Министерство науки и высшего образования Российской Федерации Федеральное государственное автономное образовательное учреждение высшего образования

ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ СИСТЕМ

УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ (ТУСУР)

Кафедра компьютерных систем в управлении и проектировании (КСУП)

**РАЗРАБОТКА ПЛАГИНА «МОНИТОР» ДЛЯ «КОМПАС-3D»**

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

по дисциплине

«Основы разработки САПР» (ОРСАПР)

Выполнил:

студент гр. 581

\_\_\_\_\_\_\_\_\_ Грахович В.В.

«\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2024 г.

Проверил:

к.т.н., доцент каф. КСУП \_\_\_\_\_\_\_\_\_\_ Калентьев А.А.

«\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2024

**РЕФЕРАТ**

Пояснительная записка, 33 страницы, 20 рисунков, 12 таблиц, 11 источников.

Ключевые слова: САПР, КОМПАС-3D, Плагин для САПР, Плагин монитора, C#, Windows Forms.

Объектом исследования являются технологии разработки плагинов для САПР.

Предметом исследования является применение технологий разработки плагинов, для автоматизации построения мониторов с разными параметрами в САПР КОМПАС-3D.

Цель работы: создание программы для автоматизации построения монитора в САПР КОМПАС-3D.

Для создания использовались Microsoft Visual Studio 2022 (Windows Forms), .NET Framework 4.7.2, NUnit 3.14.0, NUnit3TestAdapter 3.17.0, ReSharper, Fine Code Coverage, GitHub.

В результате работы было создано приложение Windows Forms, взаимодействующее с САПР КОМПАС-3D.

Областью применения являются предприятия, связанные с моделированием мониторов.

**СОДЕРЖАНИЕ**

[1 ВВЕДЕНИЕ 4](#_Toc23763)

[2 ПОСТАНОВКА И АНАЛИЗ ЗАДАЧИ 5](#_Toc22434)

[3 ОПИСАНИЕ ПРЕДМЕТА ПРОЕКТИРОВАНИЯ 7](#_Toc31758)

[4 ВЫБОР ИНСТРУМЕНТОВ И СРЕДСТВ РЕАЛИЗАЦИИ 9](#_Toc363)

[5 НАЗНАЧЕНИЕ ПЛАГИНА 10](#_Toc17893)

[6 ОБЗОР АНАЛОГОВ 10](#_Toc13689)

[7 ОПИСАНИЕ РЕАЛИЗАЦИИ 12](#_Toc16292)

[8 ОПИСАНИЕ ПРОГРАММЫ ДЛЯ ПОЛЬЗОВАТЕЛЕЙ 19](#_Toc9887)

[9 ТЕСТИРОВАНИЕ ПЛАГИНА 23](#_Toc28841)

[9.1 Функциональное тестирование 23](#_Toc10064)

[9.2 Модульное тестирование 28](#_Toc20472)

[9.3 Нагрузочное тестирование 31](#_Toc3928)

[10 ЗАКЛЮЧЕНИЕ 34](#_Toc1958)

[11 СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 35](#_Toc5152)

**1 ВВЕДЕНИЕ**

САПР – организационно-техническая система, входящая в структуру проектной организации и осуществляющая проектирование при помощи комплекса средств автоматизированного проектирования[1].

API (Application Programming Interface) – набор правил и протоколов, с помощью которых различные программные приложения могут взаимодействовать друг с другом и обмениваться данными, повышая тем самым функциональность и эффективность работы[2].

КОМПАС-3Д – российская система автоматизированного проектирования (САПР), разработанная компанией «Аскон»[3].

Для разработки плагина для САПР прежде всего необходимо выбрать объект проектирования, подходящую для выбранного объекта САПР, средства разработки плагина (язык программирования и дополнительные средства разработки, выбор может быть основан на наличии API для выбранной САПР на конкретном языке).

Плагин автоматизации построения монитора необходим и может быть использован на предприятиях, занимающихся моделированием мониторов, поскольку он упростит процесс моделирования.

**2 ПОСТАНОВКА И АНАЛИЗ ЗАДАЧИ**

Этапы проведения работ по разработке плагина «Монитор» для САПР «Компас 3D» приведены в таблице 2.1.

Таблица 2.1 − Этапы проведения работ по разработке плагина «Монитор» для САПР «Компас 3D».

