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 «ТУМБА ШВАРТОВНАЯ» ДЛЯ «КОМПАС-3D»**

Проект системы по лабораторному проекту

по дисциплине «ОСНОВЫ РАЗРАБОТКИ САПР»

«Построение тумбы швартовной в системе КОМПАС-3D»

Выполнил:

студент гр. 588-1

\_\_\_\_\_\_\_\_\_\_\_\_\_\_Севостьянов К.С.

«\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2021 г.

Руководитель:

к.т.н., доцент каф. КСУП

\_\_\_\_\_\_\_\_\_\_\_\_\_\_Калентьев А. А.

«\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2021 г.

Томск 2022

**1 Описание САПР**

* 1. **Описание программы Компас-3D**

В настоящее время проектирование в своем понимании представляет собой автоматизированный процесс и в некотором роде программно-аппаратный. Проектировщику, который занимается разработкой сложного механизма подходят системы автоматизации проектных решений — САПР [1].

САПР позволяют уменьшить финансовые затраты на разработку макета, а также сократить время, которое тратит проектировщик на создание модели объекта и составление проектной документации.

В каждой крупной САПР есть свой средства для разработки, которые предоставляются с целью дать возможность разработчикам расширить функционал данной системы под свои конкретные нужды. Данным средством является API — программный интерфейс приложения [2]. Это набор готовых средств: классов, процедур, функций, структур, констант и т.д. API позволяет определить функциональность, которую предоставляет приложение, при этом абстрагируясь от того, как она реализована.

Расширение функциональности, в основном, подразумевает разработку плагина или библиотеки на основе предоставленного API. Плагин — независимо компилируемый программный модуль, динамически подключаемый к основной программе, предназначенный для расширения или использования ее возможностей [3].

В качестве системы, которая предоставляет API и для которой стоит задача разработать плагин, была выбрана САПР «КОМПАС-3D» версии 19.

КОМПАС-3D – это российская система трехмерного проектирования, ставшая стандартом для тысяч предприятий и десятков тысяч профессиональных пользователей. Система КОМПАС-3D широко используется для проектирования изделий основного и вспомогательного производств в таких отраслях промышленности, как машиностроение, приборостроение, станкостроение и т. д.

* 1. **Обзор аналогов**

Pinion — Библиотека Зуборезных Долбяков.

Эта библиотека предназначается для тех, кто проектирует элвольвентные зуборезные добляки средних модулей (1-12 мм) (рисунок 1.1). Также она позволяет автоматически создавать графические документы в системе Компас.

