Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждения

высшего образования

ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

СИСТЕМ УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ (ТУСУР)

Кафедра компьютерных систем в управлении и проектировании (КСУП)

ПРОЕКТ СИСТЕМЫ

по дисциплине

«Основы разработки САПР» (ОРСАПР)

Студент гр. 586-1

\_\_\_\_\_\_\_\_\_\_\_\_\_С.А. Леонов

«\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2020г.

Руководитель

К.т.н., доцент каф. КСУП

\_\_\_\_\_\_\_\_\_\_А.А. Калентьев

«\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2020г.

2020

**Содержание**

1 Описание САПР 4

2 Описание API 17

3 Описание аналогов 18

3.1 Оборудование: Металлоконструкции 19

3.2 Оборудование: Кабели и жгуты 21

4 Описание предмета проектирования 27

5 Диаграмма пакетов плагина 28

6 Диаграмма классов плагина 30

7 Диаграмма прецедентов плагина 31

8 Пользовательский интерфейс 33

Список использованных источников 35

**1 Описание САПР**

Система КОМПАС-3D [1] предназначена для создания трехмерных ассоциативных моделей отдельных деталей и сборочных единиц, содержащих как оригинальные, так и стандартизованные конструктивные элементы. Параметрическая технология реализует возможность получения модели типовых изделий на основе ранее спроектированного прототипа.

Функциональные возможности проектирования:

* + - инструментальная поддержка методик проектирования “снизу-вверх” (использование готовых компонентов) и “сверху вниз” (проектирование компонентов в контексте конструкции);

Функциональные возможности моделирования:

* + - создание простых и сложных поверхностей;
    - ЗD-моделирование с учетом поля допуска;
    - поддержка всех возможностей твердотельного моделирования;
    - ассоциативное задание параметров элементов;
    - функционал для моделирования деталей из листового материала — команды создания листового тела, сгибов, отверстий, жалюзи, буртиков, штамповок и вырезов в листовом теле, замыкания углов, а также выполнения развертки полученного листового тела (в том числе формирования ассоциативного чертежа развертки);
    - булевы операции над типовыми формообразующими элементами;
    - построение вспомогательных прямых и плоскостей, эскизов, пространственных кривых (ломаных, сплайнов, различных спиралей);
    - создание конструктивных элементов — фасок, скруглений, отверстий, ребер жесткости, тонкостенных оболочек;
    - специальные возможности, облегчающие построение литейных форм — литейные уклоны, линии разъема, полости по форме детали (в том числе с заданием усадки);
    - создание любых массивов формообразующих элементов и компонентов сборок;
    - вставка в модель стандартных изделий из библиотеки, формирование пользовательских библиотек моделей;
    - наложение сопряжений на компоненты сборки (при этом возможность автоматического наложения сопряжений существенно повышает скорость создания сборки);
    - обнаружение взаимопроникновения деталей;
    - специальные средства для упрощения работы с большими сборками;
    - возможность гибкого редактирования деталей и сборок, в том числе с помощью характерных точек;
    - переопределение параметров любого элемента на любом этапе проектирования, вызывающее перестроение всей модели;
    - возможность создания таблиц переменных в моделях и графических документах;
    - работа с зеркальными исполнениями деталей и сборочных единиц, создание зеркального текста;
    - учет усадки, свойств и параметров материалов, а также технологии производства окончательного проекта.

Функциональные возможности разработки документации:

* + - разработка документации отвечающей требованием ГОСТ, ISO, DIN или стандартами, установленными на предприятии;
    - создание чертежей и технологической документации проекта;
    - создание вариантов исполнения для деталей и сборочных единиц.

**2 Описание API**

Взаимодействие внешнего приложения или подключаемого модуля с системой КОМПАС (с функциями моделирования, математическими функциями ядра системы и пр.) осуществляется посредством программных интерфейсов (API).

На данный момент в КОМПАС существуют API двух версий: API 5 [2] и API 7 [2]. Обе версии реализуют различные функции системы и не являются взаимозаменяемыми.

При создании подключаемых модулей в большей мере используется интерфейс API 5 [2].

Для начала взаимодействия с интерфейсом API необходимо выполнить следующие методы:

* Activator.CreateInstance(Type) – создание экземпляра указанного типа, используя конструктор этого типа без параметров. (метод пространства имен System);
* UIElement.ActivateControllerAPI() – активация контроллера API.

Необходимо использовать свойство видимости UI:

* UIElement.Visibility – получает или задает видимость пользовательского интерфейса данного элемента. Это свойство зависимостей (Пространство имен: System.Windows).

Методы взаимодействия:

* **Cr****eate** – создать документ-модель (деталь или сборку);

Синтаксис Autimation:

BOOL Create (BOOL invisible, BOOL typeDoc);

Синтаксис COM:

BOOL Create (BOOL invisible, BOOL \_typeDoc);

Входные параметры:

invisible – признак режима редактирования документа (TRUE – невидимый режим, FALSE – видимый режим);

typeDoc – тип документа (TRUE - деталь, FALSE – сборка).

Возвращаемое значение:

True – в случае успешного завершения.

* **Create** – создать объект в модели;

Синтаксис Automation:

BOOL Create();

Синтаксис COM:

BOOL Create();

Возвращаемое значение:

TRUE – в случае успешного завершения.

Примечание:

Данный метод необходимо вызвать для создания объекта в модели после создания интерфейса с помощью метода NewEntity и изменения параметров создаваемого объекта;

Если до вызова данного метода свойства объекта не изменены, то объект создается с параметрами, принятыми по умолчанию;

Если объект уже создан, то функция вызовет метод Update.

* **ActiveDocument3D** – получить указатель на интерфейс текущего документа трехмерной модели;

Синтаксис Automation:

LPDISPATCH ActiveDocument3D();

Возвращаемое значение:

Указатель на интерфейс документа трехмерной модели ksDocument3D.

