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)

Проект системы по лабораторному проекту

по дисциплине «Основы разработки САПР»

Выполнил:   
Студент гр. 586-2

Козырева В.М.

Проверил:

к.т.н., доцент каф. КСУП

\_\_\_\_\_\_\_\_\_ Калентьев А.А.

« » 2020 г.

Томск 2020

**Оглавление**

[1 Описание САПР 3](#_Toc36017410)

[1.1 Описание программы 3](#_Toc36017411)

[1.2 Описание API 3](#_Toc36017412)

[1.3 Обзор аналогов 7](#_Toc36017413)

[1.3.1 Glassful plagin 7](#_Toc36017414)

[1.3.2 Plug-in "Build a Glass" for Compass 3d 8](#_Toc36017415)

[2 Описание предмета проектирования 9](#_Toc36017416)

[3 Проект программы 12](#_Toc36017417)

[3.1 Диаграмма вариантов использования (Use Cases) 11](#_Toc36017418)

[3.2 Диаграмма классов 13](#_Toc36017419)

[3.3 Макет пользовательского интерфейса 13](#_Toc36017420)

[Список использованной литературы 15](#_Toc36017421)

# 1 Описание САПР

# Описание программы

САПР (Система автоматизированного проектирования) – автоматизированная система, реализующая информационную технологию выполнения функций проектирования, представляет собой организационно-техническую систему, предназначенную для автоматизации процесса проектирования, состоящую из комплекса технических, программных и других средств автоматизации его деятельности.

SOLIDWORKS[1] – программный комплекс САПР для автоматизации работ промышленного предприятия на этапах конструкторской и технологической подготовки производства. Обеспечивает разработку изделий любой степени сложности и назначения. Работает в среде Microsoft Windows. САПР поддерживает различные чертежные стандарты, такие как GOST, ANSI, ISO и т.д. Система включает программные модули собственной разработки, а также сертифицированное ПО от специализированных разработчиков (SolidWorks Gold Partners). Также имеется поддержка работы с пользовательскими макросами и сторонними библиотеками.

# Описание API

API (англ. Application Programming Interface) - описание способов (набор классов, процедур, функций, структур или констант), которыми одна программа может взаимодействовать с другой. В данном случае используется Solidworks API[2], которое содержит сотни функций, вызывающихся из Visual Basic (VB), Visual Basic for Applications (VBA), VB.NET, C++, C# или файлов макросов SOLIDWORKS. Эти функции предоставляют прямой доступ к функциональным возможностям SOLIDWORKS.

Для разработки плагина требуется набор простых функций, таких как создание эскиза, выдавливание и т.д. Набор используемых функций представлен ниже.

**SelectByID2** – Выбор указанного объекта.

***Входные параметры***:

* Name – Имя требуемого объекта;
* Type – Тип объекта;
* X – Значение Х-координаты;
* Y – Значение Y-координаты;
* Z – Значение Z-координаты;
* Append – Добавляется ли к текущему набору выбранных элементов;
* Mark – Метка;
* Callout – Указатель на связанный вынос;
* SelectOption – опция выбора.

***Выходные параметры***: True если объект успешно выбран, иначе false.

**InsertSketch** – Создание нового эскиза/выход из режима эскиза.

***Входные параметры***: UpdateEditRebuild – True для сохранения изменений и выхода из режима эскиза, иначе false

**Insert3DSketch** – Создание нового эскиза/выход из режима эскиза.

***Входные параметры***: UpdateEditRebuild – True для сохранения изменений и выхода из режима эскиза, иначе false

**CreateSketchPlane** – Создание эскиза трехмерной плоскости.

***Входные параметры***: Relation – Описание отношений (swGuideCurveInfluence\_e) между элементами детали и плоскостью для ее позиционирования. На вход подается три таких параметра.