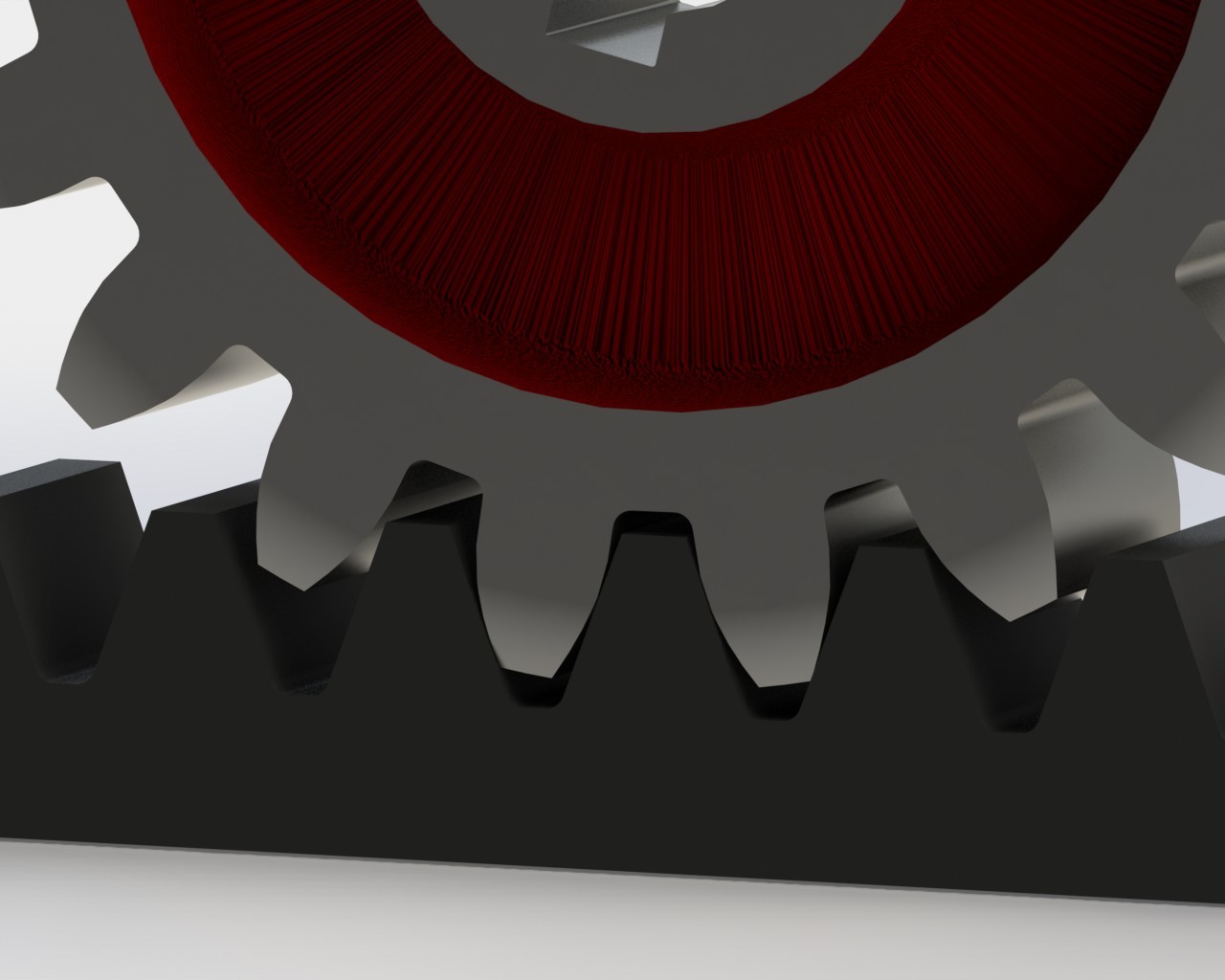


Рисунок 1.1 – Элвольвентные зуборезные добляки

Библиотека помогает в решении следующих задач:

- рассчитывает геометрические параметры долбяка;

- формирует значения показателей точности и технических требований, в соответствии с точностью нарезаемого колеса;

- строит рабочий чертеж долбяка (или изображение) с заданным видом;

- строит 3d-модель долбяка.

Удобно, что полученные графические документы автоматически можно редактировать в обычных редакторах Компас.

Pinion работает в версиях Компас 11 и выше и не предъявляет никаких требований к операционной системе и аппаратному обеспечению вашего ПК, помимо стандартных.[4]

1. **Описание API Компас-3D**

API (англ. Application Programming Interface) – описание способов, которыми одна компьютерная программа может взаимодействовать с другой программой.

В КОМПАС-3D существуют API двух версий: API 5 и API 7. Обе версии реализуют различные функции системы и дополняют друг друга. 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-3D является KompasObject. Получить указатель на этот интерфейс (на интерфейс приложения API 5) 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.

Ниже в таблице 2.1 представлены основные свойства и методы интерфейса KompasObject.

Таблица 2.1 – Методы и свойства интерфейса KompasObject

|  |  |  |  |
| --- | --- | --- | --- |
| Название | Входные параметры | Тип возвращаемых данных | Описание |
| Document3D() |  | ksDocument | Метод для получения указателя на интерфейс трехмерного графического документа (детали или сборки) |

Продолжение таблицы 2.1

|  |  |  |  |
| --- | --- | --- | --- |
| Название | Входные параметры | Тип возвращаемых данных | Описание |
| GetParamStruct(short structType) | structType – тип интерфейса параметров | StructType2D | Метод для получения указателя на интерфейс графического документа (чертежа или фрагмента) |
| Visible |  | bool | Свойство видимости приложения |
| Quit() |  |  | Метод для закрытия активного окна приложения КОМПАС |

В таблице 2.2 представлены методы интерфейса ksEntity, необходимые для разработки плагина.

