какого-либо функционала.

# 5.1 Функциональное тестирование

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

Проведено тестирование максимальных и минимальных параметров модели.

Ниже на рисунке 5.1 представлена проверка размеров модели с минимальным введенными параметрами (ширина 100 мм, длина 100 мм, высота 50 мм, ширина внутри 50 мм, длина внутри 50 мм, высота внутри 30 мм, ширина крышки 51 мм, длина крышки 51 мм и высота крышки 10 мм).

  
Рисунок 5.1 – Модель с минимальными веденными параметрами

Ниже на рисунке 5.2 представлена проверка размеров модели с максимальными введенными параметрами (ширина 200 мм, длина 200 мм, высота 150 мм, ширина внутри 195 мм, длина внутри 195 мм, высота внутри 145 мм, ширина крышки 230 мм, длина крышки 230 мм и высота крышки 30 мм).

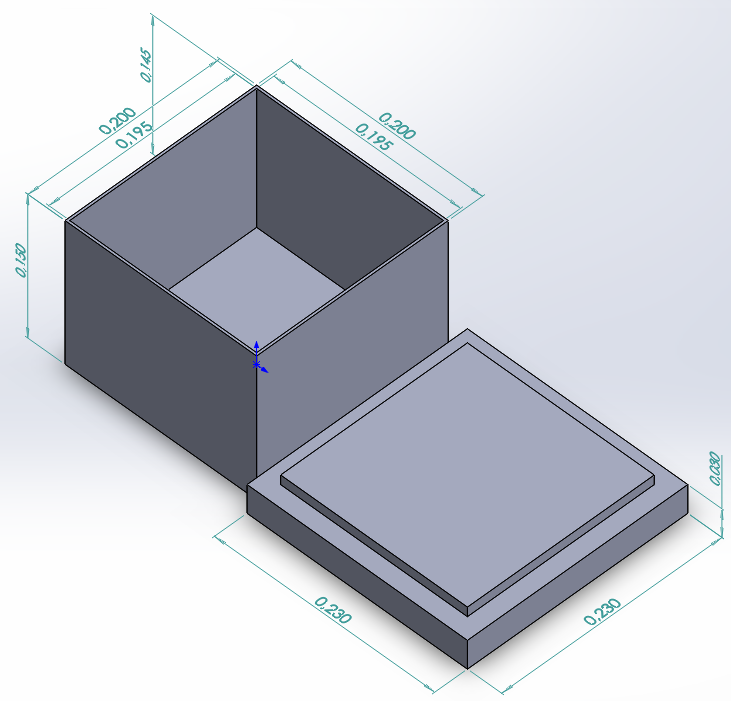
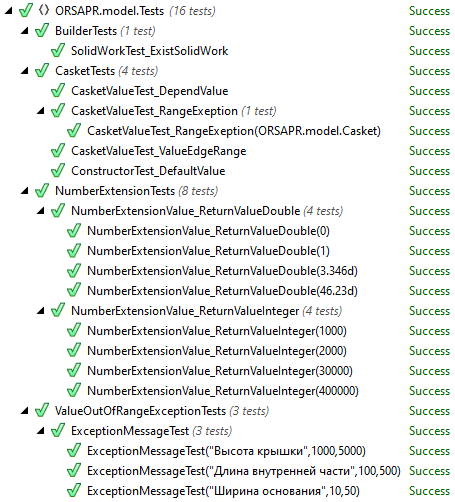


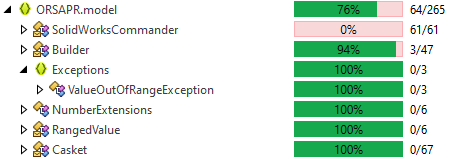
Рисунок 5.2 - Модель с максимально веденными параметрами

# 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 [11], проверялись открытые поля и методы. На рисунке 5.3 представлено тестирование классов «Builder», «Casket» и «NumberExtension», «ValueOutOfRangeException». Класс «SolidWorkCommander» не подвергался тестированию, поскольку хранить в себе API SolidWorka (описание полей и методов находится в приложении А).