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Этап | Состав работ | Наименование документа | Обозначение | Разработано согласно | Сроки выполнения |
| 1 | Создание технического задания | Техническое задание | − | ГОСТ 34.602-2020 | Не позднее 8.10.2024 |
| 2 | Создание проекта системы | Проект системы | − | ОС ТУСУР 01-2021 | Не позднее 29.10.2024 |
| 3 | Реализация плагина | Программный код | − | RSDN Magazine #1-2004 | Не позднее 10.12.2024 |
| Документ с тремя вариантами дополнительной функциональности плагина для согласования |
| Модульные тесты |

Таблица 2.1 − Продолжение

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Этап | Состав работ | Наименование документа | Обозначение | Разработано согласно | Сроки выполнения |
| 4 | 1. Доработка плагина 2. Создание пояснительной записки | Программный код | − | 1. RSDN Magazine #1-2004 2. ОС ТУСУР 01-2021 | Не позднее 31.12.2024 |
| Модульные тесты |
| Пояснительная записка |

В процессе работы над плагином были внесены некоторые изменения, которые повлекли за собой изменения конечной версии UML-диаграммы классов. Связано это с недостаточными знаниями API Kompas-3D на этапе проектирования. Это значит, что в дальнейшем необходимо более детально подходить к изучению средств разработки и API еще на этапе создания проекта системы.

# 3 ОПИСАНИЕ ПРЕДМЕТА ПРОЕКТИРОВАНИЯ

Монитор – главное периферийное устройство для ЭВМ, предназначенная для вывода изображения, графической информации [4].



Рисунок 2.1 − Модель отвёртки

***Изменяемые параметры для плагина*** (также все обозначения показаны на рисунке 2.1):

**Ширина экрана L** (400–900 мм);

**Высота экрана H**:

* Для **4:3** экранов высота составляет 75% от ширины L, то есть H=0.75⋅L;
* Для **16:10** экранов высота будет 62.5% от ширины L, то есть H=0.625⋅L;
* Для **16:9** экранов высота составляет 56% от ширины L, то есть H=0.56⋅L;
* Для **21:9** экранов высота будет 43% от ширины L, то есть H=0.43⋅L.

**Высота рамки b** (5–30 мм);

**Толщина рамки w** (5–20 мм);

**Глубина экрана g** (0–10 мм; если g=0, то экран и рамка находятся на одной плоскости);

**Высота стойки p** (50–200 мм);

**Ширина стойки m** (30–100 мм);

**Толщина стойки f** (10–50 мм);

**Высота подставки s** (50–150 мм);

**Ширина подставки D** (150–400 мм);

**Толщина подставки z** (10–50 мм);

**Длина рычага j** (0–150мм).

Также в ходе работы были добавлены параметры высоты (20-200мм) и ширины (20-200мм) рычага.

**4 ВЫБОР ИНСТРУМЕНТОВ И СРЕДСТВ РЕАЛИЗАЦИИ**

При создании плагина использовались следующие инструменты:

− WindowsForms и .NET Framework 4.7.2;

− GitHub;

− ReSharper;

− Fine Code Coverage;

− UnitTestProject

Плагин был создан на технологии Windows Forms, поддерживающей широкий набор функций для разработки приложений, включая элементы управления, графику, привязку данных и ввод пользователя[5], а также .NET Framework 4.7.2, программной платформе основанной на сервероцентрической модели.

GitHub − платформа с возможностями хранения, распространения и совместной работы над написанием кода. Git − система управления версиями, которая интеллектуально отслеживает изменения в файлах.[6]

ReSharper — это интегрированное средство разработки (IDE) и плагин для Visual Studio, разработанный компанией JetBrains.[7]

Fine Code Coverage − расширение для Microsoft Visual Studio, визуализирующий покрытие кода модульными тестами.[8]

NUnit − фреймфорк для модульного тестирования всех языков .Net.[9]

# 5 НАЗНАЧЕНИЕ ПЛАГИНА

Назначение разрабатываемого плагина обусловлено быстрым моделированием мониторов разных размеров. Используя данный плагин, можно быстро построить монитор по заданным параметрам экрана, корпуса, а также подставки.