Таблица 2.2 – Методы интерфейса ksEntity

|  |  |  |
| --- | --- | --- |
| Название | Тип возвращаемых данных | Описание |
| Create() | bool | Создать объект в модели |
| GetDefinition() | IUnkown | Получить указатель на интерфейс параметров объектов и элементов |
| Update() | bool | Изменить свойства объекта (используя ранее установленные свойства) |

В таблице 2.3 представлены свойства и методы интерфейса ksDocument2D, необходимые для разработки плагина

Таблица 2.3 – Методы интерфейса ksDocument2D

|  |  |  |  |
| --- | --- | --- | --- |
| Название | Входные параметры | Тип возвращаемых данных | Описание |
| ksRectangle(  ksRectangleParam param, int style) | param – параметры прямоугольника.  style – стиль линии. | int | Получить указатель на прямоугольник на двумерной плоскости либо 0 в случае ошибки |
| ksCircle(  double xc, double yc, double rad, int style) | xc, yc – координаты центра окружности.  rad – радиус окружности.  style – стиль линии. | int | Получить указатель на окружность на двумерной плоскости либо 0 в случае ошибки |

В таблице 2.4 представлены свойства и методы интерфейса ksDocument3D, необходимые для разработки плагина.

Таблица 2.4 – Методы интерфейса ksDocument3D

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Название | Входные параметры | Тип возвращаемых данных | Описание | |
| Create (bool invisible, bool \_typeDoc) | invisible – признак режима редактирования документа (true – невидимый режим, false – | bool | Создать документ-модель (деталь или сборку) | |
|  | видимый режим),  typeDoc – тип документа (true – деталь, false – сборка). |  | |  |
| GetPart(int type) | type – тип компонента из перечисления Типы компонентов. | ksPart | | Получить указатель на интерфейс компонента в соответствии с заданным типом |

В таблице 2.5 представлены методы интерфейса ksPart, необходимые для разработки плагина.

Таблица 2.5 – Свойства и методы интерфейса ksPart.

|  |  |  |  |
| --- | --- | --- | --- |
| Название | Входные параметры | Тип возвращаемых данных | Описание |
| EntityCollection(short objType) | objType – тип объектов, содержащихся в массиве. | ksEnintyCollection | Формирует массив объектов и возвращает указатель на его интерфейс |
| GetDefaultEntity(short objType) | objType – тип объекта | ksEntity | Получить указатель на интерфейс объекта, создаваемого системой по умолчанию |
| GetPart(int type) | Type – тип компонента | ksPart | Получить указатель на интерфейс компонента в соответствии с заданным типом |
| NewEntity(short objType) | objType – тип объекта | ksEntity | Создать новый интерфейс объекта и получить указатель на него |

В таблице 2.6 представлены типы объектов документа-модели, необходимые для разработки плагина.

Таблица 2.6 – 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 |
| o3d\_face | Грань | ksFaceDefinition |
| o3d\_baseExtrusion | Базовая операция выдавливания | ksBaseExtrusionDefinition |

1. **Описание предмета проектирования**

Швартовые тумбы (или кнехты) – это конструкции из металла (чаще всего из стали или чугуна), представляющие собой парные металлические тумбы, которые отливаются вместе с основанием (плитой). Кнехты предназначены для закрепления тросов при буксировке или швартовке и могут устанавливаться как на пристани, так и на палубе судна – в кормовой и/или носовой части, а также у бортов. Устанавливают тумбы в непосредственной близости от клюзов, обеспечивая их прочную связь с судном и между собой попарно. Расположенные у бортов судна тумбы используются исключительно для швартовки. Чтобы трос не соскальзывал, кнехты имеют специальные шляпки (головки, козырьки, приливы).

Кнехты, устанавливаемые в доках, на пристанях и т.п., также называют тумбами. Изготавливают тумбы в соответствии с требованиями ГОСТ 11265-73 (для стальных кнехтов) и ГОСТ 17424-72 (для чугунных литых тумб различного климатического исполнения из чугуна марки СЧ 18-36, а также из сортов стали и сплавов, которые по технологическим характеристикам и механическим свойствам не уступают чугуну данной марки).[5]

Под желание заказчика тумбы могут изменять параметры, не нарушая ГОСТ 17424-72, приведенные ниже:

1. H1 – Высота козырька: от 110 до 240 мм;
2. H2 – Высота тела тумбы: от 350 до 780 мм;
3. D1 – Диаметр площадки тумбы: от 600 до 1350 мм;
4. D3 – Диаметр отверстия для головки болта: от 72 до 114 мм;
5. D4 – Диаметр отверстия под резьбу болта: от 34 до 76 мм.

