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)

РАЗРАБОТКА ПЛАГИНА «ШЕСТЕРНЯ» ДЛЯ САПР «КОМПАС-3D v16»

Пояснительная записка к курсовому проекту

по дисциплине «Основы разработки САПР» на тему

Студент гр.583-1

\_\_\_\_\_\_ Скрябина Т. С.

«\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_2017 г.

К.т.н., м.н.с. ЛИКС, каф. КСУП

\_\_\_\_\_\_ Калентьев А.А.

«\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_2017 г.

Томск 2017

**Реферат**

Курсовой проект 28 страниц, 19 иллюстраций, 11 таблиц, 9 использованных источников, 2 приложения.

ПЛАГИН, ШЕСТЕРНЯ, САПР, КОМПАС-3D, ПРОЕКТИРОВАНИЕ, ПРОЕКТНАЯ ДОКУМЕНТАЦИЯ, API.

Объектом разработки является плагин «Шестерня» для САПР «КОМПАС-3D».

Цель работы – создание плагина для построения трехмерной детали (шестерня) по введенным параметрам в «КОМПАС-3D».

В процессе выполнения курсового проекта разработан плагин для построения шестерни в рабочей плоскости САПР КОМПАС-3D, а также проектная документация к ней.

Плагин предназначен для построения трехмерной модели прямозубой шестерни. Взаимодействие с пользователем производится с помощью диалогового окна.

Содержание

[Введение 4](#_Toc472681132)

[1 Постановка и анализ задачи 5](#_Toc472681133)

[1.1 Описание САПР 5](#_Toc472681134)

[1.2 Описание предмета проектирования 12](#_Toc472681135)

[1.3 Описание технических и функциональных аспектов проекта 13](#_Toc472681136)

[2 Описание программы для пользователя 15](#_Toc472681137)

[3 Тестирование программы 16](#_Toc472681138)

[3.1 Функциональные тесты 16](#_Toc472681139)

[3.2 Модульные тесты 20](#_Toc472681140)

[3.3 Нагрузочные тесты 21](#_Toc472681141)

[4 Заключение 22](#_Toc472681142)

[Список литературы 23](#_Toc472681143)

[Приложение А 24](#_Toc472681144)

[Приложение Б 28](#_Toc472681145)

# Введение

В настоящее время проектирование в своем понимании представляет собой автоматизированный процесс и в некотором роде программно-аппаратный. Проектировщику, который занимается разработкой сложного механизма, или устройства, требующего больших расчетов, математических вычислений при построении модели и высокой точности, подходят системы автоматизации проектных решений — 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. В данном курсовом проекте стоит задача разработки плагина для построения 3D модели шестерни автоматизированном режиме. Плагин — независимо компилируемый программный модель, динамически подключаемый к основной программе, предназначенный для расширения или использования ее возможностей [3].

В качестве системы, которая предоставляет API и для которой стоит задача разработать плагин была взята САПР КОМПАС-3D.

# Постановка и анализ задачи

## Описание САПР

* + 1. **Описание программы**

КОМПАС-3D — система трехмерного проектирования, ставшая стандартом для тысяч предприятий, благодаря сочетанию простоты освоения и легкости работы с мощными функциональными возможностями твердотельного и поверхностного моделирования [4]. Ключевой особенностью продукта является использование собственного математического ядра С3D и параметрических технологий, разработанных специалистами АСКОН. КОМПАС-3D обеспечивает поддержку наиболее распространенных форматов 3D-моделей (STEP, ACIS, IGES, DWG, DXF), что позволяет организовывать эффективный обмен данными со смежными организациями и заказчиками, использующими любые CAD / CAM / CAE-системы в работе.

* + 1. Описание API

В КОМПАС на данный момент существуют API двух версий: API 5 и API 7. [5]. Обе версии реализуют различные функции системы и взаимно дополняют друг друга. Отсюда очевидно, что обе версии программных интерфейсов в равной мере поддерживаются и развиваются с учетом самих изменений в системе. В основном, для создания полноценных подключаемых модулей достаточно методов и свойств интерфейсов API 5.

Главным интерфейсом API системы КОМПАС является KompasObject. Получить указатель на этот интерфейс (если быть точным, на интерфейс приложения API 5) можно при работе под управлением внешнего приложения (контроллера) - после вызова стандартной системной функции. Методы этого интерфейса, реализуют наиболее общие функции работы с документами системы, системными настройками, файлами, а также дают возможность получить указатели на другие интерфейсы (интерфейсы динамического массива, работы с математическими функциями, библиотек моделей или фрагментов и различных структур параметров определенного типа).

Ниже в таблице 1.1 представлены свойства и методы интерфейса KompasObject, которые были использованы при разработке плагина.

Таблица . – Некоторые методы и свойства интерфейса KompasObject

|  |  |  |
| --- | --- | --- |
| Название | Тип | Описание |
| Document3D() | ksDocument | Метод для получения указателя на интерфейс трехмерного графического документа (детали или сборки) |
| GetMathematic2D() | ksMathematic2D | Метод для получения указателя на интерфейс для работы с математическими функциями |
| GetParamStruct(short structType) | [StructType2D](mk:@MSITStore:C:\Program%20Files\ASCON\KOMPAS-3D%20V16\SDK\SDK.chm::/StructType2D.htm) | Метод для получения указателя на интерфейс графического документа (чертежа или фрагмента) |
| GetDynamicArray(long type) | ksDynamicArray | Метод для получения указателя на интерфейс динамического массива ksDynamicArray |
| Visible | bool | Свойство видимости приложения |

В таблице 1.2 представлены методы интерфейса ksEntity, которые были использованы при разработке плагина.

Таблица . – Некоторые методы интерфейса ksEntity

|  |  |  |
| --- | --- | --- |
| Название | Тип | Описание |
| Create() | bool | Создать объект в модели |
| GetDefinition() | IUnkown | Получить указатель на интерфейс параметров объектов и элементов |
| Update() | bool | Изменить свойства объекта (используя ранее установленные свойства) |

В таблице 1.3 представлены свойства и методы интерфейса ksDocument3D, которые были использованы при разработке плагина.

