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 «СКВОРЕЧНИК» ДЛЯ «КОМПАС-3D V18.1»**

Проект системы по лабораторному проекту

по дисциплине «ОСНОВЫ РАЗРАБОТКИ САПР»

«Построение скворечника в системе КОМПАС-3D v18.1»

|  |
| --- |
| Выполнил:  студент гр. 586-2  \_\_\_\_\_\_\_\_\_\_\_\_\_\_ К.И. Колмогорцева  «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2020г. |
| Руководитель:  к.т.н., доцент каф. КСУП:  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ А.А. Калентьев  «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2020г. |

Томск 2020

Содержание

[1 Описание САПР 3](#_Toc35912275)

[1.1 Описание программы 3](#_Toc35912276)

[1.2 Описание API 3](#_Toc35912277)

[1.3 Обзор аналогов 8](#_Toc35912278)

[2 Описание предмета проектирования 9](#_Toc35912279)

[3 Проект программы 10](#_Toc35912280)

[3.1 Диаграмма вариантов использования (Use Cases) 10](#_Toc35912281)

[3.2 Диаграмма классов 11](#_Toc35912282)

[3.3 Макет пользовательского интерфейса 12](#_Toc35912283)

[Список использованных источников 13](#_Toc35912284)

# 1 Описание САПР

# 1.1 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 [1]. 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.

# 1.2 Описание API

В КОМПАС на данный момент существуют API двух версий: API 5 и API 7 [2]. 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 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 основные свойства и методы интерфейса KompasObject.

Таблица 1.1 – Методы и свойства интерфейса KompasObject

|  |  |  |
| --- | --- | --- |
| Название | Тип возвращаемых данных | Описание |
| Document3D() | ksDocument | Дает возможность получить указатель на активный трехмерный документ |
| GetParamStruct(short structType) | [StructType2D](mk:@MSITStore:C:\Program%20Files\ASCON\KOMPAS-3D%20V16\SDK\SDK.chm::/StructType2D.htm) | Позволяет получить интерфейс структуры параметров объекта определенного типа (например параметры прямоугольника и т.д.) |
| Visible | bool | Свойство видимости приложения |

В таблице 1.2 представлены методы интерфейса ksEntity, необходимые для разработки плагина.

Таблица 1.2 – Методы интерфейса ksEntity

|  |  |  |
| --- | --- | --- |
| Название | Тип возвращаемых данных | Описание |
| Create() | bool | Дает возможность создать пустой документ (деталь или сборку) |
| GetDefinition() | IUnkown | Получить указатель на интерфейс параметров объектов и элементов |
| Update() | bool | Позволяет обновить настройки документа |

В таблице 1.3 представлены свойства и методы интерфейса ksDocument2D, необходимые для разработки плагина.

Таблица 1.3 – Методы интерфейса ksDocument2D

|  |  |  |
| --- | --- | --- |
| Название | Тип возвращаемых данных | Описание |
| ksLineSeg(double x1, double y1, double x2, double y2, int style) | int | Получить указатель на отрезок на двумерной плоскости либо 0 в случае ошибки |
| ksRegularPolygon(  ksRegularPolygonParam param, int style) | int | Получить указатель на многоугольник на двумерной плоскости либо 0 в случае ошибки |
| ksRectangle(  ksRectangleParam param, int style) | int | Получить указатель на прямоугольник на двумерной плоскости либо 0 в случае ошибки |
| ksCircle(  double xc, double yc, double rad, int style) | int | Получить указатель на окружность на двумерной плоскости либо 0 в случае ошибки |

В таблице 1.4 представлены свойства и методы интерфейса ksDocument3D, необходимые для разработки плагина.

Таблица 1.4 – Методы интерфейса ksDocument3D

|  |  |  |
| --- | --- | --- |
| Название | Тип возвращаемых данных | Описание |
| Create (bool invisible, bool \_typeDoc) | bool | Создать документ-модель (деталь или сборку) |
| GetPart(int type) | ksPart | Получить указатель на интерфейс компонента в соответствии с заданным типом |

В таблице 1.5 представлены методы интерфейса ksPart, необходимые для разработки плагина.

