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 лабораторному проекту

по дисциплине «Основы разработки САПР»

Выполнил:   
Студент гр. 586-2

Ю.А. Новичкова

Проверил:

к.т.н., доцент каф. КСУП

\_\_\_\_\_\_\_\_\_А.А. Калентьев

« » 2020 г.

Томск 2020

**Оглавление**

[1 Введение 4](#_Toc40557531)

[2 Постановка и анализ задачи 5](#_Toc40557532)

[2.1 Описание предмета проектирования 5](#_Toc40557533)

[2.2 Выбор инструментов и средств реализации 6](#_Toc40557534)

[2.3 Назначение плагина 7](#_Toc40557535)

[2.4 Описание аналогов разрабатываемого продукта 7](#_Toc40557536)

[2.4.1 САПР GRAFIS 7](#_Toc40557537)

[2.4.2 САПР Julivi 8](#_Toc40557538)

[2.4.3 САПР Грация 9](#_Toc40557539)

[2.4.4 САПР Леко 10](#_Toc40557540)

[2.4.5 RedCafe 10](#_Toc40557541)

[3 Описание реализации 12](#_Toc40557542)

[3.1 Диаграммы USE CASE 12](#_Toc40557543)

[3.2 Диаграмма классов 13](#_Toc40557544)

[4 Описание программы для пользователя 17](#_Toc40557545)

[5 Тестирование программы 20](#_Toc40557546)

[5.1 Функциональное тестирование 20](#_Toc40557547)

[5.2 Модульное тестирование 23](#_Toc40557548)

[5.3 Нагрузочное тестирование 24](#_Toc40557549)

[Заключение 27](#_Toc40557550)

[Список литературы 28](#_Toc40557551)

[Приложение А 30](#_Toc40557552)

# 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.

Практическая реализация методов и идей автоматизированного моделирования происходит в рамках систем автоматизированного проектирования (САПР). Однако мало создать высокопроизводительные современные САПР. В рамках современного «компьютеризированного» общества инженер любой специальности, занимающийся разработкой технических устройств, должен уметь использовать средства автоматизированного проектирования. Это позволяет повысить эффективность моделирования, улучшить его качество, снизить материальные затраты и уменьшить число разработчиков.

Таким образом, целью данной работы является разработка плагина, автоматизирующего построение модели ремня, для системы автоматизированного проектирования «Компас-3D» v 16 [1], с помощью интегрированной среды разработки Visual Studio Сommunity 2019 [2].

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 «Ремень». Более того, требовалось, чтоб плагин позволял изменять входные параметры в соответствии с требованиями пользователей, а именно: длина, ширина и толщина ленты, диаметр отверстий и расстояние между ними, диаметр язычка бляшки, длина и ширина бляшки, а также ее форма: прямоугольная или треугольная.

## 2.1 Описание предмета проектирования

Ремень [3] — элемент одежды, представляющий собой длинную кожаную, тканую, или составную металлическую ленту, снабженную металлической или (реже) пластиковой застежкой (пряжкой) с передней стороны. Модель ремня представлена на рисунке 2.1.

Параметры ремня:

* длина ремня A (от 800 до 1200 мм);
* ширина ремня D (от 20 до 40 мм);
* толщина ремня F (от 3 до 4 мм);
* диаметр отверстий E (4 мм);
* расстояние между отверстиями C (от 15 до 25 мм);
* ширина бляшки L (от 20 до 40 мм);
* длина бляшки B (от 40 до 50 мм);
* диаметр язычка бляшки K (от 3 до 4 мм).

Зависимые параметры ремня:

* Ширина язычка бляшки: ;
* Ширина бляшки:

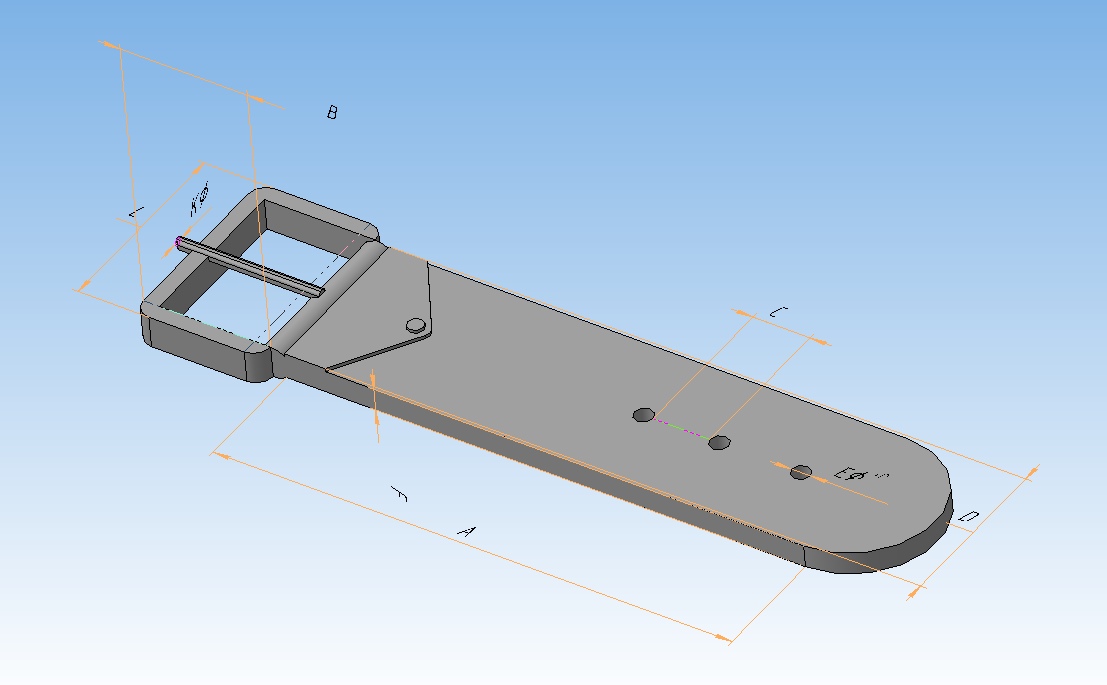


Рисунок 2.1 – Модель ремня в САПР «Компас-3D»

