Министерство образования и науки 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 к индивидуальной работе по дисциплине

«Основы разработки САПР» (ОрСАПР)

Тема проекта: Разработка библиотеки «Компьютерная мышь» для САПР «КОМПАС-3D V17»

|  |  |
| --- | --- |
|  | Студент гр. 584-2  \_\_\_\_\_\_\_\_\_\_\_\_\_ В.А. Бабынцева  «\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2017 г.  Руководитель к.т.н. доцент каф. КСУП  \_\_\_\_\_\_\_\_\_\_\_\_\_ А.А. Калентьев  «\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2017 г. |

Томск 2017

Министерство образования и науки 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)

Утверждаю:

Зав. кафедрой КСУП

\_\_\_\_\_\_\_\_Ю.А. Шурыгин

«\_\_\_»\_\_\_\_\_\_\_\_\_\_\_ 2017г.

ЗАДАНИЕ

по дисциплине «Основы Разработки САПР»

Выдано: студенту группы 584-2 Бабынцевой Виктории Александровны

1. Тема проекта: Разработка плагина «Компьютерная мышь» для «КОМПАС-3D V17».

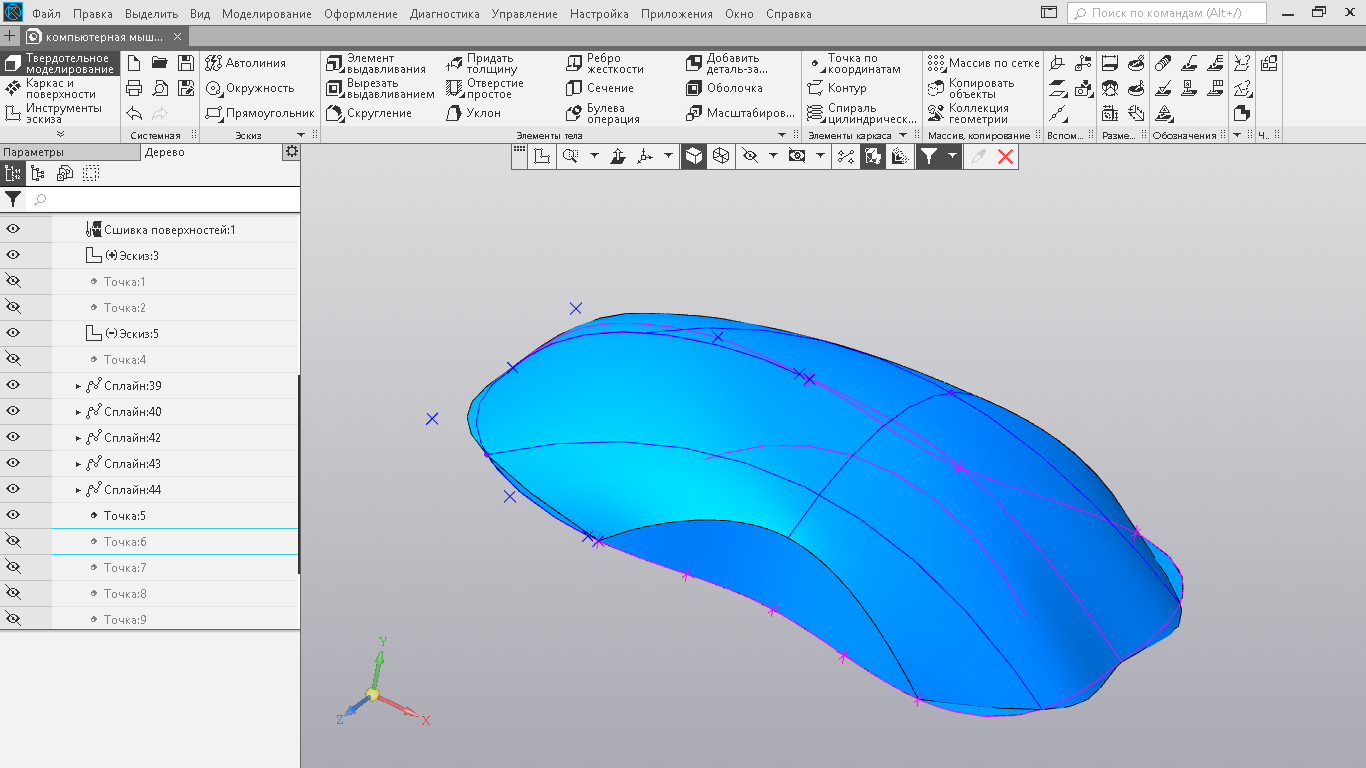


Рисунок 1.1 – Чертеж компьютерной мыши

1. Срок сдачи студентом проекта: 29.09.2017
2. Исходные данные

Разработать плагин «Компьютерная мышь» для «КОМПАС-3D V17»

1. Требования к плагину

Плагин должен обеспечивать следующую функциональность:

* выводить диалоговое окно ввода для изменения следующих параметров:
* Длина компьютерной мыши представлена на рисунке 1.2;

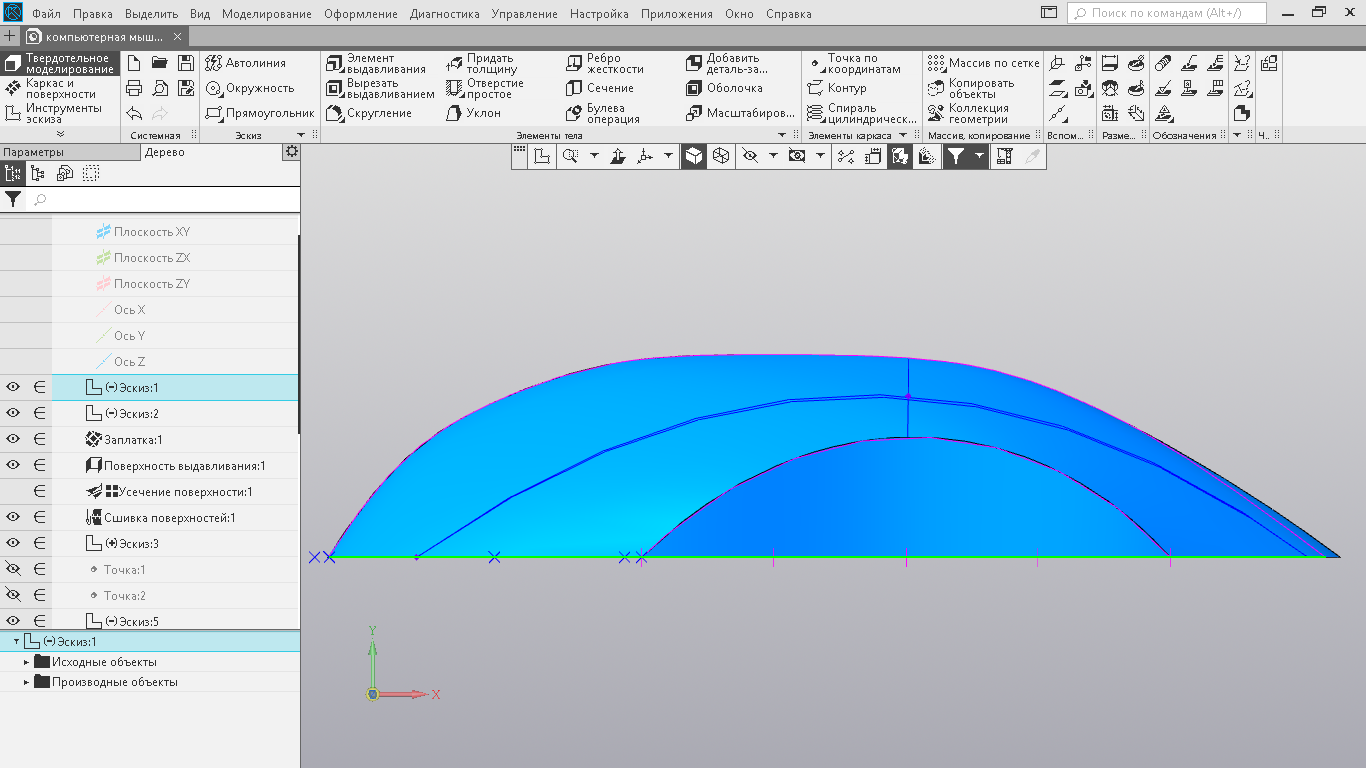


Рисунок 1.2 – Чертеж с обозначением длины компьютерной мыши

* Радиус боковой выемки (не может быть больше чем радиус бокового каркаса) представлен на рисунке 1.3;

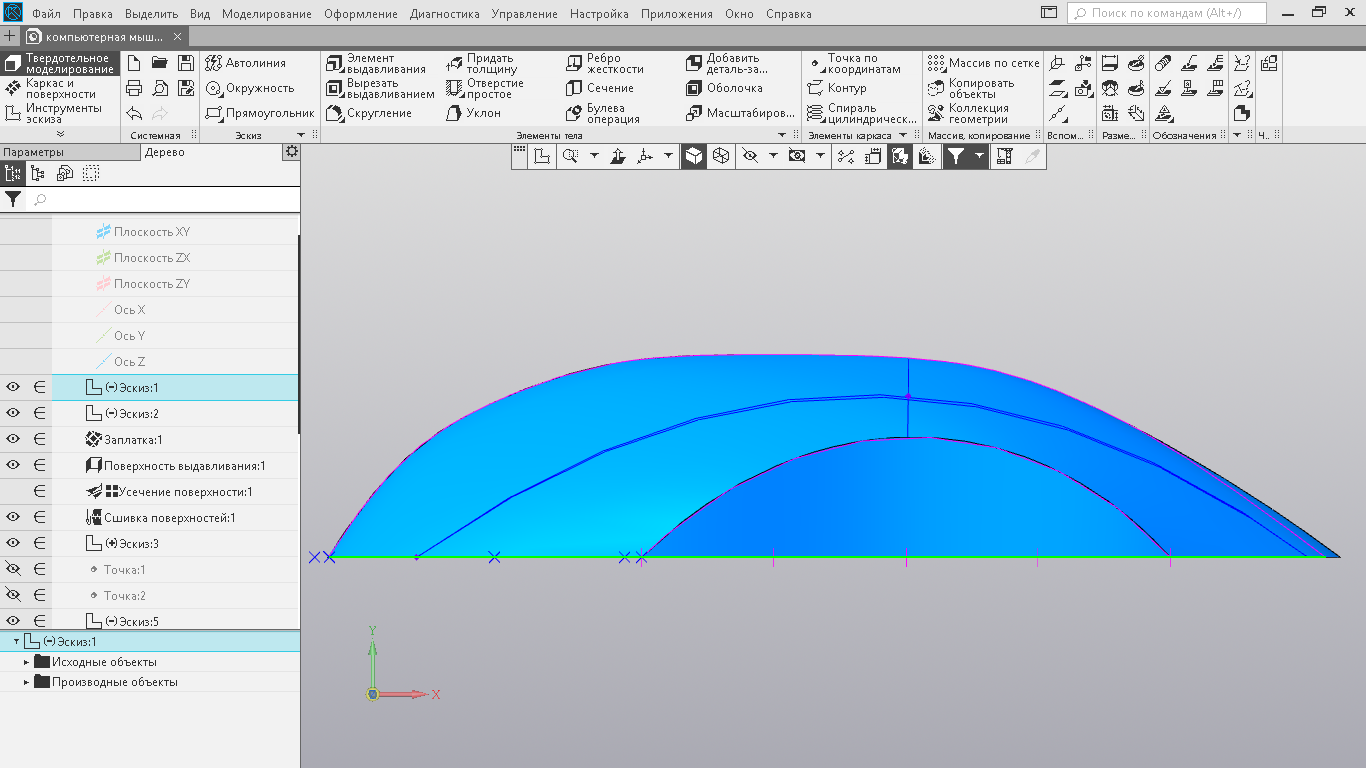


Рисунок 1.3 – Чертеж с обозначением радиуса боковой выемки

* Радиус бокового каркаса (не может быть меньше радиуса боковой выемки) представлен на рисунке 1.4;