Таблица 1.5 – Свойства и методы интерфейса ksPart

|  |  |  |
| --- | --- | --- |
| Название | Тип возвращаемых данных | 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 1.6 представлены типы объектов документа-модели, необходимые для разработки плагина.

Таблица 1.6 – Некоторые типы объектов документа-модели

|  |  |  |
| --- | --- | --- |
| Идентификатор объекта | Название объекта | Интерфейс параметров |
| o3d\_unknown | Неизвестный (включает все объекты) |  |
| o3d\_planeXOZ | Плоскость XOZ | ksPlaneParam |
| o3d\_planeYOZ | Плоскость YOZ | ksPlaneParam |
| o3d\_planeXOY | Плоскость XOY | ksPlaneParam |
| o3d\_sketch | Эскиз | [ksSketchDefinition](mk:@MSITStore:C:\Program%20Files\ASCON\KOMPAS-3D%20V16\SDK\SDK.chm::/ksSketchDefinition.htm) |
| o3d\_face | Грань | [ksFaceDefinition](mk:@MSITStore:C:\Program%20Files\ASCON\KOMPAS-3D%20V16\SDK\SDK.chm::/ksEdgeDefinition.htm) |
| o3d\_baseExtrusion | Базовая операция выдавливания | [ksBaseExtrusionDefinition](mk:@MSITStore:C:\Program%20Files\ASCON\KOMPAS-3D%20V16\SDK\SDK.chm::/ksBaseExtrusionDefinition.htm) |
| o3d\_cutExtrusion | Вырезать выдавливанием | [ksCutExtrusionDefinition](mk:@MSITStore:C:\Program%20Files\ASCON\KOMPAS-3D%20V16\SDK\SDK.chm::/ksCutExtrusionDefinition.htm) |
| o3d\_baseLoft | Создание элемента по сечениям | [ksBaseLoftDefinition](mk:@MSITStore:C:\Program%20Files\ASCON\KOMPAS-3D%20V16\SDK\SDK.chm::/ksBaseExtrusionDefinition.htm) |
|  |  |  |
| Окончание таблицы 1.6 |  |  |
| o3d\_baseEvolution | Создание кинематического элемента | [ksBaseEvolutionDefinition](mk:@MSITStore:C:\Program%20Files\ASCON\KOMPAS-3D%20V16\SDK\SDK.chm::/ksBaseExtrusionDefinition.htm) |
| o3d\_cutEvolution | Вырезать кинематический элемент | [ksCutEvolutionDefinition](mk:@MSITStore:C:\Program%20Files\ASCON\KOMPAS-3D%20V16\SDK\SDK.chm::/ksCutExtrusionDefinition.htm) |

Таким образом, создание какой-либо трехмерной операции пользовательской программой сводится к такой последовательности шагов.

1. Инициализация главного интерфейса приложения API - KompasObject. Он инициализируется один раз для всего сеанса работы программы.

2. Инициализация интерфейса трехмерного документа ksDocument3D, с последующим созданием нового документа или получением указателя на активный документ.

3. Создание компонента и получение на него указателя (интерфейс ksPart). Для сборки это может быть готовый компонент, компонент, вставленный из файла или созданный «на месте». Для детали необходимо получить указатель на компонент типа pTop\_Part.

4. Создание с помощью метода ksPart::NewEntity интерфейса нужной операции. При этом в метод передается соответствующий идентификатор (например, для выдавливания - o3d\_bossExtrusion).

5. Получение с помощью метода ksEntity::GetDifinition указателя на интерфейс параметров конкретной операции (для выдавливания этим интерфейсом является ksBossExtrusionDefinition). Настройка этих параметров необходимым пользователю образом.

6. Создание операции с помощью метода ksEntity::Create.

# 1.3 Обзор аналогов

Выбран аналог “Викс-3D-Дом” [3] построения дома из-за схожести элементов для варьирования. Небольшой перечень состава данного плагина:

Стены и их редактирование: пилястры, четверти, торцы стен и проемов. штриховка 1-3х слойных стен.