Стоит отметить, что ремни в свою очередь делятся на мужские, женские, детские и для пионерской формы. Значения основных параметров для ремня каждого типа представлены в ГОСТ 28754-90 [4] и в ГОСТ 18176-79 [5].

## 2.2 Выбор инструментов и средств реализации

В связи с требованием технического задания программа выполнена на языке C# в среде Microsoft Visual Studio 2019, с использованием .NET Framework 4.7.0, для системы КОМПАС 3D V16. Инструментом тестирования и создания модульных тестов был выбран стандартный обозреватель тестов среды Microsoft Visual Studio 2019 [2] с тестовым фреймворком NUnit версии 3.12.0.

Для реализации пользовательского интерфейса использовался WinForm.

Взаимодействие плагина с системой КОМПАС 3D [1] осуществляется посредством интерфейсов, называемых API. В КОМПАС 3D на данный момент существует API двух версий API 5 [6] и API 7 [6]. Явных преимуществ между версиями нет, поскольку обе версии реализуют различные функции системы и взаимно дополняют друг друга. Для выполнения лабораторных была выбрана версия API 5, так как для полноценной реализации плагина «Музыкальная колонка» достаточно методов и свойств интерфейсов API 5.

## 2.3 Назначение плагина

Назначение разрабатываемого плагина обусловлено быстрым моделированием ремней взрослых моделей. Благодаря данному расширению, люди, занимающиеся ручным изготовлением ремней, смогут наглядно посмотреть результат работы и при необходимости изменить некоторые параметры без расхода дорогостоящего материала.

## 2.4 Описание аналогов разрабатываемого продукта

### 2.4.1 [САПР GRAFIS](http://www.cadrus.ru/cad/advantages/)

Grafis [7] ‒ разработка компании Cadrus, которая специализируется на программных продуктах для швейного производства. У программы доступная система навигации, а также есть возможность интегрировать ранее разработанные бумажные лекала. САПР Grafis включает в себя варианты основ изделий: юбки, брюки, мужские и женские плечевые основы, трикотажные основы, детские, бельевые основы, джинсовые изделия, основы спецодежды и головные уборы. Программа умеет выполнять автоматическую градацию по размерным признакам, задавать припуски изделий и делать ручную или автоматическую раскладку деталей кроя. На рисунке 2.2 представлен интерфейс САПР Grafis.

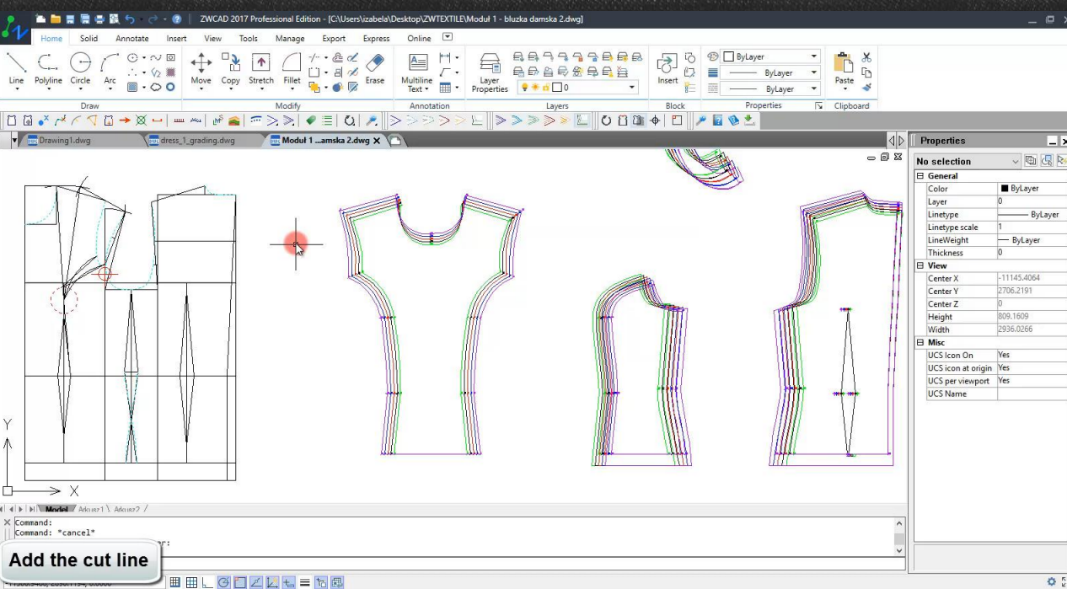


Рисунок 2.2 – Интерфейс САПР Grafis

### 2.4.2 [САПР Julivi](http://julivi.com/)

Компания «Сапрлегпром» предлагает сразу несколько программных решений для швейных производств. В систему Julivi [8], разработанную компанией, входят 2D и 3D программы проектирования одежды.

