Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего профессионального образования

ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ СИСТЕМ

УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ (ТУСУР)

Кафедра компьютерных систем в управлении и проектировании (КСУП)

ПРОЕКТ СИСТЕМЫ

к курсовому проекту по дисциплине

«Основы разработки САПР»

Тема проекта: «РАЗРАБОТКА ПЛАГИНА «ТЕРМОС» НА БАЗЕ СИСТЕМЫ «AUTODESK INVENTOR»»

|  |  |
| --- | --- |
|  | Выполнил:  Студент гр. 582-2  \_\_\_\_\_\_\_\_\_\_ Егоров А.С  «\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_2016 г.  Проверил  м.н.с. ЛИКС, каф. КСУП  \_\_\_\_\_\_\_\_\_\_\_Калентьев А.А  «\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_2016 г. |

Томск 2016

# Описание САПР

## Обзор программы «Inventor»

Autodesk Inventor — система трехмерного твердотельного и поверхностного параметрического проектирования (САПР) компании Autodesk, предназначенная для создания цифровых прототипов промышленных изделий. Инструменты Inventor обеспечивают полный цикл проектирования и создания конструкторской документации:

* 2D/3D-моделирование;
* создание изделий из листового материала и получение их разверток;
* разработка электрических и трубопроводных систем;
* проектирование оснастки для литья пластмассовых изделий;
* динамическое моделирование;
* параметрический расчет напряженно-деформированного состояния деталей и сборок;
* визуализация изделий;
* автоматическое получение и обновление конструкторской документации (оформление по ЕСКД).

**Возможности программы** «**Inventor»**

Autodesk Inventor является гибкой платформой для разработки специализированных машиностроительных приложений, вертикальных и корпоративных решений.

Интерфейс прикладного программирования Inventor построен на технологии COM Automation. Доступ к нему можно используя большинство распространенных в настоящее время средств разработки ПО: Microsoft Visual C++®, VB, C# и Delphi. Autodesk Inventor включает также и популярную среду разработки Microsoft VBA.[1]

## Описание API интерфейсов системы «Inventor»

Общая схема объектной модели AI c рядом основных компонентов представлена на рисунке 1.1

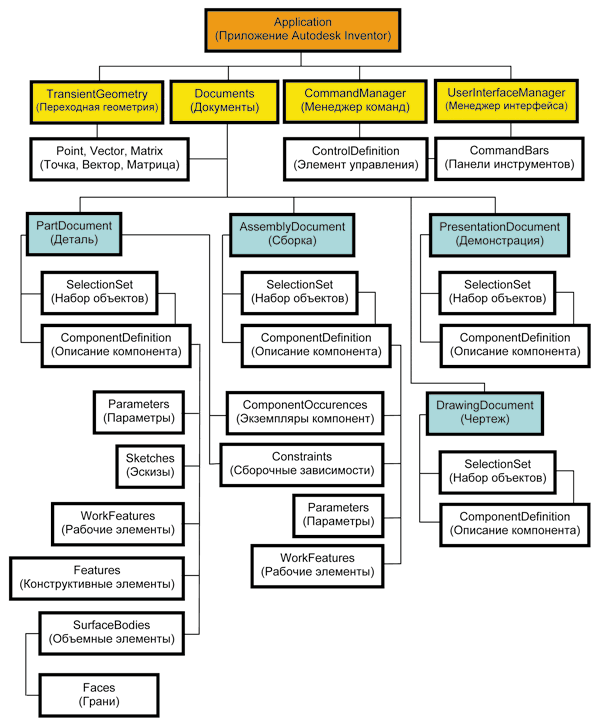


Рисунок 1.1 – Объектная модель Autodesk Inventor (фрагмент)

Проект приложения AI (Application) представляет собой систему специальных документов САПР (Documents), переходной геометрии (TransientGeometry), менеджера команд (CommandManager) и менеджера пользовательского интерфейса (UserInterfaceManager). Основные типы документов AI: сборки (AssemblyDocument), деталь (PartDocument), демонстрация (PresentationDocument), чертеж (DrawingDocument).