***Выходные параметры***: True если объект успешно создан, иначе false

**CreateCircle** – Создание окружности.

***Входные параметры***:

* XC – Х-координата центра окружности;
* YC – Y-координата центра окружности;
* ZC – Z-координата центра окружности;
* XR – Х-координата точки на окружности;
* YR – Y-координата точки на окружности;
* ZR – Z-координата точки на окружности.

***Выходные параметры***: Указатель на построенный элемент.

**ClearSelection2** – Очищение списка выделенных элементов.

***Входные параметры***: All – True для отчистки всего списка элементов, false для удаления только одного активного элемента списка.

**InsertProtrusionBlend** – Создает вытянутый элемент или выступ из выбранных профилей, осевой линии и направляющих кривых (Вытягивание по сечениям).

***Входные параметры***:

* Closed – True для закрытой верхней части, иначе false;
* KeepTangency – True для сохранения точек касания между кривыми сечения;
* ForceNonRational - True для получения более гладких поверхностей;
* TessToleranceFactor – Фактор, который контролирует количество промежуточных секций, используемых для верхней части с осевой линией; значение по умолчанию - 1,0; чем больше значение, тем больше промежуточных секций создается;
* StartMatchingType – Тип касания в начальном профиле:
  + 0 – нет;
  + 1 – касательная к нормали профиля;
  + 2 – касательная к выбранному вектору;
  + 3 – касание ко всем смежным граням, имеющим ребро с начальным профилем;
  + 4 – касательная к некоторым из выбранных граней, имеющих общий край с начальным профилем (недоступно).
* EndMatchingType – Тип касания в конечном профиле:
  + 0 – нет;
  + 1 – касательная к нормали профиля;
  + 2 – касательная к выбранному вектору;
  + 3 – касание ко всем смежным граням, имеющим ребро с начальным профилем;
  + 4 – касательная к некоторым из выбранных граней, имеющих общий край с начальным профилем (недоступно).
* StartTangentLength – Начальная длина касательной;
* EndTangentLength – Конечная длина касательной;
* StartTangentDir – True прямое направление, false - обратное;
* EndTangentDir – True прямое направление, false - обратное;
* IsThinBody – True для тонкостенной модели;
* Thickness1 – Значение толщины для первого направления;
* Thickness2 – Значение толщины для второго направления;
* ThinType – Тип построения тонкостенной модели;
  + 0 = прямое направление;
  + 1 = обратное направление;
  + 2 = средняя плоскость;
  + 3 = два направления.
* Merge – True для объединения результата построения в многотельной части;
* UseFeatScope – True если функция влияет только на выбранные тела, false если влияет на все тела;
* UseAutoSelect – True для автовыбора всех тел, на которые будет влиять функция;
* GuideCurveInfluence – Влияние направляющих кривых согласно swGuideCurveInfluence\_e.

***Выходные параметры***: Указатель на вытянутый объект.

**InsertCutBlend** – Создает вырез из выбранных профилей, осевой линии и направляющих кривых (Вырез по сечениям).

***Входные параметры***: Аналогичны предыдущему методу.

***Выходные параметры***: Указатель на вырезанный объект.

**FeatureExtrusion2** – Операция вытягивания.

***Входные параметры***:

* Sd – True для вытягивания в одно направление, false в два направления;
* Flip – True для вытягивания в обратном направлении;
* Dir – True для изменения направления выдавливания;
* T1 – Тип завершения для первого конца выдавливания согласно swEndConditions\_e;
* T2 – Тип завершения для второго конца выдавливания согласно swEndConditions\_e;
* D1 – Глубина выдавливания для первого конца выдавливания;
* D2 – Глубина выдавливания для второго конца выдавливания;
* Ddir1 – направление первого угла;
* Ddir2 – направление второго угла;
* TranslateSurface2 – второй тип завершения выдавливания;
* Merge – объединить результаты в много-тельной части;
* UseFeatScope – функция влияет на выбранные объекты и на все;
* UseAutoSelect – автоматически выбирать объекты;
* T0 – начальное условие;
* StartOffset – смещение относительно T0;
* FlipStartOffset – направление относительно T0;

