

Министерство науки и высшего образования Российской Федерации

ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ СИСТЕМ  
УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ  
(ТУСУР)

Кафедра компьютерных систем в управлении и проектировании

Проект системы по лабораторному проекту по дисциплине  
«Основы разработки САПР»

**«Разработка плагина для построения топора  
строительного в системе автоматизированного  
проектирования КОМПАС-3D»**

Руководитель:

к. т. н., доцент кафедры КСУП

Калентьев А. А.

Дата:

Выполнил:

студент гр. 588-1

Сабанова К. В.

Дата:

« » \_\_\_\_\_ 20\_\_ г.

« » \_\_\_\_\_ 20\_\_ г.

2022 г.

## Содержание

Термины и обозначения.....	3
Описание программы.....	4
Описание технического объекта.....	6
Описание задействуемых функций API.....	8
Проект программы-плагина.....	12
Список использованных источников.....	17

### **Термины и обозначения**

**Плагин** – программное обеспечение, подключаемое к другому программному обеспечению с целью расширить функциональность последней [2].

**Система автоматизированного проектирования (САПР)** – программное обеспечение, позволяющее автоматизировать процесс проектирования.

**Application Programming Interface (API)** – описание способа взаимодействия компьютерных программ друг с другом [3].

**Component Object Model (COM)** – специфический для ОС Windows стандарт, позволяющий создавать программные компоненты, с которыми могут впоследствии взаимодействовать другие программы.

**Маршалинг** – процесс преобразования информации в ОЗУ в пригодный для хранения или передачи (в частности, другим программам) формат.

**Фреймворк** – программное обеспечение, облегчающее разработку программного обеспечения (набор библиотек и инструментов).

## 1 Описание программы

Система автоматизированного проектирования «КОМПАС-3D» от компании «Аскон» представляет собой семейство программ, объединённых целью автоматизации процесса инженерного проектирования. Данная САПР позволяет создавать трёхмерные модели и чертежи технических объектов и готовить техническую документацию к ним в соответствии с единой системой конструкторской документацией, а также другими стандартами. Несмотря на то, что этим функциональность программы не ограничивается, в рамках данного проекта будет задействована лишь малая доля функциональности данной САПР, а именно трёхмерное моделирование.