**6 ОБЗОР АНАЛОГОВ**

Косвенным аналогом разрабатываемого плагина является **AutoCAD Plant 3D Toolset** [10]. Это специализированный инструмент для проектирования промышленных объектов, который включает инструменты для параметрического проектирования. В то время как он ориентирован на создание инженерных объектов, его функционал позволяет настроить параметрическое проектирование для создания различных объектов по заданным размерам и параметрам, включая экраны или другие объекты с заданными пропорциями. Стоимость **AutoCAD Plant 3D Toolset** включает подписку на AutoCAD с расширенным функционалом, и его цена начинается примерно с **3000 долларов в год.** Интерфейс взаимодействия представлен на рисунке 1.1.



Рисунок 6.1 − Интерфейс приложения «Разъёмные соединения»

# 7 ОПИСАНИЕ РЕАЛИЗАЦИИ

UML — это стандартный язык визуального моделирования, предназначенный для следующего использования:

− моделирование бизнеса и подобных процессов;

− анализ, проектирование и внедрения программных систем.

UML — это общий язык для бизнес-аналитиков, архитекторов и разработчиков программного обеспечения, используемый для описания, спецификации, проектирования и документирования существующих или новых бизнес-процессов, структуры и поведения артефактов программных систем[11]. На рисунках 7.1 и 7.2 представлены UML диаграмма классов для плагина «Монитор» до и после реализации.

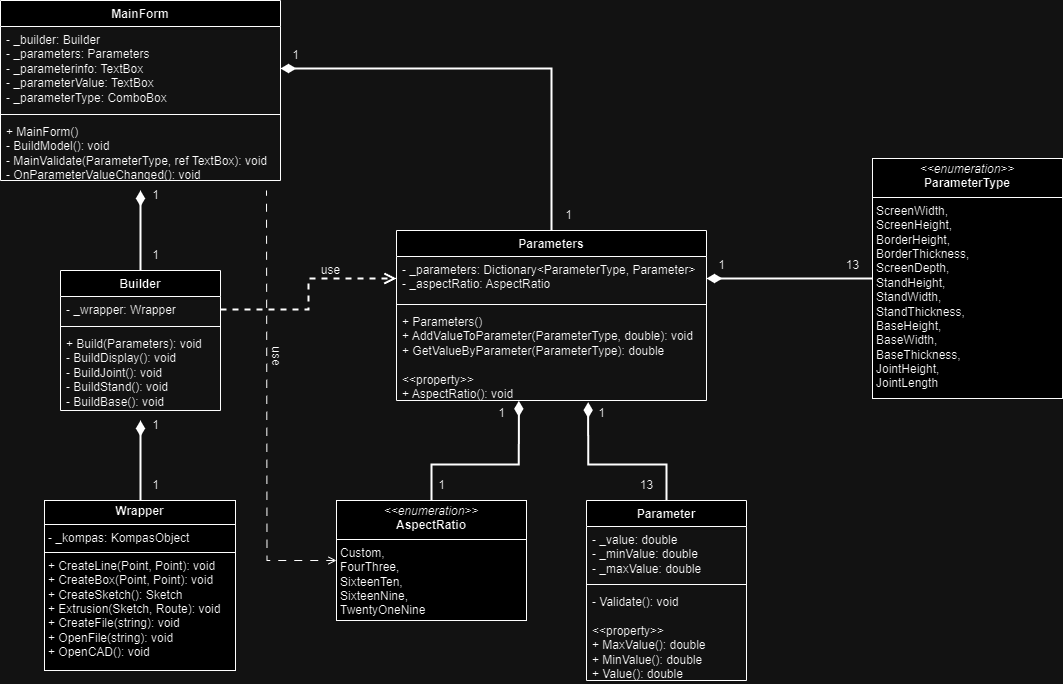


Рисунок 7.1 − UML диаграмма классов до реализации плагина «Монитор»