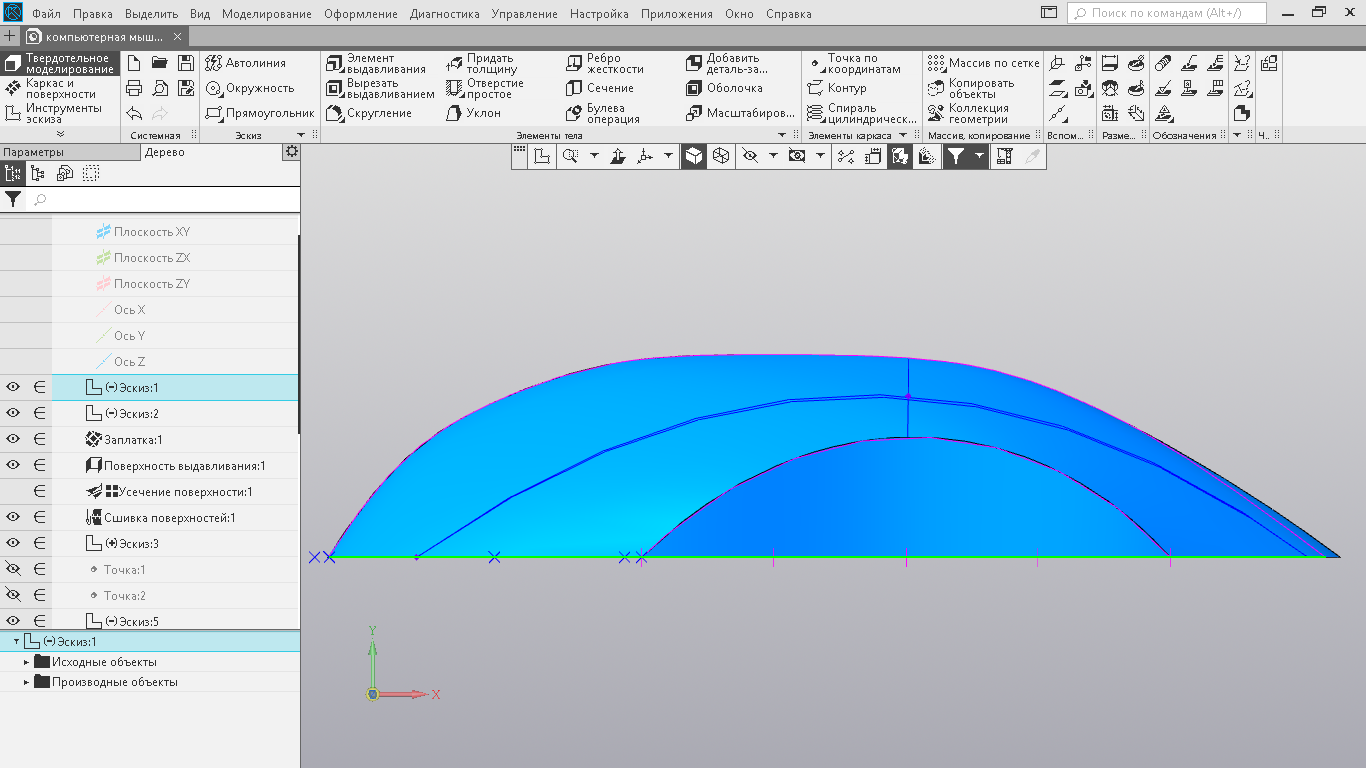


Рисунок 1.4 – Чертеж с обозначением радиуса бокового каркаса

* длина головного часть компьютерной мыши представлена на рисунке 1.5;

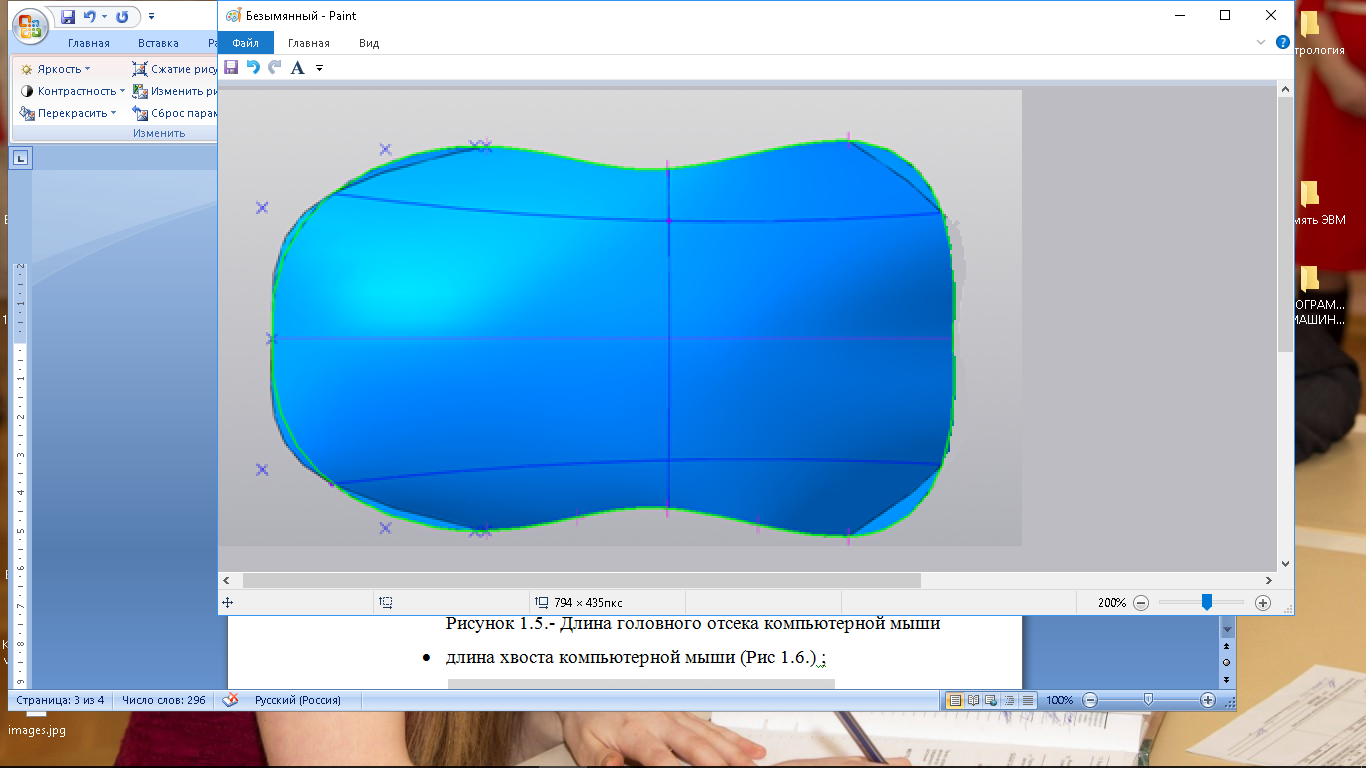


Рисунок 1.5 ­­– Чертеж с обозначением длины головной части компьютерной мыши

* длина задней части компьютерной мыши представлена на рисунке 1.6;