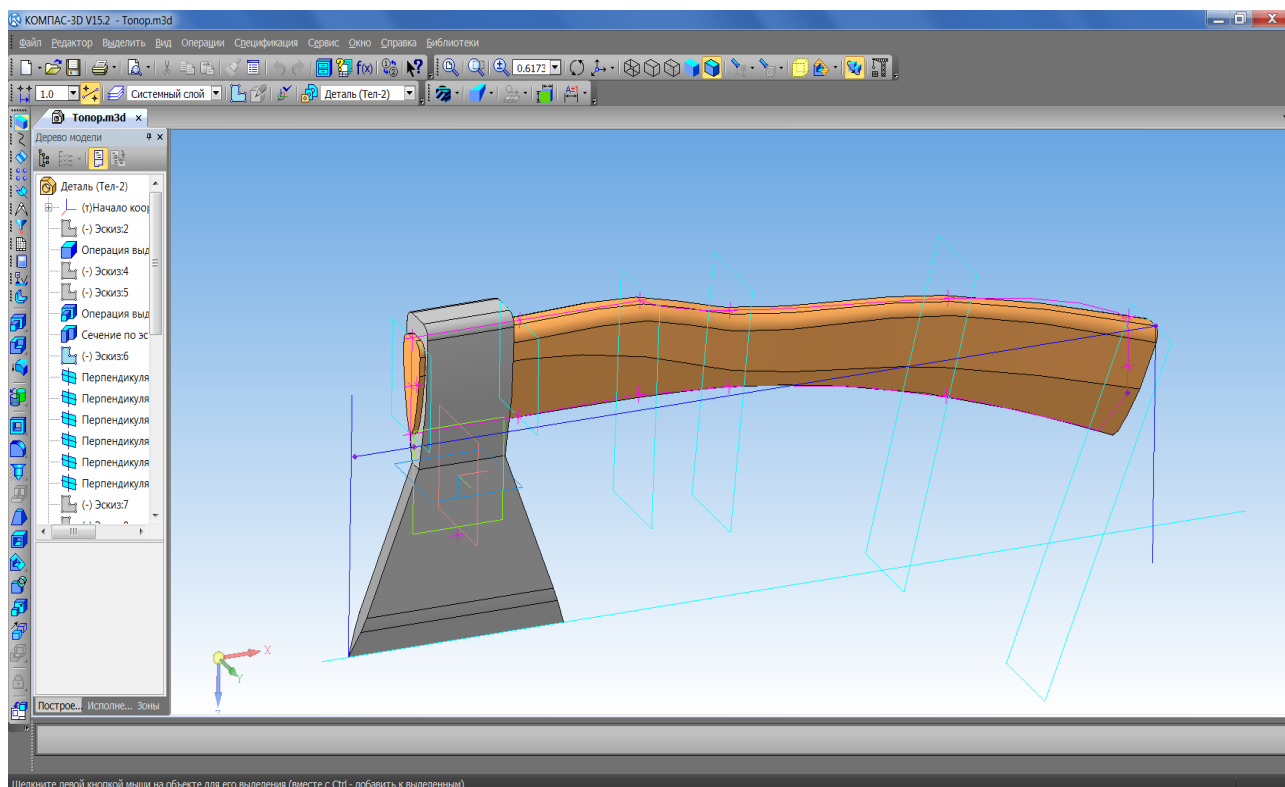


Рисунок 1.1 – КОМПАС-3D V15 в режиме трёхмерного моделирования

Программа «КОМПАС-3D» обладает стандартным для ОС Windows интерфейсом. Для трёхмерного моделирования используется множество операций: построение двумерных эскизов, выдавливание, вырезание, создание фасок, скруглений, оболочек и так далее. Все операции, совершаемые одна за другой, в итоге формируют дерево операций (на рисунке выше оно находится в

левой части окна). Модель же формируется за счёт последовательного выполнения этих операций. Кроме того, можно задать свойства модели, такие как цвет, оптические свойства, масса и так далее.

Несмотря на очень высокую степень автоматизации процесса конструирования технических объектов, «КОМПАС-3D» всё ещё предоставляет простор для дальнейшей автоматизации. В частности, создание однотипных технических объектов можно свести к генерации на основе входных параметров, дабы избежать множественного копирования модели-шаблона с ручным изменением параметров или создания их каждый раз с нуля. Осуществить такую автоматизацию «КОМПАС-3D» позволяет с помощью своего API, включающего в себя существенную часть функциональности данной САПР. В частности, с помощью данного API можно строить чертежи и трёхмерные модели. С API «КОМПАС-3D» можно работать на языках C++ и C#. В дистрибутиве «КОМПАС-3D» также поставляется справочник по API данной программы [4].

## 2 Описание технического объекта

В рамках данного проекта будет разработан плагин, позволяющий автоматизировать создание трёхмерных моделей топоров.