Таблица . – Некоторые свойства и методы интерфейса ksDocument3D

|  |  |  |
| --- | --- | --- |
| Название | Тип | Описание |
| Create (bool invisible, bool \_typeDoc) | bool | Создать документ-модель (деталь или сборку) |
| UpdateDocumentParam() | bool | Активизировать измененные параметры документа |
| GetPart(int type) | ksPart | Получить указатель на интерфейс компонента в соответствии с заданным типом |
| fileName | string | Имя файла документа (трехмерной модели) |

В таблице 1.4 представлены методы интерфейса ksPart, которые были использованы при разработке плагина.

Таблица . – Некоторые свойства и методы интерфейса ksPart

|  |  |  |
| --- | --- | --- |
| Название | Тип | Описание |
| EntityCollection(short objType) | ksEnintyCollection | Формирует массив объектов и возвращает указатель на его интерфейс |
| GetDefaultEntity(short objType) | ksEntity | Получить указатель на интерфейс объекта, создаваемого системой по умолчанию |
| GetPart(int type) | ksPart | Получить указатель на интерфейс компонента в соответствии с заданным типом |
| NewEntity(short objType) | ksEntity | Создать новый интерфейс объекта и получить указатель на него |

В таблице 1.5 представлены методы интерфейса ksMathematic2D, которые были использованы при разработке плагина.

Таблица . – Некоторые методы интерфейса ksMathematic3D

|  |  |  |
| --- | --- | --- |
| Название | Тип | Описание |
| ksAngle (double x1, double y1, double x2, double y2) | double | Получить угол (в градусах) между осью OX и вектором, заданным двумя точками |
| ksIntersectArcArc (double xac, double yac, double rada, double fa1, double fa2,short directa, double xbc, double ybc, double radb, double fb1,double fb2, short directb, LPDISPATCH param) | bool | Получить координаты точек пересечения двух дуг окружностей |
| ksIntersectLinSArc (double x1,double y1, double x2, double y2, double xc,double yc, double rad, double f1, double f2, short direct,LPDISPATCH param) | bool | Получить координаты точек пересечения отрезка и дуги |
| ksSymmetry (double x,double y,double x1, double y1, double x2, double y2,double\* xc, double\* yc) | bool | Получить координаты точки, симметричной относительно заданной оси |

В таблице 1.6 представлены типы объектов документа-модели, которые были использованы при разработке плагина.

Таблица . – Некоторые типы объектов документа-модели

|  |  |  |
| --- | --- | --- |
| Идентификатор объекта | Название объекта | Интерфейс параметров |
| o3d\_unknown | Неизвестный (включает все объеткы) |  |
| o3d\_planeXOZ | Плоскость XOZ | ksPlaneParam |
| o3d\_sketch | эскиз | [ksSketchDefinition](mk:@MSITStore:C:\Program%20Files\ASCON\KOMPAS-3D%20V16\SDK\SDK.chm::/ksSketchDefinition.htm) |
| o3d\_planeOffset | смещённая плоскость | [ksPlaneOffsetDefinition](mk:@MSITStore:C:\Program%20Files\ASCON\KOMPAS-3D%20V16\SDK\SDK.chm::/ksPlaneOffsetDefinition.htm) |
| o3d\_edge | ребро | [ksEdgeDefinition](mk:@MSITStore:C:\Program%20Files\ASCON\KOMPAS-3D%20V16\SDK\SDK.chm::/ksEdgeDefinition.htm) |
| o3d\_chamfer | операция "фаска" | [ksChamferDefinition](mk:@MSITStore:C:\Program%20Files\ASCON\KOMPAS-3D%20V16\SDK\SDK.chm::/ksChamferDefinition.htm) |
| o3d\_cutExtrusion | вырезать выдавлива­нием | [ksCutExtrusionDefinition](mk:@MSITStore:C:\Program%20Files\ASCON\KOMPAS-3D%20V16\SDK\SDK.chm::/ksCutExtrusionDefinition.htm) |
| o3d\_axisOY | Ось OY |  |
| o3d\_circularCopy | операция копирова­ния по концентриче­ской сетке | [ksCircularCopyDefinition](mk:@MSITStore:C:\Program%20Files\ASCON\KOMPAS-3D%20V16\SDK\SDK.chm::/ksCircularCopyDefinition.htm) |
| o3d\_baseExtrusion | базовая операция выдавливания | [ksBaseExtrusionDefinition](mk:@MSITStore:C:\Program%20Files\ASCON\KOMPAS-3D%20V16\SDK\SDK.chm::/ksBaseExtrusionDefinition.htm) |

* + 1. Обзор аналогов
       1. **Плагин PDF**

Данный плагин позволяет производить экспорт моделей и сборок из КОМПАС-3D в формат PDF формат [6]. Основной особенностью является возможность интерактивного взаимодействия пользователя с сохраненной 3D сценой внутри PDF файла. Например, пользователь может вращать, масштабировать, передвигать детали и сборки внутри 3D PDF файла. Также доступно создание анимации сборки и разборки изделий. Это полезно для подготовки интерактивных сборочных инструкций, создания маркетинговых материалов, презентаций, а также для налаживания взаимодействия между проектировщиками и заказчиками. В подобных ситуациях традиционным подходом являлся экспорт сборки или детали КОМПАС-3D в промежуточный формат и дальнейшее сохранение в формат 3D PDF. Используемый подход в плагине исключает использование промежуточных файлов для осуществления 3D преобразования, что существенно повышает качество выходной 3D модели в формате PDF.Ключевые возможности:

* сохранение деталей и сборок в формате 3D PDF для интерактивного просмотра при помощи бесплатной программы Adobe Reader;
* создание анимаций имитирующих естественный порядок сборки и разборки создание имитации анимации гибки листовых тел;
* вставка в существующие PDF документы, содержащие основной текст, фоновые картинки, таблицы спецификаций, эмблемы, логотипы;
* пакетный режим для поочередной конвертации всех файлов.
  + - 1. **Экспорт из КОМПАС-3D в формат 3D PDF**

Пользователям КОМПАС-3D стала доступна функция экспорта созданных трехмерных моделей и дальнейшего их использования в создании технической документации. Экспорт происходит в формате 3D PDF [7].

Главной особенностью является то, что пользователь по-прежнему имеет возможность интерактивно взаимодействовать с 3D сценой, находясь внутри файла 3D PDF. То есть пользователь может передвигать детали, вращать их, масштабировать, передвигать сборки внутри самого файла. Пользователь также может создать анимацию сборки и разборки изделия. Этот функционал очень удобен. Он используется при создании презентаций, маркентиговых материалов, при подготовке интерактивных сборочных конструкций. Он значительно упрощает взаимодействие между заказчиками и проектировщиками.