***Выходные параметры***: Указатель на вытянутый объект.

**FeatureCut4** – Операция вырезания.

***Входные параметры***: Аналогичны предыдущему методу.

***Выходные параметры***: Указатель на вырезанный объект.

# Обзор аналогов

# 1.3.1 Glassful plugin

Данный плагин[3] позволяет построить модель стеклянного стакана по заданным параметрам. В отличии от разрабатываемого продукта, здесь всего пять настраиваемых параметров:

* Толщина стенки стакана (см)
* Диаметр верхней окружности (см)
* Высота стакана (см)
* Толщина дна стакана (см)
* Диаметр нижней окружности (см)

Кроме того, есть функционал, позволяющий построить граненый стакан.

На рисунке 1.1 представлен интерфейс плагина.

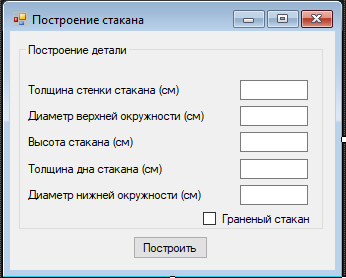


Рисунок 1.1 – Интерфейс плагина для построения стакана

# 1.3.2 Plug-in "Build a Glass" for Compass 3d

Данный плагин[4] позволяет построить модель стеклянного стакана по заданным параметрам. Доступные для заполнения параметры меняются в зависимости от выбранного типа стакана. Здесь их три:

* Гофрированный (рис. 1.2);
* Гладкий (рис. 1.3);
* Граненый (рис. 1.4).

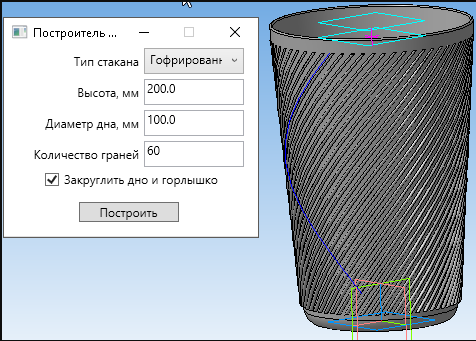


Рисунок 1.2 – Интерфейс плагина и результат построения гофрированного стакана

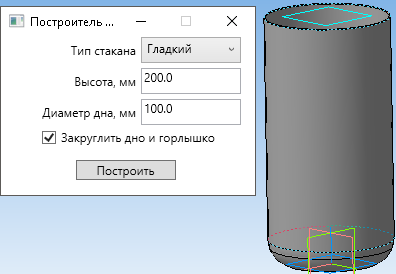


Рисунок 1.3 – Интерфейс плагина и результат построения гладкого стакана

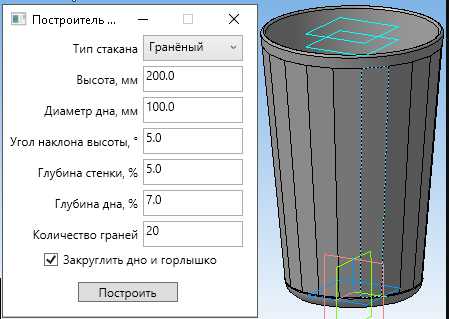


Рисунок 1.4 – Интерфейс плагина и результат построения граненого стакана

# 2 Описание предмета проектирования

Стакан – обычно стеклянный сосуд, близкий по форме к цилиндру или усечённому конусу, без ручки и без ножки. Применяется для холодных и горячих напитков.

* + толщина дна (0 < Hb < 0,4\*H);
  + толщина стенок (0 < Ts < 0,2\*Rb);
  + ширина горлышка (0 ≤ Ht < 0,15\*H);
  + толщина горлышка (0 < Tt <0,3\*Rb);
  + радиус дна (0 < Rb);
  + радиус горлышка (Rb < Rt < 1,5\*Rb);
  + высота стакана (2\*Rb < H);
  + материал.