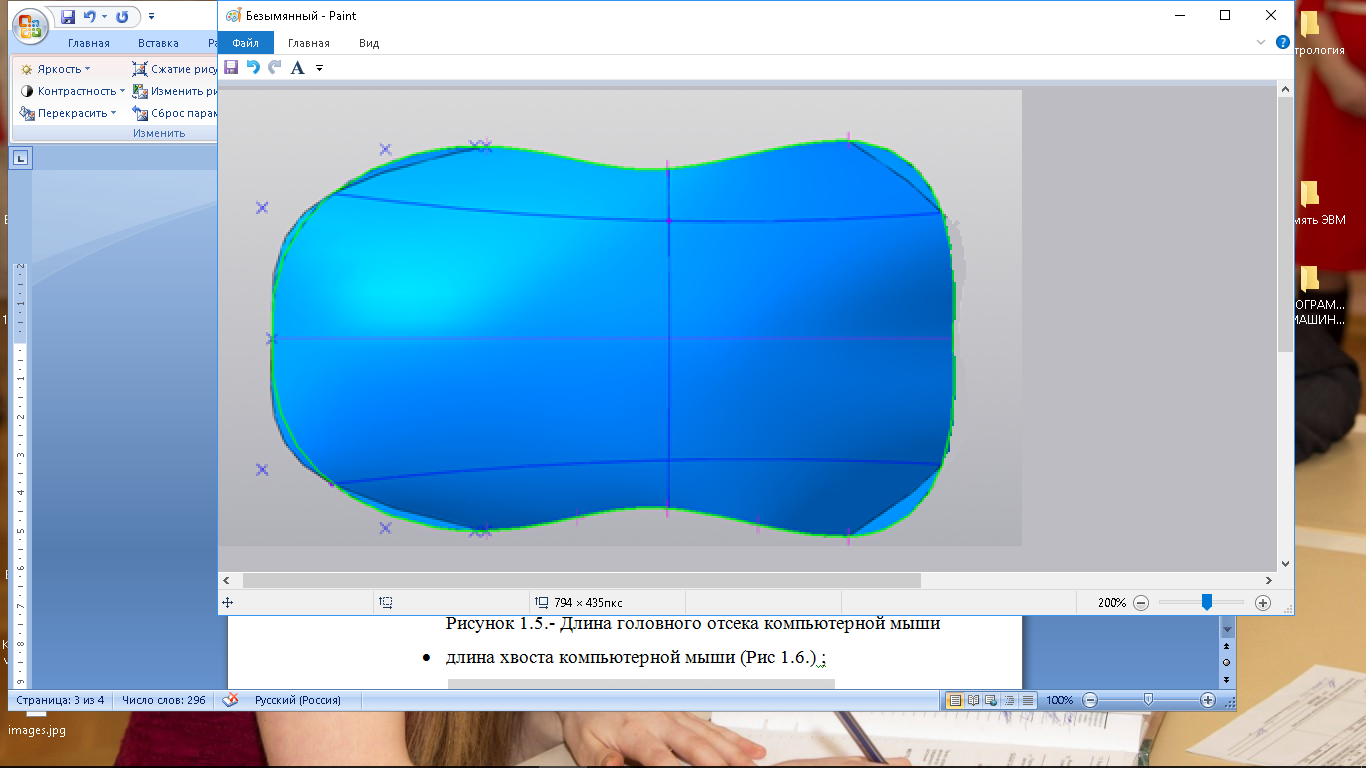


Рисунок 1.6 – Чертеж длины задней части компьютерной мыши

В сумме длина задней части и длина головной части равна длине компьютерной мыши.

* обеспечивать построение трехмерной модели на графическом окне системы «КОМПАС-3D» на основе введенных значений параметров;
* обеспечить проверку корректности ввода данных;
* вывод информационного сообщения в случае ввода некорректных данных.

1. Сфера применения

Плагин применим при проектировании комплектующих для персональных компьютеров в системе «КОМПАС-3D V17».

1. Минимальные требования к программной и аппаратной частям:

* операционная система: Windows 10 (x32/x64);
* процессор: 32-разрядный с тактовой частотой 1.75 ГГц;
* ОЗУ: 2 Гб;
* графические устройства с поддержкой DirectX 9 с драйвером WDDM 1.0;

1. Инструменты разработки:

* язык программирования C#, .NET framework 4.5;
* IDE Microsoft Visual Studio 2015;
* система контроля версий Git;

Руководитель к.т.н. доцент каф. КСУП:  
Калентьев А.А. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Задание принял к исполнению  
Бабынцева В.А. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Введение**

Инженерная компьютерная графика является одной из более интенсивно развивающихся отраслей технических знаний. Современные CAD- подсистемы, входящие в состав интегрированных СAD/CAM/CAE- систем и системы твердотельного параметрического моделирования механических объектов, отражающие последние достижения инженерной компьютерной графики, представляют собой наиболее важные разработки в области новых технологий по автоматизации деятельности инженеров, конструкторов и технологов. Это оценка была сделана еще в конце 80-ых годов прошлого века. К тому времени в мире было разработано 40-50 трехмерных систем автоматизированного проектирования (САПР) [1].

САПР позволяют уменьшить финансовые затраты на разработку макета (модели) проекта (объекта), а также сократить время, которое тратить проектировщик на создание модели объекта и составление проектной документации.

В каждой крупной САПР есть свой средства для разработки, который предоставляются с целью дать возможность разработчикам расширить функционал данной системы под свои конкретные нужды. Данным средством является API — программируемый интерфейс приложения [2]. Это набор готовых средств: классов, процедур, функций, структур и т.д. API позволяет определить функциональность, которую предоставляет приложение, при этом абстрагируясь от того, как она реализована.

Расширение функционала в основном подразумевает разработку плагина или библиотеки на основе предоставленного API.

Цель индивидуальной работы задача разработки плагина для построения 3D модели компьютерной мыши в автоматизированном режиме. В качестве системы, которая предоставляет API и для которой стоит задача разработать плагин была взята САПР КОМПАС 3D.

* + - 1. **UML-диаграммы**

После того как плагин был сделан произошли несоответствия с заявленными UML-диаграммами (Рисунки 1.1-1.4).

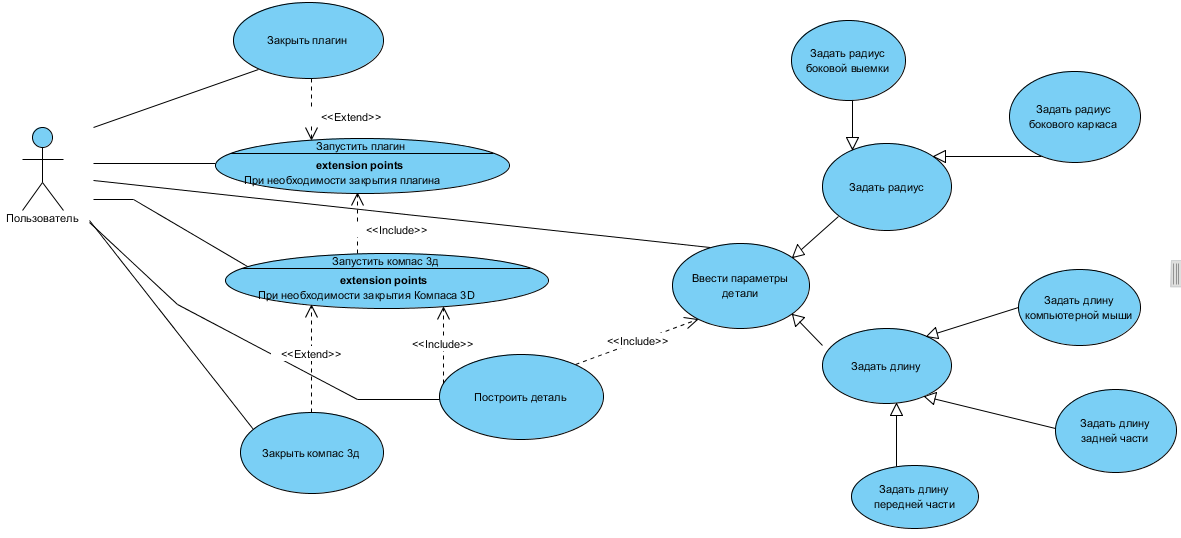


Рисунок 1.1 – Первоначально предоставленная диаграмма вариантов использования

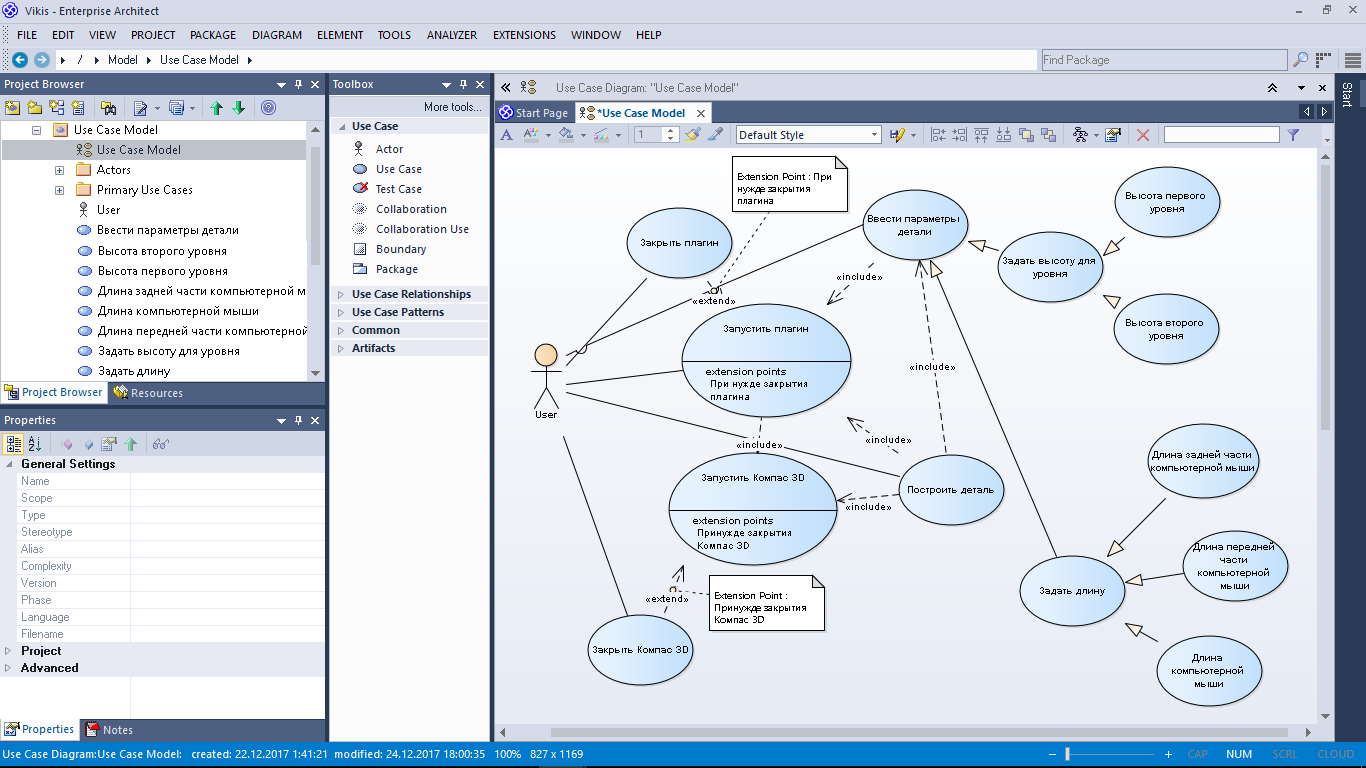


Рисунок 1.2 – Итоговая диаграмма вариантов использование

Как уже заметно диаграммы до и после вариантов использования практически не отличаются. Произошла замена названия параметров и пользователь при помощи нашего плагине не может открыть Компас 3D

(так как при нажатии кнопки «Построить деталь» плагин сам открывает Компас 3D).