Компания Visual Technology Services Ltd. Из Великобритании разработала плагин PDF3D, предоставляющий доступ к описанным выше возможностям. Основной функционал плагина:

- сохранение сборок и деталей для интерактивного просмотра в формате 3D PDF с помощью программы Adobe Reader;

- создание анимации, имитирующей естественный порядок разборки и сборки;

- создание анимации, имитирующей гибки листового тела;

- возможность вставки в PDF документ, содержащий основной текст, фоновых картинок, таблиц, логотипов, эмблем, спецификаций и т.д.

## Описание предмета проектирования

Предметом проектирования является модель шестерни [8]. Зубчатое колесо или шестерня – основная деталь [зубчатой передачи](https://ru.wikipedia.org/wiki/%D0%97%D1%83%D0%B1%D1%87%D0%B0%D1%82%D0%B0%D1%8F_%D0%BF%D0%B5%D1%80%D0%B5%D0%B4%D0%B0%D1%87%D0%B0) в виде диска с [зубьями](https://ru.wikipedia.org/w/index.php?title=%D0%97%D1%83%D0%B1_(%D1%82%D0%B5%D1%85%D0%BD%D0%B8%D0%BA%D0%B0)&action=edit&redlink=1) на [цилиндрической](https://ru.wikipedia.org/wiki/%D0%A6%D0%B8%D0%BB%D0%B8%D0%BD%D0%B4%D1%80) или [конической](https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BD%D1%83%D1%81) поверхности, входящими в зацепление с зубьями другого зубчатого колеса. В машиностроении принято малое зубчатое колесо называть шестернёй, а большое — колесом. Однако часто все зубчатые колёса называют шестернями. На рисунке 1.1 представлен вид на 3D модель шестерни.

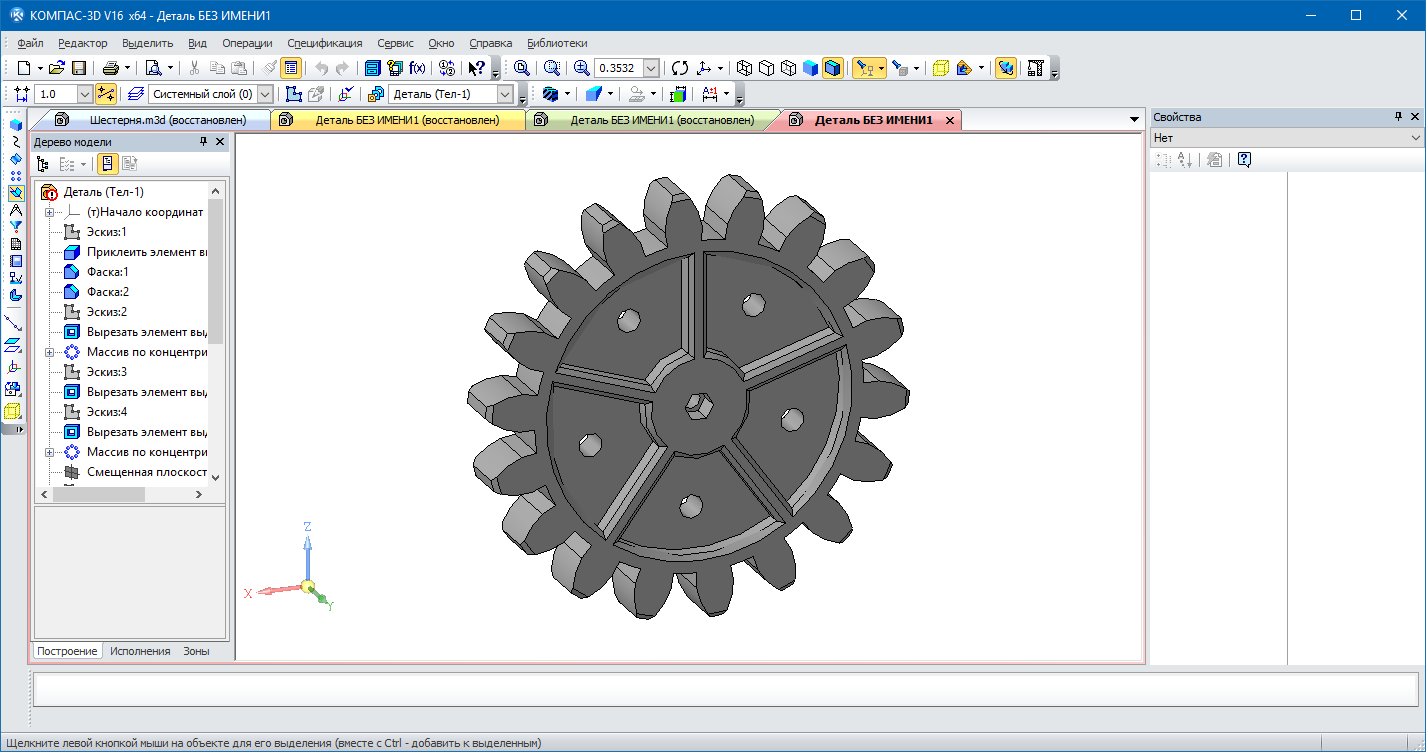


Рисунок . – Вид на 3D модель шестерни

## Описание технических и функциональных аспектов проекта

Для графического описания абстрактной модели проекта, а также пользовательского взаимодействия (сценария действий) использован стандарт UML[9].

UML язык графического описания для объектного моделирования в области разработки программного обеспечения. UML является языком широкого профиля, это – открытый стандарт, использующий графические обозначения для создания абстрактной модели системы, называемой UML-моделью. UML был создан для определения, визуализации, проектирования и документирования, в основном, программных систем. UML не является языком программирования, но на основании UML-моделей возможна генерация кода.

При использовании UML были построены: диаграмма использования и диаграмма классов.

* + 1. Диаграмма вариантов использования (Use Cases)

На рисунке 1.2 представлена диаграмма вариантов использования.



Рисунок . – Диаграмма вариантов использования

* + 1. Диаграмма классов

На рисунке 1.3 показана диаграмма классов.



Рисунок . – Диаграмма классов

В приложении А приведены описания данных классов.