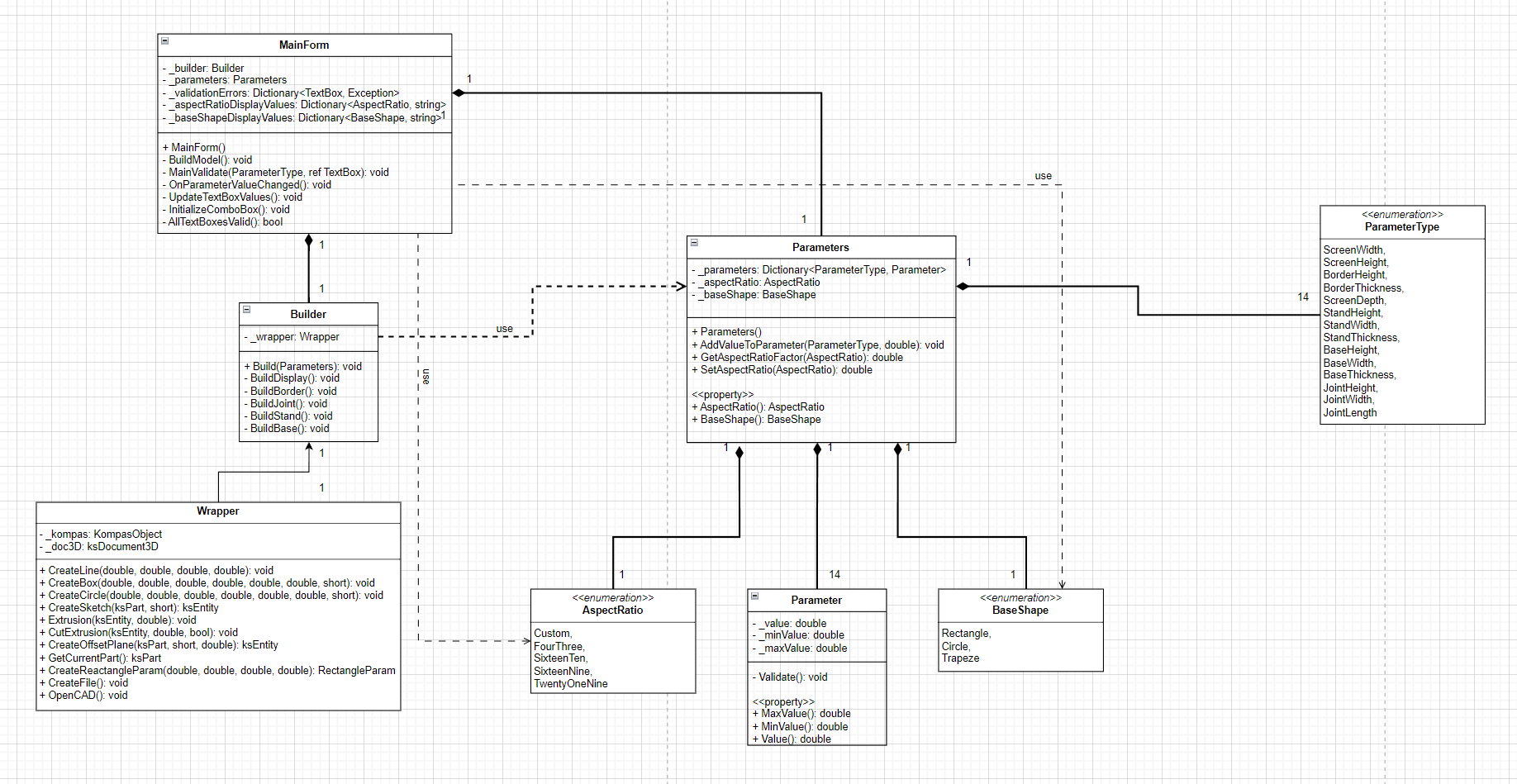


Рисунок 7.2 − UML диаграмма классов после реализации плагина «Монитор»

В таблицах ниже представлена информация о свойствах и методах каждого из классов.

Таблица 7.1 − Свойства класса MainForm

|  |  |  |
| --- | --- | --- |
| Название | Тип данных | Описание |
| \_builder | Builder | Хранит в себе объект построения |
| \_parameters | Parameters | Хранит в себе параметры для объекта построения |
| \_validationErrors | Dictionary<TextBox, Exception> | Хранит список ошибок валидации |
| \_aspectRatioDisplayValues | Dictionary<AspectRatio, string> | Хранит отображаемые значения для comboBoxRatio |
| \_baseShapeDisplayValues | Dictionary<BaseShape, string> | Хранит отображаемые значения для comboBoxShape |

