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)

**РАЗРАБОТКА ПЛАГИНА «СВЕРЛО» ДЛЯ «КОМПАС-3D v18.1»**

Пояснительная записка к лабораторному проекту

по дисциплине «Основы разработки САПР»

Выполнил:

студент гр.587-2

\_\_\_\_\_\_\_\_\_\_\_\_\_\_Пилипенко П. С.

«\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2021 г.

Руководитель:

к.т.н., доцент каф. КСУП

\_\_\_\_\_\_\_\_\_\_\_\_\_\_Калентьев А. А.

«\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2021 г.

# 1 Описание САПР КОМПАС 3D

## Описание программы

Компас 3D – это система трёхмерного проектирования, используемая для конструирования трехмерных моделей. В его разработку входят объекты различного уровня сложности: от простых деталей и небольших сборных механизмов до инженерных сетей, высотных зданий, суперсовременных машин и высокоточных приборов.

Программа обеспечивает создание объемных ассоциативных моделей, как простейших предметов, так и сложных сборочных конструкций, содержащих стандартизированные и оригинальные элементы. Применяется при разработке технологической и конструкторской документации, ускоряет переход от составления проекта к непосредственному производству или строительству реального объекта.

Компас-3D содержит набор инструментов, упрощающих построение:

– поверхностей различных конфигураций;

– структурных элементов (ребер жесткости, отверстий, желобков и пр.);

– вспомогательных элементов (прямых и ломаных линий, спиралей и т.д.).

Компас распознает популярные форматы IGES, DWG, ACIS, STEP и DXF, которые используются для хранения трехмерных моделей, созданных в приложении, и облегчают взаимодействие между пользователями систем CAD, CAE и CAM [1].

## Описание API САПР КОМПАС 3D

API (англ. Application Programming Interface — программный интерфейс приложения) представляет собой совокупность различных инструментов, функций, реализованных в виде интерфейса для создания новых приложений, благодаря которому одна программа будет взаимодействовать с другой. API-технология предоставляет программисту набор процедур и функций для управления САПР, но не дает прямого доступа к свойствам и методам объектов внутри САПР.

Расчетный модуль помимо собственно расчетов должен выполнять следующие функции:

– подключение к КОМПАС 3D и загрузка в него параметрической сборки;

– получение текущих значений переменных деталей, входящих в сборку, названий деталей и названия самой сборки;

– изменение значений переменных, перестроение и сохранение модели.

Система КОМПАС в API 5 описывается интерфейсом KompasObject. Методы данного 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) [2].

Главные методы данного интерфейса приведены в таблице 1.1.

Таблица 1.1 – Методы интерфейса KompasObject

|  |  |  |  |
| --- | --- | --- | --- |
| Метод | Тип входных данных | Тип выходных данных | Описание |
| ActiveDocument3D |  | указатель на интерфейс до­кумента трехмерной модели ksDocument3D | Дает возможность получить указатель на активный трехмерный документ |
| Document3D |  | указатель на интерфейс документа | Дает возможность получить указатель на интерфейс трехмерного |
| Продолжение таблицы 1.1 | | | |
|  |  | трехмерной модели ksDocument3D | документа (детали или сборку) |
| Visible |  |  | Свойство видимости приложения |

Графические документы имеют собственный интерфейс – ksDocument2D, со своими специфическими свойствами и методами. С помощью функций, присутствующих в ksDocument2D, создаются изображения в эскизах трехмерных операций.

Методы этого интерфейса позволяют программно управлять (без выполнения трехмерных операций) трехмерным документом, как сборкой и ее компонентами, так и отдельной деталью. Наиболее используемые из них приведены в таблице 1.2.

Таблица 1.2 – Методы интерфейса ksDocument3D

|  |  |  |  |
| --- | --- | --- | --- |
| Метод | Тип входных данных | Тип выходных данных | Описание |
| Create | invisible-признак режима редактирования документа,  typeDoc-тип документа | TRUE - в случае успешного завершения | Дает возможность создать пустой документ (деталь или сборку) |
| GetDefaultEntity | objType - тип объекта | указатель на интерфейс ksEntity или IEntity | Получить указатель на интерфейс объекта, создаваемого системой по умолчанию |
| NewEntity | objType - тип объекта | указатель на | Создать новый интерфейс объекта |
| Продолжение таблицы 1.2 | | | |
|  |  | интерфейс ksEntity или IEntity | и получить  указатель на него |

Метод ksDocument3D::GetPart возвращает указатель на интерфейс детали или компонента сборки – ksPart. Свойства и методы этого интерфейса управляют состоянием компонентов сборки, они почти полностью дублируют команды контекстного меню и панели свойств, доступные пользователю при работе с тем или иным компонентом.

Некоторые методы приведены в таблице 1.3.