Программа «Дизайн» умеет строить базовые конструкции с нуля по любой методике (Мюллер, ЕМКО СЭВ и т.д.) в одном или нескольких размерах. Перечень возможностей в базовой комплектации включает построение чертежа конструкции, наращивание припусков и оформление срезов, автоматическую градацию, а также возможность разработать конструкцию на индивидуальную фигуру.

Другой комплекс программ, «Конструктор», подойдет для построения лекал одежды, обуви, головных уборов, мягкой мебели на швейном производстве. Позволяет работать от базовой или типовой конструкции до запуска в производство, размножает лекала, моделирует, строит подкладки и клеевые. Программа адаптирована для конструирования изделий из текстиля, трикотажа, кожи, меха. На рисунке 2.3 представлен интерфейс САПР Julivi.

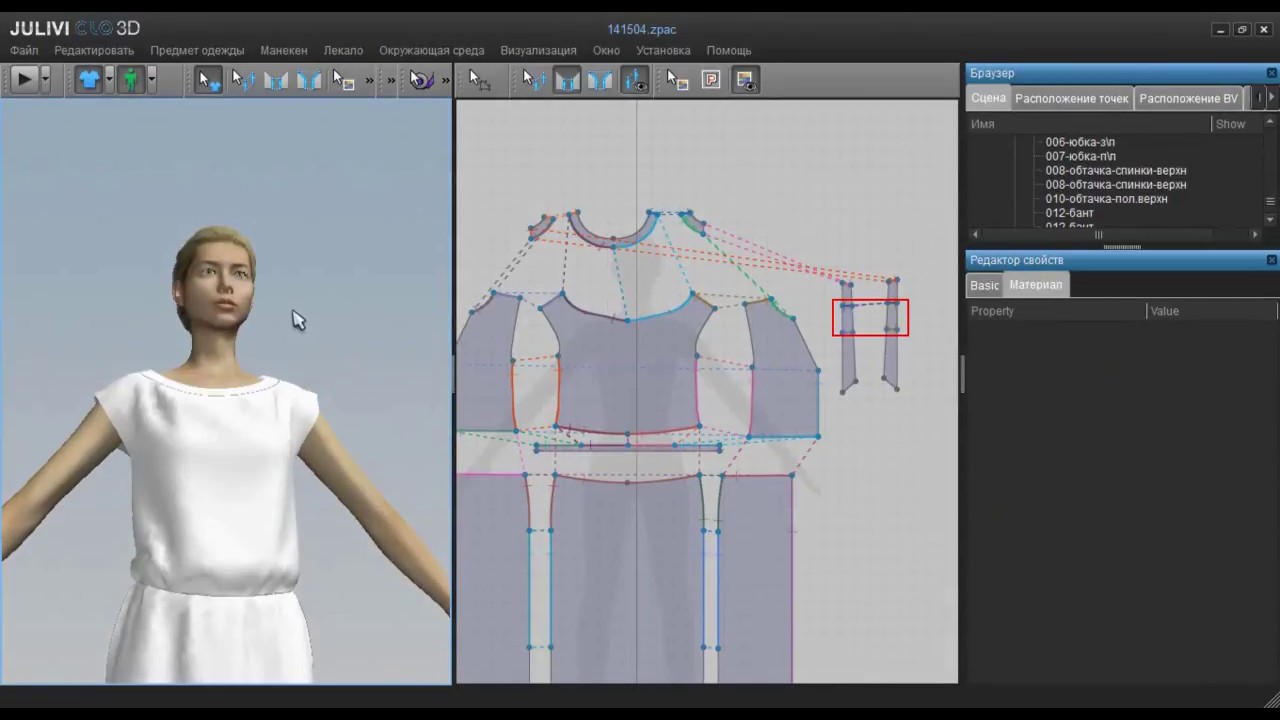


Рисунок 2.3 – Интерфейс САПР Julivi

### 2.4.3 [САПР Грация](http://www.saprgrazia.com/)

САПР Грация [9] ‒ это несколько комплектов, рассчитанных на разные типы производства. «Комплект для предприятий» состоит из систем дизайна, конструирования и моделирования и раскладки лекал. Строит оптимальные для экономии материалов схемы раскладки деталей кроя, а также высчитывает раскрой по настилам материала. Кроме нее есть версии «Комплект для ателье», «Комплект для фрилансеров» и «Комплект для студентов». На рисунке 2.4 представлен интерфейс САПР Грация.