* + 1. Макет пользовательского интерфейса

Плагин представляет собой пользовательскую форму с ячейками для ввода параметров. Запуск построения осуществляется кнопкой «Build». На рисунке 1.4 представлен интерфейс плагина.

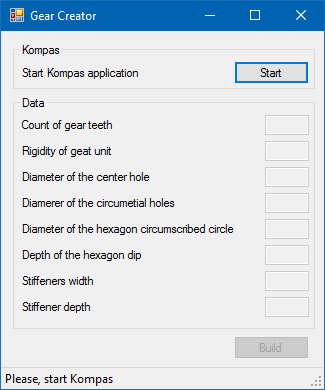


Рисунок . – Пользовательский интерфейс при запуске программы

# Описание программы для пользователя

Чтобы построить модель шестерни, используя данный плагин, необходимо запустить плагин. В запущенном окне нажать на кнопку «Start», чтобы запустить приложение КОМПАС-3D или сделать его активным, в случае если он уже запущен. Далее необходимо ввести данные - параметры шестерни. При вводе некорректных значений программа выдаст сообщение об ошибке, в котором будет указан диапазон корректных значений. После ввода всех значений, нужно нажать на кнопку «Build», чтобы создать модель шестерни на рабочей плоскости программы КОМПАС-3D.

# Тестирование программы

## Функциональные тесты

Функциональное тестирование — это тестирование в целях проверки реализуемости функциональных требований, то есть способности программного обеспечения (ПО) в определённых условиях решать задачи, нужные пользователям. Функциональные требования определяют, что именно делает ПО, какие задачи оно решает.

При запуске плагина открывается окно, показанное на рисунке 3.1. Ячейки для ввода данных недоступны, а строка состояния уведомляет о необходимости запуска программы «КОМПАС-3D» нажатием на кнопку «Start». После нажатия кнопки пользовательский интерфейс изменится на вид, показанный на рисунке 3.2. Если при запуске плагина «КОМПАС-3D» был уже открыт, то при запуске плагина сразу отобразится окно, представленное на рисунке 3.2. Появляется возможность вводить данные – основные параметры шестерни.

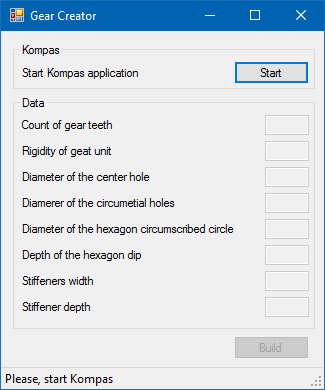


Рисунок . – Окно плагина без запущенной программы

«КОМПАС-3D»

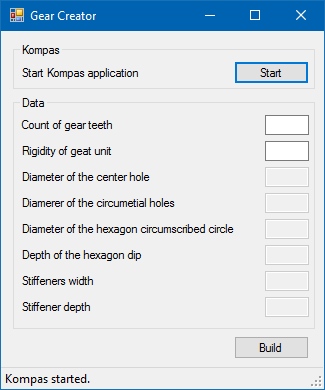


Рисунок . – Окно плагина с запущенной программой «КОМПАС-3D»

Далее, при попытке ввода неправильных символов в ячейки ввода параметров, программа выдает нам ошибку проверки данных (рисунок 3.3). После ввода независимых параметров, появляется возможность ввода зависимых параметров, кроме диаметра шестиугольника (рисунок 3.4), так как он дополнительно зависит от диаметра отверстия в центре.