Плагин имеет следующие зависимости:

1. Изменяемые величины не должны быть больше требований ГОСТ;
2. Высота H2 не должна быть больше высоты H1;
3. Сумма H1 и H2 не должна быть больше H;
4. Диаметр D4 не должен быть больше D3;
5. Диаметр D2 не должен быть больше D1.

На рисунке 3.1 и 3.2 представлены чертежи с указанными параметрами.

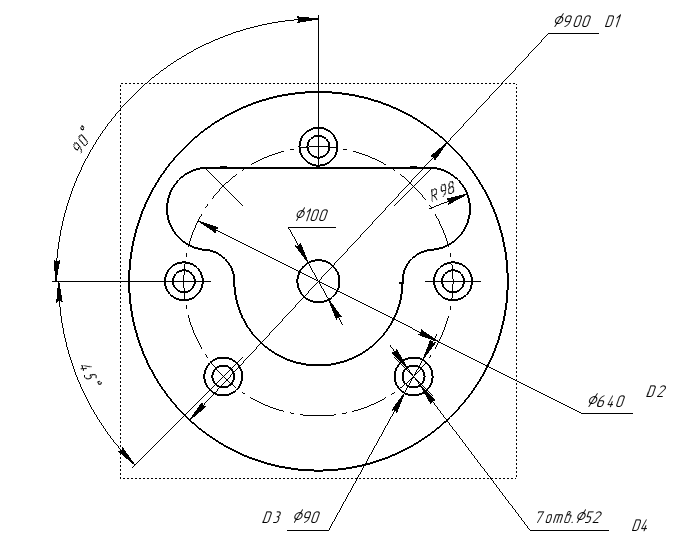


Рисунок 3.1 – Чертеж тумбы, вид сверху

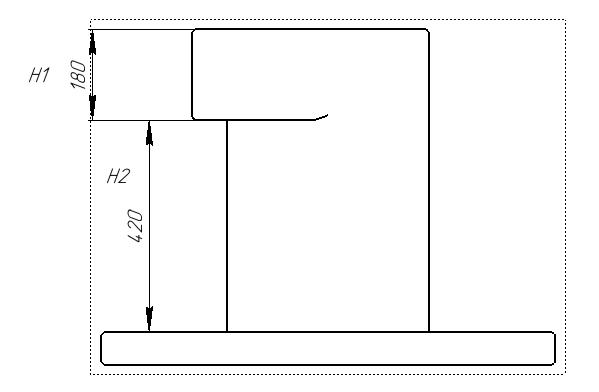


Рисунок 3.2 – Чертеж тумбы, вид сбоку

1. **Проект программы**
   1. **Диаграмма классов**

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

UML 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. UML не является языком программирования, но на основании UML возможна генерация кода и наоборот.

Диаграмма классов – структурная диаграмма языка моделирования UML, демонстрирующая общую структуру иерархии классов системы, их коопераций, атрибутов (полей), методов, интерфейсов и взаимосвязей между ними.[6]

Диаграмма классов плагина представлена на рисунке 4.1.