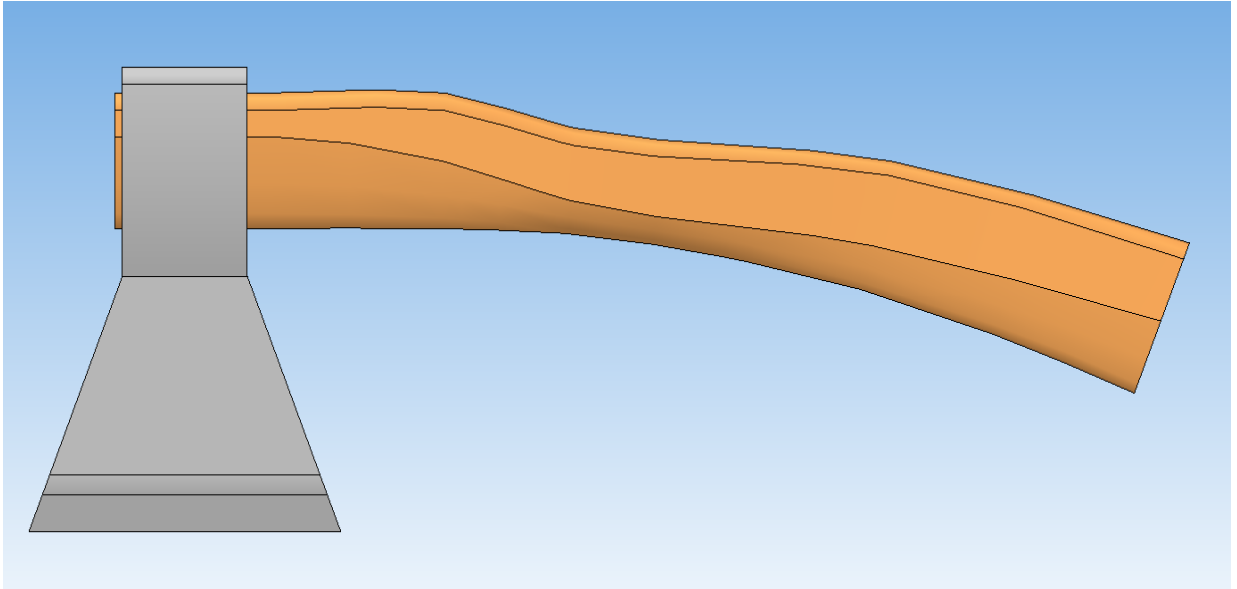


Рисунок 2.1 – Топор строительный

У данной модели есть пять изменяемых параметров.

- 1 Общая высота рубящей части  $H$  (от 145 мм до 185 мм);
- 2 Общая длина изделия  $L$  (от 400 мм до 550 мм);
- 3 Длина острой кромки  $L_1$  (от 110 мм до 150 мм);
- 4 Длина тупой части  $L_2$  (от 45 мм до 60 мм);
- 5 Высота края топорика  $K$  (от 55 мм до 95 мм);

Следующие параметры являются зависимыми:

- 1 Высота края топорика  $K = H - 90$ ;
- 2 Длина острой кромки  $L_1 < L / 3$ .

### 3 Описание задеиствуемых функций API

Для создания данной модели потребуется задеиствовать ряд операций моделирования, в числе которых создание эскизов и плоскостей, выдавливание (в том числе с вырезанием), скругление, создание оболочки, операция по сечениям и зеркальное отражение.

Взаимодействие с «КОМПАС-3D» можно обеспечить несколькими способами, в частности, получив объект его экземпляра посредством COM. Если «КОМПАС-3D» установлен на компьютере, то получить его экземпляр можно с помощью маршалинга. Если получить объект работающего экземпляра «КОМПАС-3D» получилось, то необходимо его привести к интерфейсу *KompasObject* для дальнейшей работы. Если нет, то необходимо попытаться запустить «КОМПАС-3D», после чего получить его экземпляр. Если и это не удалось, то это означает, что «КОМПАС-3D» не установлен на компьютере должным образом.

Объект *KompasObject* позволяет производить множество операций, из которых потребуются, в частности, создание двумерных и трёхмерных документов, а также геометрические расчёты.

В самой программе необходимо создать деталь. Получить её экземпляр можно с помощью функции *Document3D()*, он будет иметь интерфейс *ksDocument3D*. Для создания детали необходимо вызвать метод *Create()*.

Также через *KompasObject* необходимо создавать и получать объекты различных структур. Например, через него можно получить структуры геометрических объектов (прямоугольников, линий, окружностей) для их дальнейшего построения, создать динамические массивы и коллекции тех или иных объектов или же получить специальный объект для геометрических расчётов.

Полная функциональность *KompasObject* для данного проекта избыточна. Список используемых функций *KompasObject* приведён в таблице ниже.

Таблица 3.1 – Функции KompasObject

Функция	Тип	Аргументы	Описание
Document3D	ksDocument3D	Нет	Возвращает новый объект трёхмерной модели.
GetParamStruct	Требуется приведения	structType (short) – константа, определяющая тип структуры	Возвращает объект структуры указанного типа.
GetMathematic2D	ksMathematic2D	Нет	Возвращает объект, позволяющий осуществлять геометрические расчёты на плоскости.
GetDynamicArray	ksDynamicArray	type (int) – константа, определяющая тип элементов в массиве	Возвращает новый динамический массив объектов указанного типа.