Синтаксис COM:

Экспертная функция LPDOCUMENT3D ksGetActive3dDocument();

Возвращаемое значение:

Указатель на интерфейс документа трехмерной модели IDocument3D.

Примечание:

Если документ трехмерной модели не активен, функция возвращает NULL.

* **Document3D** – получить указатель на интерфейс документа трехмерной модели;

Синтаксис Automation:

LPDISPATCH Document3D();

Возвращаемое значение:

Указатель на интерфейс документа трехмерной модели ksDocument3D.

Синтаксис COM:

Экспертная функция LPDOCUMENT3D ksGet3dDocument();

Возвращаемое значение:

Указатель на интерфейс документа трехмерной модели IDocument3D.

* **GetPart** – получить указатель на интерфейс компонента в соответствии с заданным типом;

Синтаксис Automation:

LPDISPATCH GetPart(long type);

Синтаксис COM:

LPPART GetPart(int type);

Входные параметры:

type - тип компонента из перечисления “Типы компонентов”.

Типы компонентов:

pInPlace\_Part – компонент, редактируемый на месте;

pNew\_Part – новый компонент;

pEdit\_Part – редактируемый компоненты;

pTop\_Part – главный компонент, в составе которого находится новый или редактируемый или указанный компонент.

Возвращаемое значение:

Указатель на интерфейс компонента ksPart или IPart.

Примечание:

Тип задается для нового, редактируемого или главного компонента либо тип равен номеру компонента в документе;

Функция используется, чтобы получить доступ к компоненту документа;

Деталь или сборка являются компонентами. Сборка, в свою очередь, состоит из компонентов - деталей и подсборок.

* **GetDefaultEntity** –получить указатель на интерфейс объекта, создаваемого системой по умолчанию;

Синтаксис Automation:

LPDISPATCH GetDefaultEntity(short objType);

Синтаксис COM:

LPENTITY GetDefaultEntity(short objType);

Входные параметры:

objType – тип объекта.

Возвращаемое значение:

Указатель на интерфейс ksEntity или IEntity.

Типы объектов (objType):

o3d\_planeXOY 1 - плоскость XOY

o3d\_planeXOZ 2 - плоскость XOZ

o3d\_planeYOZ 3 - плоскость YOZ

o3d\_pointCS 4 - точка начала системы координат

o3d\_axisOX 71 - ось OX

o3d\_axisOY 72 - ось OY

o3d\_axisOZ 73 - ось OZ

* **N****e****wEntity** – создать новый интерфейс объекта и получить указатель на него;

Синтаксис Automation:

LPDISPATCH NewEntity(short objType);

Синтаксис COM:

LPENTITY NewEntity(short objType);

Входные параметры:

objType – тип объекта.

Возвращаемое значение:

Указатель на интерфейс ksEntity или IEntity.

Примечание:

Реальный объект создается в модели после вызова метода Create.

* **B****egin****Edit** – войти в режим редактирования эскиза;

Синтаксис Automation:

BeginEdit();

Возвращаемое значение:

SketchDoc – указатель на интерфейс IFragmentDocument.

Синтаксис COM:

HRESULT BeginEdit([out, retval] IFragmentDocument \*\* SketchDoc);

Возвращаемое значение:

SketchDoc – указатель на интерфейс IFragmentDocument.

Примечание:

После редактирования необходимо вызвать функцию ISketch::EndEdit.

* **EndEdit** – выйти из режима редактирования эскиза;

Синтаксис Automation:

BOOL EndEdit();

Синтаксис COM:

HRESULT EndEdit([out, retval] VARIANT\_BOOL \* res);

Возвращаемое значение:

TRUE – в случае успешного завершения;

FALSE – в случае неудачи.

* **ksLineSeg** – создать отрезок;

Синтаксис Automation:

long ksLineSeg(double x1, double y1, double x2,   
 double y2, long style);

Входные параметры:

x1, y1 – координаты первой точки отрезка;

x2, y2 – координаты второй точки отрезка;

style – стиль линии.

Возвращаемое значение:

Указатель на отрезок – в случае удачного завершения;

0 – в случае неудачи.

* **ExtrusionParam** – получить указатель на интерфейс параметров элемента выдавливания;

Синтаксис Automation:

LPDISPATCH ExtrusionParam();

Синтаксис COM:

LPEXTRUSIONPARAM ExtrusionParam();

Возвращаемое значение:

Указатель на интерфейс объекта ksExtrusionParam или IExtrusionParam.

* **SetSketch** – изменить указатель на интерфейс эскиза элемента;

Синтаксис Automation:

BOOL SetSketch(LPDISPATCH sketch);

Синтаксис COM:

BOOL SetSketch(LPENTITY sketch);

Входные параметры:

sketch – указатель на интерфейс эскиза ksEntity или IEntity.

Возвращаемое значение:

TRUE – в случае успешного завершения.

* **ksCircle** – создать окружность;

Синтаксис Automation:

long ksCircle(double xc, double yc, double rad,   
long style);

Входные параметры:

xc, yc – координаты центра окружности;

rad – радиус окружности;

style – стиль линии.

Возвращаемое значение:

указатель на окружность – в случае удачного завершения;

0 – в случае неудачи.

* **array** – получить указатель на интерфейс массива скругляемых объектов;

Синтаксис Automation:

LPDISPATCH Array();

Синтаксис COM:

LPENTITYCOLLECTION Array();

Возвращаемое значение:

Указатель на интерфейс массива объектов ksEntityCollection или IEntityCollection.

Примечание:

В возвращаемом массиве включен контроль, не позволяющий добавить в массив нулевые указатели на элементы.