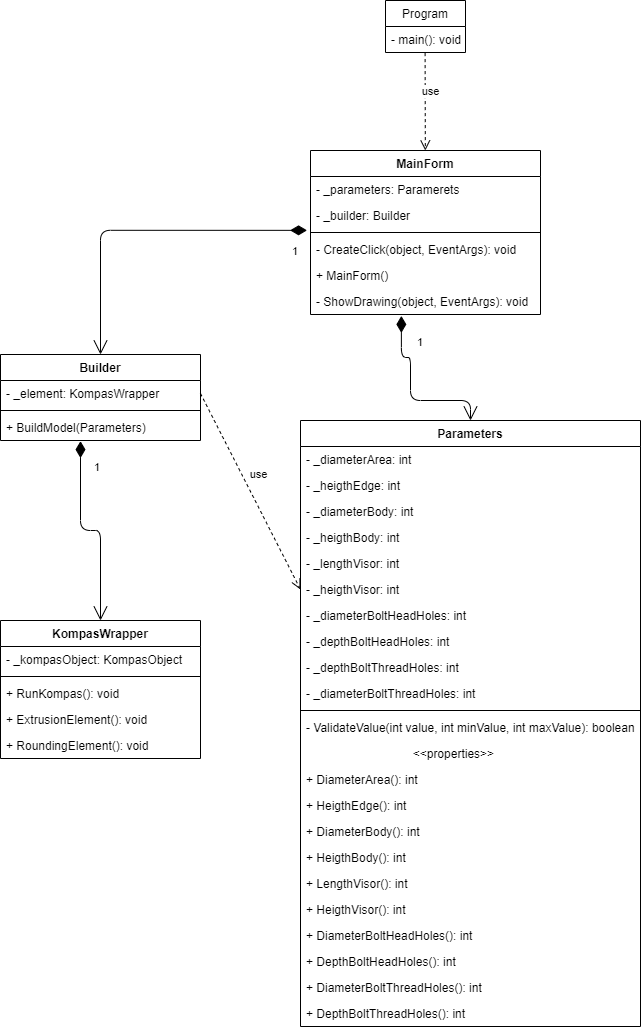


Рисунок 4.1 – Диаграмма классов плагина «Тумба швартовая»

Для реализации был выбран следующий набор классов:

Класс «Program» использует «MainForm» для обработки действий в графическом интерфейсе. «Builder» содержит в себе метод BuildModel() для создания 3D модели в «Компас 3D», которая также подключается к САПР при помощи «KompasWrapper». Класс «Parameters» содержит введенные значения в графическом интерфейсе. При передаче значений свойствам класса «Parameters» в сеттерах при помощи метода «ValidateValue» проверяется правильность диапазона значения. В случае выхода из диапазона вызывается исключение. «KompasWrapper» имеет публичные методы выдавливания (ExtrusionElement()) и скругливания (RoundingElement()) для создания модели в САПР.

* 1. **Макет пользовательского интерфейса**

Макет пользовательского интерфейса представляет собой форму для ввода параметров тумбы швартовой (рисунок 4.1).

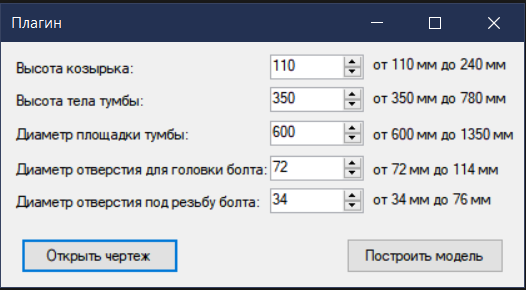


Рисунок 4.1 – Главное окно плагина

Построение модели осуществляется путем нажатия на кнопку «Построить модель». При попытке ввода недопустимых символов, они не будут вводиться в строку (если необходимо ввести цифры, то невозможно будет ввести другие символы). Также при вводе в строку цифр из недопустимого диапазона, который указан справа от каждой области ввода, при попытке построить модель значения будут сбрасываться до минимально или максимально допустимых, а также появится диалоговое окно с уточняющим вопросом (рисунок 4.2).

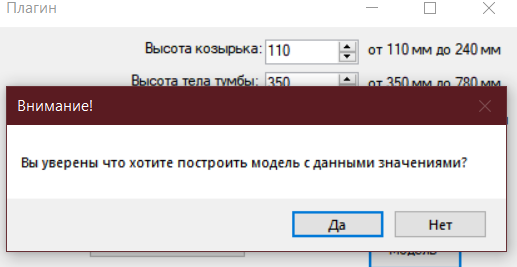


Рисунок 4.2 – Диалоговое окно

При нажатии кнопки «Открыть чертеж» открывается окно с изображением чертежа тумбы (рисунок 4.3).