  
Рисунок 5.3 – Тестовые классы «Builder», «Casket», «NumberExtension», «ValueOutOfRangeException»

Ниже, на рисунке 5.4 представлено процентное соотношение покрытия библиотеки Model тестами, оно определено с помощью расширения для Visual Studio – dotCover.

  
Рисунок 5.4 – Покрытие библиотеки тестами

# 5.3 Нагрузочное тестирование

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

Процессор: AMD FX-8320E Eight-Core Processor 3.20 GHz

Видеоадаптер: AMD Radeon R9 200 Series

ОЗУ: 12 GB DDR 1333 MHz

На рисунке 5.5 для проведения нагрузочного тестирования был добавлен секундомер («Stopwatch»), который засекал время от начала построения, с каждым успешным построением-перестроением фигуры («\_build.BuilModel(casket, \_commander)») производилась запись результатов в текстовый файл «log.txt».



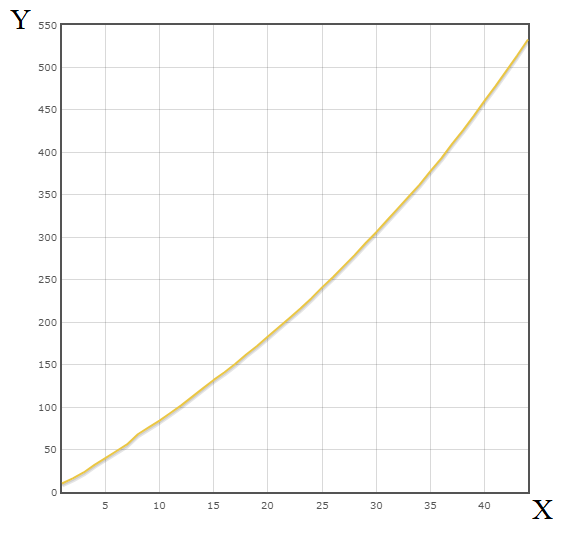
Рисунок 5.5 – Зацикливание перестроения фигуры

На представленных графиках в текущей главе ось «X» – время в секундах, ось «Y» – количество построенных-перестроенных деталей. На протяжении всех тестов (продолжительностью до вылета SolidWorks) общая загруженность процессора была в пределах 16 процентов, потребление ОЗУ плагином прямолинейное от 15мб до 31мб.

На рисунке 5.6 представлено тестирование зацикленного перестроения фигуры со следующими максимальными параметрами:

* Ширина 200 мм;
* Длина 200 мм;
* Высота 150 мм;
* Ширина внутри 195 мм;
* Длина внутри 195 мм;
* Высота внутри 145 мм;
* Ширина крышки 230 мм;
* Длина крышки 230 мм;
* Высота крышки 30 мм.

Заметно, что на графике (рисунок 5.6) на 200 секунде производительность падает.

  
Рисунок 5.6 – График зависимости времени от количества построенных фигур с максимальными параметрами

На рисунке 5.7 представлено тестирование зацикленного перестроения фигуры со следующими минимальными параметрами:

* Ширина 100 мм;
* Длина 100 мм;
* Высота 50 мм;
* Ширина внутри 50 мм;
* Длина внутри 50 мм;
* Высота внутри 30 мм;
* Ширина крышки 51 мм;
* Длина крышки 51 мм;
* Высота крышки 10 мм.

На графике (рисунок 5.7) на 200 секунде производительность падает.

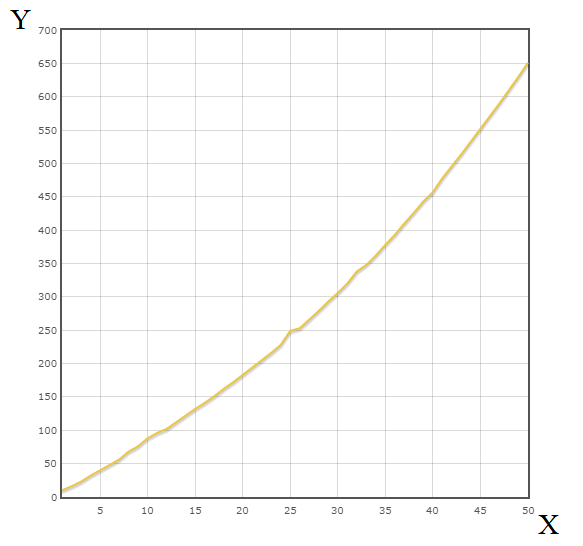


Рисунок 5.7 – График зависимости времени от количества построенных фигур с минимальными параметрами