* **A****ddMateConstraint** – добавить сопряжение в сборку;

Синтаксис Automation:

BOOL AddMateConstraint(long constraintType,   
LPDISPATCH object1, LPDISPATCH object2, short direction,   
short fixed, double value);

Синтаксис COM:

BOOL AddMateConstraint(long constraintType,   
LPENTITY obj1, LPENTITY obj2, short direction, short fixed,   
double val);

Входные параметры:

|  |  |
| --- | --- |
| constraintType | - тип сопряже­ния из перечи­сления MateConstraintType; |
| object1 | - указатель на интерфейс первого объ­екта, на кото­рый наклады­вается сопряжение (ksEntity или IEntity); |
| object2 | - указатель на интерфейс второго объек­та, на который накладывает­ся сопряжение (ksEntity или iEntity); |
| direction | - ориентация  (1 - объекты однонаправ­ленные,   0 - направле­ние не учиты­вается,  -1 - объекты разнонаправ­ленные); |
| fixed | - признак фик­сации деталей перед выпол­нением  (0 - детали не фиксируются,   1 - фиксирует­ся первая де­таль,  2 - фиксирует­ся вторая де­таль); |
| val | - параметр для ограничений (расстояние или угол меж­ду объектами). |

Возвращаемое значение:

TRUE - в случае успешного завершения.

Примечание:

Направление задаётся знаком параметра val.

* **M****at****eConstraintType** – типы сопряжений;

|  |  |  |
| --- | --- | --- |
| mc\_Coincidence | 0 | совпадение объектов |
| mc\_Parallel | 1 | параллельность |
| mc\_Perpendicular | 2 | перпендикулярность |
| mc\_Tangency | 3 | касательность |
| mc\_Concentric | 4 | концентричность |
| mc\_Distance | 5 | постоянное расстояние между объек­тами |
| mc\_Angle | 6 | постоянный угол между объектами |
| mc\_InPlace | 7 | создание компонента "на месте" (эквивалентно совпадению системы координат плоскости, на которой со­здается компонент, и системы коорди­нат плоскости первого эскиза этого компонента) |

* **ksCreateInsertionFragment** – создать вставку фрагмента;

Синтаксис Automation:

int LIB\_FUNC ksCreateInsertionFragment(char\*fileName);

Входные параметры:

fileName – полное имя файла фрагмента.

Возвращаемое значение:

1 – в случае удачи.

* **SaveAs** – сохранить документ – модель (деталь или сборку) с новым именем;

Синтаксис Automation:

BOOL SaveAs(BSTR fileName);

Синтаксис COM:

BOOL SaveAs(LPOLESTR fileName);

Входные параметры:

filename – имя файла документа.

Возвращаемое значение:

TRUE – в случае успешного завершения.

* **SaveAsEx** – сохранить документ с новым именем файла;

Синтаксис Automation:

BOOL SaveAsEx(BSTR fileName, long saveMode);

Синтаксис COM:

BOOL SaveAsEx(LPOLESTR fileName, long saveMode);

Входные параметры:

|  |  |
| --- | --- |
| fileName | – новое имя файла доку­мента; |
| saveMode | – версия для сохранения: -1 – в предыду­щую версию; 0 – в текущую версию; 1 – в версию 5.11. |

Возвращаемое значение:

TRUE – в случае успешного завершения;

FALSE – в случае неудачи.

Примечание:

Если fileName = NULL, то используется имя файла из документа. Если же и в документе отсутствует имя файла, то выводится ошибка.

Используемые свойства:

* **thin** – признак формирования тонкой стенки элемента;

Тип данных: BOOL.

Значения свойства:

|  |  |
| --- | --- |
| TRUE | – тонкая стенка формируется; |
| FALSE | – тонкая стенка не формируется. |

Синтаксис Automation:

|  |  |
| --- | --- |
| thin = iThinParam.thin | Получить свойство |
| iThinParam.thin = thin | Установить свойство |
| thin = iThinParam.GetThin() | Получить свойство |
| iThinParam.SetThin(thin) | Установить свойство |

* **radius** – радиус скругления;

Тип данных: double.

Синтаксис Automation:

|  |  |
| --- | --- |
| radius = iFilletDefinition.radius | Получить свойство |
| iFilletDefinition.radius = radius | Установить свойство |
| radius= iFilletDefinition.GetRadius() | Получить свойство |
| iFilletDefinition.SetRadius(radius) | Установить свойство |

* **directionType** – направление выдавливания;

Тип данных: short.

Синтаксис Automation:

|  |  |
| --- | --- |
| directionType= iBaseExtrusion.directionType | Получить свойство |
| iBaseExtrusion.directionType= directionType | Установить свойство |
| directionType= iBaseExtrusion.GetDirectionType() | Получить свойство |
| iBaseExtrusion.SetDirectionType  (directionType) | Установить свойство |

Значения свойства:

dtNormal – прямое направление (для тонкой стенки наружу);

dtReverse – обратное направление (для тонкой стенки внутри);

dtBoth – в обе стороны;

dtMiddlePlane – от средней плоскости.

Примечание:

1.Нормаль, проведенная к грани, всегда направлена наружу ("из тела детали");

2.Прямое направление совпадает с нормалью, проведенной к плоскости эскиза;

3.Для вырезаемого элемента выдавливания направление противоположно нормали.

* **Depth** – глубина выдавливания;

Тип данных: double.

Синтаксис Automation:

|  |  |
| --- | --- |
| Depth = iObject.Depth(normal); | Получить свойство |
| iObject.Depth(normal) = Depth; | Установить свойство |
| Depth = iObject.GetDepth(normal); | Получить свойство |
| iObject.SetDepth(normal, Depth); | Установить свойство |

Синтаксис COM:

|  |  |
| --- | --- |
| iObject->get\_Depth(normal,&Depth); | Получить свойство |
| iObject->put\_Depth(normal,Depth); | Установить свойство |

Входные параметры:

TRUE - глуби­на выдавлива­ния в прямом направлении;

FALSE - глу­бина выдавли­вания в обрат­ном направлении.