Окна, двери, проемы, отверстия в стенах и их редактирование. Витражи. Створки дверей. Фасадные окна или двери. Редактор фасадного заполнения проема.

На рисунке 1.1 представлены параметры для изменений.

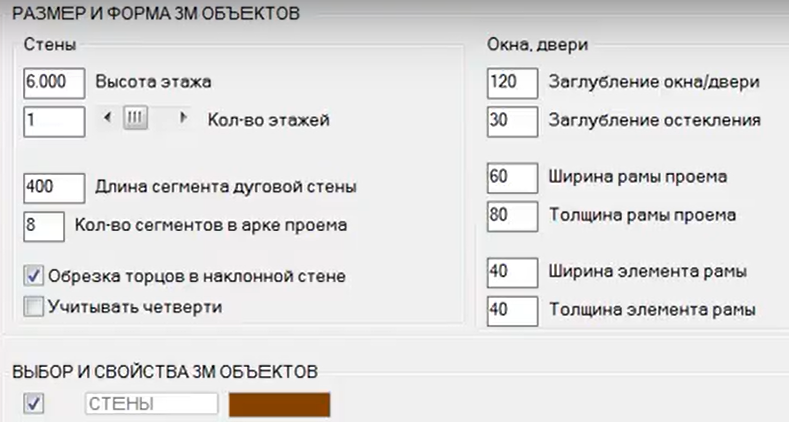


Рисунок 1.1 ­­– Параметры для варьирования при строительстве дома

На рисунке 1.2 представлен построенный дом с выбранными параметрами.

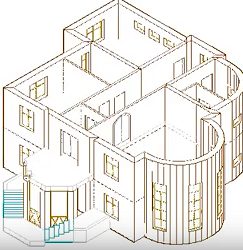


Рисунок 1.2­­ – Построенный дом в программе

# 2 Описание предмета проектирования

Предметом проектирования является скворечник.

Помещение для птиц в виде ящика с отверстием, укрепленного на высоком шесте или на дереве. Также дополнительными элементами являются жердочка (перекладина для сидения птицы) и навес в виде крыши, который должен выступать за габариты самого скворечника со стороны фасада, чтобы внутрь скворечника не попадали осадки.

В качестве крепежа выбрана дополнительная балка, прибитая с задней стороны изделия.

На рисунке 2.1 представлена 3D модель скворечника.

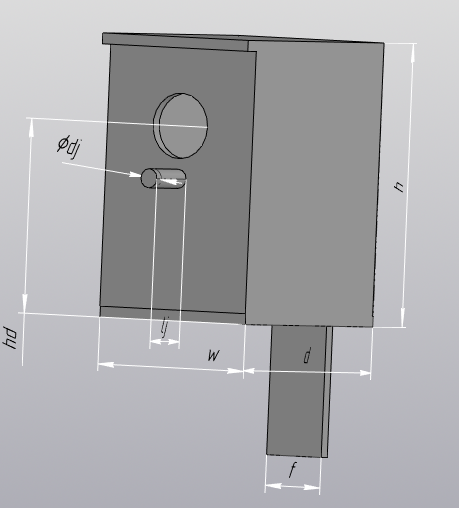


Рисунок 2.1 – 3D модель скворечника

# 3 Проект программы

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

UML – это язык графического описания для объектного моделирования в области разработки программного обеспечения. 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 [4].

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

# 3.1 Диаграмма вариантов использования (Use Cases)

Вариант использования (use case) — это описание множества последовательных действий (включая вариации), которые выполняются некоторым субъектом с целью получения результата, значимого для некоторого действующего лица [5].

На рисунке 3.1 представлена диаграмма вариантов использования.

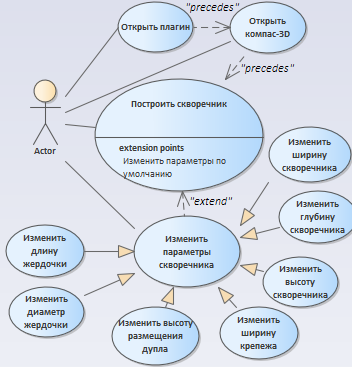


Рисунок 3.1 – Диаграмма вариантов использования

# 3.2 Диаграмма классов

Диаграмма классов (class diagram) служит для представления статической структуры модели системы в терминологии классов объектно-ориентированного программирования. Диаграмма классов может отражать, в частности, различные взаимосвязи между отдельными сущностями предметной области, а также описывает их внутреннюю структуру и типы отношений [6].