Таблица 7.2 − Методы класса MainForm

|  |  |  |  |
| --- | --- | --- | --- |
| Название | Входные параметры | Выходные параметры | Описание |
| MainForm | − | − | Конструктор MainForm |
| BuildModel | − | − | Строит модель монитора с текущими параметрами |
| BuildButton\_Click | object sender, EventArgs e | − | Обработчик нажатия кнопки "Построить модель" |
| MainValidate | ParameterType, textBoxTemp | − | Проверка введённых данных по формату |
| textBox\_Leave | object sender, EventArgs e | − | Обработчик выхода из текстбоксов |
| comboBoxRatio\_SelectedIndexChanged | object sender, EventArgs e | − | Обработчик изменения значения выпадающего списка для выбора соотношения сторон экрана |
| comboBoxShape\_SelectedIndexChanged | object sender, EventArgs e | − | Обработчик изменения значения выпадающего списка для выбора формы подставки |
| UpdateTextBoxValues | − | − | Обновляет визуальное представление textBox’ов |
| InitializeComboBox | − | − | Обновляет содержимое выпадающих списков |
| AreAllTextBoxesValid | − | bool | Проверяет наличие ошибок в словаре с ошибками |

Таблица 7.3 − Свойства класса Parameters

|  |  |  |
| --- | --- | --- |
| Название | Тип данных | Описание |
| \_parametersDict | Dictionary<ParameterType, Parameter> | Словарь с параметрами |
| \_aspectRatio | AspectRatio | Перечисление с соотношениями сторон |
| \_baseShape | BaseShape | Перечисление с формами подставки |

Таблица 7.4 − Методы класса Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Название | Входные параметры | Выходные параметры | Описание |
| ParametersDict | − | Dictionary<ParameterType, Parameter> | Свойство для \_parametersDict |
| AddValueToParameter | ParameterType, double | − | Устанавливает значение для параметра |
| SetAspectRatio | AspectRatio | − | Задать соотношение сторон |
| GetAspectRatioFactor | AspectRatio | double | Получает коэффициент соотношения сторон |
| BaseShape | BaseShape | BaseShape | Свойство для \_baseShape |

Таблица 7.5 − Свойства класса Builder

|  |  |  |
| --- | --- | --- |
| Название | Тип данных | Описание |
| \_wrapper | Wrapper | Хранит в себе объект обёртки API |

Таблица 7.6 − Методы класса Builder

|  |  |  |
| --- | --- | --- |
| Название | Входные параметры | Описание |
| Build | Parameters | Построение модели по заданным параметрам |
| BuildDisplay | Parameters | Построение экрана |
| BuildBorder | Parameters | Выдавливание рамки |
| BuildJoint | Parameters | Построение крепления |
| BuildStand | Parameters | Построение стойки |
| BuildBase | Parameters | Построение подставки |

Таблица 7.7 − Свойства класса Parameter

|  |  |  |
| --- | --- | --- |
| Название | Тип данных | Описание |
| \_maxValue | int | Максимальное значение параметра |
| \_minValue | int | Минимальное значение параметра |
| \_value | int | Значение параметра |

Таблица 7.8 − Методы класса Parameter

|  |  |  |
| --- | --- | --- |
| Название | Выходные параметры | Описание |
| Value | double | Свойство для поля \_value |
| MaxValue | double | Свойство для поля \_maxValue |
| MinValue | double | Свойство для поля \_minValue |
| Validate | − | Сравнивает полученное значение с максимальным и минимальным возможными |

Таблица 7.9 − Свойства класса Wrapper

|  |  |  |
| --- | --- | --- |
| Название | Тип данных | Описание |
| \_kompas | KompasObject | Поле, хранящее в себе экземпляр программы Компас |
| \_doc3D | ksDocument | Поле, хранящее в себе документ |

Таблица 7.10 − Методы класса Wrapper

|  |  |  |  |
| --- | --- | --- | --- |
| Название | Входные параметры | Выходные параметры | Описание |
| CreateLine | double, double, double, double | − | Создание линии на плоскости XOZ по координатам |
| CreateBox | double, double, double, double, double, double, short | − | Создание цилиндра на плоскости short planeType |
| CreateCircle | double, double, double, double, double, short | − | Создание эскиза (по int выбираем базисную плоскость) |
| CreateSketch | ksPart, short | ksEntity | Создает эскиз на указанной плоскости |
| CutExtrusion | ksEntity, double, bool | − | Создание выреза по эскизу |
| Extrusion | ksEntity, double | − | Выдавливание эскиза на глубину double |
| GetCurrentPart | − | ksPart | Получить деталь из документа \_doc3D |
| CreateRectangleParam | double, double, double, double | RectangleParam | Создание параметров прямоугольника |
| CreateFile | − | − | Создание файла |
| OpenCAD | − | − | Открытие Компас3D |

В отличии от диаграммы классов проекта системы диаграмма классов после реализации плагина имеет следующие отличия:

­− Добавлен перечислимый тип BaseShape, включающий формы Rectangle, Circle, Trapeze. Теперь система поддерживает разные формы базы или других элементов.