Если последние три функции чаще всего находят своё применение в частных ситуациях, то на первой стоит остановиться подробнее. Полученный объект *ksDocument3D* содержит в себе корневую деталь, которая и станет в итоге корпусом розетки. Деталь – объект *ksPart* – можно получить методом *GetPart(int type)* объекта *ksDocument3D*, передав в качестве *type* константу верхней детали. Объект *ksPart* предоставляет множество методов для работы с ним, из которых стоит рассмотреть самые необходимые.

Таблица 3.2 – Функции ksPart

Функция	Тип	Аргументы	Описание
NewEntity	ksEntity	objType (short) – константа типа объекта	Возвращает новый объект в дереве модели.
GetDefaultEntity	ksEntity	objType (short) – константа типа объекта	Возвращает стандартный объект в дереве модели.
EntityCollection	ksEntityCollection	objType (short) – константа типа объекта	Возвращает коллекцию объектов указанного типа в дереве модели.

Под объектами *ksEntity* понимается всё, что находится в дереве модели под деталью, то есть разного рода операции над моделью. Объекты



*ksEntityCollection* представляют собой коллекции с присущими им функциями добавления, удаления, очистки и так далее. Отдельно стоит упомянуть лишь функцию *SelectByPoint(x, y, z)*, позволяющую оставить в коллекции лишь те операции, которые в результате своего выполнения порождают геометрические объекты, пересекающиеся с точкой, имеющей координаты  $x$ ,  $y$  и  $z$ . С помощью данной функции можно выделять ребро, зная координаты лежащей на нем точки.

Таблица 3.3 – Функции *ksEntity*

Функция	Тип	Аргументы	Описание
GetDefinition	Требуется приведения	Нет	Возвращает интерфейс параметров операции.
Create	bool	Нет	Заканчивает операцию и добавляет её в дерево модели. Возвращает true в случае успеха.

Интерфейсы параметров операций имеют разные типы в зависимости от самих операций. Например, для операции вырезания выдавливанием интерфейс параметров будет иметь тип *ksCutExtrusionDefinition*. Некоторые из этих типов имеют специфичные лишь для себя методы, но есть и несколько общих.

Таблица 3.4 – Функции параметров операции

Функция	Тип	Аргументы	Описание
SetPlane	bool	plane (object) – плоскость	Задаёт плоскость.
BeginEdit	ksDocument2D	Нет	Открывает эскиз для редактирования и возвращает его.
EndEdit	bool	Нет	Закрывает редактирование эскиза. Возвращает true при успехе.
SetSketch	bool	sketch (object) – эскиз	Назначает эскиз.
GetSketch	ksDocument2D	Нет	Возвращает эскиз.

Эскизы (*ksDocument2D*) имеют ряд методов для черчения на плоскости.

Таблица 3.5 – Функции параметров операции

Функция	Тип	Аргументы	Описание
ksRectangle	int	rectParams (ksRectangleParam) – параметры прямоугольника	Создаёт прямоугольник с заданными параметрами.
ksCircle	int	xc, yc, rad (double) – координаты центра и радиус; style (int) – стиль линии	Создаёт окружность с заданными центром и радиусом.
ksLineSeg	int	x1, y1, x2, y2 (double) – конца отрезка, style (int) – стиль линии	Создаёт отрезок с указанными концами.
ksLine	int	x, y, angle (double) – координата точки и угол относительно OX	Создаёт вспомогательную прямую.
ksArcBy3Points	int	x1, y1, x2, y2, x3, y3 (double) – координаты трёх точек дуги	Создаёт дугу по трём точкам.
ksArcByPoint	int	xc, yc, rad, x1, y1, x2, y2 (double) – координаты центра и двух точек, direction, style (int) – направление и стиль линии соответственно	Создаёт дугу по центру и радиусу окружности и двум точкам.

#### 4 Проект программы-плагина

Исходя из назначения программы, логично выделить в ней несколько архитектурных единиц: графический интерфейс пользователя, бизнес-логику и внешние библиотеки. В число последних входят фреймворки, используемые при разработке, а также библиотеки API «КОМПАС-3D». Кроме того, необходимо покрыть бизнес-логику юнит-тестами.