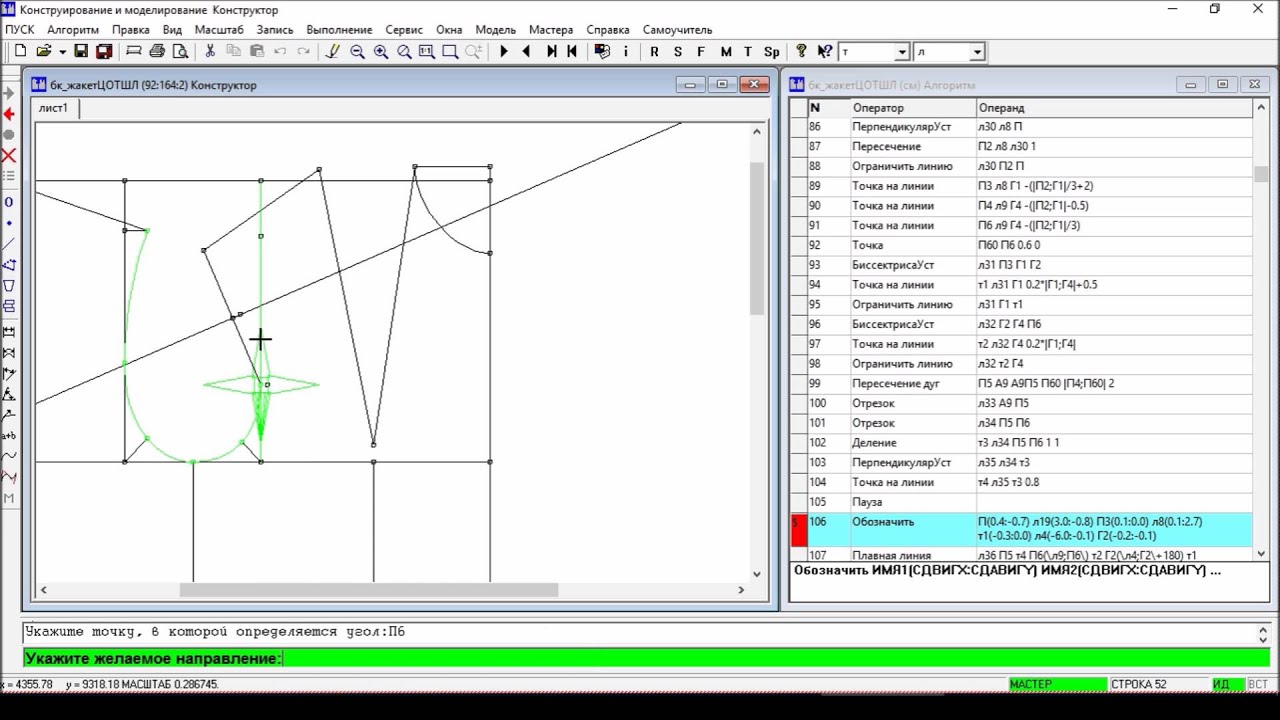


Рисунок 2.4 – Интерфейс САПР Грация

### 2.4.4 [САПР Леко](http://lekala.info/leko/dn.html)

Разработчик «Леко» [10], компания «Вилар», предлагает три версии программного обеспечения. Сокращенная версия подойдет для небольших организаций или молодых марок. В эту версию входит установочный диск с программой, описание, книга по конструированию и базы данных конструкций и методик, а также типовых и индивидуальных размерных признаков (ОСТ, ГОСТ и возможность расчета произвольного размера). Можно создавать сетку по нескольким размеро-ростам, строить ручные линии и градации, делать раскладку комплекта лекал для печати. На рисунке 2.5 представлен интерфейс САПР Леко.

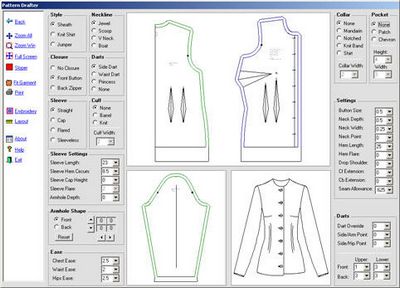


Рисунок 2.5 – Интерфейс САПР Леко

### 2.4.5 [RedCafe](http://redcafestore.com/)

Редактор одежды Redcafe [11] позволяет работать с чертежом на уровне линий, точек и объектов, строить и моделировать выкройки одежды. Программа умеет задавать припуски на швы, делать градацию и разведение лекал. Выкройки можно масштабировать, изменять, перемещать. Также программа позволяет оцифровывать бумажные лекала, выкройки из книг и журналов. Redcafe включает базу типовых размеров с возможностью добавлять индивидуальные. На рисунке 2.6 представлен интерфейс RedCafe.

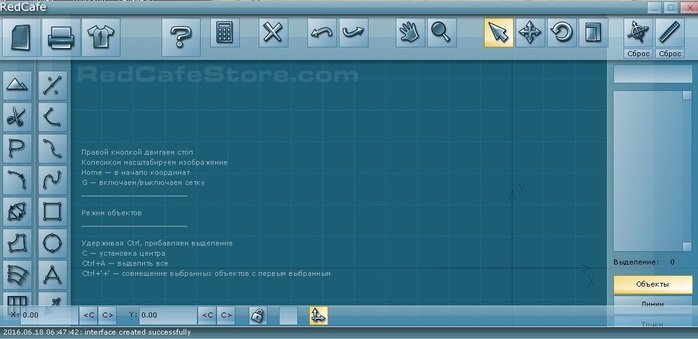


Рисунок 2.6 – Интерфейс САПР RedCafe

# 3 Описание реализации

## 3.1 Диаграммы USE CASE

Use Case [12] – диаграмма, отражающая отношения между актерами и прецедентами и являющаяся составной частью модели прецедентов, позволяющей описать систему на концептуальном уровне. На рисунке 3.1 представлена первоначальная диаграмма вариантов использования для реализуемого плагина.

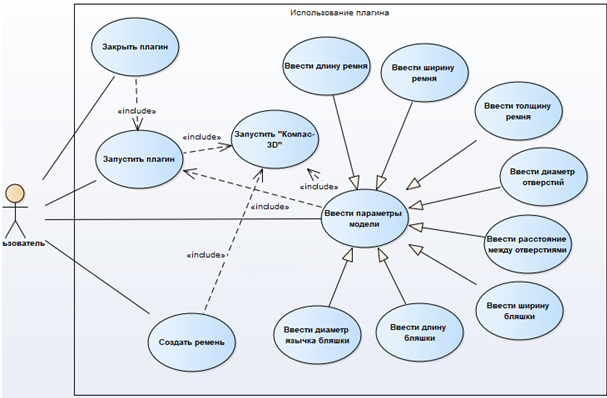


Рисунок 3.1 – Первоначальная диаграмма вариантов использования

На рисунке 3.2 представлена финальная версия диаграммы, в которой, в отличии от первоначальной, появилась возможность выбора формы бляшки.