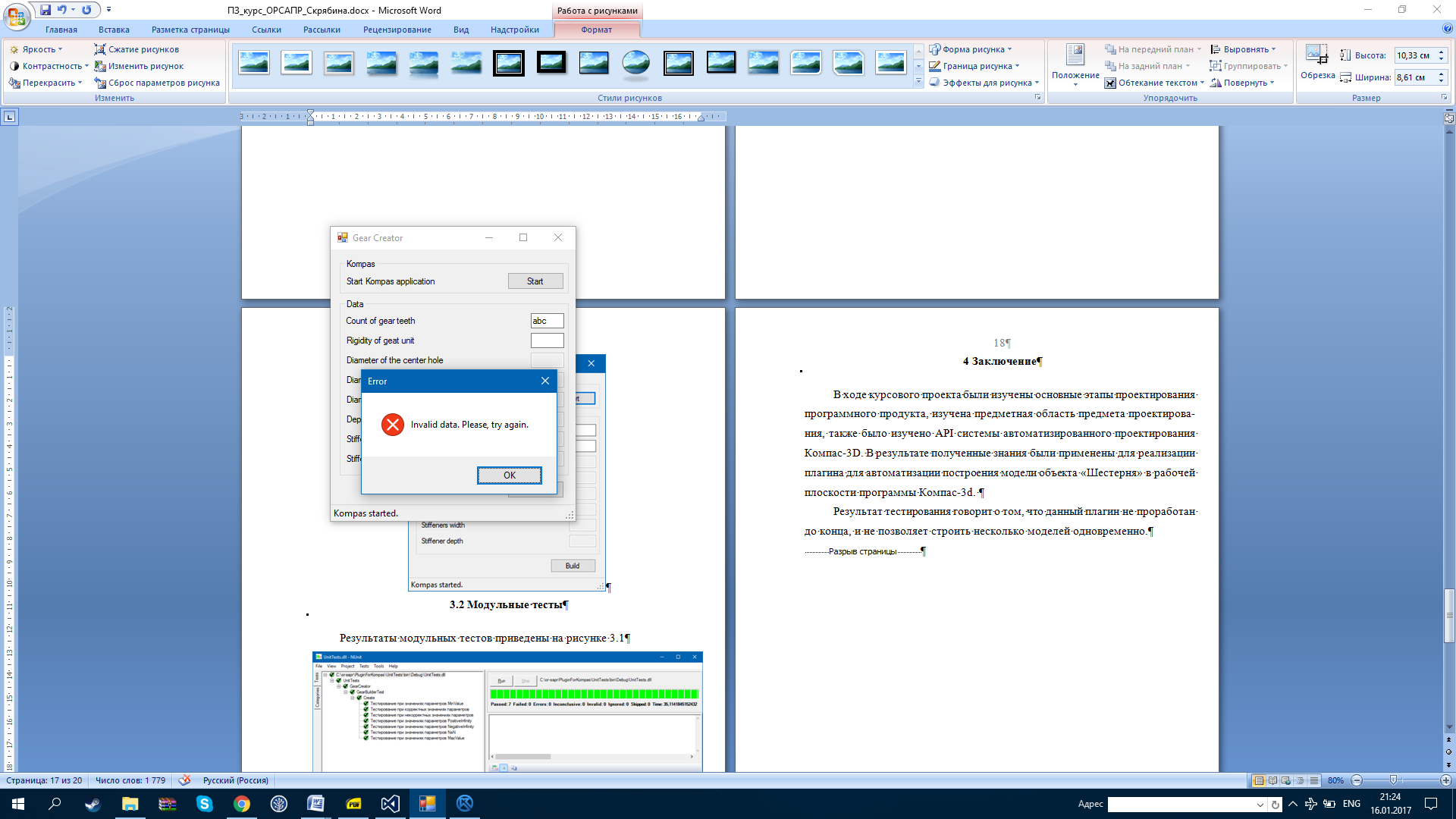


Рисунок . – Ошибка проверки

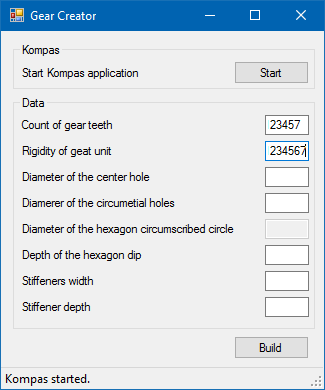


Рисунок . – Реакция программы при вводе допустимых символов

После ввода диаметра центрального отверстия, становятся доступны все ячейки ввода данных (рисунок 3.5). Тестируется ввод некорректных данных (рисунок 3.6).

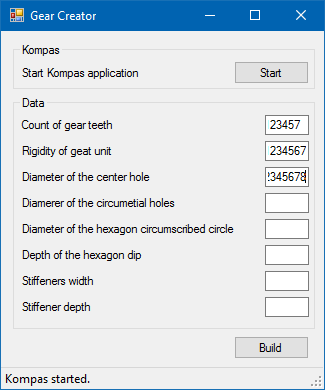


Рисунок . – Ввод диаметра отверстия

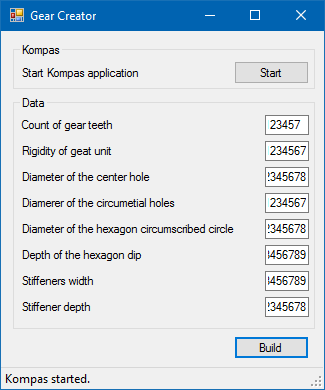


Рисунок . – Ввод недопустимых данных

При нажатии на кнопку «Build» программа выдаст ошибку (рисунок 3.7).

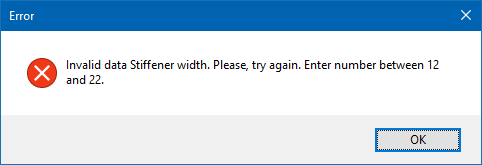
****

Рисунок . – Ошибка недопустимых значений

Данная ошибка будет выводиться до тех пор, пока пользователь не введет допустимые значения параметров во все ячейки ввода данных (рисунок 3.8).

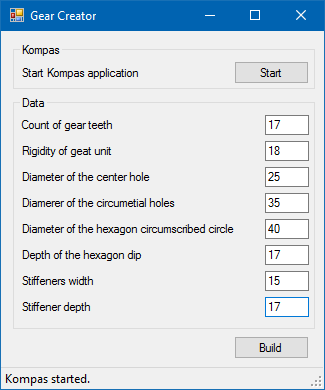
****

Рисунок . – Один из допустимых вариантов значения параметров

При вводе корректных данных и нажатии кнопки «Build», на рабочей плоскости «КОМПАС-3D» построится модель детали (рисунок 3.9).