Пример проектируемого изделия приведен ниже, на рисунках 2.1 – 2.2.

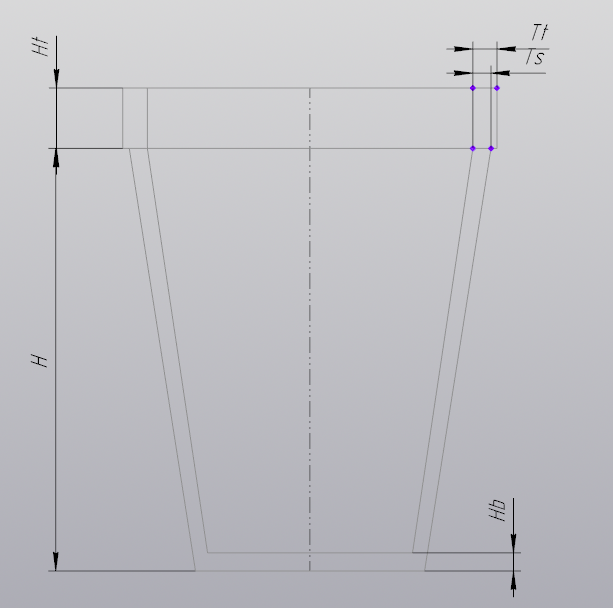


Рисунок 2.1 – Размерные выноски на виде сбоку

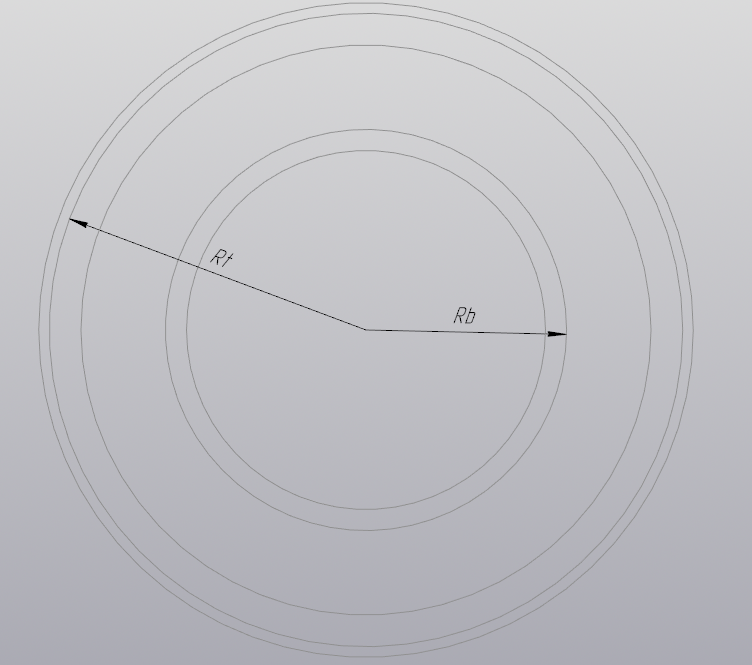


Рисунок 2.2 – Размерные выноски на виде сверху

# 3 Проект программы

# 3.1 Диаграмма вариантов использования (Use Cases)

Диаграмма вариантов использования в [UML](https://ru.wikipedia.org/wiki/UML) — диаграмма, отражающая отношения между [актерами](https://ru.wikipedia.org/wiki/%D0%90%D0%BA%D1%82%D0%BE%D1%80_(UML)) и [прецедентами](https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%B5%D1%86%D0%B5%D0%B4%D0%B5%D0%BD%D1%82_(UML)) и являющаяся составной частью модели прецедентов, позволяющей описать систему на концептуальном уровне. Основное назначение диаграммы — описание функциональности и поведения, позволяющее заказчику, конечному пользователю и разработчику совместно обсуждать проектируемую или существующую систему. [5]

На рисунке 3.1 представлена диаграмма вариантов использования.