Примечание:

Позволяет считывать и устанавливать глубину выдавливания.

* **Plane** – опорная плоскость;

Тип данных: указатель на интерфейс IModelObject.

Синтаксис Automation:

|  |  |
| --- | --- |
| Plane = iObject.Plane; | Получить свойство |
| iObject.Plane = Plane; | Установить свойство |
| Plane = iObject.GetPlane(); | Получить свойство |
| iObject.SetPlane(Plane); | Установить свойство |

Синтаксис COM:

iObject->get\_Plane(&Plane); Получить свойство

iObject->put\_Plane(Plane); Установить свойство

* **A****ngle** –угол поворота эскиза относительно проекции системы координат модели на плоскость эскиза (в градусах).

Тип данных: double.

Синтаксис Automation:

|  |  |
| --- | --- |
| Angle = iObject.Angle; | Получить свойство |
| iObject.Angle = Angle; | Установить свойство |
| Angle = iObject.GetAngle(); | Получить свойство |
| iObject.SetAngle(Angle); | Установить свойство |

Синтаксис COM:

iObject->get\_Angle(&Angle); Получить свойство

iObject->put\_Angle(Angle); Установить свойство

**3 Описание аналогов**

**3.1 Оборудование: Металлоконструкции**

Оборудование: Металлоконструкции [3] — приложение для КОМПАС-3D, предназначенное для автоматизации работ по проектированию конструкций из профильного металлопроката. Приложение позволяет быстро проектировать всевозможные рамы и каркасы, автоматически создавать комплект документации.

Создание металлоконструкции в приложении начинается с построения Трехмерного каркаса — геометрических осей, которые являются эскизом конструкции. После чего для каждой из осей назначается профиль. Сортамент профиля может выбираться из нового Каталога профилей, входящего в комплект поставки приложения, либо из Справочника Материалы и Сортаменты для КОМПАС. Для удобства построения и редактирования металлоконструкции в приложении реализован механизм Характерных точек, который позволяет задавать длину и угол поворота профиля непосредственно в окне построения. При изменении Трехмерного каркаса металлоконструкция перестроится автоматически.

После назначения профилей необходимо проработать отдельные узлы металлоконструкции. Для этого в приложении есть специальные инструменты. Можно корректировать длины деталей, задавать угловую или стыковую разделки, строить дополнительные элементы в виде ребер жесткости или фасонок.

Для созданной с помощью приложения конструкции можно автоматически получить спецификацию либо любые другие виды настраиваемых отчетов. Металлоконструкция, спроектированная в приложении, может быть проверена на наличие пересечений. Трехмерная модель позволяет сразу выявить возможные нестыковки. Оборудование: Металлоконструкции позволяет избежать дополнительных затрат на материал и инструмент.

На рисунке 3.1 представлен интерфейс, каталога выбора структуры швеллера, приложения “Оборудование: Металлоконструкции”.

Рисунок 3.1 – Интерфейс, каталога выбора структуры швеллера, приложения “Оборудование: Металлоконструкции”

**3.2 Оборудование: Кабели и жгуты**

Оборудование: Кабели и жгуты [4] – специализированное приложение для КОМПАС-3D, предназначенное для автоматизации процесса трехмерного моделирования электрических кабелей и жгутов, а также для выпуска конструкторской документации на эти изделия.

Приложение автоматически создает:

* трассы прокладки кабелей и жгутов в пространстве изделия;
* позиционирование кабельных частей соединителей по их приборным (блочным) частям;
* трехмерные модели кабелей и жгутов с учетом количества и диаметра проводников, проходящих по трассам;
* скругления в точках поворота трассы с учетом автоматически рассчитываемого условного диаметра кабеля или ветви жгута;
* сборочный чертеж кабеля или жгута;
* спецификацию к чертежу с подсчетом длин всех проводников и количества различных материалов;
* позиционные обозначения на сборочном чертеже;
* объекты спецификации типа «сборочная единица» для жгутов или кабелей.

В качестве базы данных компонентов используется любой источник трехмерных моделей. Это могут быть отдельные файлы 3D-моделей, библиотеки моделей, а также каталоги Библиотеки Стандартные Изделия.

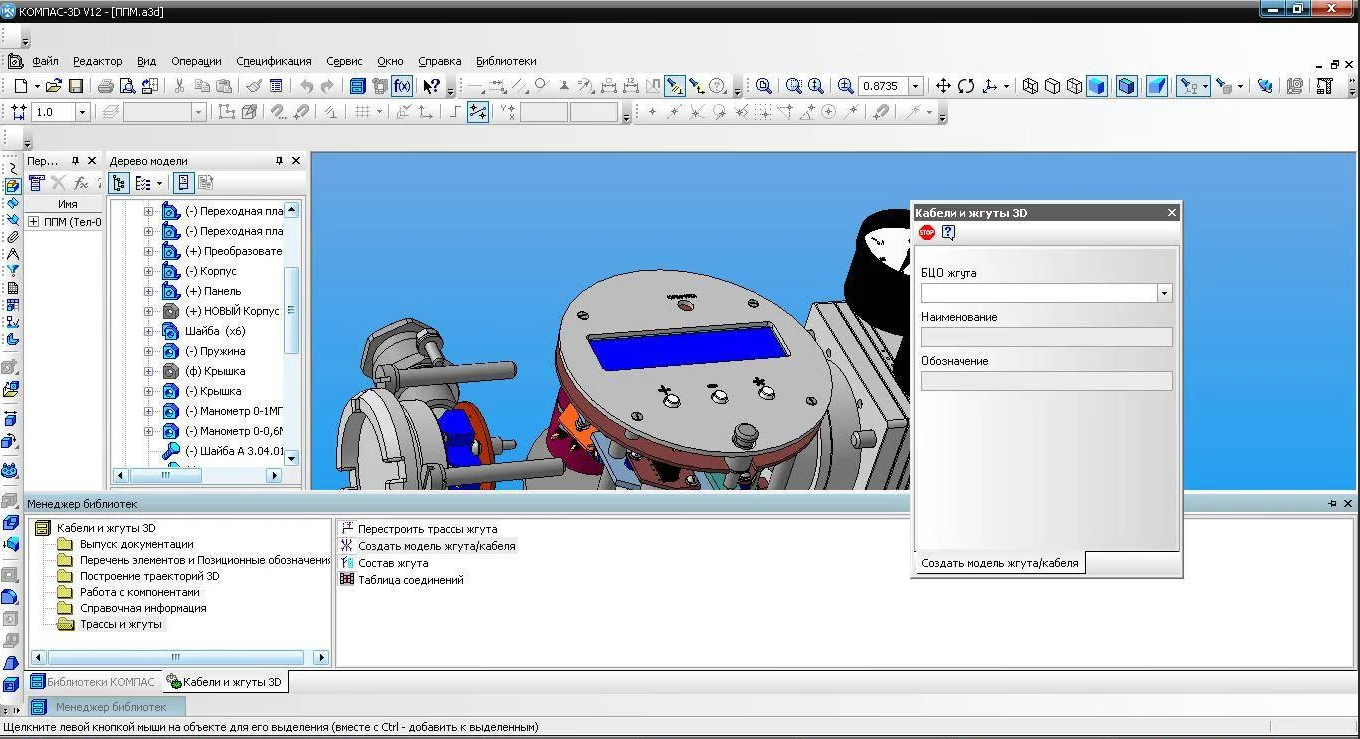
В качестве базы данных проводов и кабелей используются как специальные текстовые файлы в составе приложения, так и Библиотеку Материалы и Сортаменты.