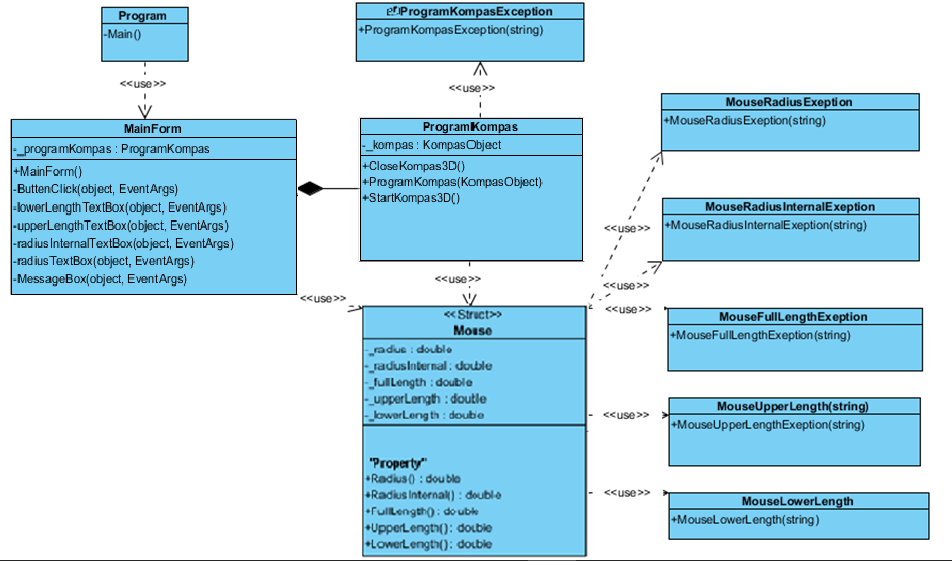


Рисунок 1.3 – Первоначальная диаграмма классов

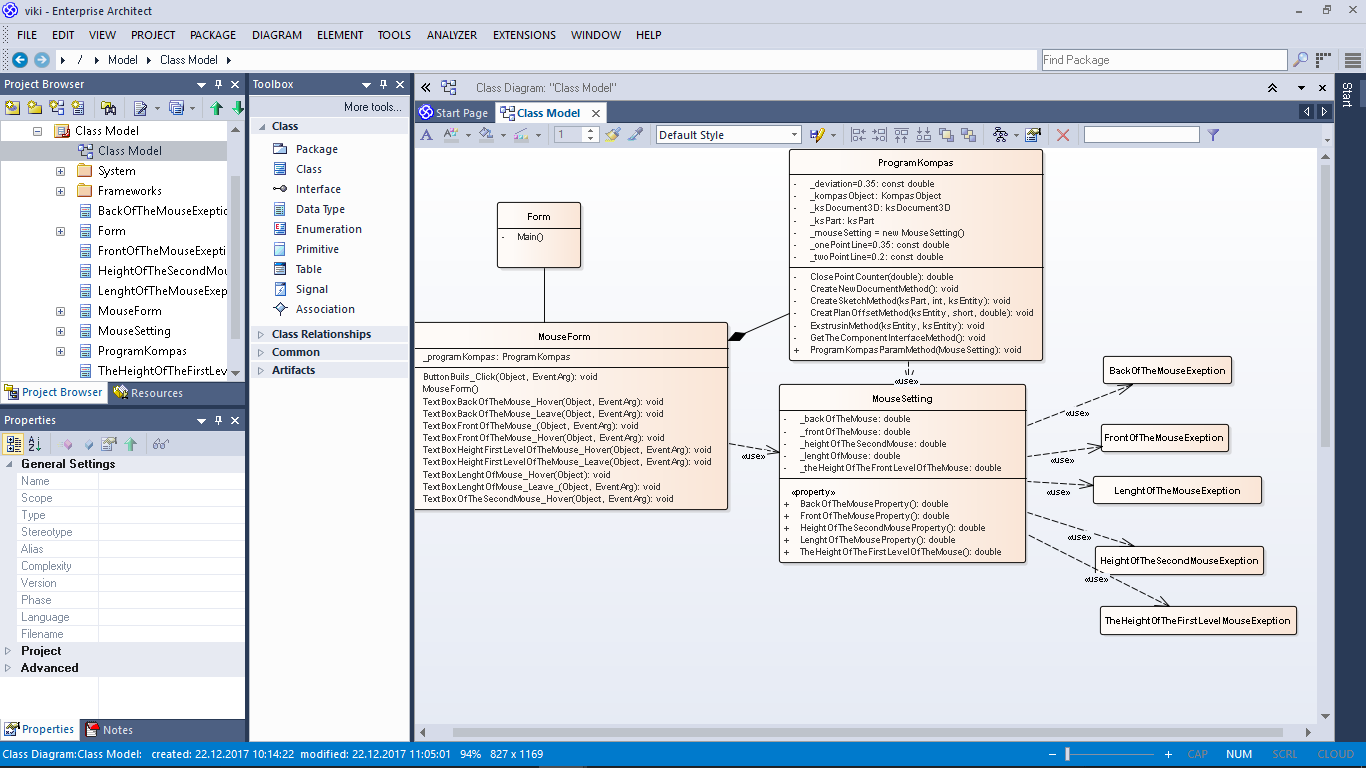


Рисунок 1.4 – Конечная диаграмма классов

Как видно из рисунков 1.3 и 1.4 сама структура не изменилась, но изменились название классов, параметров и более подробно описаны классы.

* + - 1. **Описание пользовательского интерфейса**

Интерфейс пользователя заключается в общении человека с компьютером. В нашем плагине это возможно при помощи макета пользовательского интерфейса (Рисунок 2.1)

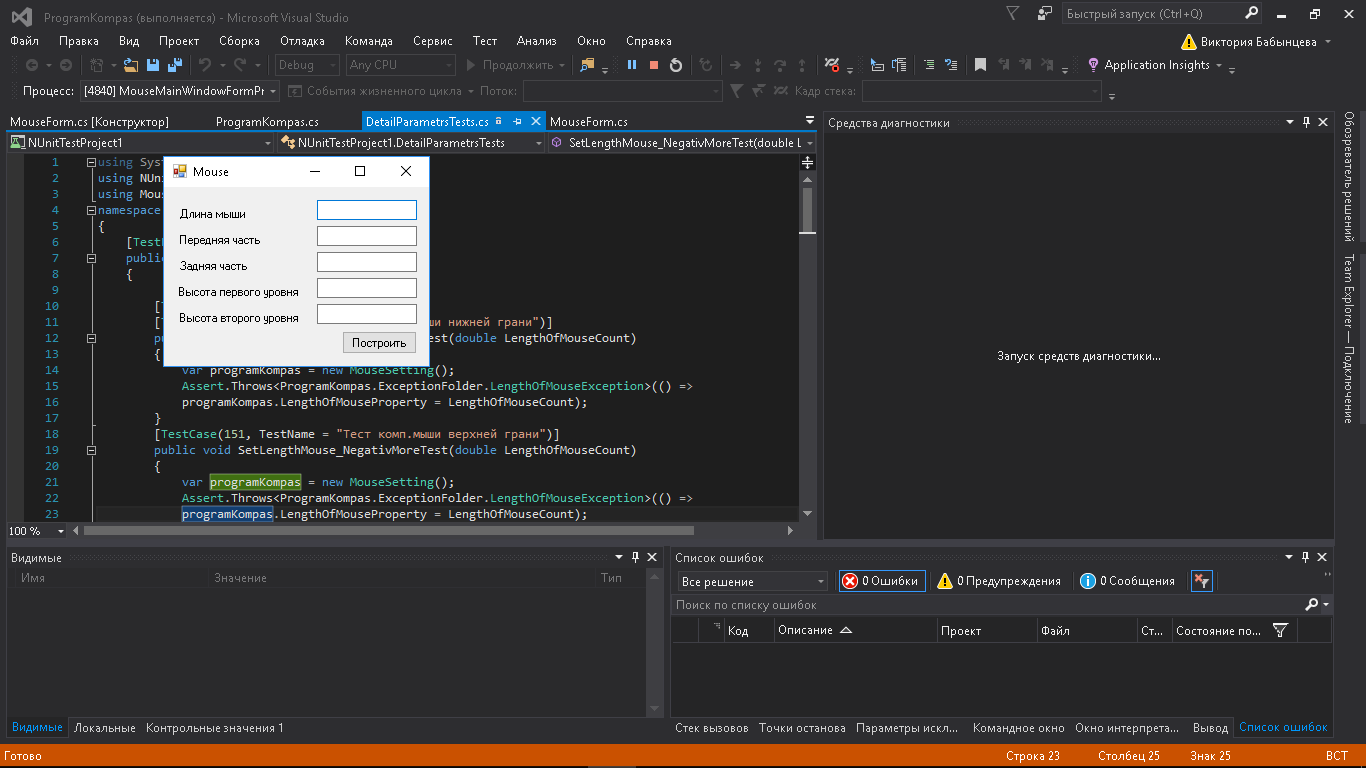
****

Рисунок 2.1 – Макет пользовательского интерфейса

Параметры вводятся в поля ввода. В поле ввода, подсвеченное синим цветом, требуется ввод параметра. При вводе символа и просто оставление пустой строки выводится ошибка (Рисунок 2.1 а).

