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 "ЗАБОР" ДЛЯ САПР INVENTOR

Проект системы по лабораторному проекту

по дисциплине «ОСНОВЫ РАЗРАБОТКИ САПР»

Выполнил:

студент гр. 588-2

\_\_\_\_\_\_\_\_\_\_\_\_\_\_В.С. Швоев

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

Руководитель:

к.т.н., доцент каф. КСУП

\_\_\_\_\_\_\_\_\_\_\_\_\_\_А.А. Калентьев

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

# 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 [1].

Autodesk Inventor — система трёхмерного 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).[2]

# Описание API

API (англ. Application Programming Interface) – описание способов, которыми одна компьютерная программа может взаимодействовать с другой программой.[3]

Для Inventor существует API под названием Inventor. Для построения объекта в Inventor через API будут использоваться следующие классы: Application (таблица 1.1), TransientGeometry (таблица 1.2), PartComponentDefinition (таблица 1.3), PlanarSketches (таблица 1.4).

Таблица 1.1 — Используемые методы класса Application

|  |  |  |  |
| --- | --- | --- | --- |
| Название | Входные параметры | Тип возвращаемых данных | Описание |
| Documents |  | Documents | Свойство, содержащие все открытые документы в Inventor |
| FileManager |  | FileManager | Свойство, позволяющее работать с файлами Inventor |
| TransientGeometry |  | TransientGeometry | Свойство, содержащие методы геометрии |

Таблица 1.2 — Используемые методы класса TransientGeometry

|  |  |  |  |
| --- | --- | --- | --- |
| Название | Входные параметры | Тип возвращаемых данных | Описание |
| CreatePoint2d | int X, int Y | Point2d | Создает точку на 2D эскизе |

Таблица 1.3 — Используемые методы класса PartComponentDefinition

|  |  |  |  |
| --- | --- | --- | --- |
| Название | Входные параметры | Тип возвращаемых данных | Описание |
| WorkPlanes |  | WorkPlanes | Свойство, позволяющее работать с плоскостью |
| Sketches |  | PlanarSketches | Свойство, позволяющее работать с планарным (плоским, поверхностным) эскизом |
| WorkAxes |  | WorkAxes | Свойство, позволяющее работать с осями |
| Features |  | PartFeatures | Свойство, позволяющее работать с деталями |

Таблица 1.4 — Используемые методы класса PlanarSketches

|  |  |  |  |
| --- | --- | --- | --- |
| Название | Входные параметры | Тип возвращаемых данных | Описание |
| SketchLines |  | SketchLines | Свойство, позволяющее работать с линями на эскизе |
| Profiles |  | Profiles | Свойство, позволяющее работать с профилем эскиза |

Таблица 1.5 — Используемые методы класса ExtrudeDefinition

|  |  |  |  |
| --- | --- | --- | --- |
| Название | Входные параметры | Тип возвращаемых данных | Описание |
| SetDistanceExtent | double width, PartFeatureExtentDirectionEnum (вариант выдавливания симметричный, ассиметричный и т.д.) |  | Метод выдавливания |

# Обзор аналогов

**Плагин Instant Fence and Railing для Sketchup**

SketchUp — программа для 3D дизайна и архитектурного проектирования. В основном используется для моделирования жилых домов, мебели, интерьера. Есть инструменты для проектирования лестниц, электропроводки, санитарно-технических коммуникаций и оборудования.[4]

Данный плагин содержит более трехсот предустановленных стилей, разделенных на библиотеки. Пользователи могут создавать собственные стили и сохранять изображения и значки. Применяя этот плагин, пользователи могут моделировать все заборы или перила в 3D, 2D линиях расположенных с основной стороны или 2d скрытых линиях.

Плагин генерирует множество моделей заборов, балконных перил, перил, перил. Пикеты, перила и ограждения могут быть расположены по центру или с выбранной стороны.[5]

Интерфейс плагина показан на рисунке 1.1.



Рисунок 1.1 — Интерфейс плагина Instant Fence and Railing

# 2 Описание предмета проектирования

Забор (также городьба, изгородь, огорожа, ограда, палисад, плетень, тын, частокол) — сооружение, которое охватывает территорию, как правило, на открытом воздухе, и обычно состоящее из столбов и перекрытий из различных материалов, и служащее для ограждения (защиты) и обрамления (обозначения границы) той или иной территории.[6]

На рисунке 2.1 представлен чертеж забора.