Таблица 1.3 – Методы интерфейса ksPart

|  |  |  |  |
| --- | --- | --- | --- |
| Метод | Тип входных данных | Тип выходных данных | Описание |
| BeginEdit |  | указатель на интерфейс эскиза ksDocument2D | Запускает режим редактирование компонента |
| EndEdit |  | TRUE - в случае успешного завершения | Закрывает режим редактирование компонента |
| GetPart | type - тип компонента objectTypе - тип объекта | указатель на интерфейс компонента ksPart или IPart указатель на интерфейс ksEntity или IEntity | Позволяет получить указатель на интерфейс компонента Создает интерфейс нового трехмерного объекта и возвращает указатель на него |

## 1.3 Обзор аналогов

### 1.3.1 Валы и механические передачи 3D

Приложение предназначено для автоматизации проектирования и построения трехмерных моделей валов, втулок, элементов механических передач и различных конструктивных элементов в среде КОМПАС-3D.

Встроенные расчетные модули, каталоги материалов и стандартных изделий помогают инженеру создавать модели узлов и механизмов в кратчайшие сроки. Результаты проектировочных и прочностных расчетов могут быть представлены в виде отчетов и сохранены в любом удобном формате. Стандартные средства КОМПАС-График позволяют быстро оформить конструкторскую документацию в соответствии с требованиями нормативных документов.

С помощью средств приложения «Валы и механические передачи 3D**»** могут быть спроектированы следующие элементы механических передач: шестерни цилиндрические с внешним и внутренним зацеплением;

* шестерни цилиндрические винтовых эвольвентных передач;
* зубчатые рейки;
* шестерни конические с прямым зубом;
* шкивы клиноременных передач;
* звездочки приводных роликовых цепей;
* червяки и червячные колёса (цилиндрическая червячная передача);
* зубчатые глухие муфты.

Пользователям доступны геометрические и проектные расчеты, расчеты передач на прочность и долговечность, а также оптимизационные расчеты.

«Валы и механические передачи 3D. Дополнительный модуль» расширяет список проектируемых передач. К основному перечню элементов добавляются элементы следующих передач:

* конической передачи с круговыми зубьями;
* конической передачи с тангенциальными зубьями;
* червячной глобоидной передачи;
* зубчаторемённой передачи;
* зубчатой соединительной муфты;
* цевочной передачи.

Кроме того, средствами дополнительного модуля может быть выполнен восстановительный расчет цилиндрической передачи внешнего зацепления.

Другой дополнительный модуль «Часовые механизмы» предназначен для проектирования приборов времени.

Особенности работы с данным приложением следующие:

1. Возможность начать проектирование изделия с его 3D-модели или с чертежа.

2. Возможность оптимизационного проектирования.

3. Проектировать детали можно не только в соответствии с отечественной нормативной базой (ГОСТ, ОСТ), но и по стандартам других стран (AGMA, ASA, DIN, ISO, JIS, GBT).

1. При проектировании зубчаторемённых и клиноременных шкивов ремень можно выбрать как по отечественным стандартам, так и по каталогам компании Optibelt (Германия).
2. 2D-профили зубчатых венцов и генерируемые 3D-модели абсолютно правильны и геометрически корректны. Они безо всяких ограничений могут быть использованы при создании программ для станков с ЧПУ [3].

# 2 Описание предметов проектирования

Предметом данного проектирование является сверло спиральное.

Сверло – режущий инструмент, который фиксируется в патроне дрели или станке, предназначенный для сверления отверстий в различных материалах.

Сверло состоит из двух основных частей: корпус и хвостовик.

Корпус – часть сверла от хвостовика до вершины режущей кромки. На данной модели эту часть занимает рабочая часть (l) и шейка – часть корпуса с уменьшенным диаметром (d).

В свою очередь хвостовик – часть сверла, предназначенная для закрепления и передачи крутящего момента. На данной модели в эту часть входит лапка – плоский конец конического хвостовика, предназначенный для крепления в прорези переходной втулки (а).

Параметры спирального сверла:

– Длина сверла L (от 3 до 145 мм);

­­– Длина рабочей части l (от 1,5 до 129 мм);

– Диаметр сверла D (от 0,25 до 22 мм);

– Длина лапки a (от 0,45 до 22 мм);

– Ширина лапки b (от 0,24 до 20 мм);

– Длина шейки d (от 0,45 до 10 мм);

– Ширина шейки c (от 0,24 до 20 мм);

Параметры имеют следующие зависимости:

1. D > c;

2. D > b;

3. l > D.

4. L – (l + a + d) > 5 мм.

На рисунке 2.1 представлен чертеж сверла с указанными параметрами.