AI — среда ассоциативного проектирования, поэтому каждый из типов документов, входящих в проект на всем протяжении разработки, сохраняет взаимосвязь с другими документами, в которых он используется.

Так, изменения в файле детали отражаются в документе сборки, содержащей эту деталь, или файле чертежа, имеющем виды этой детали.

Каждый из типов документов позволяет программно формировать и работать с наборами составляющих объектов (SelectionSet) и отдельными компонентами (ComponentDefinition). Для отдельного компонента доступны все характеризующие его объекты и свойства. Например, описание компонента (ComponentDefinition) детали (PartDocument) включает эскизы (плоские и трехмерные контуры), объемные элементы (вращения, выдавливания, сдвига, оболочки и др.) с опорными эскизами, конструктивные элементы (фаски, сопряжения, резьбы и др.), рабочие элементы (вспомогательные плоскости, оси, точки), таблицы параметров, управляющих геометрией. Описание компонента (ComponentDefinition) документа сборки (AssemblyDocument) в качестве экземпляра компонента (ComponentOccurences) может включать документы деталей (PartDocument).

Элементы переходной геометрии (TransientGeometry) — это объекты (точки, векторы, матрицы), не имеющие графического выражения в моделях, но используемые при создании геометрии в документах AI.

Отдельно следует сказать об объектах типа сборочных зависимостей (Constraints), с помощью которых в документе сборки между деталями устанавливаются взаимные ограничения позиционирования и движения, что позволяет моделировать в сборке функциональность объекта-прототипа. Сборочные зависимости (Constraints) ассоциативно доступны как из объектов деталей, так и из применяющих их сборок[2].

## Обзор аналогов

### Генератор компонентов вала

Проектирование формы вала, добавление и вычисление нагрузок и опор, а также других расчетных параметров выполняется в генераторе компонентов вала программы. Выполнение проверки прочности и создание вала в Autodesk Inventor. Вал состоит из отдельных секций (цилиндра, конуса и многоугольника), включая конструктивные элементы (фаски, сопряжения, шейки и т.д.).

Генератор компонентов вала полезен для выполнения следующих действий:

* Проектирование и вставка валов с неограниченным числом сечений (цилиндры, конусы, многоугольники) и конструктивных элементов (сопряжений, фасок, резьбы и т. д.);
* Проектирование полых валов;
* Вставка конструктивных элементов (сопряжений, фасок, резьбы) во внутренние отверстия;
* Разделение цилиндра вала с сохранением длины сечения вала;
* Сохранение валов в библиотеке шаблонов;
* Добавление в проект вала неограниченного числа нагрузок и опор.

Для некоторых элементов в генераторе валов используются данные из Библиотеки компонентов.

Прим.: Для вставки компонентов с использованием генератора валов необходимо подключение к Библиотеке компонентов и настройка Библиотеки компонентов на компьютерах пользователей.

Выберите в Библиотеке компонентов следующие элементы:

* Шпоночные канавки;
* Плоские шпоночные канавки;
* Шпоночные канавки со скругленным торцом;
* Стопорные кольца.

При проектировании валов с целью ускорения процесса проектирования при работе в графическом окне воспользуйтесь 3D-ручками, чтобы изменить размеры сечений.

**Конструктивные элементы ребра**

Конструктивные элементы, которые можно разместить только на ребре вала: фаски, сопряжения, стопорные гайки, резьба. На ребре вала можно разместить только один конструктивный элемент.(см. рисунок 1.2):

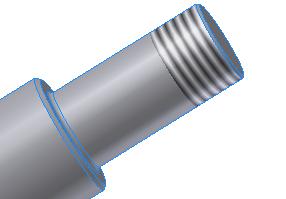


Рисунок 1.2 – Демонстрация резьбы на вале

**Средние конструктивные элементы**

Конструктивные элементы, которые можно разместить на сечении вала: шпоночные пазы, стопорные кольца, надрезы и т. д. На сечении вала можно разместить неограниченное количество таких конструктивных элементов.(см. рисунок 1.3):