Размеры на получаемых чертежах полностью соответствуют размерам кабелей и жгутов, измеренным по осям их трехмерных моделей. Суммарная длина проводников в спецификации дополнительно учитывает все припуски на монтаж и провисание, заданные пользователем еще на этапе формирования трасс. Пользователи могут добавлять к моделям разъемов таблицы с контактами, соответствующими им сигналами и маркировкой подводящих проводов. При этом данные из блоков и приборов об адресах электрической связи и сигналах автоматически передаются в кабельные части жгутов.

На сборочном чертеже жгута или кабеля конструктор при необходимости может размещать таблицы распайки разъемов.

Приложение применяется в приборостроении для моделирования соединений между блоками и разъемами в целом, а также в электротехнике, где должны моделироваться связи между отдельными контактами.

На рисунке 3.2 представлен интерфейс создания модели кабеля/жгута приложения “Оборудование: Кабели и жгуты”.

Рисунок 3.2 – Интерфейс создания модели кабеля/жгута приложения “Оборудование: Кабели и жгуты”

**4 Описание предмета проектирования**

Акустической системой [5] называется громкоговоритель, предназначенный для использования в качестве функционального звена в бытовой радиоэлектронной аппаратуре. Под «громкоговорителем» понимается «устройство для эффективного излучения звука в окружающее пространство в воздушной среде, содержащее одну или несколько головок громкоговорителей, при наличии акустического оформления, электрических устройств (фильтры, трансформаторы, регуляторы и т.п.).

Так как при проектировании внешнего вида модели, невозможно учесть технические характеристики звукового оборудования, пункты ГОСТ связанные с качеством звучания рассматриваться не будут.

Разрабатываемая модель музыкальной колонки должна соответствовать ГОСТ 23262-88 [6] Системы акустические бытовые. Общие технические условия.

Рассмотрим пункты, связанные в внешним видом модели.

Пункт 2.1.5 По внешнему виду и качеству исполнения АС должна соответствовать образцу – эталону утвержденному в соответствии с ГОСТ 15.009 – 86 (заменен на ГОСТ 15.009-91).

Обратимся к ГОСТ 15.009-91 [7].

ГОСТ 15.009-91 [7] Система разработки и постановки продукции на производство (СРПП). Непродовольственные товары народного потребления.

1.Непродовольственные товары народного потребления (далее - продукция), подлежащие разработке и постановке на производство, должны удовлетворять запросам населения с учетом возрастных, социальных и национальных групп потребителей, направлений развития ассортимента и моды, а также отвечать требованиям, обеспечивающим безопасность для жизни и здоровья населения и охрану окружающей среды.

Исследуем характеристики проектируемой модели.

Акустическое оформление: закрытый ящик.

Описание: Оформление оказывает непосредственное влияние на качество звука. Корпус у акустики закрытого типа представляет собой герметично закрытый ящик с выведенным на фронтальную панель диффузором (излучателем звука) динамика. Недостатки такой акустики – низкая чувствительность, не очень глубокий бас, преимущества – простота конструкции и хорошие переходные характеристики, которые гарантируют низкий уровень искажений и точное воспроизведение звука.

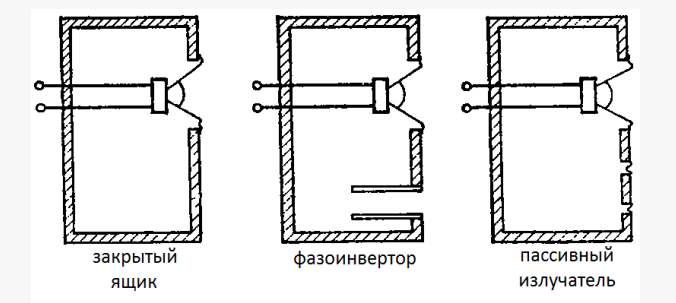


Рисунок 4.1 – Различные виды низкочастотных оформлении

Количество полос фронтальных колонок: 1.

Описание: Количество полос, на которое разбит частотный диапазон фронта колонок.

Ширина фронтальных колонок: от 150 до 200 мм.

Высота фронтальных колонок: от 100 до 500 мм.