При наведении на поле ввода высвечивается подсказки (Рисунок 2.1 б)

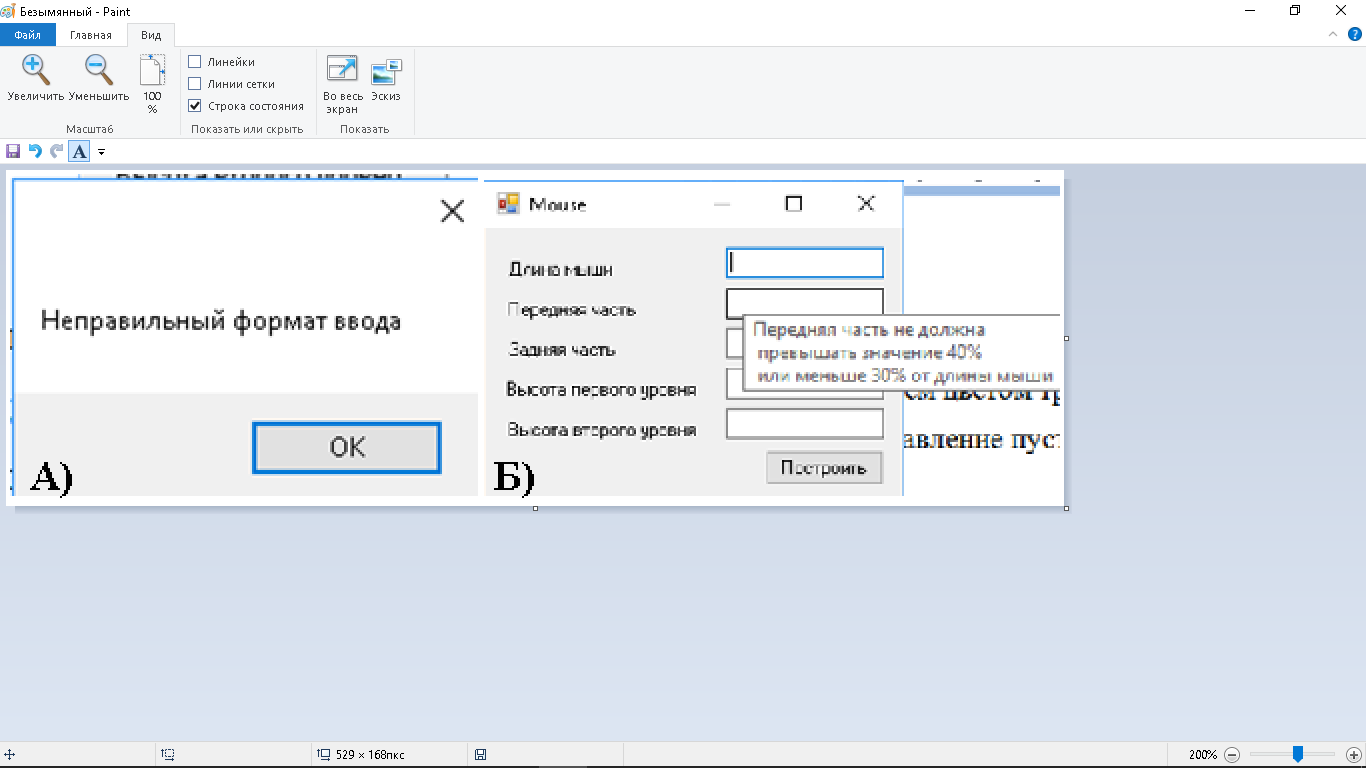


Рисунок 2.1 – а) Ошибка о неправильном формате б) Подсказка при наведении

При дальнейшем использовании подсказок можно корректно построить модель компьютерной мыши в противном случае вылезет ошибка, которая содержит пояснение этой ошибки. (Рисунок 2.3)

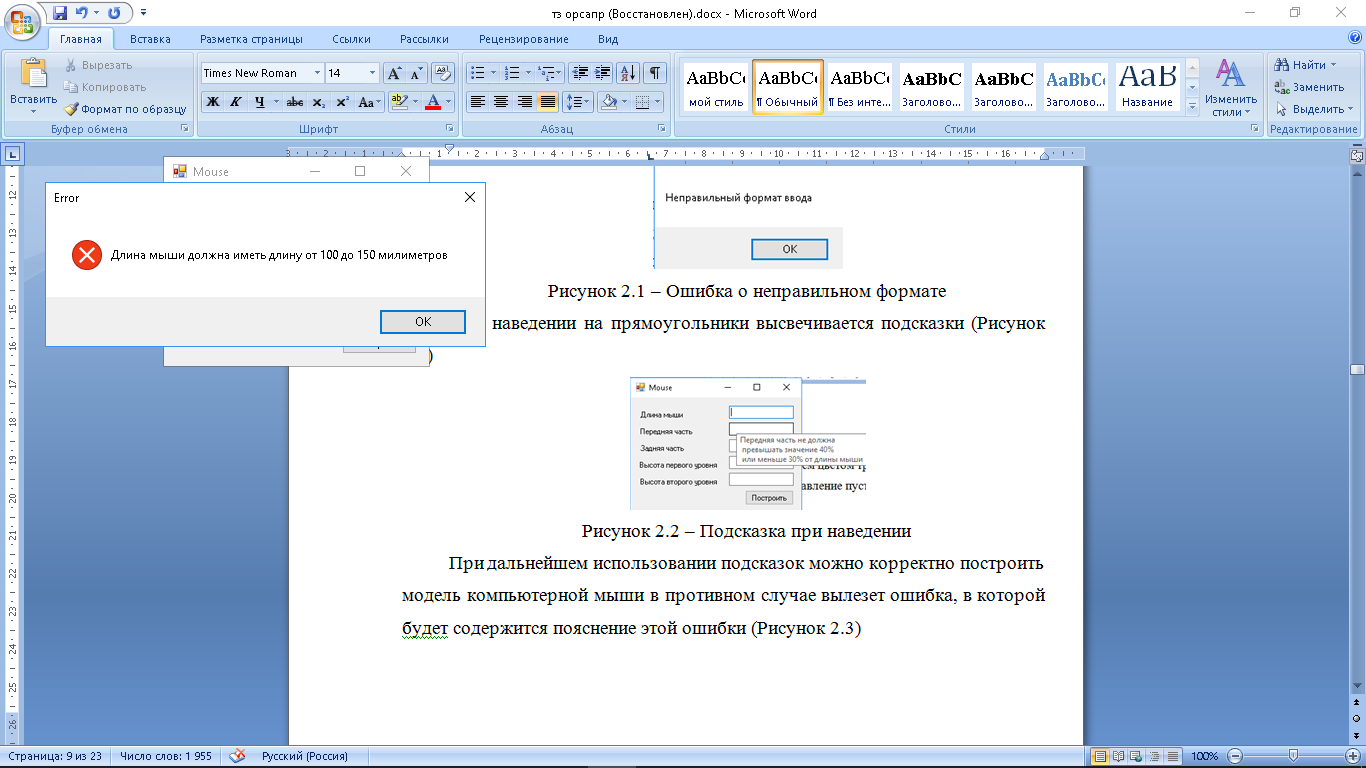


Рисунок 2.3 – Ошибка при несоблюдении подсказки

# Тестирование программы

Тестирование — это проверка соответствия объекта желаемым критериям. Несоответствие объекта желаемым критериям называется ошибкой[4].

Существует много классификаций тестов по разным признакам. Тут же стоит отметить, что данные классификации пересекаются и один тест может одновременно относиться к нескольким классификациям [4].

В данном проекте будет применятся три вида тестирования: функциональное тестирование, модульное тестирование, нагрузочное тестирование.

## Функциональные тесты

Функциональное тестирование — тестирование функциональности объекта, т. е. правильно ли объект выполняет свои функции. Фактически, выполняется проверка правильности выходных данных при соответствующих входных данных [4]. В данном случае будет проверяться правильность построения детали при различных входных параметрах (таблица 3.1):

Таблица 3.1 – Тестовые случаи и результаты работы программы на них

|  |  |  |
| --- | --- | --- |
| **Тестовый случай** | **Значения** | **Результат** |
| Параметр «Длина мыши» меньше наименьшего значения 100 мм или наибольшего значения 150 мм | (99, -, -, -,-;);  (151, -. -. -, -). | Корректно. Программа выводит MessageBox с предупреждением о том, что длина мыши может быть от 100 до 150мм. |

Окончание таблицы 3.1

|  |  |  |
| --- | --- | --- |
| **Тестовый случай** | **Значения** | **Результат** |
| Параметр «Передняя часть мыши» больше 40% или меньше 30% | (100, 41, -,-,-);  (100,29,-,-,-). | Корректно. Программа выводит MessageBox с предупреждением о том, что длина передней части мыши не должна превышать 40% или быть меньше 30% от длины мыши |
| Параметр «Передняя часть мыши» больше 70% или меньше 60% | (100,-,71,-,-);  (100,-,59,-,-). | Корректно. Программа выводит MessageBox с предупреждением о том, что длина задней части мыши не должна превышать 70% или быть меньше 60% от длины мыши |
| Параметр «Высота первого уровня» больше 30% или меньше 20% | (100,39,61,31,-)  (100,39,61,19,-) | Корректно. Программа выводит MessageBox с предупреждением о том, что высота первого уровня не должна превышать 30% или быть меньше 20% от длины мыши |
| Параметр «Высота второго уровня» больше 20% или меньше 10% | (100,39,61,20,21)  (100,39,61,20,9) | Корректно. Программа выводит MessageBox с предупреждением о том, что высота второго уровня не должна превышать 20% или быть меньше 10% от длины мыши |

Результат работы программы после ввода минимальных корректных значений представлен на рисунке 3.1.