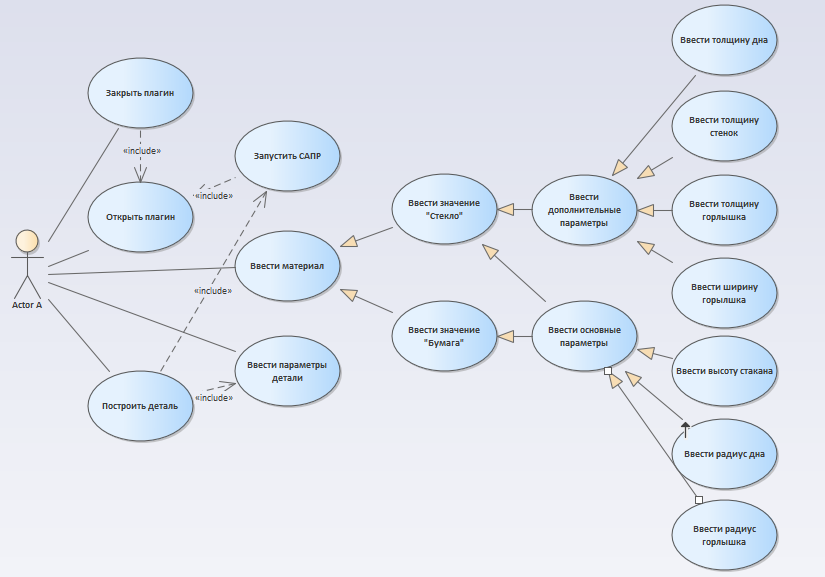


Рисунок 3.1- Диаграмма вариантов использования

# 3.2 Диаграмма классов

UML – это язык графического описания для объектного моделирования в области разработки программного обеспечения. UML является языком широкого профиля, это – открытый стандарт, использующий графические обозначения для создания абстрактной модели системы, называемой UML-моделью. UML был создан для определения, визуализации, проектирования и документирования, в основном, программных систем. [6]

На рисунке 3.2 представлена диаграмма классов.

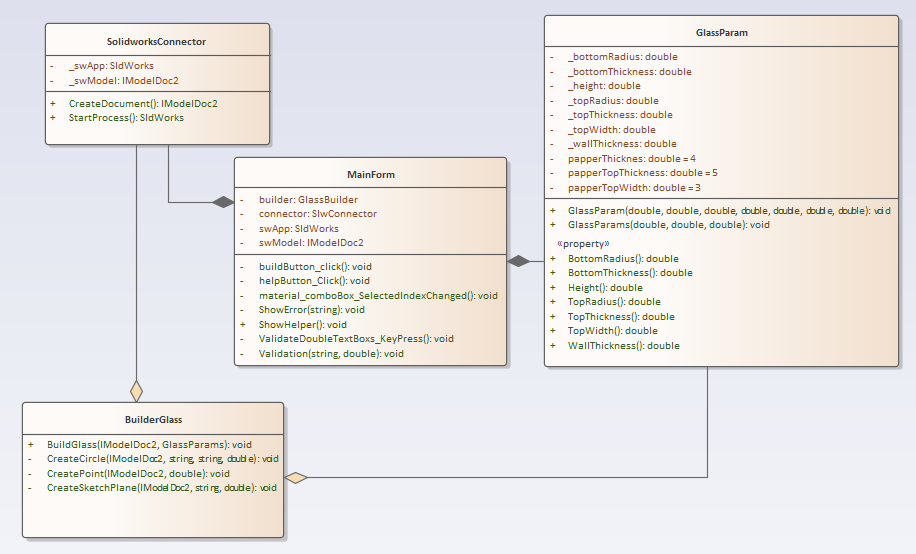


Рисунок 3.2 – Диаграмма классов

Для реализации подсистемы были спроектированы следующие классы:

* SolidworksConnector – класс, отвечающий за работу с Solidworks API;
* Builder – класс, отвечающий за вызов методов Solidworks API, необходимых для постройки объекта проектирования;
* MainForm – класс диалогового окна, отвечающий за взаимодействие между пользователем и программой через форму;
* GlassParam − класс, отвечающий за хранение и валидацию параметров.