Длинна фронтальной колонки: от 200 до 300 мм.

На рисунке 4.2 приведена 3D модель музыкальной колонки с обозначенными параметрами L, H, W, D, LS, HS, WS.

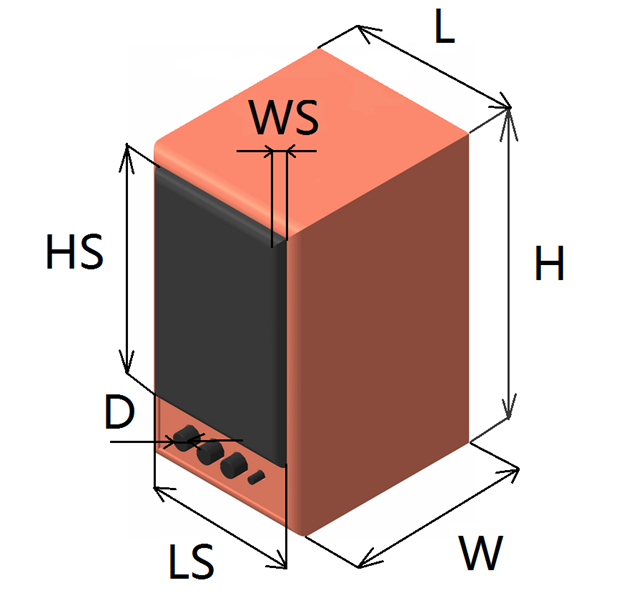


Рисунок 4.2 – 3D модель музыкальной колонки с обозначенными параметрами

Обозначенные параметры:

H – Высота корпуса;

L – Длинна корпуса;

W – Ширина корпуса;

HS – Высота динамик;

WS – Ширина наружного выступа динамика;

LS – Длинна динамика;

D – Диаметр реле регулировки.

Рассмотрим чертеж музыкальной колонки с максимально допустимыми размерами.

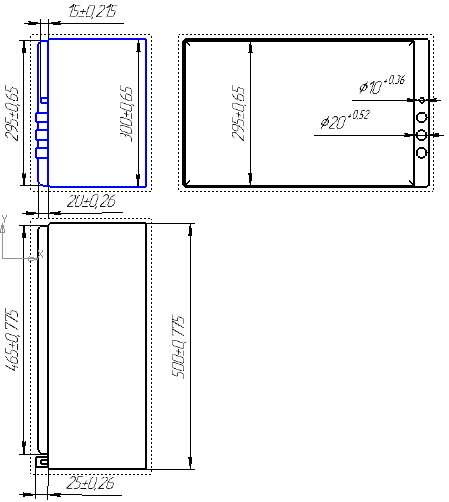


Рисунок 4.3 – Чертеж музыкальной колонки в 3-х проекциях

Проведем анализ аналогичных моделей, распространяемых сетью магазинов DNS. Данные модели соответствуют запросам населения с учетом возрастных, социальных и национальных групп потребителей, направлений развития ассортимента и моды, а также отвечают требованиям, обеспечивающим безопасность для жизни и здоровья населения и охрану окружающей среды.

1.Колонки 2.0 Genius SP-HF160 [8]



Рисунок 4.4 – Изображение колонок 2.0 Genius SP-HF160

Характеристики:

Акустическое оформление: закрытый ящик;

Ширина фронтальных колонок: 84 мм;

Высота фронтальных колонок: 129 мм;

Длинна фронтальной колонки: 84 мм;

Количество полос фронтальных колонок: 1.

2.Колонки 2.0 Defender SPK-240 [9]

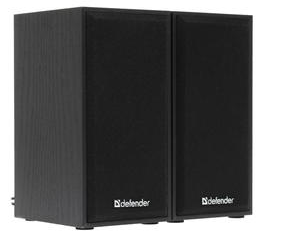


Рисунок 4.5 – Изображение колонок 2.0 Defender SPK-240

Характеристики:

Акустическое оформление: закрытый ящик;

Ширина фронтальных колонок: 146 мм;

Высота фронтальных колонок: 150 мм;

Длинна фронтальной колонки: 95 мм;

Количество полос фронтальных колонок: 1.

Характеристики проектируемой модели и моделей, распространяемых магазинами бытовой техники аналогичны.

Исходя из сравнительного анализа разрабатываемой модели и моделей, распространяемых в магазинах бытовой техники можно сделать вывод, что колонки с такими характеристиками и подобным дизайном удовлетворяют запросам населения с учетом возрастных, социальных и национальных групп потребителей, направлений развития ассортимента и моды, и будут востребованы на рынке. Особенно популярны модели с деревянным корпусом и MDF, выполненные в темном цвете.

При рассмотрении не учитываются параметры:

* Качество звучания;
* Масса;
* Цвет.

**5 Диаграмма пакетов плагина**

Пакет (package) [10] – это инструмент группирования, который позволяет взять любую конструкцию UML объединить ее элементы в единицы высокого уровня. В основном пакеты служат для объединения классов в группы, но могут применяться для любой другой конструкции языка UML.

Рассмотрим диаграмму пакетов плагина на рисунке 5.1.

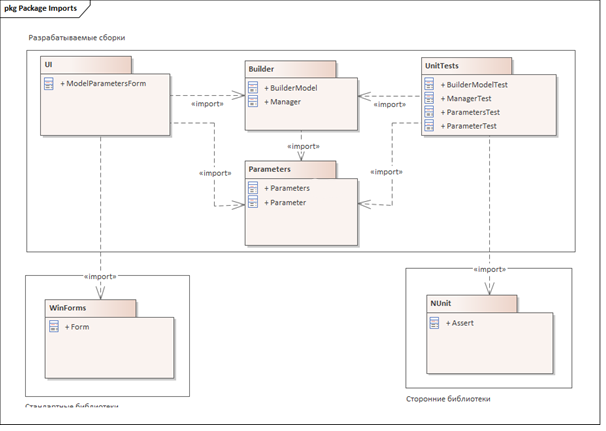


Рисунок 5.1 – Диаграмма пакетов плагина

Пакеты Builder и Parameters содержат классы бизнес логики плагина. Пакет Builder использует Parameters.