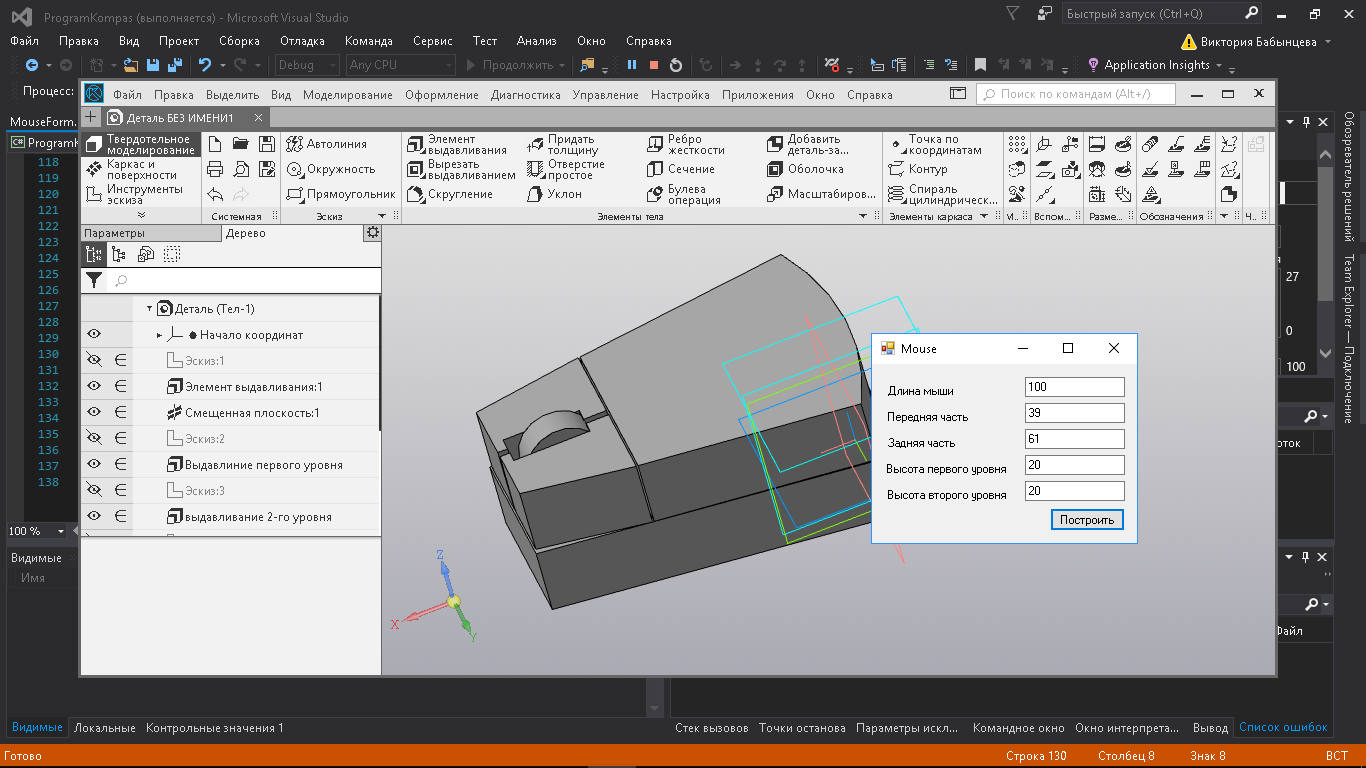
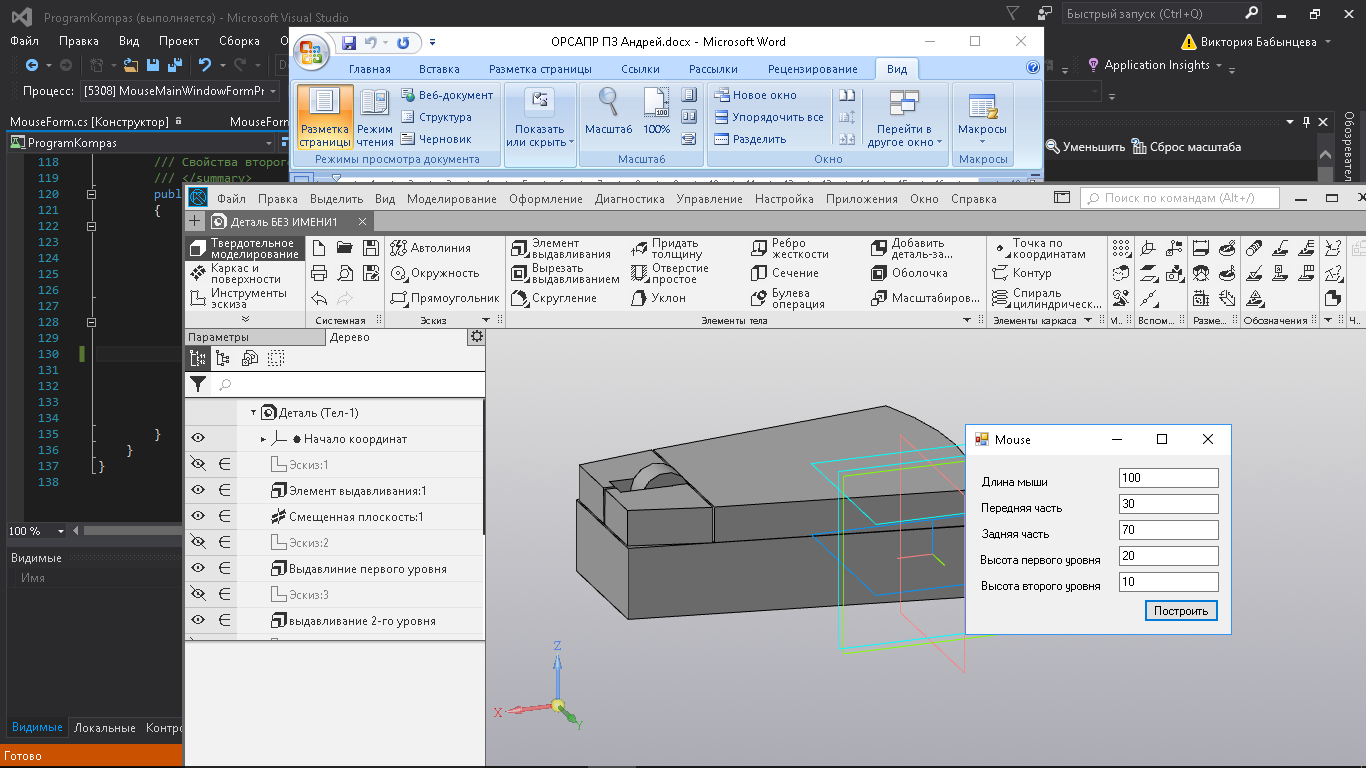


Рисунок 3.1 – Результат работы программы после ввода корректных значений корректными значениями

Результат работы программы после ввода минимально корректных значений представлен на рисунке 3.2



Результат работы программы после ввода максимально корректных значений представлен на рисунке 3.3.

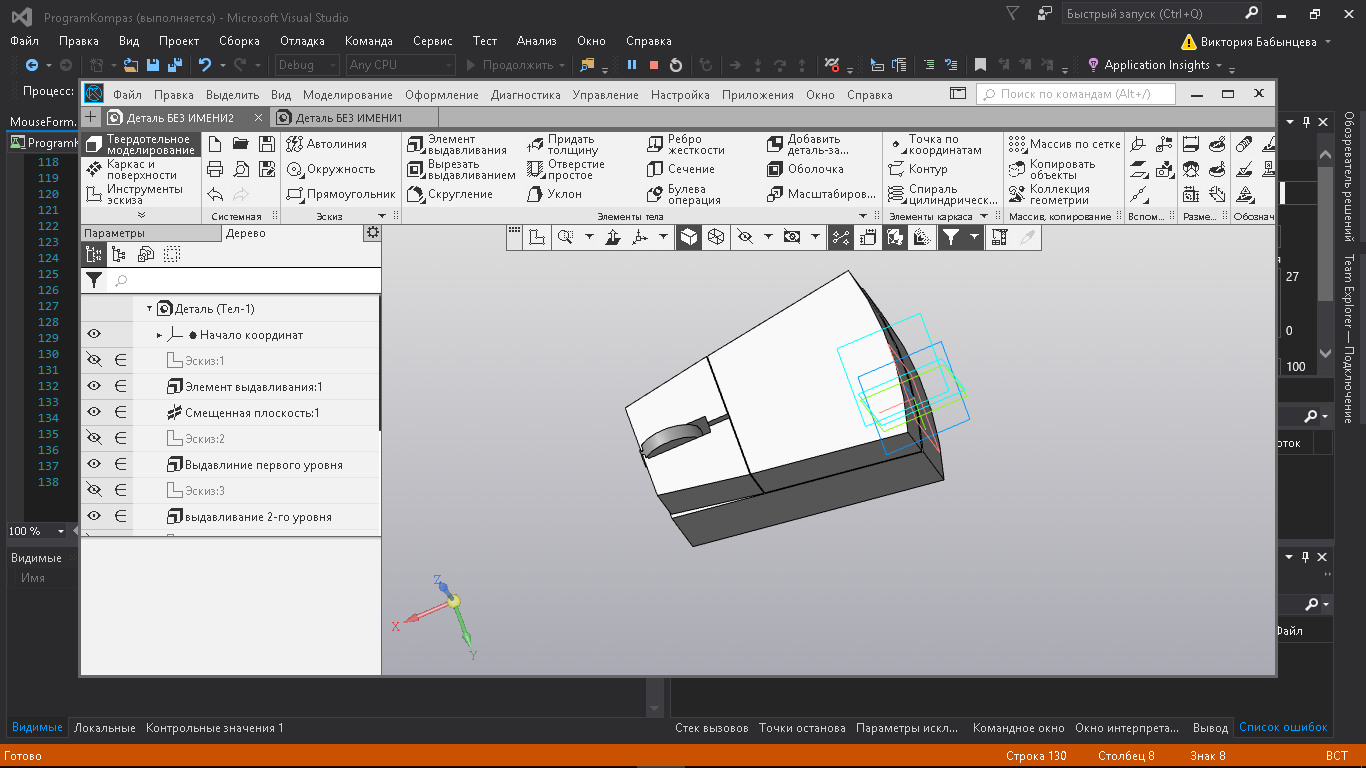


Рисунок 3.3 – Результат работы программы после ввода корректных значений с максимальными корректными значениями

* 1. **Модульное тестирование**

Юнит-тестирование (англ. «*unit*-*testing*», или модульное тестирование) — тестирование отдельного элемента изолированно от остальной системы. Относительно парадигмы объектно-ориентированного программирования системой является вся программа, а отдельным элементом — класс или его метод. Юнит-тестирование предназначено для проверки правильности работы отдельно взятого элемента. Чтобы исключить из результатов тестирования влияние потенциальных ошибок других элементов, тестируемый элемент должен быть максимально изолирован, то есть не использовать объекты и методы других классов [4].

На основе следующих тестовых сценариев (таблица 3.2), проводилось тестирование корректности входных параметров и правильности работы плагина с системой «Компас 3D».

Таблица 3.2 – Тестовые сценарии класса DetailParametrsTests

|  |  |  |
| --- | --- | --- |
| **Название тестового метода** | **Назначение** | |
| SetLengthMouse\_PositivTest(double LengthOfMouseCount) | Присвоение корректных данных компьютерной мыши | |
| SetFrontLengthMouse\_PositivTest  (double FrontOfTheMouse, double lengthMouseCount) | Присвоение корректных данных передней компьютерной мыши | |
| SetBackLengthMouse\_PositivTest  (double BackOfTheMouse, double lengthMouseCount) | Присвоение корректных данных задней компьютерной мыши | |
| SetTheHeightOfTheFirstLevelOfTheMouse\_PositivTest  (double TheHeightOfTheFirstLevelOfTheMouse,  double lengthMouseCount) | Присвоение корректных данных первого уровня компьютерной мыши | |
| SetHeightOfTheSecondMouse\_PositivTest  (double HeightOfTheSecondMouse, double lengthMouseCount) | Присвоение корректных данных второго уровня компьютерной мыши | |
| SetLengthMouse\_NegativLessTest  (double LengthOfMouseCount) | Присвоение некорректных данных компьютерной мыши нижней грани | |
| SetFrontLengthMouse\_NegativLessTest  (double FrontOfTheMouse, double lengthMouseCount) | Присвоение некорректных данных передней части компьютерной мыши нижней грани |
| SetBackLengthMouse\_NegativLessTest  (double BackOfTheMouse, double lengthMouseCount) | Присвоение некорректных данных задней части компьютерной мыши нижней грани |
| SetTheHeightOfTheFirstLevelOfTheMouse\_NegativLessTest  (double TheHeightOfTheFirstLevelOfTheMouse,  double lengthMouseCount) | Присвоение некорректных данных первого уровня компьютерной мыши нижней грани |
| SetHeightOfTheSecondMouse\_NegativLessTest  (double HeightOfTheSecondMouse, double lengthMouseCount) | Присвоение некорректных данных второго уровня компьютерной мыши нижней грани |