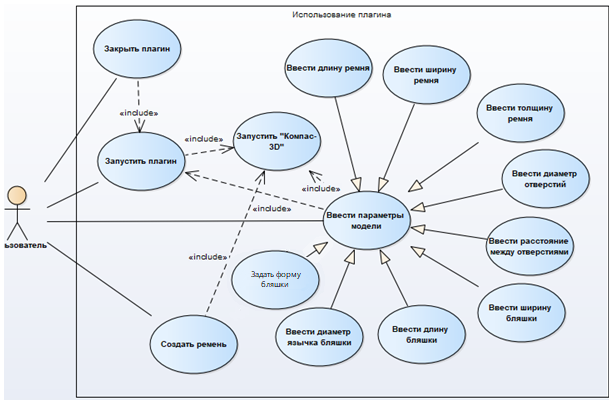


Рисунок 3.2 – Финальная диаграмма вариантов использования

## 3.2 Диаграмма классов

UML (Unified Modeling Language) [12] – унифицированный язык моделирования. На рисунке 3.3 представлена первоначальная версия UML-диаграммы, которая была построена до написания кода программы. На рисунке 3.4 представлена финальная UML-диаграмма для построения ремня в программе «Компас-3D».

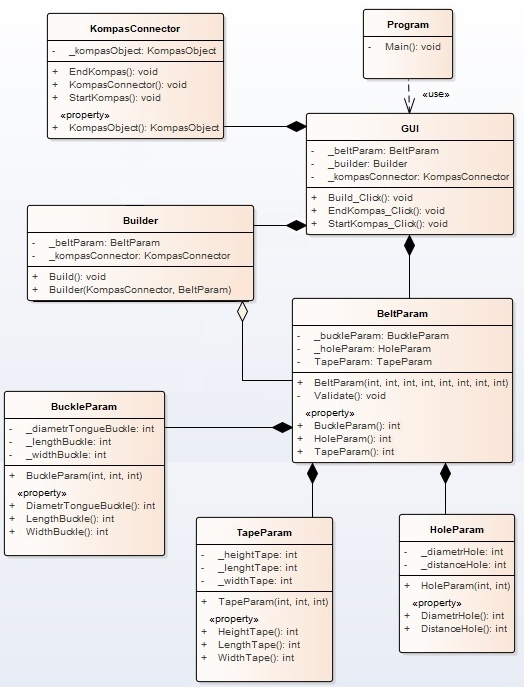


Рисунок 3.3 – Первоначальная UML-диаграмма

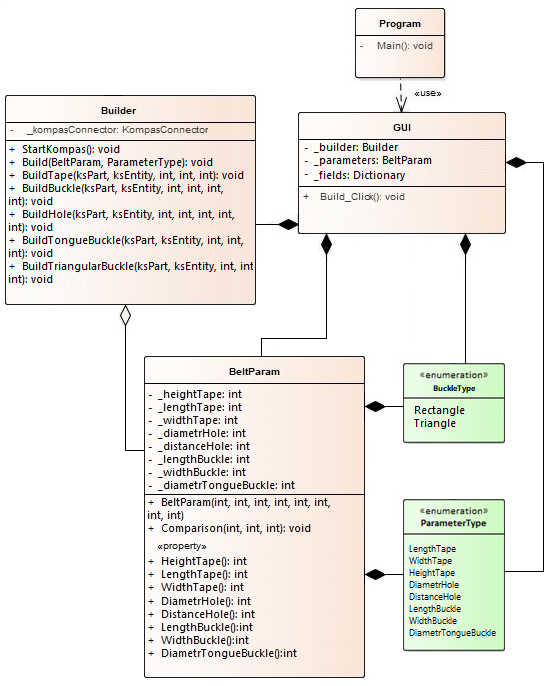


Рисунок 3.4 – Финальная UML-диаграмма

Для реализации подсистемы были спроектированы следующие классы:

* Builder – класс, отвечающий за вызов методов API КОМПАС 3D, необходимых для постройки объекта проектирования;
* GUI – класс диалогового окна, обеспечивающий взаимодействие между пользователем и программой через форму;
* BeltParams − класс, хранящий в себе все параметры модели, осуществляет проверку зависимых параметров;
* BuckleType – список, хранящий в себе возможные формы бляшки;
* ParameterType – список, хранящий в себе параметры ремня.

Основные отличия финальной версии UML-диаграммы от первоначальной заключается в следующим:

1. Был удален класс KompasConnector из-за отсутствия необходимости в нем.
2. Теперь за взаимодействие с программой «Компас-3D» отвечает класс Builder.
3. Метод Build() был разделен на 4 метода, каждый из которых строит определенную часть ремня.
4. Все параметры модели инкапсулируются в одном классе для упрощения логики хранения.
5. Добавлено перечисление форм бляшек.

# 4 Описание программы для пользователя

Макет пользовательского интерфейса [13] – это визуальное статическое представление концепции интерфейса пользователя. Интерфейс не только решает проблему взаимодействия с приложением, но и делает это взаимодействие максимально комфортным.

Макет пользовательского интерфейса для построения ремня в программе КОМПАС-3D представлен на рисунке 4.1.

После запуска плагина пользователь увидит интерфейс, представленный на рисунке 4.1, который в свою очередь разбит на 2 основных блока:

* Блок задания параметров модели ремня;
* Блок запуска построения модели ремня.