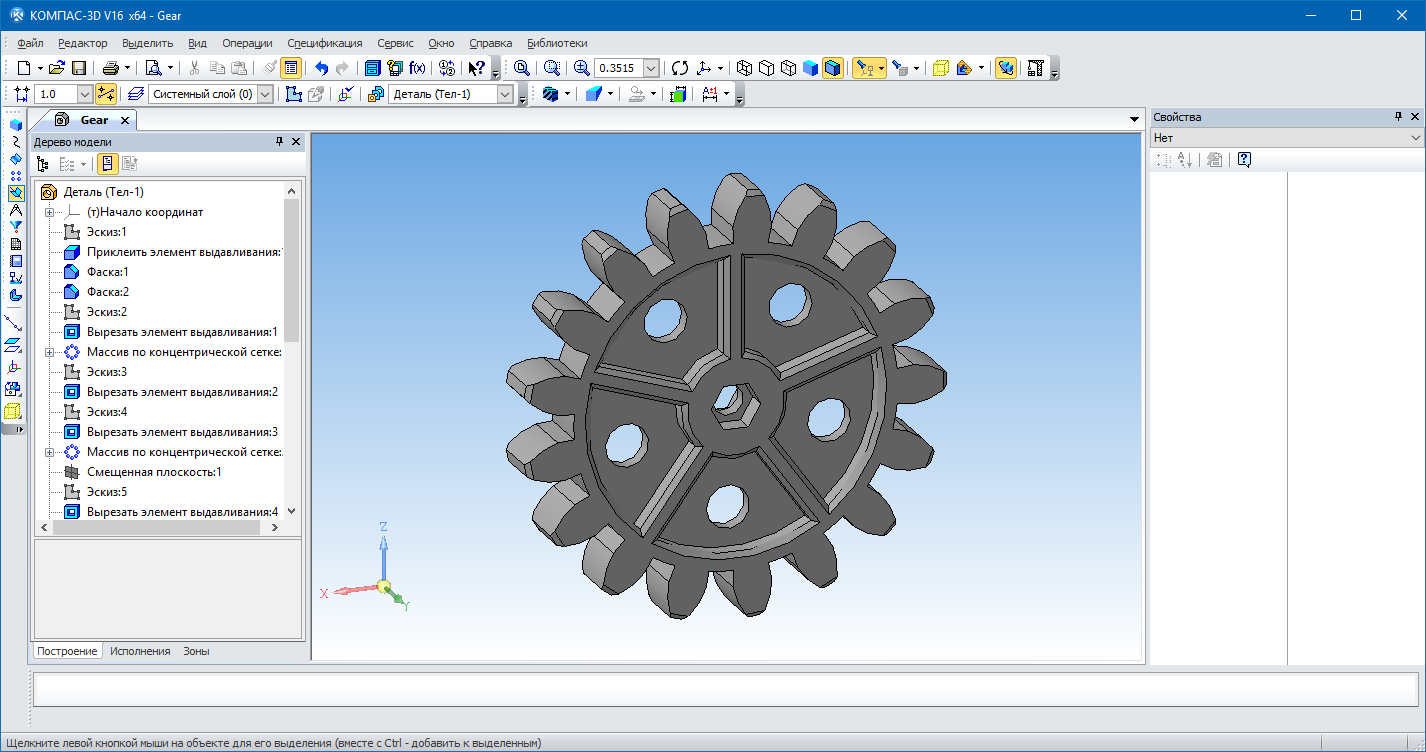


Рисунок . – Построенная модель шестерни

Ввод минимальных допустимых значений (рисунок 3.10).

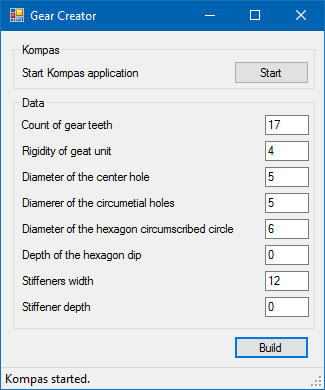


Рисунок . – Ввод минимально допустимых значений

Результат при вводе минимально допустимых значений показан на рисунке 3.11.

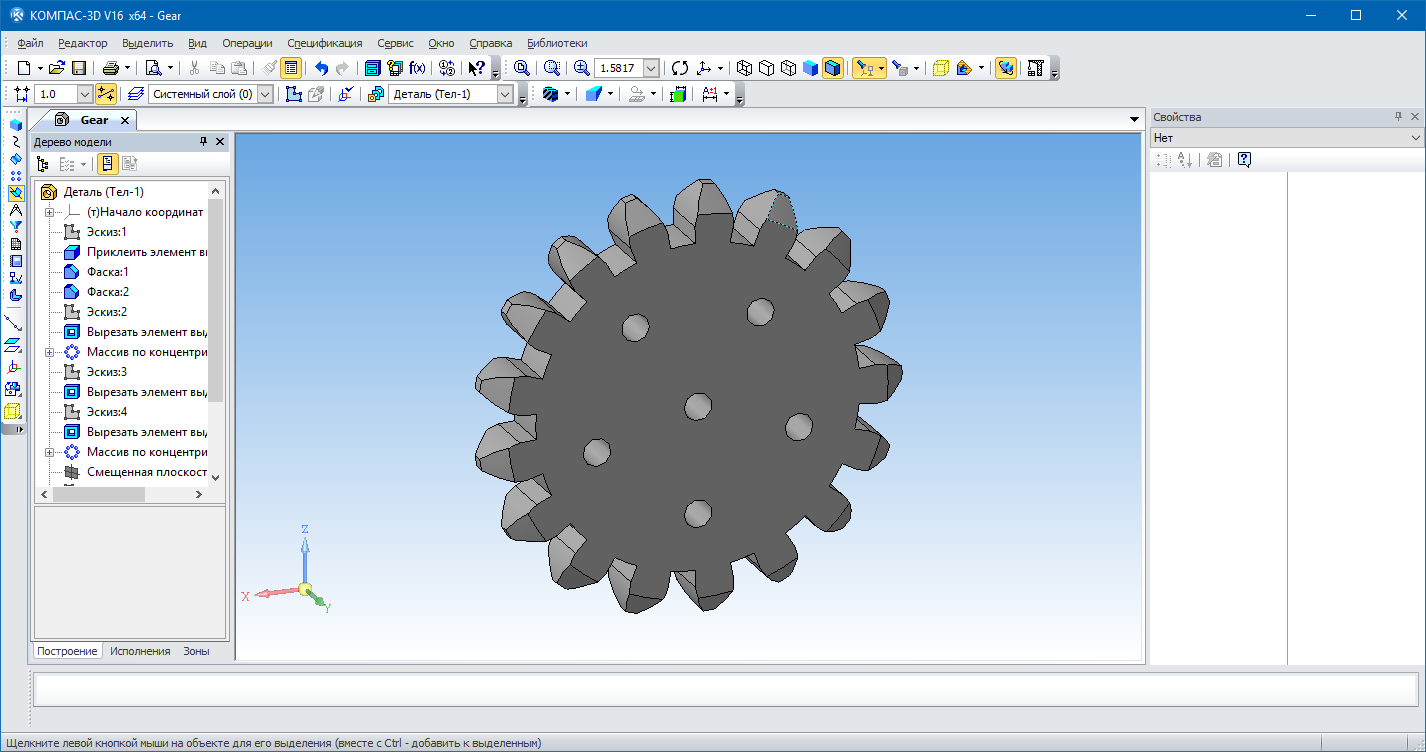


Рисунок . – Результат ввода минимально допустимых значений

Ввод максимально допустимых значений (рисунок 3.12).

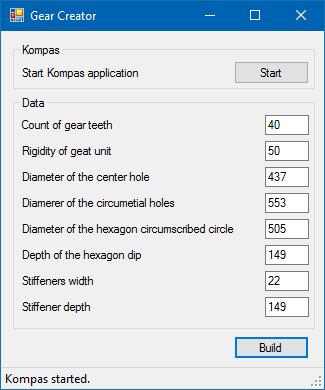


Рисунок . – Ввод максимально допустимых значений

Результат при вводе максимально допустимых значений показан на рисунке 3.13.