Продолжение таблицы 3.2 – Тестовые сценарии класса DetailParametrsTests

|  |  |
| --- | --- |
| **Название тестового метода** | **Назначение** |
| SetLengthMouse\_NegativMoreTest  (double LengthOfMouseCount) | Присвоение некорректных данных компьютерной мыши верхней грани |
| SetFrontLengthMouse\_NegativMoreTest  (double FrontOfTheMouse, double lengthMouseCount) | Присвоение некорректных данных передней части компьютерной мыши верхней грани |
| SetBackLengthMouse\_NegativMoreTest  (double BackOfTheMouse, double lengthMouseCount) | Присвоение некорректных данных задней части компьютерной мыши верхней грани |
| SetTheHeightOfTheFirstLevelOfTheMouse\_NegativMoreTest  (double TheHeightOfTheFirstLevelOfTheMouse,  double lengthMouseCount) | Присвоение некорректных данных первого уровня компьютерной мыши верхней грани |
| SetHeightOfTheSecondMouse\_NegativMoreTest  (double HeightOfTheSecondMouse, double lengthMouseCount) | Присвоение некорректных данных второго уровня компьютерной мыши верхней грани |

Тестирование проводилось с помощью фреймворка модульного тестирования NUnit 3.9 для языков платформы .NET.

Ниже приведен рисунок успешного прохождения всех описанных модульных тестов (см. рисунок 3.4).

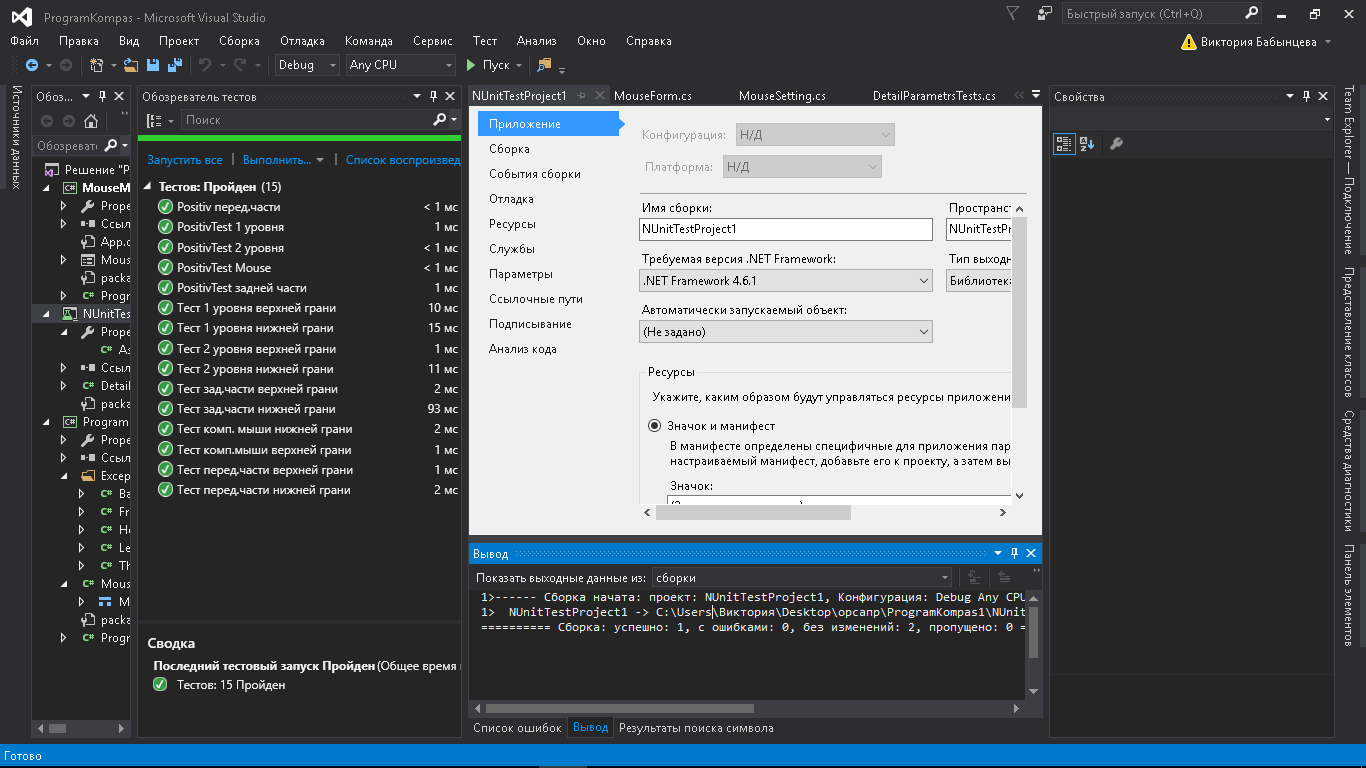


Рисунок 3.4 – Прохождение модульного тестирования приложения

* 1. **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 (устройству) [5].

Максимальное количество запросов на одновременный запуск плагина не может превышать 22 раз. При попытке построить больше «КОМПАС-3D» в автономном режиме закрывается и вызывается исключение о непредвиденной ошибке.

На рисунке 3.5 нарисованы график зависимости от количества строящихся деталей (ось х) и объема памяти в MB.

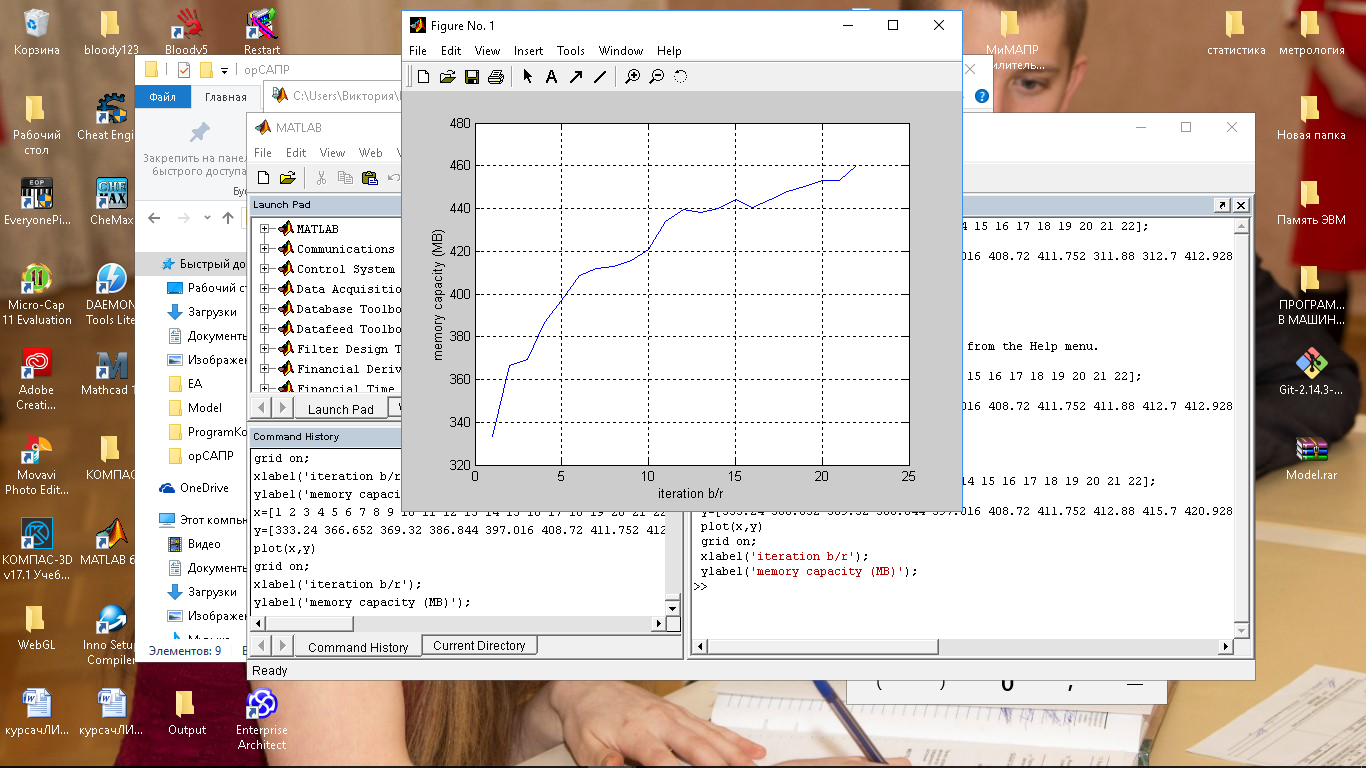


Рисунок 3.5 – График зависимости от количества строящихся деталей (ось х) и объема памяти в MB

По графику можно определить, что зависимость прямая. То есть используемая память, при увеличении количества деталей, увеличивается линейно.

На рисунке 3.6 нарисованы график зависимости от количества строящихся деталей (ось х) и время построения в секундах.

Первая итерация составляла 190.71 секунд. Это связано с тем, что по- мимо построение самой детали тратится время на открытия «КОМПАСА-3D». Поэтому она не была включена в график, чтобы наглядно можно было посмотреть на график других итераций.