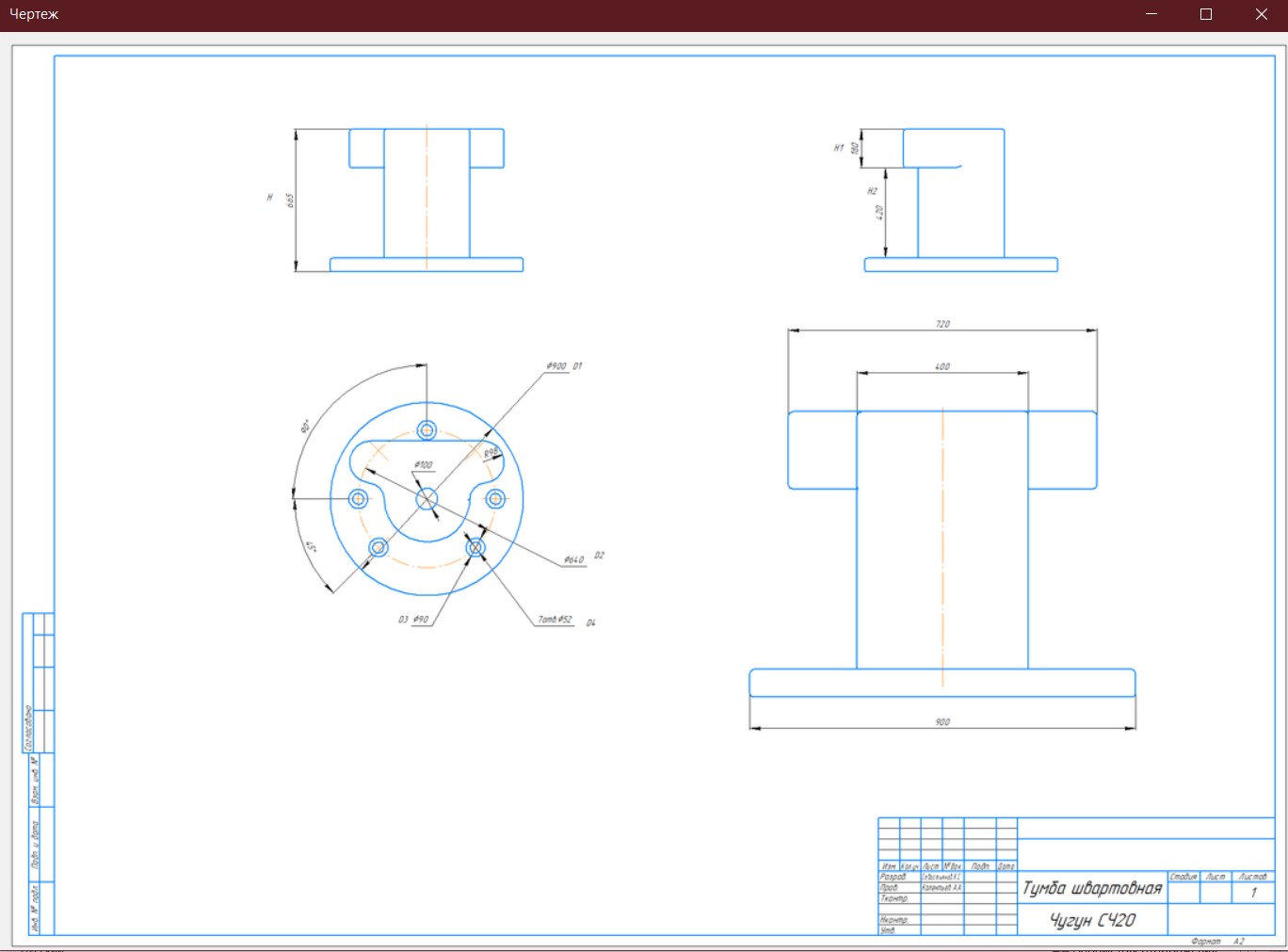


Рисунок 4.3 – Окно с чертежом

1. **Список литературы**

Норенков И.П. «Основы автоматизированного проектирования». Издательство: МГТУ; Москва:, 2002 – 336 с.

API – Википедия. [Электронный ресурс]. — Режим доступа: <https://ru.wikipedia.org/wiki/API> (дата обращения 23.10.2021)

Плагин – Википедия. [Электронный ресурс]. – Режим доступа: <https://ru.wikipedia.org/wiki/Плагин> (дата обращения 25.10.2021)

Модели, чертежи, библиотеки для Компас 3D. [Электронный ресурс]. – Режим доступа: <https://stylingsoft.com/sapr/kompas3d/dopolneniu-kompas-3d> (дата обращения 11.11.2021)

Тумбы швартовые – Описание изделия [Электронный ресурс] – Режим доступа: <http://irontub.ru/tumby-shvartovye> (дата обращения 01.11.2021)

UML. [Электронный ресурс]. – Режим доступа: http://www.uml.org/ (дата обращения 9.11.2021)