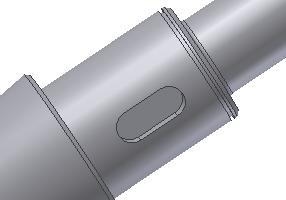


Рисунок 1.3 – Демонстрация размещения шпоночного паза на вале

**Маркеры положения**

Определение точек сечения, в которых можно разместить нагрузки и опоры. Для перемещения силы или опоры и их назначения другому маркеру положения используйте перетаскивание, удерживая клавишу ALT. (см. рисунок 1.4):

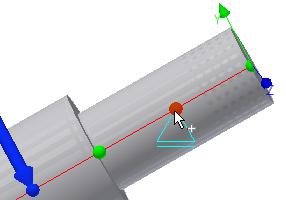


Рисунок 1.4 – Определение точек для размещения нагрузки и опоры

**Графический просмотр**

|  |
| --- |
| При проектировании вала с помощью генератора валов в окне Autodesk Inventor можно просмотреть графическое изображение в трехмерном виде (со всеми сечениями, конструктивными элементами и 3D ручками), а на вкладке "Модель" — изображение в двухмерном виде. Просмотр в двухмерном режиме является дополнительным. |

При переходе на вкладку "Расчет" графическое изображение в окне Autodesk Inventor и на вкладке "Расчет" меняется на предварительное изобрежение нагрузок и опор. [3] (см. рисунок 1.5):

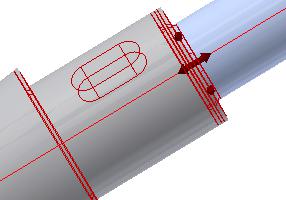


Рисунок 1.5 – Предварительный просмотр изображения нагрузок и опор

# Описание предмета проектирования

Термос — специальный сосуд для хранения содержимого в постоянной температуре.

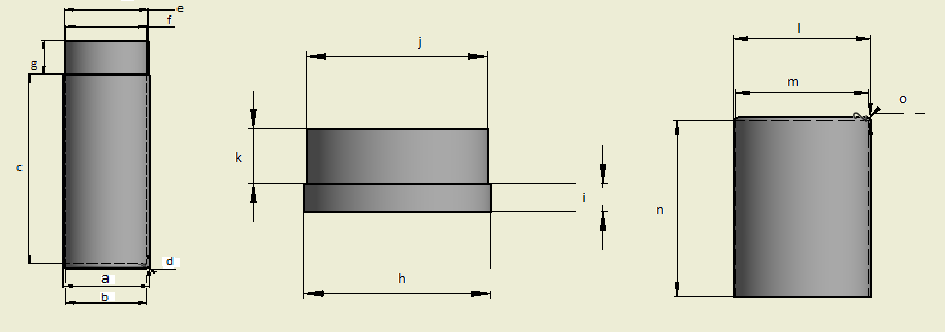


Рисунок 2.1 – Термос

# Проект программы

## Для описания проектной части программы используется унифицированный язык моделирования (UML). UML — является графическим языком для визуализации, описания параметров, конструирования и документирования различных систем (программ в частности). Диаграммы создаются с помощью специальных CASE средств, например Rational и Enterprise Architect. В данном случае используется второй программный продукт. На основе UML построены: диаграммы вариантов использования (use case) и диаграмма классов с точки зрения реализации.[3]

## Use Case диаграмма

Проектируемая система представляется в виде множества сущностей или актеров, взаимодействующих с системой с помощью вариантов использования. При этом актером (actor) или действующим лицом называется любая сущность, взаимодействующая с системой извне. Это может быть человек, техническое устройство, программа или любая другая система, которая может служить источником воздействия на моделируемую систему так, как определит сам разработчик. Вариант использования служит для описания сервисов, которые система предоставляет актеру (см. рисунок 3.1):