Пользовательский интерфейс содержится в пакете UI, при работе данный пакет использует бизнес логику приложения (пакеты Builder и Parameters) и графический интерфейс (пакет WinForms).

Пакет UnitTests хранит классы с юнит-тестами приложения, для работы используется вспомогательный пакет NUnit.

**6 Диаграмма классов плагина**

Диаграмма классов [10] описывает типы объектов системы и различного рада статические отношения, которые существуют между ними. На диаграммах классов отображаются также свойства классов, операции классов и ограничения, которые накладываются на связи между объектами. В UML термин функциональность применяется в качестве основного термина, описывающего и свойства, и операции класса. Целью создания диаграммы классов является графическое представление статической структуры декларативных элементов системы.

Рассмотрим диаграмму классов плагина на рисунке 6.1.

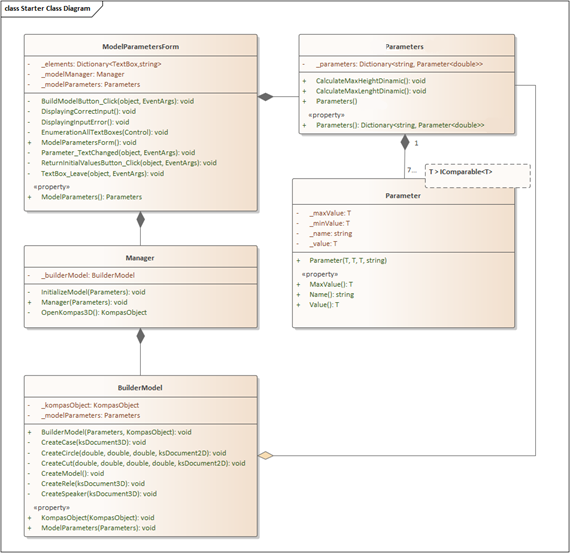


Рисунок 6.1 – Диаграмма классов плагина

Класс Parameter хранит информацию об одном параметре проектируемой модели.

Класс ModelParameters хранит словарь со всеми параметрами модели, также поддерживаются методы расчета максимальных значений параметров высоты и длинны динамик.

Класс BuilderModel хранит параметры модели, введенные пользователем. Реализуются методы используемы для построения 3D модели, данные методы использую параметры модели.

Класс Manager хранит объект класса построителя 3D модели (BuilderModel), реализует методы для начала работы с САПР.

Класс ModelParametersForm является формой пользовательского интерфейса. Реализует методы, используемые для взаимодействия с пользователем.

**7 Диаграмма прецедентов плагина**

Прецеденты [10] – это технология определения функциональных требований к системе. Работа прецедентов заключается в описании типичных взаимодействий между пользователем системы и самой системой.

Прецеденты представляют собой ценный инструмент для понимания функциональных требований к системе.

Рассмотрим диаграмму прецедентов плагина на рисунке 7.1.

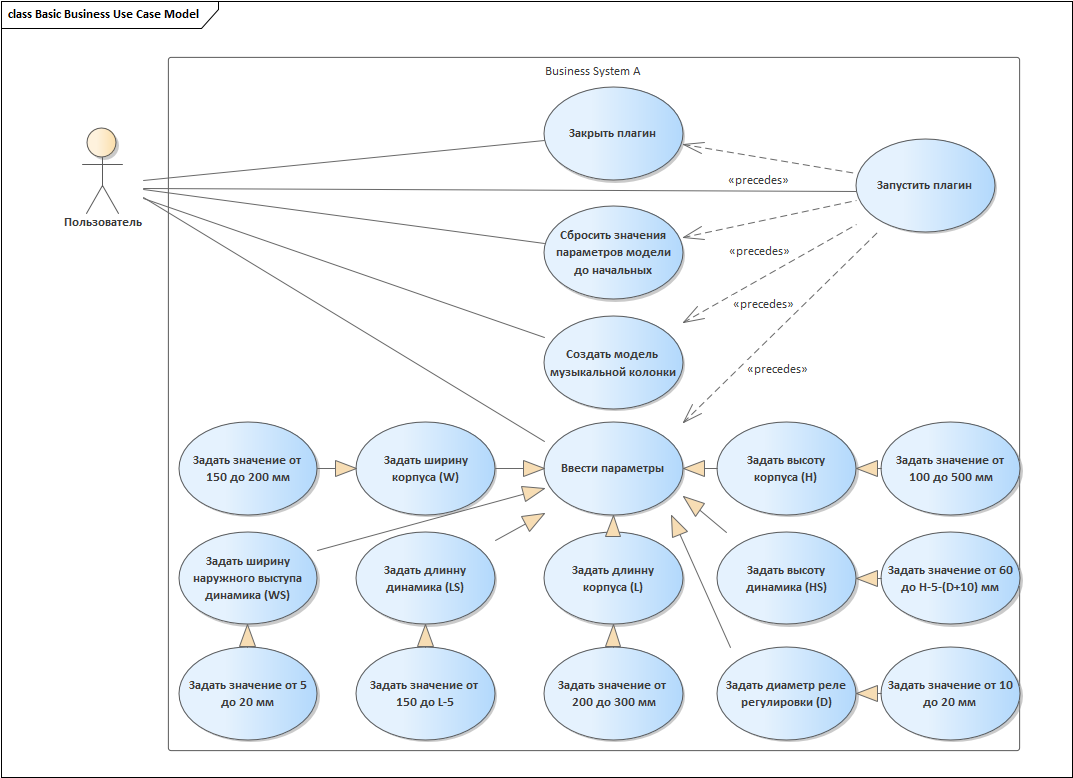


Рисунок 7.1 – Диаграмма прецедентов плагина

Действующим лицом выступает пользователь.