# 3.3 Макет пользовательского интерфейса

Пользовательский интерфейс состоит из отдельных элементов и форм, которые собираются в единое целое. Проектирование интерфейса заставляет думать не только о расположении элементов, но и о динамике перехода пользователя от одного подобного элемента к другому таким образом, чтобы это было максимально удобно и эффективно. Это нетривиальная задача, и для её решения необходимо понимать, как именно пользователь будет действовать при работе с программой. [5]

На рисунке 3.3 представлен макет пользовательского интерфейса.

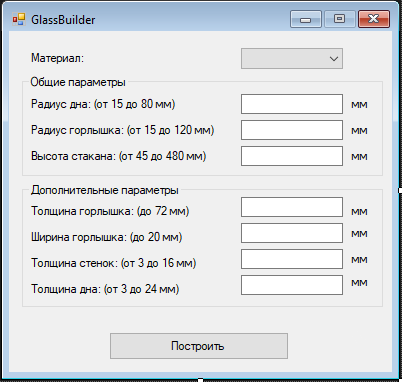


Рисунок 3.3 – Макет пользовательского интерфейса

Работа с интерфейсом разбивается на три части:

1. Выбор материала стакана.
   1. Бумага. В этом случае блок дополнительных параметров скрывается/блокируется.
   2. Стекло. В этом случае для заполнения доступны оба блока.
2. Заполнение доступных параметров стакана.
3. Нажатие кнопки «Построить».

Открытие Solidworks и построение модели происходит автоматически.

Для оповещения пользователя о некорректно введенных данных используется система подсказок, которая сообщает о том, как правильно заполнить данное поле. Кроме того, поле с некорректно введенными данными подсвечивается красным, а также блокируется кнопка «Построить». Пример приведен на рисунке 3.4.

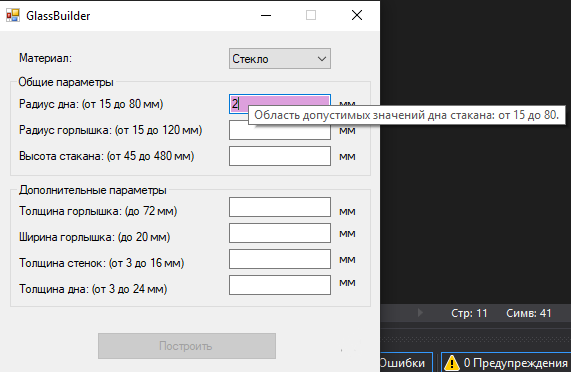


Рисунок 3.4 – Реакция программы на некорректный ввод данных

# Список использованной литературы

1. Solidworks [Электронный ресурс]. – Режим доступа: https://www.solidworks.com/ (дата обращения 10.03.2020)
2. Solidworks API Help [Электронный ресурс]. – Режим доступа: https://help.solidworks.com/2019/English/api/sldworksapiprogguide/Welcome.htm (дата обращения 10.03.2020)
3. Glassful Plugin [Электронный ресурс] – Режим доступа: https://github.com/Sapchanskiy/CADCUP (дата обращения 10.03.2020)
4. Plug-in "Build a Glass" for Compass 3d [Электронный ресурс] – Режим доступа https://github.com/GregoryGhost/plugin-glass-for-compass3d (дата обращения 10.03.2020)
5. Новые технологии в программировании: учебное пособие / А.А. Калентьев, Д. В. Гарайс, А. Е. Горяинов. – Томск, 2014. − 176 стр.\
6. UML. Основы, 3-е издание – Мартин Фаулер, Кендалл Скотт. Издательство Символ Плюс, Москва, 2004 год – 192 с.