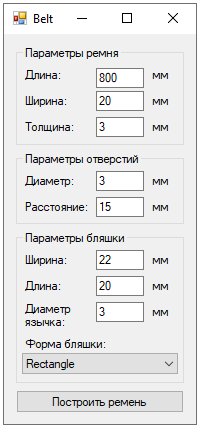


Рисунок 4.1 – Макет пользовательского интерфейса

Пользователь может сразу построить модель, используя изначально заданные минимальные параметры, или задать свои.

Поля, в которые пользователь может вписывать свои значения параметров, можно обозначить как TextBox. На эти поля наложены следующие ограничения:

Пользователь может ввести только положительные целочисленные значения;

При вводе значения, выходящего за допустимы диапазон, кнопка «Построить ремень» становится автоматически недоступна;

Если некорректное значение все же было введено, то значение изменяет свой цвет на красный;

Только при корректном заполнении всех полей кнопка «Построить ремень» будет доступна.

В программе также присутствуют зависимые параметры:

1. Ширина ленты должна быть меньше ширины бляшки, если это условие будет нарушено, то пользователь увидит сообщение, представленное на рисунке 4.2.

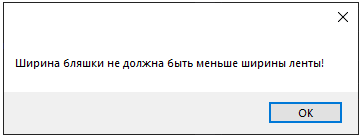


Рисунок 4.2 – Сообщение об ошибке

1. Диаметр язычка бляшки должен быть меньше диаметра отверстий в ленте ремня. При нарушении данного условия, пользователь увидит сообщение об ошибки, представленное на рисунке 4.3.

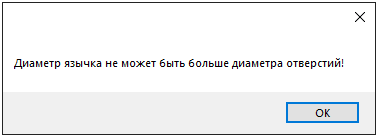


Рисунок 4.3 – Сообщение об ошибке

После ввода корректных значений и нажатия кнопки «Построить ремень», пользователь увидит результат построения в программе «Компас-3D». Пример результата работы представлен на рисунке 4.4.

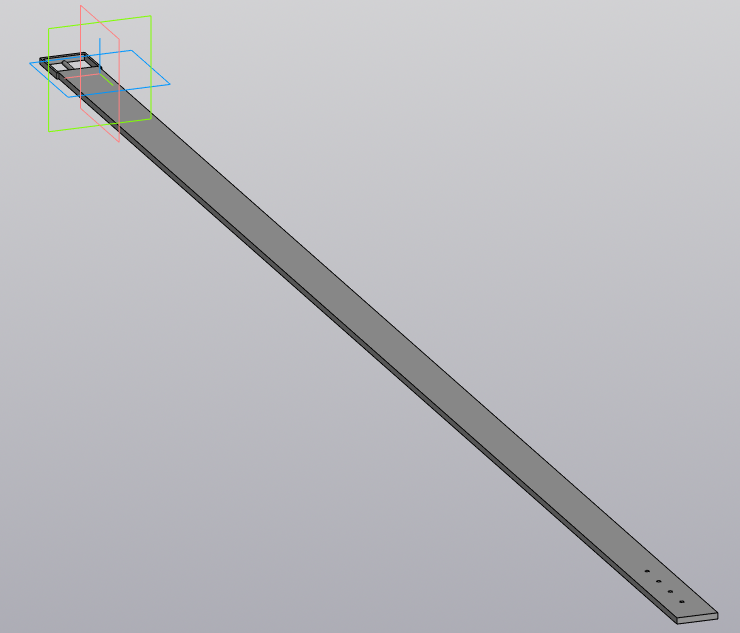


Рисунок 4.4 – Результат работы плагина

# 5 Тестирование программы

Тестирование позволяет выявлять ошибки в программе в процессе разработки и при выпуске промежуточных и финальных версий приложения.

## 5.1 Функциональное тестирование

При функциональном тестировании [14] проверялась корректность работы плагина “Ремень”, а именно, соответствие полученного результата в виде трехмерной модели, с входными параметрами.

Проведем тестирование максимальных и минимальных параметров модели.

Минимальные параметры:

1. Длина ремня = 800;
2. Ширина ремня = 20;
3. Толщина ремня = 3;
4. Диаметр отверстий = 3;
5. Расстояние между отверстиями = 15;
6. Длина бляшки = 22;
7. Ширина бляшки =20;
8. Диаметр язычка = 3.

Максимальные параметры:

1. Длина ремня = 1200;
2. Ширина ремня = 40;
3. Толщина ремня = 4;
4. Диаметр отверстий = 5;
5. Расстояние между отверстиями = 25;
6. Длина бляшки = 42;
7. Ширина бляшки =30;
8. Диаметр язычка = 5.

Модель с минимальными параметрами представлена на рисунке 5.1.

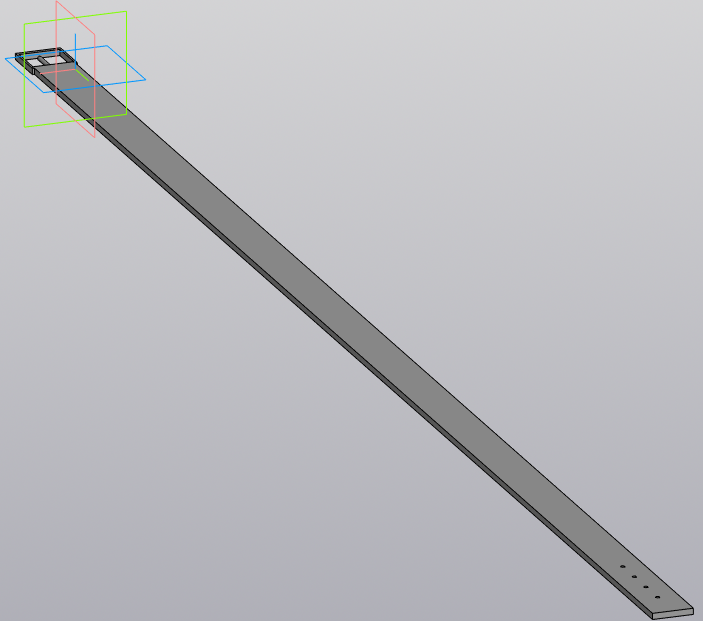


Рисунок 5.1 – Модель ремня с минимальными возможными параметрами

Модель с максимальными параметрами представлена на рисунке 5.2.