Для пользователя доступны следующие возможности:

* Сбросить значения параметров модели до начальных;
* Ввести параметры;
* Создать модель музыкальной колонки;
* Запустить плагин;
* Закрыть плагин.

**8 Пользовательский интерфейс**

Пользовательский интерфейс [11] – интерфейс, обеспечивающий передачу информации между пользователем – человеком и программно-аппаратными компонентами компьютерной сети.

На рисунке 8.1 представлен макет пользовательского интерфейса плагина с обозначением групп разделения по элементам проектируемой модели.

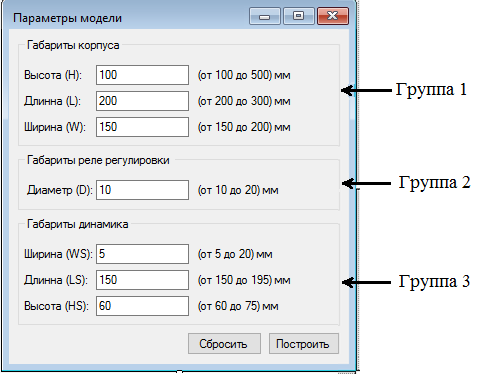


Рисунок 8.1 – Макет пользовательского интерфейса плагина с обозначением групп разделения по элементам проектируемой модели

После запуска плагина перед пользователем появляется главное окно (Рисунок 5), содержащее 3 элемента GroupBox. Каждый элемент содержит одну из групп параметров 3D модели музыкальной колонки.

Параметры распределены по группам согласно их принадлежности элементам проектируемой модели.

Группа 1: Габариты корпуса.

Группа 2: Габариты реле регулировки.

Группа 3: Габариты динамика.

Для каждого параметра предусмотрено 3 элемента:

- TextBox принимает значение параметра, вводимое пользователем. Окрашивается в красный цвет, при вводе некорректного значения и в зеленый цвет при вводе корректного параметра. При вводе некорректного значения и потере фокуса элементом, значение возвращается к последнему введенному корректному значению.

* Lable указывающий название параметра, располагается слева от элемента TextBox;
* Lable указывающий интервал возможных значений вводимого параметра. Располагается справа от элемента TextBox.

После запуска для ввода доступно 7 параметров:

* Высота корпуса (H);
* Длинна корпуса (L);
* Ширина корпуса (W);
* Диаметр реле регулировки (D);
* Ширина динамика (WS);
* Длинна динамика (LS);
* Высота динамика (HL).

Все поля предварительно заполнены минимально возможными значениями параметров.

В правом нижнем углу располагаются 2 элемента Button.

При клике по элементу с текстом “Сбросить”, все элементы формы возвращаются к начальным значениям.

Элемент с текстом “Построить” не требует предварительного ввода параметров и доступен всё время работы с параметрами. После клика по данному элементу происходит открытие САПР КОМПАС – 3D и построение модели музыкальной колонки согласно заданным пользователем параметрам. После завершения построения все элементы формы возвращаются к начальным значениям.

Верстка главного окна адаптивна.

# Список использованных источников

# 1. КОМПАС – 3D [Электронный ресурс]. – URL: <https://kompas.ru/> (дата обращения: 27.02.2020);

# 2. Базовые интерфейсы API системы КОМПАС [Электронный ресурс]. – URL: <https://it.wikireading.ru/23741> (дата обращения: 27.02.2020);

# 3. Приложение “Оборудование: Металлоконструкции” [Электронный ресурс]. – URL: <https://kompas.ru/kompas-3D/application/machinery/steel-constructions-3d/> (дата обращения: 27.02.2020);

# 4. Приложение “Оборудование: Кабели и жгуты” [Электронный ресурс]. – URL: <https://kompas.ru/kompas-3D/application/instrumentation/equipment-cables-and-harness/> (дата обращения: 27.02.2020);

# 5. Акустическая система. Основные понятия [Электронный ресурс]. – URL: <http://ldsound.ru/akusticheskaya-sistema-osnovnye-ponyatiya/> (дата обращения: 27.02.2020);

# 6. ГОСТ 23262-88 Системы акустические бытовые. Общие технические условия [Электронный ресурс]. – URL:<https://internet-law.ru/gosts/gost/11778/>

# (дата обращения: 27.02.2020);

# 7. ГОСТ 15.009-91 Система разработки и постановки продукции на производство. Непродовольственные товары народного потребления [Электронный ресурс]. – URL: <https://internet-law.ru/gosts/gost/38525/> (дата обращения: 27.02.2020);

# 8. Колонки 2.0 Genius SP-HF160 [Электронный ресурс]. – URL: <https://www.dns-shop.ru/product/f3cbaea6eb083330/kolonki-20-genius-sp-hf160>

# (дата обращения: 27.02.2020);

# 9. Колонки 2.0 Defender SPK-240 [Электронный ресурс]. – URL: <https://www.dns-shop.ru/product/cad4739395ec3330/kolonki-20-defender-spk-240/> (дата обращения: 27.02.2020);

# 10. Мартин Фаулер UML Основы. Краткое руководство по стандартному языку объектного моделирования [Электронный ресурс]. – URL: <litportal.ru›trial/pdf/24500318.pdf> (дата обращения: 27.02.2020);

# 11. Интерфейс пользователя [Электронный ресурс]. – URL: <https://ru.wikipedia.org/wiki/%D0%98%D0%BD%D1%82%D0%B5%D1%80%D1%84%D0%B5%D0%B9%D1%81_%D0%BF%D0%BE%D0%BB%D1%8C%D0%B7%D0%BE%D0%B2%D0%B0%D1%82%D0%B5%D0%BB%D1%8F>

# (дата обращения: 27.02.2020);