Исходя из всего вышеперечисленного, проект программы-плагина должен включать в себя три пакета, реализующих бизнес-логику, интерфейс и юнит-тесты соответственно. Диаграмма пакетов будет выглядеть следующим образом.

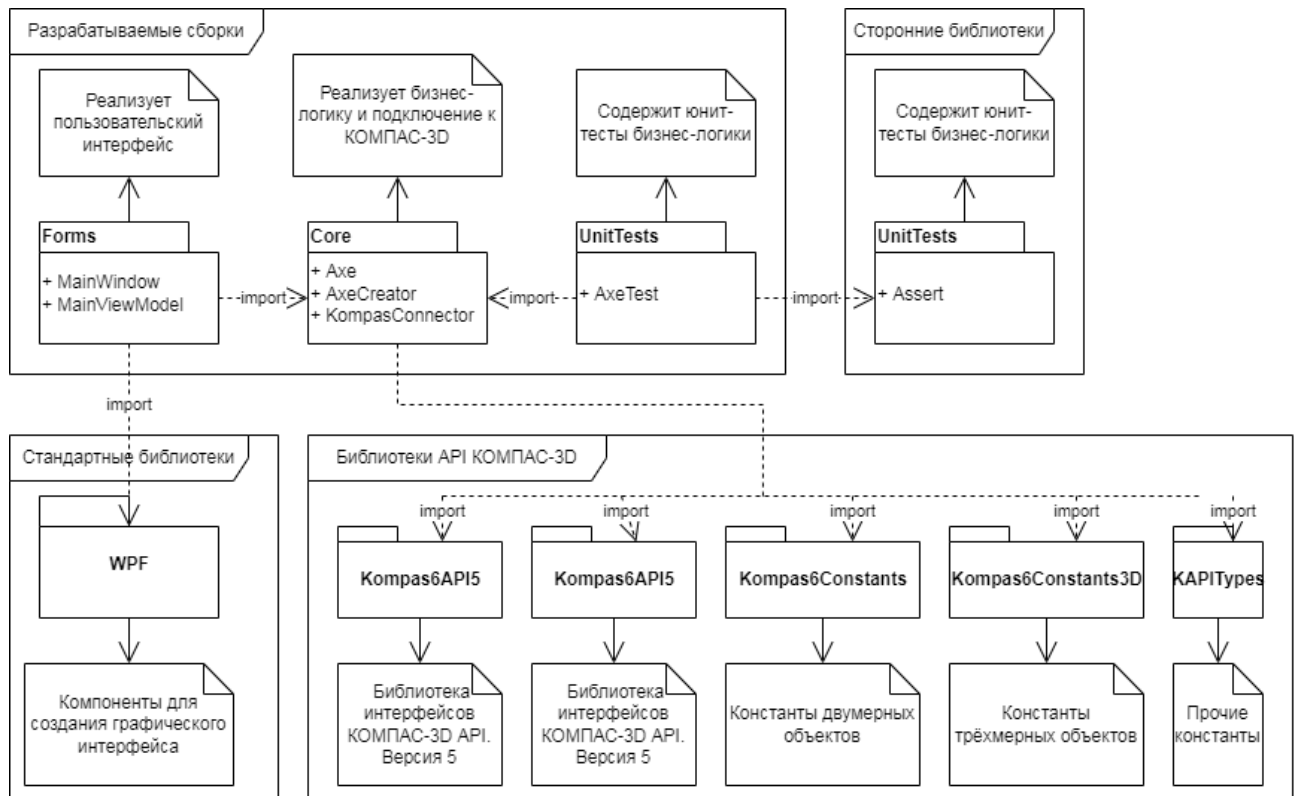


Рисунок 4.1 – Диаграмма пакетов

Пакет бизнес-логики будет включать в себя следующие классы.

1 KompasConnector – класс, реализующий подключение к экземпляру «КОМПАС-3D».

2 Sketcher – класс, реализующий построение двумерных объектов через API «КОМПАС-3D».

3 Modeler – класс, реализующий построение трёхмерных объектов через API «КОМПАС-3D».

Помимо этого, в пакете бизнес-логики будут присутствовать вспомогательные сущности (перечисления, модели).