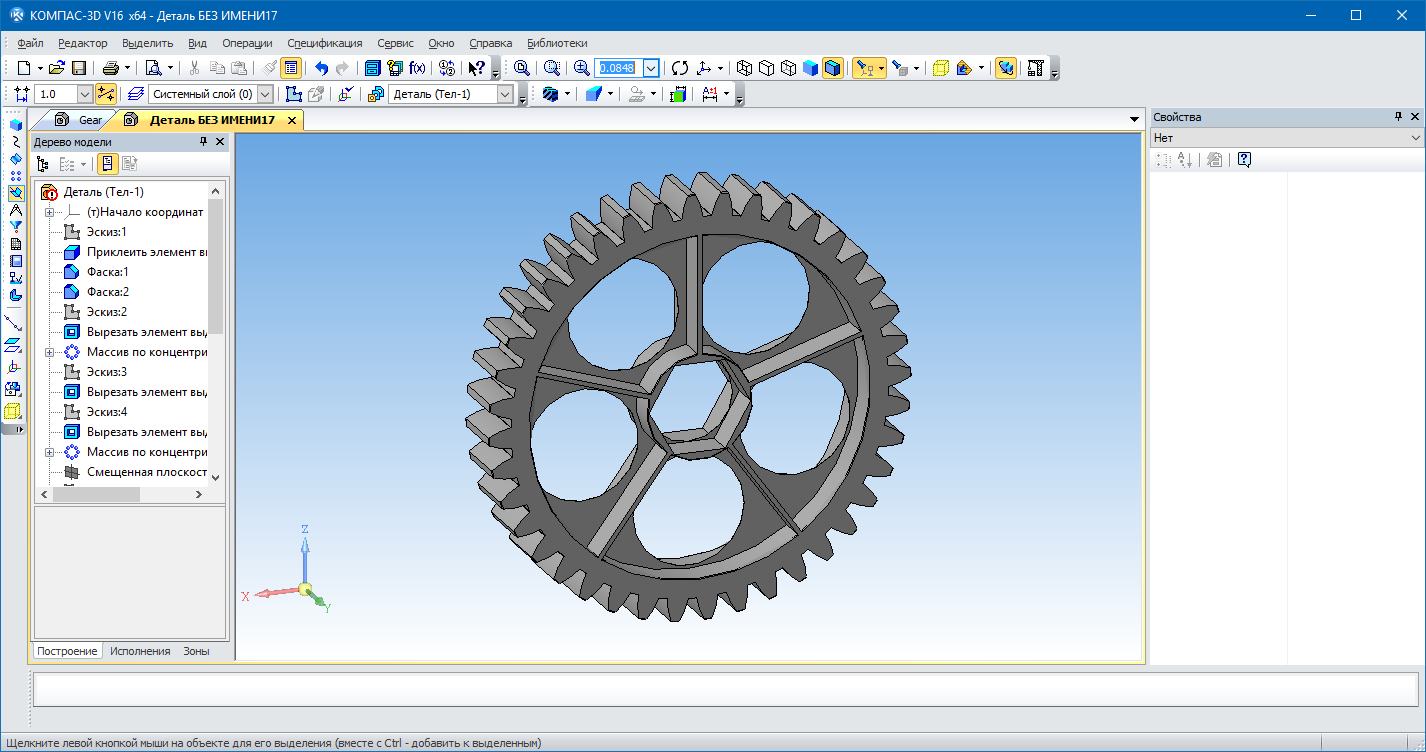


Рисунок . – Результат ввода максимально допустимых значений

## Модульные тесты

В приложении Б приведены значения входных параметров при модульном тестировании в виде листинга части программы. Результаты модульных тестов приведены на рисунке 3.14.

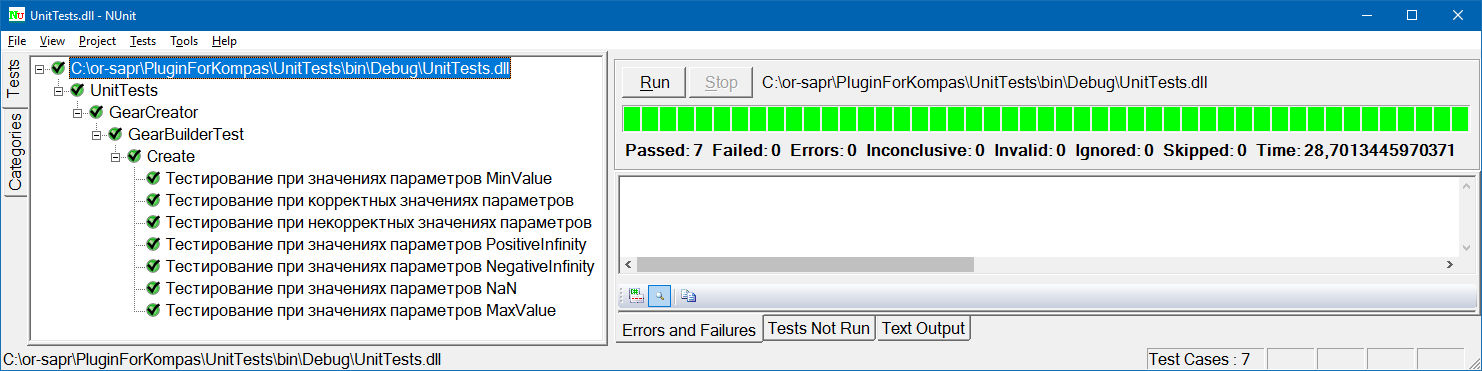


Рисунок . – Результаты модульных тестов

## Нагрузочные тесты

На рисунке 3.15 нарисованы графики зависимости от количества непрерывно-строящихся деталей (ось х) объема памяти в MB, используемой «КОМПАС-3D» – g(x) и уровень загрузки ЦП процессом в процентах – f(x).



Рисунок . – Результаты нагрузочных тестов

По графикам можно определить, что зависимость прямая. То есть используемая памяти и загрузка на ЦП, при увеличении количества деталей, увеличивается линейно.

# Заключение

В ходе курсового проекта были изучены основные этапы проектирования программного продукта, изучена предметная область предмета проектирования, также было изучено API системы автоматизированного проектирования КОМПАС-3D. В результате полученные знания были применены для реализации плагина для автоматизации построения модели объекта «Шестерня» в рабочей плоскости программы КОМПАС-3D.

# Список литературы

Норенков И.П. «Основы автоматизированного проектирования». Издательство: МГТУ; Москва.:, 2002. – 336 с.

API – Википедия. [Электронный ресурс]. — URL: <https://ru.wikipedia.org/wiki/API> (дата обращения 27.12.2016)

Плагин – Википедия. [Электронный ресурс]. — URL: <https://ru.wikipedia.org/wiki/Плагин> (дата обращения 27.12.2016)

КОМПАС-3D: О программе. Официальный сайт САПР КОМПАС [Электронный ресурс]. – URL: <http://kompas.ru/kompas-3d/about/> (дата обращения 27.12.2016).