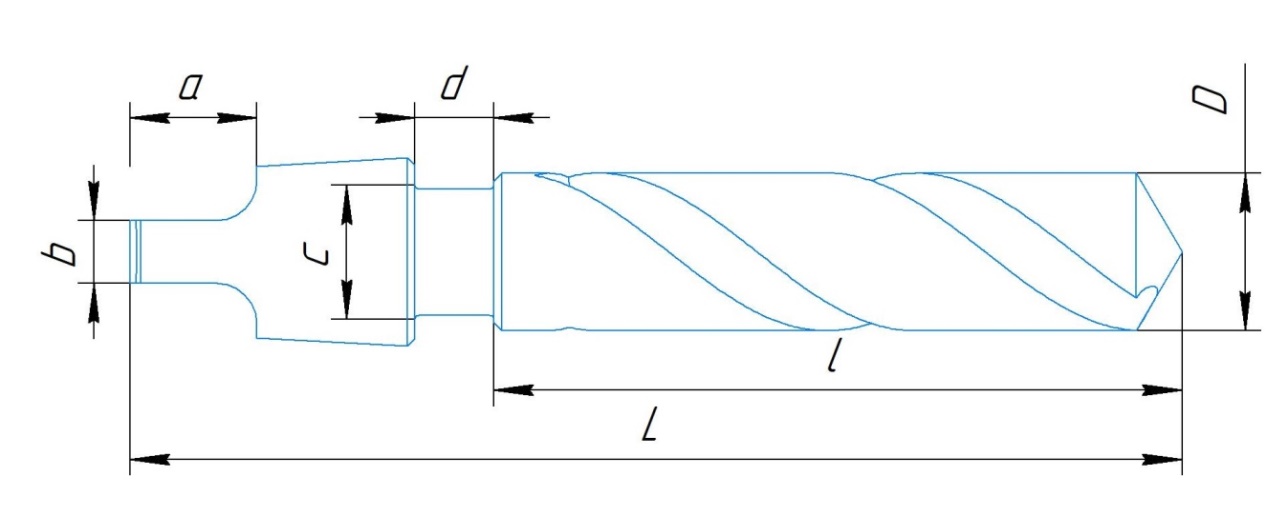


Рисунок 2.1 – 2D чертеж сверла

# 3 Проект программы

## 3.1 Диаграмма классов

Диаграмма классов описывает типы классов системы и различного рода статические связи, которые существуют между ними. На диаграммах классов изображаются также свойства классов, операции классов и ограничения, которые накладываются на связи между классами [4].

Диаграмма классов для плагина изображена на рисунке 3.1, на рисунке 3.2 изображена диаграмма классов после реализации и добавления дополнительной функциональности.