В пакете пользовательского интерфейса предполагается реализовать классы:

1 Axe – класс, реализующий модель топора. Данный класс будет включать в себя набор изменяемых и неизменных свойств.

2 KompasConnector – класс, реализующий подключение к экземпляру «КОМПАС-3D».

3 Sketcher – класс, реализующий построение двумерных объектов через API «КОМПАС-3D».

4 Modeler – класс, реализующий построение трёхмерных объектов через API «КОМПАС-3D».

5 AxeCreator – класс, реализующий построение топора с помощью класса Designer.

6 MainViewModel – класс, выступающий прослойкой между главным окном и бизнес-логикой приложения.

7 MainWindow – окно программы, содержащее поля для задания размеров и кнопку для начала построения. Также на нём будет размещена схема топора. Ввод неправильных значений будет блокировать кнопку, появится предупреждение. Поля при запуске будут иметь значения по умолчанию.

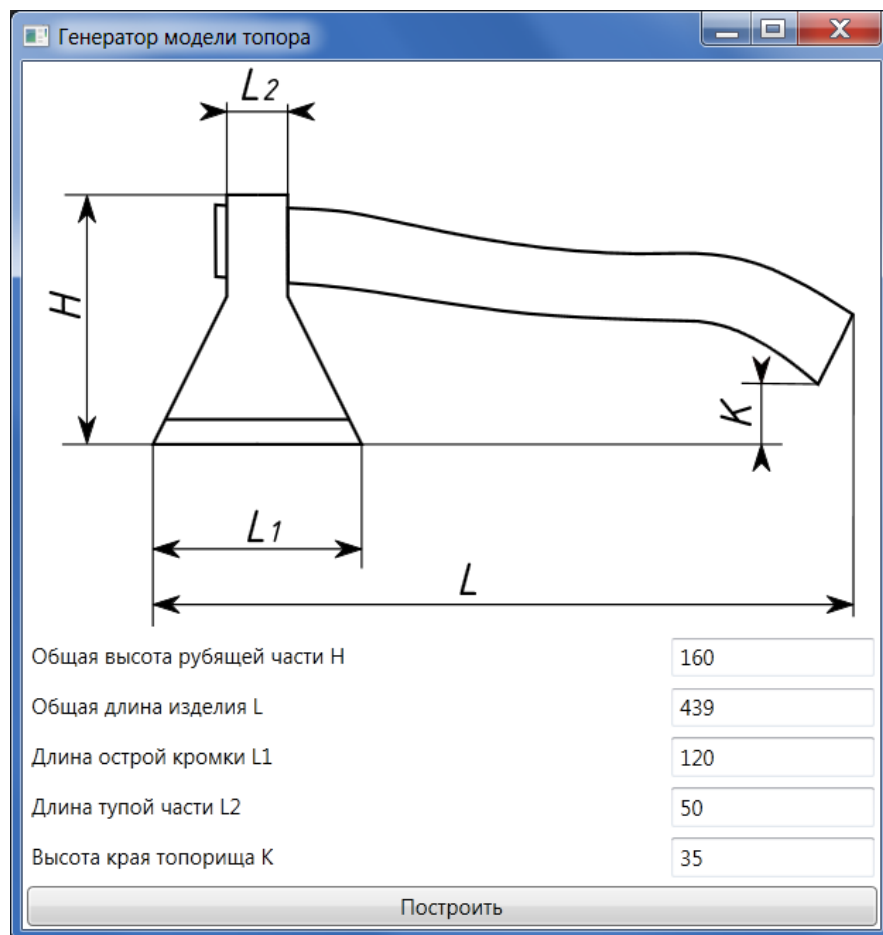


Рисунок 4.2 – Главное окно

Диаграмма классов приведена на рисунках ниже.