Для реализации был выбран следующий набор классов:

* MainForm – класс, в котором осуществляется взаимодействие между пользователем и программой;
* HouseBuilder – класс, осуществляющий вызов необходимых методов API САПР;
* HouseParameters – класс, осуществляющий хранение параметров скворечника;
* KompasConnector – класс, отвечающий за работу с API Kompas.

На рисунке 3.2. показана диаграмма классов.

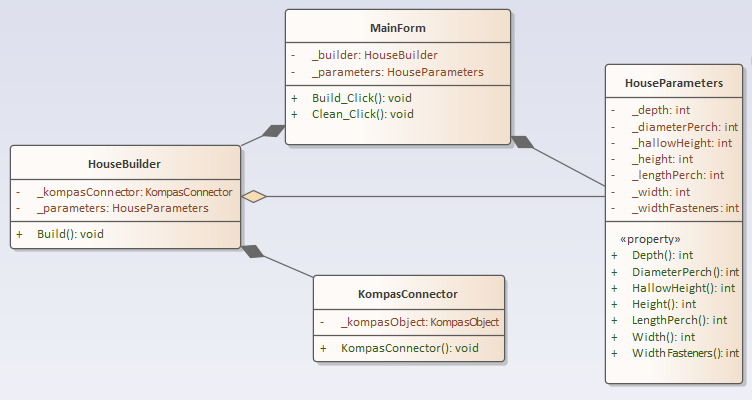


Рисунок 3.2 – Диаграмма классов

# 3.3 Макет пользовательского интерфейса

Плагин представляет собой пользовательскую форму с ячейками для ввода параметров: длина, высота, ширина корпуса, длина и диаметр жердочки, ширина крепежа скворечника, а также высота размещения дупла. Все численные значения вводятся в миллиметрах.

Запуск построения скворечника с измененными или параметрами по умолчанию осуществляется кнопкой «Build».

В полях красной области осуществляется ввод вышеперечисленных параметров скворечника.

В области, выделенной синим, осуществляется очищение вводимых параметров и само построение модели в компасе.

На рисунке 3.3 представлен макет интерфейса плагина.

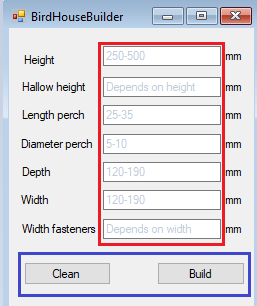


Рисунок 3.3 – Макет пользовательского интерфейса при запуске

Сообщения об ошибках будут выводиться после нажатия на кнопку «Build» при каждом несоответствии параметра с эталонным, то есть если пользователь допустит больше одной ошибки, при последующем нажатии на кнопку Build будет всплывать одно окно об ошибке, пока пользователь не введет все верно.

# Список использованных источников

КОМПАС-3D: О программе. Официальный сайт САПР КОМПАС [Электронный ресурс]. – Режим доступа: <http://kompas.ru/kompas-3d/about/> (дата обращения 15.02.2020)

Кидрук Максим. КОМПАС-3D V17 на 100% / М. Кидрук. – СПб.: Питер, 2009 – 560 с. (дата обращения 19.02.2020)

ВИКС-3D-Дом [Электронный ресурс]. – Режим доступа: <https://apps.autodesk.com/ACD/ru/Detail/Index?id=6781901237034763332&appLang=ru&os=Win32_64> (дата обращения 15.02.2020)

UML [Электронный ресурс]. — Режим доступа: <https://ru.wikipedia.org/wiki/UML> (дата обращения 26.02.2020)

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

М. Фаулер. UML. Основы, 3-е издание. Книга по UML для начинающих – 2018 – 192 с. (дата обращения 10.03.2020)