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

Плагин PDF [Электронный ресурс]. — URL: <http://gkmsoft.ru/allcatalog/pdf2dkompas_plugin/> (дата обращения 27.12.2016)

3D PDF [Электронный ресурс]. — URL: <http://sapr-journal.ru/novosti/eksport-iz-kompas-3d-v-formate-3d-pdf/> (дата обращения 27.12.2016)

Зубчатое колесо – Википедия. [Электронный ресурс]. – URL: <http://ru.wikipedia.org/wiki/Зубчатое_колесо> (дата обращения 27.12.2016).

UML. [Электронный ресурс]. — Режим доступа: <http://www.uml.org/> (дата обращения 27.12.2016).

# Приложение А

(справочное)

Описания классов

В таблицах приложения для обозначения модификаторов доступа полей приняты следующие условные знаки:

- «#» − обозначение protected (защищенного) поля;

- «−» − обозначение private (открытого) поля;

- «+» − обозначение public (открытого) поля.

Таблица A.1 – Описание класса GearConstuctingForm

|  |  |  |
| --- | --- | --- |
| Название | Тип | Описание |
| Описание класса | | |
| Класс GearConstuctingForm – сущность для описания пользовательского интерфейса | | |
| Поля | | |
| - \_rigidity | double | Переменная для хранения значения модуля жесткости шестерни |
| - \_teethCount | double | Переменная для хранения значения количества зубьев шестерни |
| Методы | | |
| + GearConstuctingForm () |  | Конструктор класса |
| - buildButton\_Click(object, EventArgs) | void | События при нажатии на кнопку «Build» |
| - changeEnable() | void | Метод для изменения параметра видимости объектов на окне |
| - dataValidating(object, EventArgs) | void | Проверка вводимых символов |
| - readData() | Parameter[] | Метод возвращает динамический массив введенных данных |

Таблица А.2 – Описание класса GearBuilder

|  |  |  |
| --- | --- | --- |
| Название | Тип | Описание |
| Описание класса | | |
| Класс GearConstuctingForm – сущность для описания пользовательского интерфейса | | |
| Поля | | |
| - \_angle54 | double | Угла 54 градусов |
| - \_baseCircle | double | Диаметр делительной окружности |
| - \_chamferWidth | double | Длина фаски |
| - \_externalArcOfDip | double | Диаметр внешней окружности углубления |
| - \_internalArcOfDip | double | Диаметр внутренней окружности углубления |
| - \_mainCircle | double | Диаметр основной окружности |
| - \_projectionCircle | double | Диаметр окружности выступов |
| - \_troughsCircle | double | Диаметр окружности впадин |
| - \_list | Parameter[] | Входные параметры детали |
| Методы | | |
| + GearBuilder(Parameter[]) |  | Конструктор класса с параметром |
| + CreateGear() | bool | Метод для создания нового документа для построения детали |
| - DoDetail() | void | Метод для пошагового построения детали |

Таблица А.3 – Описание класса Parameter

|  |  |  |
| --- | --- | --- |
| Название | Тип | Описание |
| Описание класса | | |
| Класс Parameter – структура для параметров шестерни | | |
| Поля | | |
| + Description | string | Описание параметра |
| + Value | double | Значение параметра |
| Методы | | |
| + Parameter (string, double) |  | Конструктор класса |

Таблица А.4 – Описание класса DataController

|  |  |  |
| --- | --- | --- |
| Название | Тип | Описание |
| Описание класса | | |
| Класс DataController – сущность для проверки данных | | |
| Поля | | |
| - \_centerHoleDiam | double | Диаметр отверстия в центре |
| - \_rigidity | double | Модуль жесткости |
| - \_teethCount | double | Количество зубъев |
| Методы | | |
| + Validating(double, double, double, double, Parameter[]) | bool | Метод для проверки списка параметров |
| CheckData(Parameter, double, double) | bool | Метод проверяет параметр |

Таблица А.5 – Описание класса KompasApp

|  |  |  |
| --- | --- | --- |
| Название | Тип | Описание |
| Описание класса | | |
| Класс KompasApp – сущность для работы с приложением КОМПАС | | |
| Поля | | |
| + basePlane | ksEntity | Указатель на базовую плоскость |
| + Doc3d | ksDocument3D | Указатель на интерфейс трехмерного графического документа |
| + Kompas | KompasObject | Указатель на главный интерфейс API |
| + Mat | ksMathematic2D | Указатель на интерфейс для работы с математическими функциями |
| + Part | ksPart | Указатель на интерфейс главного компонента |
| Методы | | |
| + CreateNewDoc | bool | Метод для создания нового трехмерного графического докумета |
| + GetActiveApp | bool | Метод получает указатели активного приложения или создает новые |

# Приложение Б

(справочное)

Входные параметры модульных тестов

[TestCase(true, 18, 15, 17, 35, 40, 30, 17, 18, TestName = "Тестирование при корректных значениях параметров")]

[TestCase(false, 18, 15, 17, 35, 40, 30, 17, 1, TestName = "Тестирование при некорректных значениях параметров")]

[TestCase(false, double.PositiveInfinity, double.PositiveInfinity, double.PositiveInfinity, double.PositiveInfinity, double.PositiveInfinity, double.PositiveInfinity, double.PositiveInfinity, double.PositiveInfinity, TestName = "Тестирование при значениях параметров PositiveInfinity")]

[TestCase(false, double.NegativeInfinity, double.NegativeInfinity, double.NegativeInfinity, double.NegativeInfinity, double.NegativeInfinity, double.NegativeInfinity, double.NegativeInfinity, double.NegativeInfinity, TestName = "Тестирование при значениях параметров NegativeInfinity")]

[TestCase(false, double.NaN, double.NaN, double.NaN, double.NaN, double.NaN, double.NaN, double.NaN, double.NaN, TestName = "Тестирование при значениях параметров NaN")]

[TestCase(false, double.MaxValue, double.MaxValue, double.MaxValue, double.MaxValue, double.MaxValue, double.MaxValue, double.MaxValue, double.MaxValue, TestName = "Тестирование при значениях параметров MaxValue")]

[TestCase(false, double.MinValue, double.MinValue, double.MinValue, double.MinValue, double.MinValue, double.MinValue, double.MinValue, double.MinValue, TestName = "Тестирование при значениях параметров MinValue")]