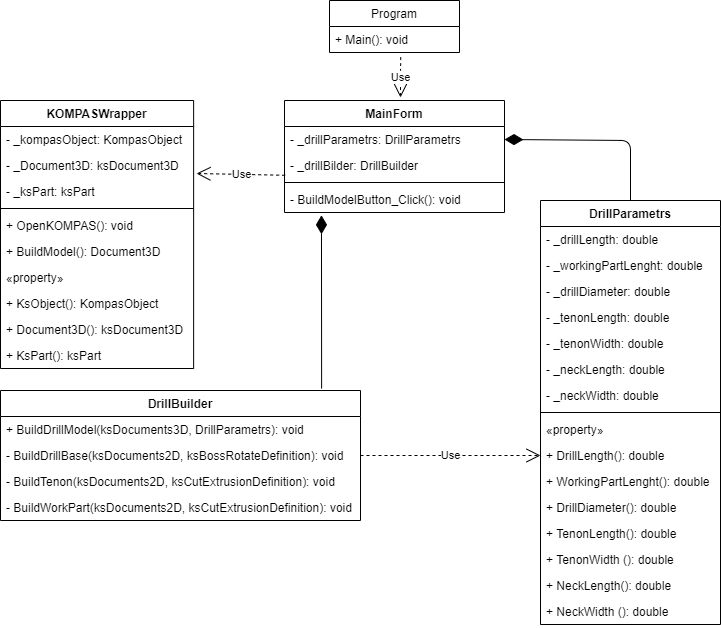


Рисунок 3.1 – UML-диаграмма классов

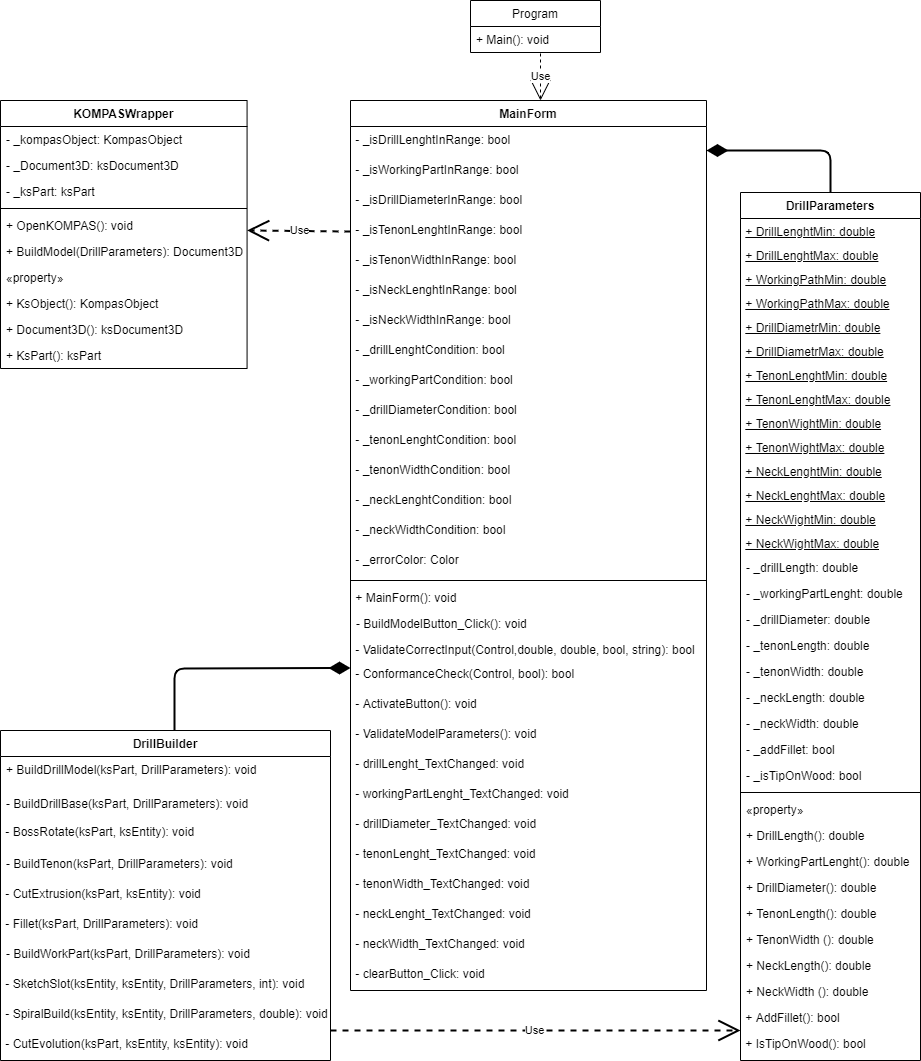


Рисунок 3.2 – UML-диаграмма классов после реализации

После разработки плагина, диаграмма классов была дополнена.

В класс Main были добавлены поля-флаги для каждого из параметров, отражающие отвечает ли параметр установленным зависимостям и входит ли значение в допустимый диапазон.

Добавлено поле, отвечающие за цвет, в которой окрашиваются текстовые поля при ошибке.

Написаны два метода ValidationCorrectInput и ConformanceCheck, проверяющие входит ли параметр, введённый пользователем, в диапазон и удовлетворяет ли он условиям, установленными зависимостями. При неудачной проверке соответствующий TextBox окрашивается в цвет, и рядом с ним всплывает ToolTip с ошибкой.

Добавлен метод ActivButton, который активирует кнопку, если все условия ввода параметров удовлетворены.

Написан метод ValidationModelParameters, передающий параметры с формы в проект.

Так же написаны методы для каждого TextBox, которые при изменении текста проверяют введённые в него символы на удовлетворение всем условиям и ограничениям.

Добавлен метод ClearButton, который реализует функцию отчистки полей.

В класс DrillParametrs добавлены константные поля, отвечающие за минимальное и максимальное значения каждого параметра. Добавлены автосвойства, отвечающие за тип кончика сверла и за присутствие округления на лапке и шейки.

В класс DrillBuider были добавлены методы BossRotate, CutExtrusion, Fillet, SketchSlot, SpiralBuid, CutEvolution, реализующие отдельные операции моделирования для каждого из этапов построения.

## 3.2 Макет пользовательского интерфейса

Пользовательский интерфейс представляет собой форму для ввода параметров. На форме присутствует чертеж для демонстрации параметров на детали, поля для ввода пользовательских значений, прописанные ограничения для этих параметров и две флаговые кнопки, реализующие дополнительную функциональность.

Пользователю разрешено вводить в поля только числа, количество которых установлено в зависимости от диапазона значений.

После верного заполнения пользователем всех полей активируется кнопка «Построить 3D-модель сверла».

При нажатии на кнопку «Построить 3D-модель сверла», строиться 3D-модель сверла по параметрам в заполненных полях. Иначе редактируемое поле подсвечивается цветом и выводится сообщение об ошибке.

На рисунке 3.3 представлен макет пользовательского интерфейса

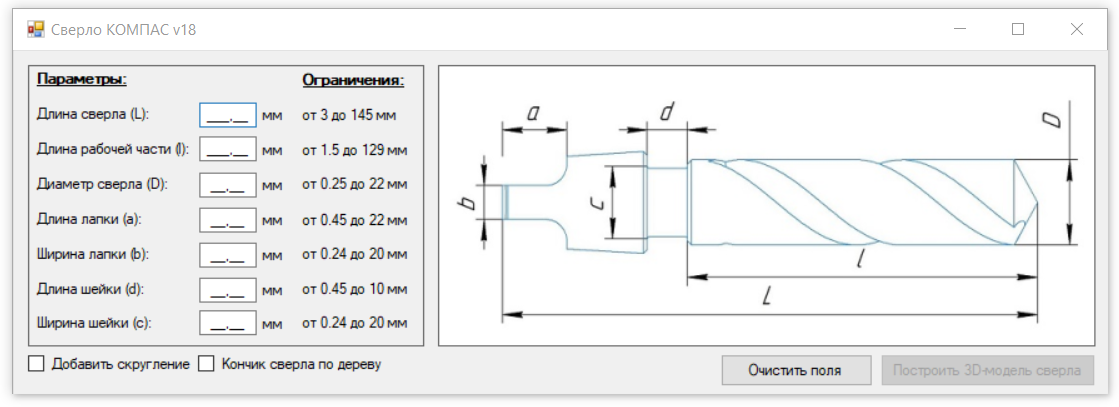


Рисунок 3.3 – Макет пользовательского интерфейса для плагина сверла

Обработка ошибок производится следующим образом. Если при вводе определенного параметра был допущен выход за границы диапазона допустимых значений или была нарушена зависимость параметров, то поле становится цветным, а рядом с ним отображается подсказка об ошибке (toolTip), как продемонстрированно на рисунке 3.4 и 3.5