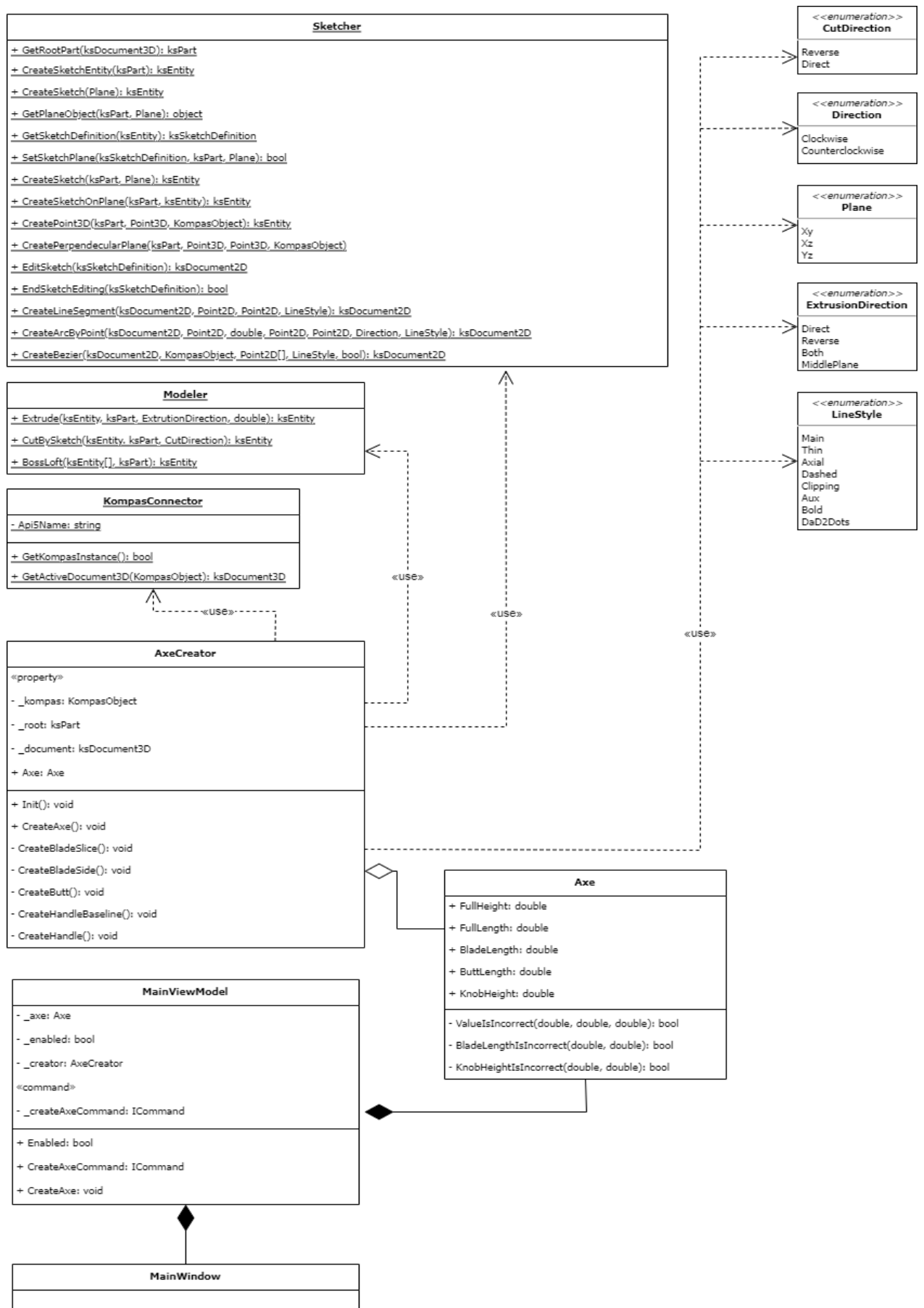


Рисунок 4.4 – Диаграмма классов

**Список использованных источников**

- 1 Новые технологии в программировании: учебное пособие / А. А. Калентьев, Д. В. Гарайс, А. Е. Горяинов – Томск: Эль Контент, 2014. – 176 с.
- 2 Плагин – Википедия. [Электронный ресурс]. – Режим доступа: <https://ru.wikipedia.org/wiki/API> (дата обращения 18.03.2021)
- 3 API – Википедия. [Электронный ресурс]. – Режим доступа: <https://ru.wikipedia.org/wiki/API> (дата обращения 18.03.2021)
- 4 Справочная система КОМПАС-МАСТЕР. [Электронный ресурс]. 2014. 11 декабря.