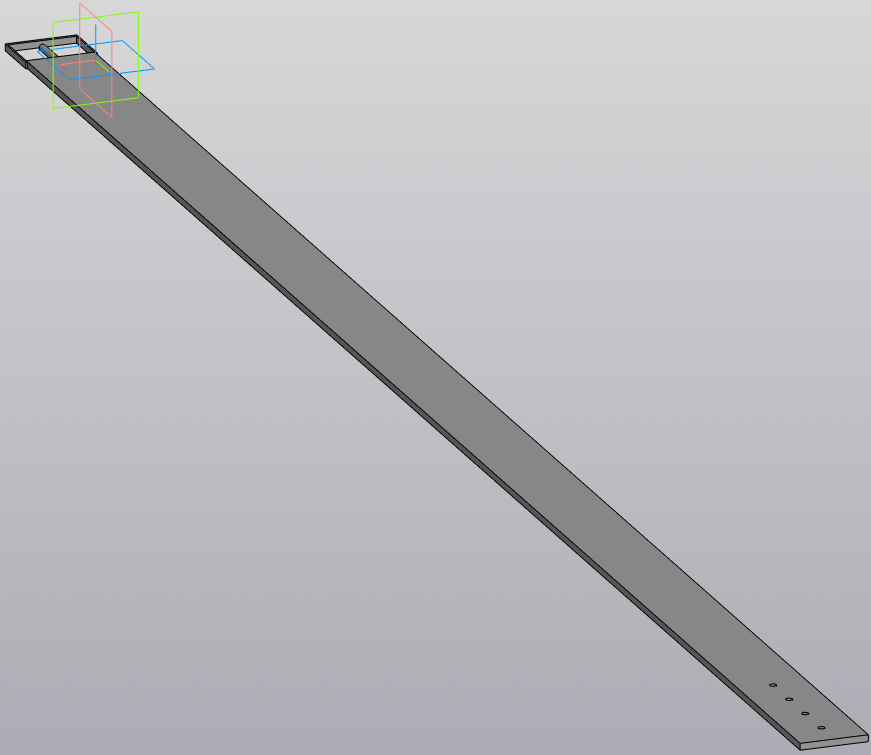


Рисунок 5.2 – Модель ремня с максимальными возможными параметрами

Проведем тестирование построения модели с бляшкой треугольной формы. Результаты тестирования представлены на рисунке 5.3.

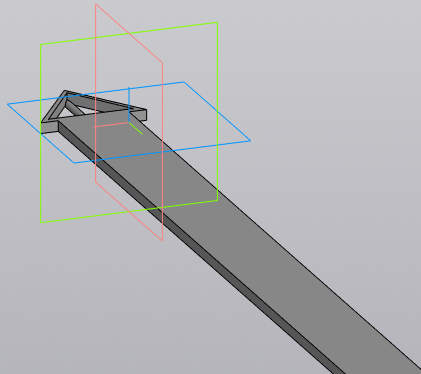


Рисунок 5.3 – Модель ремня с бляшкой треугольной формы

## 5.2 Модульное тестирование

В целях проверки корректности работы методов и свойств классов при помощи обозревателя тестов Visual Studio было проведено модульное тестирование [15], проверялись открытые поля и методы, для этого были создан тестовый класс BeltModelTests, тестовые случаи которого описаны в приложении А.1:

* BeltTests – класс тестирующий свойства и методы класса BeltParam.

Диалоговое окно состояний запущенных тестов для класса BeltModelTests изображен на рисунке 5.4.

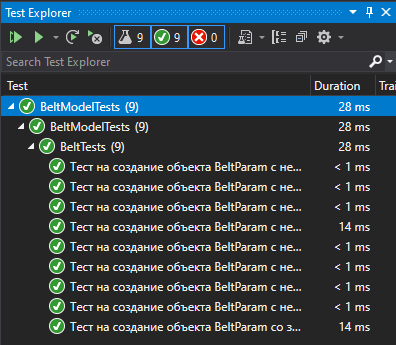


Рисунок 5.4 - Диалоговое окно состояний запущенных тестов для класса BeltModelTests

## 5.3 Нагрузочное тестирование

В целях проверки производительности работы плагина, было проведено нагрузочное тестирование [16]. Для проведения нагрузочного тестирования был добавлен секундомер, который засекал время от начала построения, с каждым успешным построением-перестроением фигуры производилась запись результатов в текстовый файл «log.txt».

Конфигурация ПК, на котором проводилось тестирование:

1. Центральный процессор: Intel Core i5-8400;
2. ОЗУ: 16 ГБ;
3. SSD 240ГБ;
4. Графический процессор: Intel UHD Graphics 630/

На рисунке 5.5 представлен график, где ось «ОX» – количество построенных-перестроенных деталей, ось «ОY» – время в секундах. На рисунке 5.6 представлен график, где ось «ОX» – количество построенных-перестроенных деталей, ось «ОY» – количество физической памяти, выделенной для программы «КОМПАС 3D» в мегабайтах.