# 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 моделей «Шкатулка» в САПР SOLIDWORKS 2020.

# Список использованных источников

1. SOLIDWORKS 2020 [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 2016 [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. Шкатулка [Электронный ресурс]. – URL: <https://www.giftman.ru/klientam/statyi-o-tovarah/istoriya-shkatulki/> (дата обращения: 20.04.2020);
7. 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);
8. ReSharper. [Электронный ресурс]. – URL: <https://www.jetbrains.com/ru-ru/resharper/> (дата обращения: 20.04.2020);
9. NUnit [Электронный ресурс]. – URL: <https://nunit.org/> (дата обращения: 20.04.2020);
10. dotCover. Инструмент для запуска юнит-тестов .NET и оценки покрытия кода [Электронный ресурс]. – URL: <https://www.jetbrains.com/ru-ru/dotcover/> (дата обращения: 20.04.2020);
11. Начало работы с WinForms [Электронный ресурс]. – URL: <https://docs.microsoft.com/ru-ru/dotnet/framework/winforms/> (дата обращения: 25.04.2020);
12. Макрос «Конструктор Шкатулок». Руководство пользователя [Электронный ресурс]. – URL: <http://cncsoft.starfair.ru/wp-content/uploads/2018/Руководство-пользователя-макроса-шкатулки.pdf> (дата обращения 12.03.2020).
13. Плагин PDF [Электронный ресурс]. – URL: <http://gkmsoft.ru/allcatalog/pdf2dkompas_plugin/> (дата обращения: 20.04.2020);
14. 3D PDF [Электронный ресурс]. – URL: <http://sapr-journal.ru/novosti/eksport-iz-kompas-3d-v-formate-3d-pdf/> (дата обращения: 20.04.2020);
15. UML [Электронный ресурс]. – URL: <http://www.uml.org/> (дата обращения: 13.04.2020);
16. Нагрузочное тестирование: с чего начать и куда смотреть [Электронный ресурс]. – URL: <https://habr.com/ru/company/jugru/blog/329174/> (дата обращения: 20.04.2020);

# Приложение А

(Справочное)

Таблица, представленная ниже, является описанием вспомогательных классов для проведения модульного тестирования. Каждый метод в классе является тестовым случаем. В каждой таблице описаны поля, методы класса и их описание.

Таблица 1 – Описание классов тестирования

|  |  |  |
| --- | --- | --- |
| Название класса | Наименование | Описание |
| BuilderTests | \_commander: ISolidWorkCommander | API Команды к программе SOLIDWORKS |
| \_builder: Builder | Построитель фигуры |
| \_casket: Casket | Шкатулка |
| InitializeForTest() | Инициализировать каждый тест |
| SolidWorkTest\_ExistSolidWork() | Проверка на существования SolidWork 20202 на PC |
| CasketTests | \_casket: Casket | Шкатулка |
| \_casketDefault: Casket | Шкатулка с параметрами по умолчанию |
| InitializeForTest() | Инициализировать каждый тест |
| ConstructorTest\_DefaultValue() | Проверка значений шкатулки по умолчанию |
| CasketValueTest\_DependValue() | Проверка зависимых значений шкатулки |
| CasketValueTest\_ValueEdgeRange () | Проверка границ диапазонов шкатулки |
| CasketValueTest\_RangeExeption (Casket casket) | Проверка на выбрасования исключения ValueOutOfRangeException |

Продолжение таблицы А.1

|  |  |  |
| --- | --- | --- |
| NumberExtensionTests | NumberExtensionValue\_ReturnValueInteger(int value) | Перевод метров в миллиметры для целых чисел |
| NumberExtensionValue\_ReturnValueDouble(double value) | Перевод метров в миллиметры для не целых чисел |
| ValueOutOfRangeExceptionTests | ExceptionMessageTest(string nameValue, int min, int max) | Проверка на вывод корректного сообщения исключения ValueOutOfRangeException |