Проектируемая система представляется в виде множества сущностей или актеров, взаимодействующих с системой с помощью, так называемых прецедентов. При этом актером (actor) или действующим лицом называется любая сущность, взаимодействующая с системой извне. Другими словами, каждый вариант использования определяет некоторый набор действий, совершаемый системой при диалоге с актером. При этом ничего не говорится о том, каким образом будет реализовано взаимодействие актеров с системой.[5]

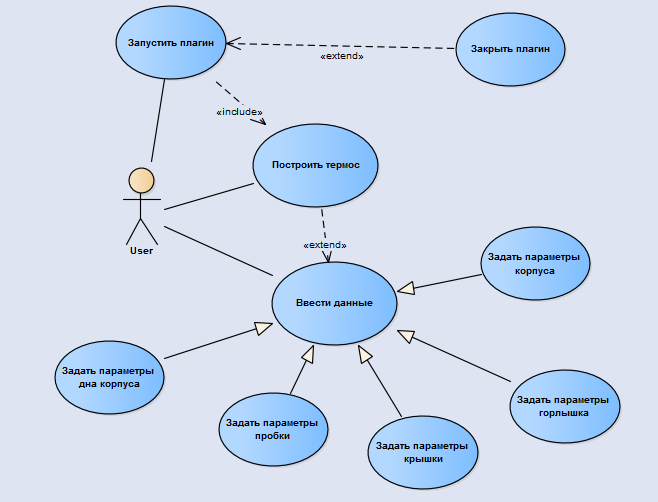


Рисунок 3.1 – Диаграмма вариантов использования

## UML диаграмма классов

Диаграмма классов 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, такими как объекты и подсистемы, а также описывает их внутреннюю структуру (поля, методы и т.п.) и типы отношений (наследование, реализация интерфейсов и т.п.). На данной диаграмме не указывается информация о временных аспектах функционирования системы. С этой точки зрения диаграмма классов является дальнейшим развитием концептуальной модели проектируемой системы. На этом этапе принципиально знание ООП подхода и паттернов проектирования.[6]

Для реализации подсистемы были спроектированы следующие классы (рисунок 3.2):

* ThermosForm — диалоговый класс, обеспечивающий взаимодействие между пользователем и программой через форму, подробное описание в Приложении A (Таблица A.1);
* InventorApi — класс, обеспечивающий взаимодействие с необходимыми методами Inventor API, подробное описание в Приложении A (Таблица A.2);
* ThermosProperties — класс, хранящий в себе все параметры модели термоса, и осуществляет проверку зависимых параметров, подробное описание в Приложении A (Таблица A.3);
* Parametr — класс предназначен для хранения и обработки параметра модели термоса, подробное описание в Приложении A (Таблица A.4);
* ThermosModel — класс построения модели термоса в системе Autodesk Inventor, подробное описание в Приложении A (Таблица A.5);
* ParametrType — перечисление всех параметров модели термоса, подробное описание в Приложении A (Таблица A.).

Описание классов (см. рисунок 3.2).