Рисунок 2.1 – Чертеж забора

Параметры забора:

1. длина забора ***L*** (1 — 3м);
2. общая высота забора ***H*** (1.5 — 2м);
3. глубина погружения столба ***h1*** (1/3 — 1/2 от общей высоты забора);
4. высота верхней части забора ***h2***(1/2 — 2/3 от общей высоты забора);
5. ширина столбика ***W*** (10-70мм);
6. расстояние между нижними перегородками ***g1*** (не больше длины забора и не меньше ширины одного столбика. Количество столбиков будет определяться автоматически исходя из данного параметра и длины забора);
7. расстояние между верхними перегородками ***g2*** (не больше длины забора и не меньше ширины одного столбика. Количество столбиков будет определяться автоматически исходя из данного параметра и длины забора).

# 3 Проект программы

# 3.1 Описание технических и функциональных аспектов проекта

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.[7]

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

# 3.2 Диаграмма классов

Диаграмма классов – структурная диаграмма языка моделирования UML, демонстрирующая общую структуру иерархии классов системы, их коопераций, атрибутов (полей), методов, интерфейсов и взаимосвязей между ними.[7]

На рисунке 3.2 представлена диаграмма классов.

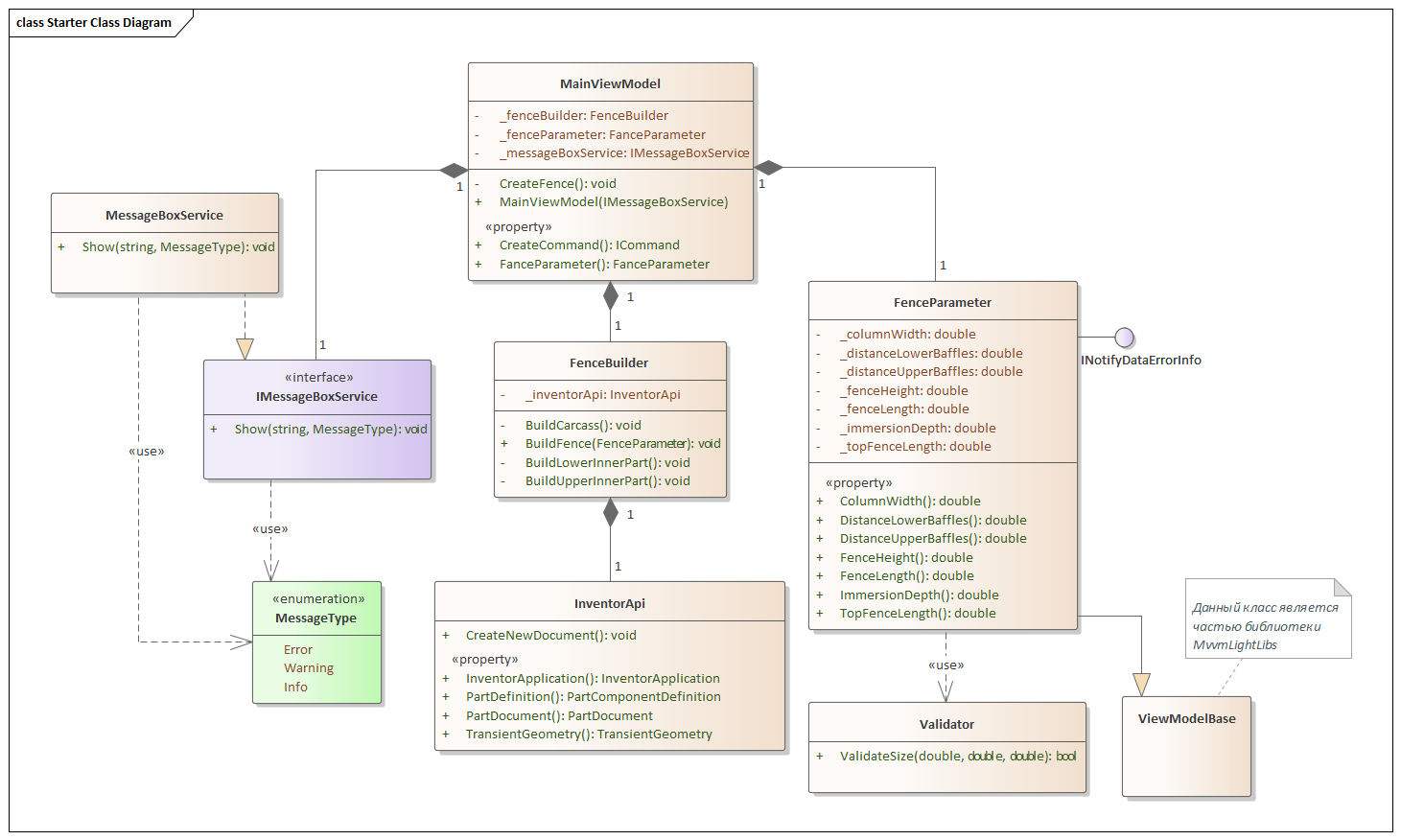


Рисунок 3.1 – Диаграмма классов

Главное окно связывается с MainViewModel для управления данными. MainViewModel содержит экземпляры следующие экземпляры классов:

* FenceParameter — содержит параметры забора, которые валидируются с помощью класса Validator;
* FenceBuilder — класс создания забора в Inventor, хранящий класс InventorApi, который содержит нужные экземпляры класса из всей Inventor API;
* IMessageBoxService — сервис для использования уведомляющих окон, реализация сервиса находится в классе MessageBoxService. Метод Show принимает в качестве одного из аргументов элемент перечисления MessageType, который отвечает за тип уведомляющего окна.

Будет использоваться дополнительная библиотека **MvvmLightLib** для более удобного использования паттерна MVVM. В данной библиотеке уже есть реализация таких интерфейсов, как:

* *INotifyPropertyChanged* — сообщает клиенту об изменении значения свойства;
* *ICommand* — определяет комманду.

Также будет использоваться объект класса *ServiceProvider* из пространства имен *Microsoft.Extensions.DependencyInjection* для более простого и удобного внедрения сервисов.

# 3.3 Макет пользовательского интерфейса

Макет пользовательского интерфейса представляет собой форму для ввода параметров забора. Построение модели осуществляется путем нажатия на кнопку «Построить». При попытке ввода недопустимых символов, они не будут вводиться в строку (если необходимо ввести цифры, то невозможно будет ввести другие символы).

На рисунке 3.3 представлен макет пользовательского интерфейса.

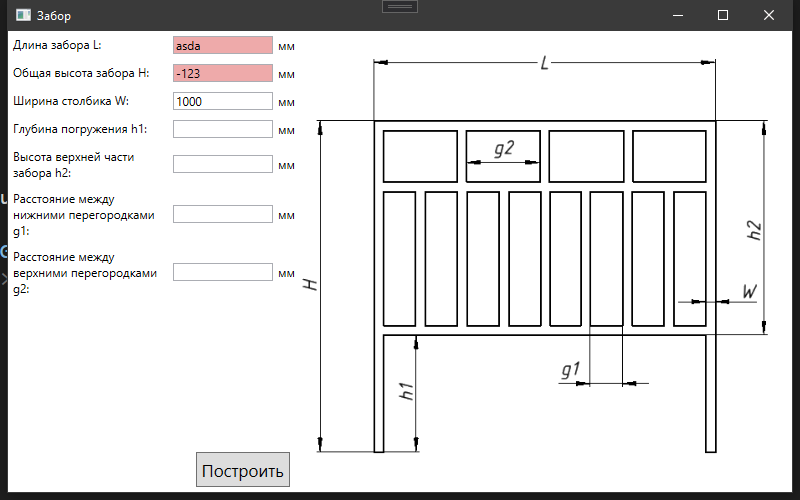


Рисунок 3.2 – Макет пользовательского интерфейса

# Список литературы

1. САПР — Википедия. [Электронный ресурс]. — Режим доступа: https://ru.wikipedia.org/wiki/Система\_автоматизированного\_проектирования (дата обращения 22.10.2021).

2. Autodesk Inventor— Википедия. [Электронный ресурс]. — Режим доступа: https://ru.wikipedia.org/wiki/Autodesk\_Inventor (дата обращения 22.10.2021).

3. API — Википедия. [Электронный ресурс]. — Режим доступа: https://ru.wikipedia.org/wiki/API (дата обращения 22.10.2021).

4. Sketchup — Википедия. [Электронный ресурс]. — Режим доступа: https://ru.wikipedia.org/wiki/SketchUp (дата обращения 22.10.2021).

5. Плагин Instant Fence and Railing — Sketchup3DConstruction. [Электронный ресурс]. — Режим доступа: sketchup3dconstruction.com/skp/instant-fenceand-railing-plugin-for-sketchup.htm (дата обращения 22.10.2021).

6. Забор — Википедия. [Электронный ресурс]. — Режим доступа: https://ru.wikipedia.org/wiki/%D0%97%D0%B0%D0%B1%D0%BE%D1%80 (дата обращения 22.10.2021).

7. UML. [Электронный ресурс]. – Режим доступа: http://www.uml.org/ (дата обращения 22.10.2021).