Рисунок 3.4 – Пример обработки ошибки при выходе из диапазона значений

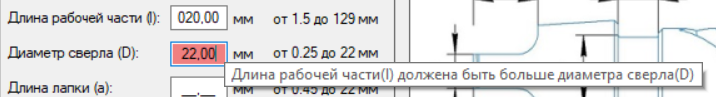


Рисунок 3.4 – Пример обработки ошибки при не соблюдении зависимостей параметров

# 4 Тестирование

## 4.1 Функциональное тестирование

В ходе данного функционального тестирования будет проверятся построения 3D-модели сверла при минимальных значения, максимальных и стандартных значениях, а так же реакцию плагина на ввод некорректных значений.

Минимальные параметры модели:

– Длина рабочей части (L): 3 мм;

– Длина рабочей части (l): 1.5 мм;

– Диаметр сверла (D): 0.25 мм;

– Длина лапки (a): 0.45 мм;

– Ширина лапки (b): 0.24 мм;

– Длина шейки (d): 0.45 мм;

– Ширина шейки (c): 0.24 мм.

На рисунках 4.1 и 4.2 изображены 3D-модели сверла, построенные по минимальным параметрам. Для данной модели нельзя построить операцию скругления, так как минимальный радиус скругления 1 мм.

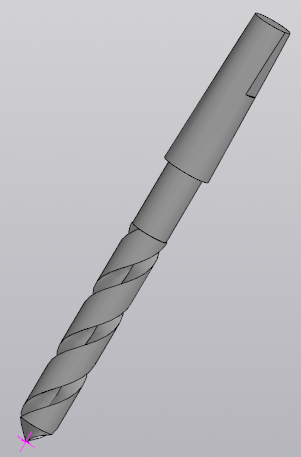


Рисунок 4.1 – 3D-модель сверла, построенная по минимальным параметрам, имеющая стандартный кончик

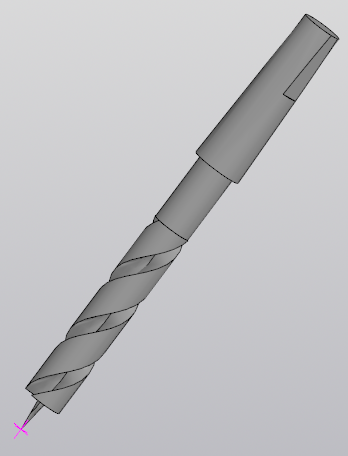


Рисунок 4.1 – 3D-модель сверла, построенная по минимальным параметрам, имеющая кончик сверла по дереву

Максимальные параметры модели:

– Длина рабочей части (L): 145 мм;

– Длина рабочей части (l): 129 мм;

– Диаметр сверла (D): 22 мм;

– Длина лапки (a): 5 мм;

– Ширина лапки (b): 18 мм;

– Длина шейки (d): 5 мм;

– Ширина шейки (c): 18 мм.

На рисунках 4.3 и 4.4 изображены 3D-модели сверла, построенные по максимальным параметрам.

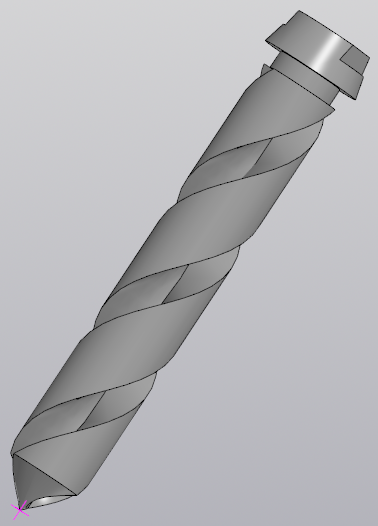


Рисунок 4.3 – 3D-модель сверла, построенная по максимальным параметрам, имеющая стандартный кончик и не имеющая скругления

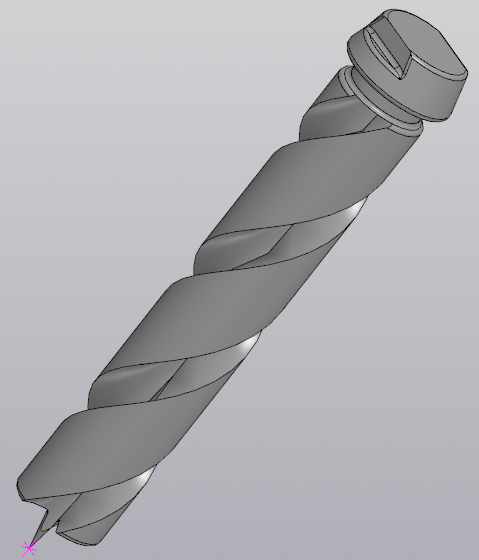


Рисунок 4.3 – 3D-модель сверла, построенная по максимальным параметрам, имеющая кончик сверла по дереву и скругление

Стандартные параметры:

– Длина рабочей части (L): 90 мм;

– Длина рабочей части (l): 61 мм;

– Диаметр сверла (D): 15.5 мм;

– Длина лапки (a): 10 мм;

– Ширина лапки (b): 06.50 мм;

– Длина шейки (d): 10 мм;

– Ширина шейки (c): 08.50 мм.