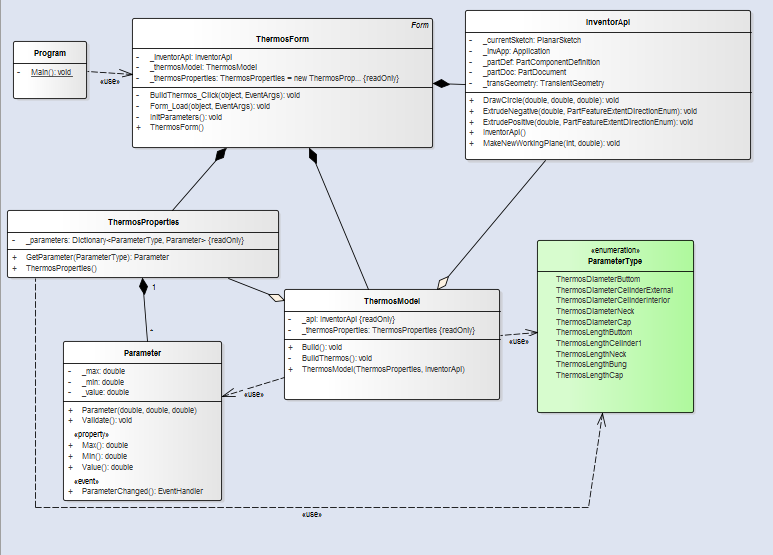


Рисунок 3.2 – UML диаграмма классов

# Макет пользовательского интерфейса

В текущем проекте будет использоваться Windows Form. На рисунке 4.1 представлен проект графического интерфейса для ввода параметров модели.

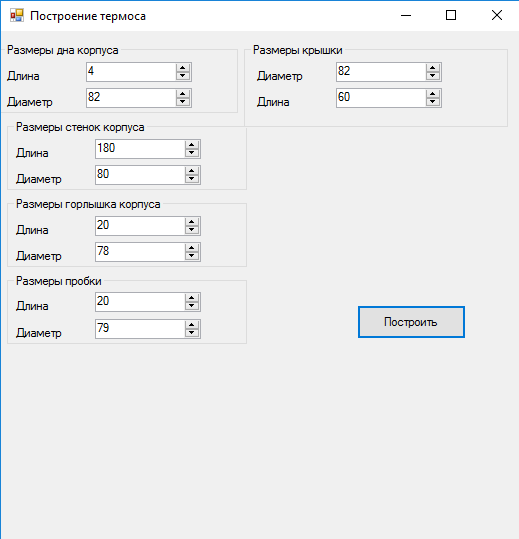
****

Рисунок 4.1 – Макет пользовательского интерфейса

Пользователю предлагается ввести ряд параметров модели. Данный интерфейс имеет 3 основных компонента:

1. Полный перечень вводимых параметров;
2. Поля для ввода значений вводимых параметров;
3. Кнопка для построения модели.

## Список используемых источников

1. Разработка приложений для Autodesk Inventor. [Электронный ресурс]. – Режим доступа: http://www.autodesk.ru/adsk/servlet/index?id=22740406&siteID=871736 (дата обращения: 29.05.16)
2. API Inventor. [Электронный ресурс]. – Режим доступа: http://www.sapr.ru/article.aspx?id=20656&iid=941 (дата обращения: 29.05.16)
3. UML tools for software development and modelling – Enterprise Architect [Электронный ресурс]. – Режим доступа: UML modeling tool. <http://www.sparxsystems.com.au/> (дата обращения: 5.05.16)
4. Проектирование программного обеспечения/Хабрахабр. [Электронный ресурс]. – Режим доступа: <http://habrahabr.ru/post/74330> (дата обращения: 29.05.16)
5. Фаулер М. UML. Основы, 3-е издание. – Пер. с англ. – СПб: Символ-Плюс, 2004. – 192с., ил.

# Приложение А

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

«−» − обозначение private (закрытого) поля;

«+» − обозначение public (открытого) поля.

Таблица А.1 – Описание полей и методов класса ThermosForm

|  |  |
| --- | --- |
| Класс\поля\методы | Описание |
| +ThermosForm() | Конструктор класса ThermosForm, для инициализации компонентов и параметров |
| –InitParameters():void | Метод инициализация параметров |
| –Form\_Load(object, EventArgs):void | Метод загрузка формы |
| –BuildThermos\_Click(object, EventArgs):void | Метод, обрабатывающий функции построения термоса |
| –\_inventorApi: InventorApi | Интерфейс API Inventor |
| –\_thermosModel: ThermosModel | Модель термоса |
| –\_thermosProperties: ThermosProperties = new ThermosProperties() {readOnly} | Параметры модели термоса |