Тестирование проводилось в течении 9 минут и 57 секунд, в течении которого происходило зацикленное построение 234 моделей со стандартными минимальными значениями параметров, которые можно увидеть на рисунке 4.1. Заметное снижение скорости построения было зафиксировано перед последним построением модели.

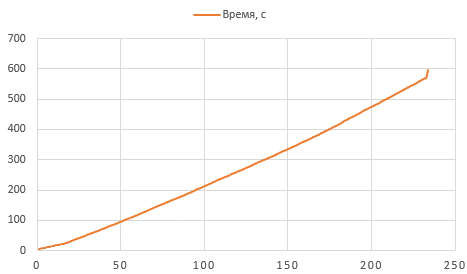


Рисунок 5.5 – График зависимости времени от количества построенных моделей

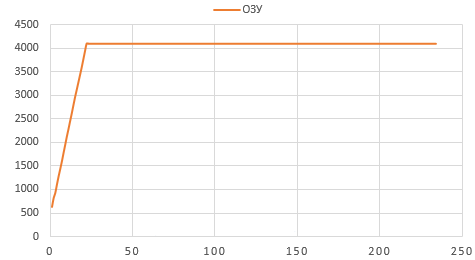


Рисунок 5.6 – График зависимости потребляемой ОЗУ от количества построенных моделей

После построения 22 детали ОС начинает выгружать часть старых окон «Компас-3D» из ОЗУ в раздел подкачки. Новые окна же окна загружаются в ОЗУ. Так как объекты моделирования одинаковые, то потребление ОЗУ остаётся на прежнем уровне продолжительное время.

# Заключение

В ходе выполнения лабораторных работ были изучены предметная область проектирования, предмет проектирования, аналоги предмета проектирования, API и на основании полученных данных были спроектированы архитектура и макет системы, создан плагин «Ремень», проведены модульные, функциональные и нагрузочные тесты.

# Список литературы

1. Компас-3D. [Электронный ресурс]. – Режим доступа: https://kompas.ru/ (дата обращения: 19.02.2020);
2. Microsoft Visual Studio 2019 [Электронный ресурс]. – URL: https://visualstudio.microsoft.com/ru/vs/ (дата обращения: 25.02.2020);
3. Ремень. [Электронный ресурс]. – Режим доступа: https://ru.wikipedia.org/wiki/%D0%A0%D0%B5%D0%BC%D0%B5%D0%BD%D1%8C (дата обращения: 19.02.2020);
4. ГОСТ 28754-90. [Электронный ресурс]. – Режим доступа: http://docs.cntd.ru/document/gost-28754-90 (дата обращения: 19.02.2020);
5. ГОСТ 18176-79. [Электронный ресурс]. – Режим доступа: http://docs.cntd.ru/document/gost-18176-79 (дата обращения: 19.02.2020);
6. API 5,7. [Электронный ресурс]. – Режим доступа: https://forum.ascon.ru/index.php/board,4.0.html (дата обращения: 19.02.2020);
7. Grafis. [Электронный ресурс]. – Режим доступа: http://www.cadrus.ru/cad/advantages/ (дата обращения: 19.02.2020);
8. Julivi. [Электронный ресурс]. – Режим доступа: http://julivi.com/ (дата обращения: 19.02.2020);
9. САПР Грация. [Электронный ресурс]. – Режим доступа: http://www.saprgrazia.com/ (дата обращения: 19.02.2020);
10. САПР Леко. [Электронный ресурс]. – Режим доступа: http://lekala.info/leko/dn.html (дата обращения: 19.02.2020);
11. Redcafe. [Электронный ресурс]. – Режим доступа: https://redcafestore.com/ (дата обращения: 19.02.2020);
12. Мартин Фаулер. UML. Основы. Краткое руководство по стандартному языку объектного моделирования. / М. Фаулер – Изд: Символ-Плюс,2011, с.192 (3-е издание);
13. Макет пользовательского интерфейса. [Электронный ресурс]. – Режим доступа: http://www.visualpharm.ru/design\_faq/kak-vyiglyadit-maket-interfeysa.html (дата обращения: 19.02.2020);
14. Функциональное тестирование [Электронный ресурс]. – URL: https://daglab.ru/funkcionalnoe-testirovanie-programmnogo-obespechenija/ (дата обращения: 13.04.2020);
15. Модульное тестирование [Электронный ресурс]. – URL: http://espressocode.top/unit-testing-software-testing/ (дата обращения: 13.04.2020);
16. Нагрузочное тестирование [Электронный ресурс]. – URL: https://daglab.ru/nagruzochnoe-testirovanie-programmnogo-obespechenija/ (дата обращения: 13.04.2020).

# Приложение А

(Справочное)

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

Таблица А.1 – Тестовый класс

|  |  |
| --- | --- |
| Наименование | Описание |
| +public void BeltParamConstructor\_NegativeTest (int lengthTape, int heightTape, int widthTape, int diametrHole, int distanceHole, int lengthBuckle, int widthBuckle, int diametrTongueBuckle) | Метод для создания объекта BeltParam c некорректными значениями |
| + public void BeltParamConstructor\_PositiveTest (int lengthTape, int heightTape, int widthTape, int diametrHole, int distanceHole, int lengthBuckle, int widthBuckle, int diametrTongueBuckle) | Метод для создания объекта BeltParam с корректными значениями |