На рисунках 4.4 и 4.5 изображены 3D-модели сверла, построенные по максимальным параметрам.

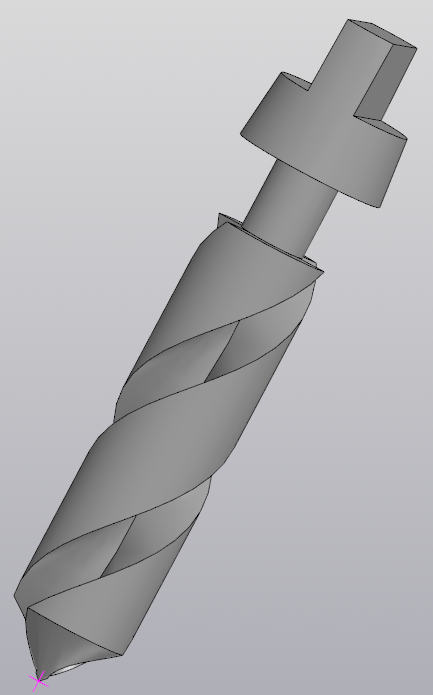


Рисунок 4.4 – 3D-модель сверла, построенная по стандартным параметрам, имеющая обычный кончик и не имеющая скругления

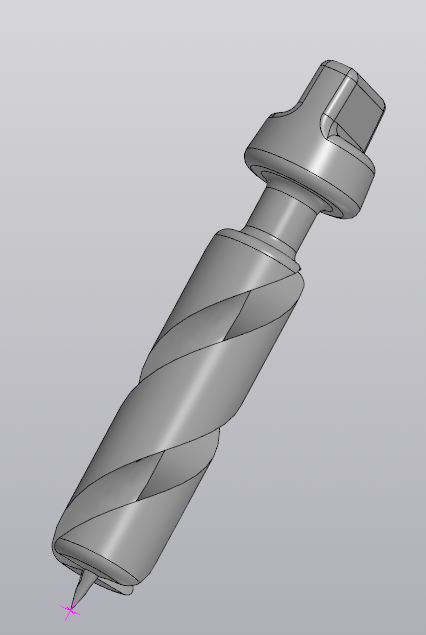


Рисунок 4.5 – 3D-модель сверла, построенная по стандартным параметрам, имеющая кончик сверла по дереву и имеющая скругления

При вводе некорректных данных программа окрашивает текстовое поле, в которое было неправильно введено значение и отображает подсказку об ошибке.

Результаты данного тестирования отражены на рисунках 4.6 ­– 4.8.

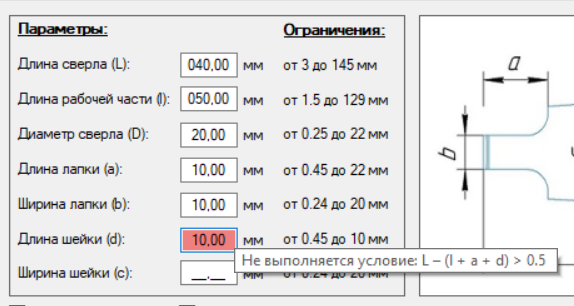


Рисунок 4.6 – Результат невыполнения зависимости: L – (l + a + d) > 0.5

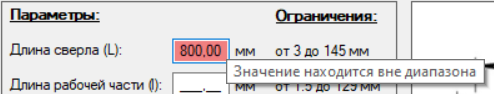


Рисунок 4.7 – Результат при введении слишком большого значения в параметр

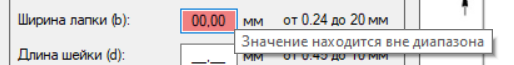


Рисунок 4.8 – Результат при введении слишком маленького значения в параметр

## 4.2 Модульное тестирование

Тестовые сценарии для класса DrillParameters, проверяющие корректность введенных параметров 3D-модели сверла, оформлены в таблицу А (приложение А).

Тестирование проводилось с помощью фреймворка модульного тестирования NUnit 3.13.1 для языков платформы .NET

На рисунке 4.9 изображен результат прохождения всех написанных модульных тестов. На рисунке 4.10 отражено покрытие данных модульных тестов.