Таблица А.2 – Описание полей и методов класса InventorApi

|  |  |
| --- | --- |
| Класс\поля\методы | Описание |
| +InventorApi() | Класс, в котором содержатся используемые методы Inventor API |
| +MakeNewWorkingPlane(int n, double offset) | Метод, создающий новую рабочую плоскость |
| +DrawCircle(double centerPointX, double centerPointY, double diameter) | Метод, рисующий круг на активном скетче |

Продолжение таблицы А.2

|  |  |
| --- | --- |
| +ExtrudePositive (double distance, PartFeatureExtentDirectionEnum extrudeDirection = PartFeatureExtentDirectionEnum.kPositiveExtentDirection) | Метод, выдавливающий текущий эскиз |
| +ExtrudePositive (double distance, PartFeatureExtentDirectionEnum extrudeDirection = PartFeatureExtentDirectionEnum.kPositiveExtentDirection) | Метод, выдавливающий с вычитанием |
| –\_currentSketch: PlanarScetch | Текущий эскиз |
| –\_invApp: Application | Ссылка на приложение Inventor |
| –\_partDef: PartComponentDefinition | Описание документа |
| –\_pertDoc: PartDocument | Создание документа в приложении |
| –\_transGeometry: TransientGeometry | Геометрия приложения |

Таблица А. 3– Описание полей и методов класса ThermosProperties

|  |  |
| --- | --- |
| Класс\поля\методы | Описание |
| +ThermosProperties | Класс, хранящий в себе все параметры модели, так же осуществляет проверку зависимых параметров |
| –ParameterChangedValue(object sender, EventArgs e) | Событие, вызываемое при изменении какого-либо параметра модели |
| +GetParameter(ParameterType parameterType) | Возвращает указанный параметр |
| –\_parametrs: Dictionery<ParametrType, Parametr> | Словарь параметров термоса |

Таблица А.4 – Описание полей и методов класса Parameter

|  |  |
| --- | --- |
| Класс\поля\методы | Описание |
| +Parameter | Класс предназначен для хранения и обработки параметра термоса |

Продолжение таблицы А.4

|  |  |
| --- | --- |
| +Validate(): void | Данный метод меняет значение параметра на корректное в случае выхода его за пределы допустимого диапазона |
| +ParameterChanged(): EventHandler | Данное событие вызывается в случае, если значение параметра было изменено |
| +Value(): double | Данное свойство позволяет установить или запросить текущее значение параметра |
| +Min(): double | Данное свойство позволяет установить или запросить минимальное значение параметра |
| +Max(): double | Данное свойство позволяет установить или запросить максимальное значение параметра |
| –\_max: double | Максимально возможное значение параметра |
| –\_min: double | Минимальное возможное значение параметра |
| –\_value: double | Значение параметра |

Таблица А.5– Описание полей и методов класса ThermosModel

|  |  |
| --- | --- |
| Класс\поля\методы | Описание |
| +ThermosModel(ThermosProperties, InventorApi) | Класс для построения модели |
| +Build(): double | Метод, строящий модель |
| –BuildThermos(): void | Метод, строящий ступени термоса |
| –\_api: InventorApi | Интерфейс API Inventor |
| –\_thermosProperties: thermosProperties | Параметры модели термоса |

Таблица А.6– Описание перечислителей перечисления ParametrType

|  |  |
| --- | --- |
| Список перечислителей | Описание |
| ThermosDiameterButtom | Диаметр дна корпуса |
| ThermosDiameterCelinderExternal | Внешний диаметр корпуса |
| ThermosDiameterCelinderInterior | Внутренний диаметр корпуса |
| ThermosDiameterNeck | Диаметр горлышка и пробки |
| ThermosDiameterCap | Диаметр крышки |

Продолжение таблицы А.6

|  |  |
| --- | --- |
| ThermosLengthButtom | Длина дна корпуса |
| ThermosLengthCelinder1 | Длина корпуса |
| ThermosLengthNeck | Длина горлышка |
| ThermosLengthBung | Длина пробки |
| ThermosLengthCap | Длина крышки |