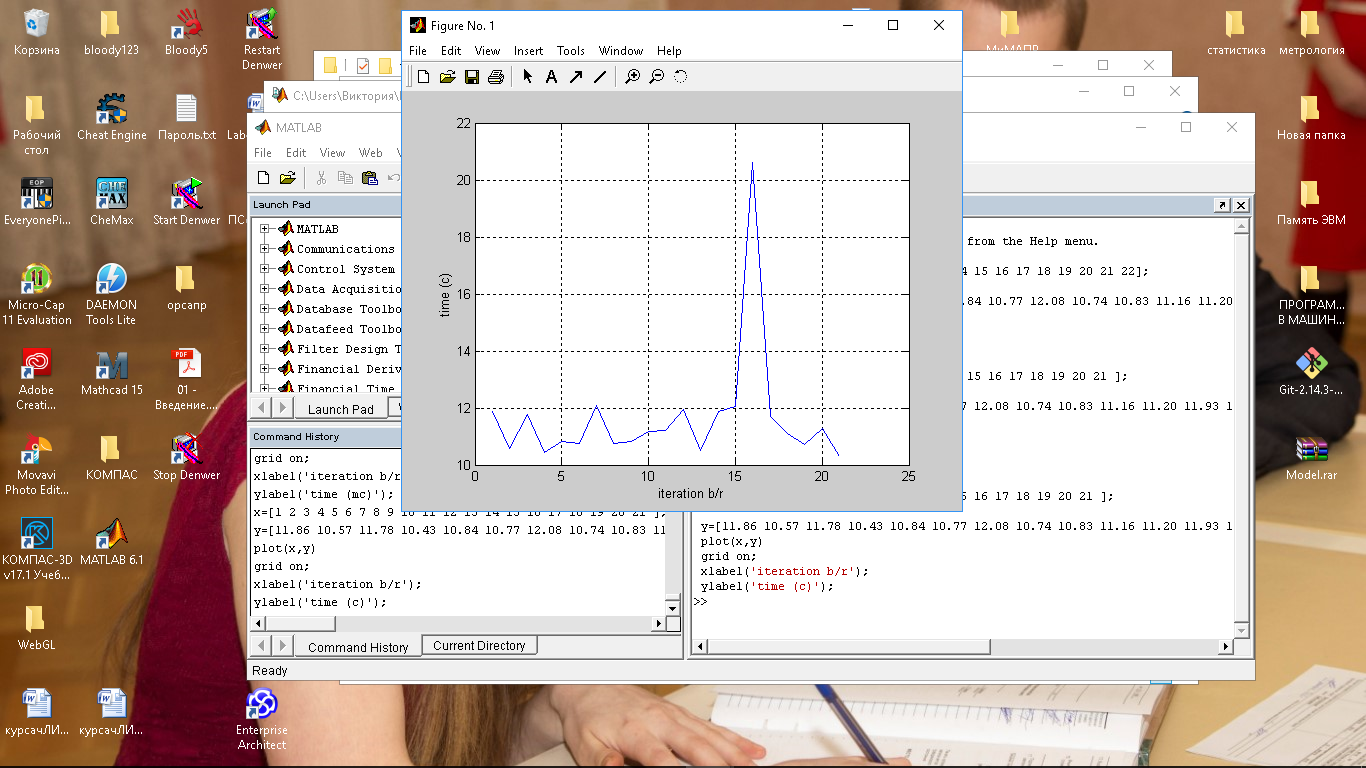


Рисунок 3.6 – график зависимости от количества строящихся деталей (ось х) и время построения в секундах

По графику можно определить, что зависимость время построения деталей от количества итераций нет. Это зависит от процента времени на обработку диском запросов чтения или записи. На 17 итерации диск был перегружен, поэтому время построение детали потребовалось больше.

**3.4 Конфигурационное тестирование**

В настройках сборки проекта выберем конфигурацию «Release» и платформу «x86», начнем сборку проекта. После успешной сборки попробуем запустить приложение на операционной системе Windows 10 (x64). Результаты запуска и работы программы на операционной системе Windows 10 представлены на рисунке 3.7.

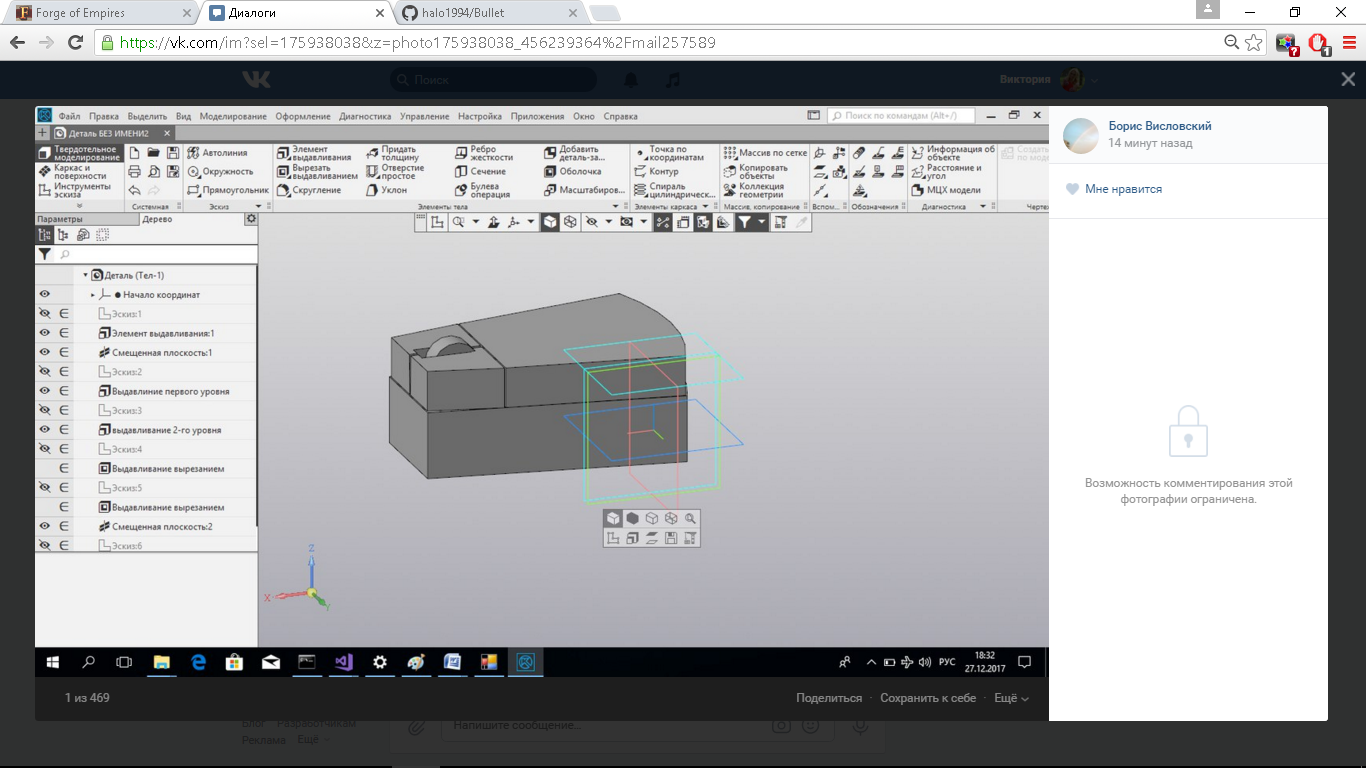


Рисунок 3.7 – Тестирование программы на другом компьютере

**Заключение**

В ходе выполнения индивидуальной работы были изучены основные этапы проектирования программного продукта и его реализации, изучена предметная область объекта проектирования. Реализованы модульные тесты для заданных параметров, изменение параметров модели и проверки корректности вводимы данных «Компьютерной мыши» для системы «Компас 3D V17».

**СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ**

Большаков В.П. Бочков А.Л. «Основы 3D-моделирования». Издательство: Питер;2013. – 300 с.

API – Википедия. [Электронный ресурс]. — URL: <https://ru.wikipedia.org/wiki/API> (дата обращения 26.10.2017)

UML. [Электронный ресурс]. — URL: <http://www.uml.org/> (дата обращения 26.10.2017).

Новые технологии в программировании: учебное пособие /А.А. Калентьев, Д. В. Гарайс, А. Е. Горяинов. – Томск, 2014. − ­ 176 стр.

Нагрузочное тестирование – Википедия. [Электронный ресурс]. – URL: <https://ru.wikipedia.org/wiki/Нагрузочное_тестирование> (дата обращения 27.12.2017)

ПРИЛОЖЕНИЕ А

ОПИСАНИЕ КЛАССОВ

Таблица А.1 ­– Описание класса MouseSetting

|  |  |  |
| --- | --- | --- |
| **Cвойство/метод** | **Тип / Тип возвращаемого значения** | **Описание** |
| FrontOfTheMouseProperty | double | Свойство для передней части компьютерной мыши |
| BackOfTheMouseProperty | double | Свойство для задней части компьютерной мыши |
| LengthOfMouseProperty | double | Свойство для длины компьютерной мыши |
| TheHeightOfTheFirstLevelOfTheMouseProperty | double | Свойство для первого уровня (фундамента) компьютерной мыши |
| HeightOfTheSecondMouseProperty | double | Свойство для второго уровня компьютерной мыши |

Таблица А.2 – Описание класса ProgramKompas

|  |  |  |
| --- | --- | --- |
| **Cвойство/метод** | **Тип / Тип возвращаемого значения** | **Описание** |
| RunningTheCompassMethod | void | Проверка и запуск компаса |
| CreateNewDocumentMethod | void | Получение интерфейса документа модели и ее создание |
| GetTheComponentInterfaceMethod | void | Интерфейс компонента |
| CreatSketchMethod | void | Создание эскиза |
| ExstrusionMethod | void | Элемент выдавливания |
| CreatPlaneOffsetMethod | void | Сдвиг плоскости вверх |
| СlosePointCouter | double | Высчитывание ближайшей точки |
| CreatExtrusionOffsetCutMethod | void | Выдавливание вырезанием |
| Cоnstruct | void | Сборка всех методов |

Продолжение таблицы А.3 – Описание класса MouseForm

|  |  |  |
| --- | --- | --- |
| **Cвойство/метод** | **Тип / Тип возвращаемого значения** | **Описание** |
| ButtonBuildClick | void | Построение модели «компьютерная мышь», обращение к методу COnstruct |
| TextBoxLengthOfMouseLeave | void | Валидация длины компьютерной при переходе на другой textBox |
| TextBoxFrontOfTheMouseLeave | void | Валидация передней части компьютерной мыши при переходе на другой textBox |
| TextBoxBackOfTheMouseLeave | void | Валидация задней части компьютерной мыши при переходе на другой textBox |
| TextBoxHeightFirstLevelOfTheMouseLeave | void | Валидация первого уровня при переходе на другой textBox |
| TextBoxHeightOfTheSecondMouseLeave | void | Валидация второго уровня при переходе на кнопку «Построить плагин» |
| TextBoxLengthOfMouseHover | void | При наведении на TextBoxLengthOfMouse вылезет подсказка |
| TextBoxFrontOfTheMouseHover | void | При наведении на TextBoxFrontOfTheMouse вылезет подсказка |
| TextBoxBackOfTheMouseHove | void | При наведении на TextBoxBackOfTheMouse вылезет подсказка |
| TextBoxTheHeightOfTheFirstLevelOfTheMouseHover | void | При наведении на TextBoxTheHeightOfTheFirstLevelOfTheMouse вылезет подсказка |