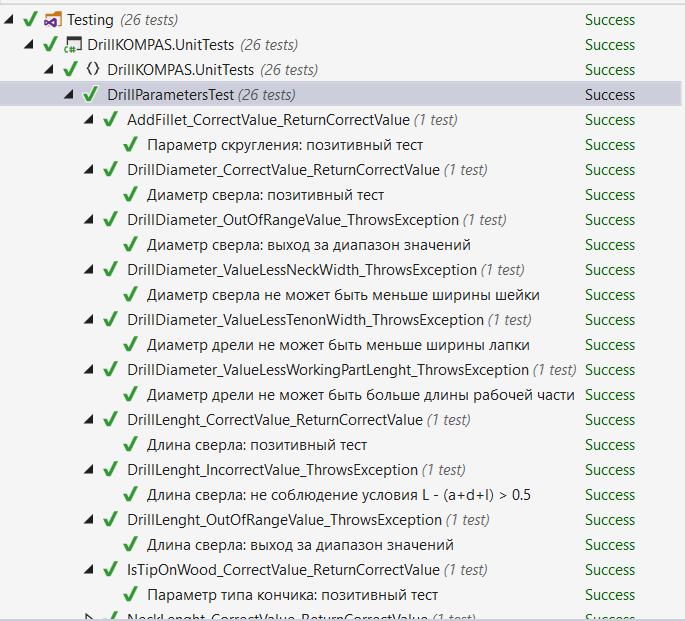


Рисунок 4.9 – Список Unit-тестов

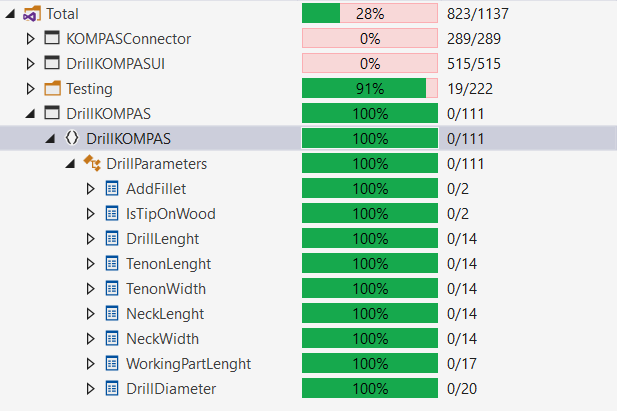


Рисунок 4.10 – Покрытие Unit-тестами класса DrillParameters

## 4.3 Нагрузочное тестирование

Данное тестирование проводилось на компьютере с характеристиками:

– Процессор: Intel Core i3-7020U 2.30 GHz

– Оперативная память: 5,88ГБ DDR4 2133 MHz;

– Графический процессор: NVIDIA GeForce MX110;

– Операционная система: Windows 10 Home 64-bit.

Перед тестированием было установлено максимальное количество последовательно строящихся моделей равным 100. Во время построения 50 модели «КОМПАС-3D v18» остановил свою работу и при просмотре в диспетчере задач отображался нерабочий статус. На этом тестирование было завершено и количество построенных деталей было равным 49.

Зависимость потребления оперативной памяти компьютера «КОМПАС-3D v18» от количества построенных моделей отражена на рисунке 4.11.

Рисунок 4.11 – Зависимость потребления ОЗУ от количества построенных модели

Потребление памяти росло линейно, кроме промежутка с 12 до 25 построения, в котором наблюдается резкий скачок потребления памяти. В данном скачке самое большое потребление памяти отмечено при 21 построении и равно 1032 МБ. При анализе данных видно, что пик потребление был достигнут в 49 построении равен 1263 МБ. Именно на этом построении «КОМПАС-3D v18» завершил свою работу в аварийном режиме.

Первое построение модели заняло больше времени (8 секунд), относительно последующих ближних точек, так как кроме построения непосредственной модели сверла необходимо было запустить «КОМПАС-3D v18». Время построение модели медленно росло, занимая в среднем 3 секунды, но имея несколько скачков. Первые два небольших скачка при построении 31 и 35 время равно примерно 13 секунд. Второй скачок при построении 42 и 43 модели затраченное время было рано 19 секунд. И третий скачок при построении 46 и 47 модели. При построении 47 модели было затрачено 35 секунд, а при построении 46 модели было потрачено самое большое количество времени 44 секунды.

Зависимость затраченного времени от количества простроенных деталей отражена на рисунке 4.12.

Рисунок 4.12 – Зависимость затраченного времени от количества построенных деталей

# Заключение

В ходе выполнения лабораторных работ были изучены предметная область проектирования, предмет проектирования, аналоги предмета проектирования, API .

На основании полученных данных были спроектированы UML-диаграмма и макет пользовательского интерфейса, использованные для написания плагина для КОМПАС v18 «Сверло».

Для данного плагина были проведены модульные, функциональные и нагрузочные тесты.

# Список использованных источников

1. Большаков В. П. Создание трехмерных моделей и конструкторской документации в системе КОМПАС-3D / В. П. Большаков. – СПб. БХВ-Петербург, 2010. – 496 с.

2. Кидрук М. И. КОМПАС-3D V10 на 100 % (+CD) / М. И. Кидрук. ­– СПб.: Питер, 2009. – 560 с.

3. КОМПАС-3D Home: Дистанционное обучение [Электронный ресурс]. – Режим доступа: https://kompas.ru (дата обращения 13.03.2021)

4. Фаулер, М. UML. Основы, 3-е издание / М. Фаулер; пер. с англ. А. Петухов – СПб: Издательство Символ-Плюс, 2018. – 192 с.

# Приложение А

Таблица А – Тестовые сценарии

|  |  |  |
| --- | --- | --- |
| Тестовый метод | Входные параметры | Описание |
| AddFillet\_CorrectValue\_ReturnCorrectValue | false | Позитивный тест параметра: наличие скругления |
| IsTipOnWood\_CorrectValue\_ReturnCorrectValue | true | Позитивный тест для параметра: тип кончика сверла |
| DrillLenght\_CorrectValue\_ReturnCorrectValue | 100.80 | Позитивный тест для параметра: длина сверла |
| DrillLenght\_OutOfRangeValue\_ThrowsException | 999.99 | Длина сверла: выход за диапазон значений |
| DrillLenght\_IncorrectValue\_ThrowsException | DrillLenght = 40.00, WorkingPartLenght = 20.00, TenonLenght = 5.00,  NeckLenght = 5.00,  10.00 | Длина сверла: не соблюдение условия L - (a+d+l) > 0.5 |
| WorkingPartLenght\_CorrectValue\_ReturnCorrectValue | DrillLenght = 40.00, 20.80 | Позитивный тест для параметра: рабочая часть |
| WorkingPartLenght\_OutOfRang eValue\_ThrowsException | 0.00 | Рабочая часть: выход за диапазон |
| Продолжение таблицы А | | |
|  |  | значений |
| WorkingPartLenght\_IncorrectValue\_ThrowsException | DrillLenght = 20.00, 26.50 | Рабочая часть: не соблюдение условия L-(a+d+l) > 0.5 |
| WorkingPartLenght\_ValueLessDrillDiametr\_ThrowsException | DrillLenght = 20.00,  WorkingPartLenght = 14, DrillDiameter = 12, 10.50 | Длина рабочей части не может быть меньше диаметра сверла |
| DrillDiameter\_CorrectValue\_ReturnCorrectValue | DrillLenght = 100,  WorkingPartLenght = 60, 20.80 | Позитивный тест для параметра: диаметр сверла |
| DrillDiameter\_OutOfRangeValue\_ThrowsException | 50.50 | Диаметр сверла: выход за диапазон значений |
| DrillDiameter\_ValueLessNeckWidth\_ThrowsException | DrillLenght = 70.00, WorkingPartLenght = 30, DrillDiameter = 20, NeckWidth = 10.00, 5.50 | Диаметр сверла не может быть меньше ширины шейки |
| DrillDiameter\_ValueLessTenonWidth\_ThrowsException | DrillLenght = 100, WorkingPartLenght = 50, DrillDiameter = 20, TenonWidth = 10.00, 5.50 | Диаметр дрели не может быть меньше ширины лапки |
| DrillDiameter\_ValueLessWorkingPartLenght\_ThrowsException | DrillLenght = 40.00, WorkingPartLenght = 14, DrillDiameter = 10, NeckWidth = 10.00, | Диаметр дрели не может быть больше длины рабочей части |
| Продолжение таблицы А | | |
|  | TenonWidth = 10.00, 15.50 |  |
| TenonLenght\_CorrectValue\_ReturnCorrectValue | DrillLenght = 100.50, 20.50 | Позитивный тест для параметра: длина лапки |
| TenonLenght\_OutOfRangeValue\_ThrowsException | 105.50 | Длина лапки: выход за диапазон значений |
| TenonLenght\_IncorrectValue\_ThrowsException | DrillLenght = 50.00, WorkingPartLenght = 20.00, NeckLenght = 10.00, 20.50 | Длина лапки: не соблюдение условия L - (a+d+l) > 0.5 |
| TenonWidth\_CorrectValue\_ReturnCorrectValue | DrillLenght = 100, WorkingPartLenght = 50, DrillDiameter = 20.00, 10.50 | Позитивный тест для параметра: ширина лапки |
| TenonWidth\_OutOfRangeValue\_ThrowsException | 100.50 | Ширина лапки: выход за границы диапазона |
| TenonWidth\_ValueLessDrillDia meter\_ThrowsException | DrillLenght = 100, WorkingPartLenght = 50, DrillDiameter = 15.00, 19.50 | Ширина лапки не может быть больше диаметра сверла |
| NeckLenght\_CorrectValue\_ReturnCorrectValue | DrillLenght = 20.50, 9.50 | Позитивный тест для параметра: длина шейки |
| NeckLenght\_OutOfRangeValue\_ThrowsException | 200.00 | Длина шейки: выход за диапазон массива |
| Продолжение таблицы А | | |
| NeckLenght\_IncorrectValue\_ThrowsException | DrillLenght = 50.00, WorkingPartLenght = 20.00, TenonLenght = 20.00, 10.00 | Длина шейки: не выполнение условия L - (a+d+l) > 0.5 |
| NeckWidth\_CorrectValue\_ReturnCorrectValue | DrillLenght = 100, WorkingPartLenght = 50, DrillDiameter = 20.50, 9.50 | Позитивный тест для параметра: ширины шейки |
| NeckWidth\_OutOfRangeValue\_ThrowsException | 250.00 | Шириан шейки: выход за диапазон значений |
| NeckWidth\_ValueLessDrillDiameter\_ThrowsException | 9.00 | Ширина шейки не может быть больше диаметра сверла |