− В MainForm добавлены поля \_baseShapeDisplayValues (словарь с отображением форм базы) и метод InitBaseComboBox(), который инициализирует выпадающий список для выбора формы.

− В Parameters появились методы SetBaseShape и GetBaseShape, которые позволяют задавать и получать текущую форму базы.

− Во Wrapper добавлены методы CreateCircle и CreateTrapeze. Теперь можно строить круги и трапеции, в отличие от старой версии, где был только метод CreateBox для прямоугольников.

− В Builder изменений не произошло, структура осталась прежней.

− Перечисление ParameterType, вероятно, расширено для работы с дополнительными параметрами, если таковые появились.

**8 ОПИСАНИЕ ПРОГРАММЫ ДЛЯ ПОЛЬЗОВАТЕЛЕЙ**

При запуске приложения открывается форма для заполнения параметров объекта (рисунок 8.1), а также приложение KOMPAS-3D.

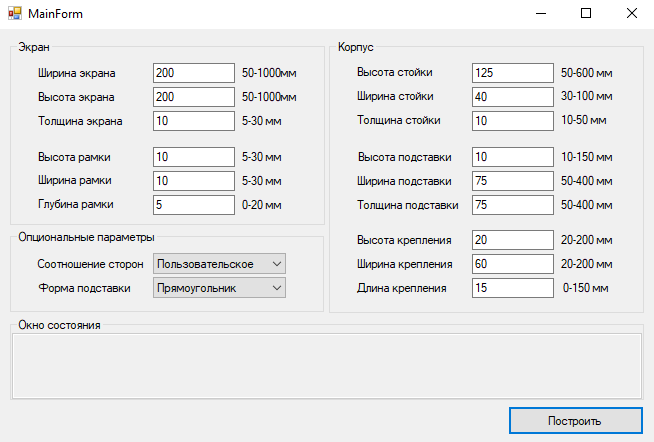


Рисунок 8.1 – Начальная форма в момент запуска приложения

Сразу после запуска программы, выставляются стандартные допустимые параметры для построения.

Пользователь может настраивать размеры и параметры всех основных компонентов монитора, таких как:

− Экран: ширина, высота, толщина, рамка по высоте и ширине, глубина.

− Корпус: параметры стойки (высота, ширина, толщина), подставки (высота, ширина, толщина) и крепления (высота, ширина, длина).

− Дополнительные параметры: выбор соотношения сторон экрана (например, 16:9, 4:3 и другие) и форма подставки.

При выставлении параметров, выходящих за установленный диапазон, окно с текстом будет выделено красным, а в окно состояния внизу формы выведется описание ошибки (рисунок 8.2).

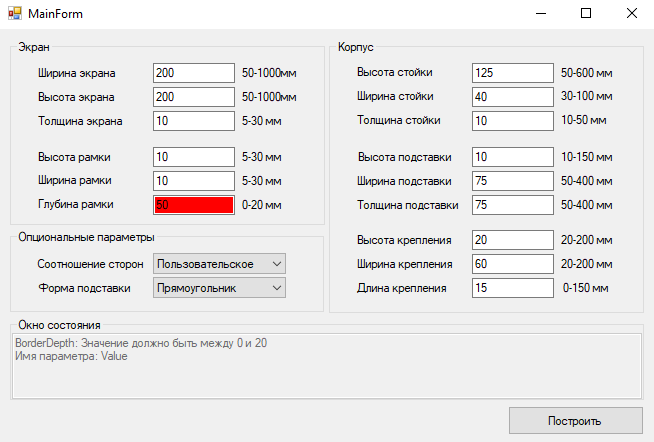


Рисунок 8.2 – Глубина рамки выходит за минимальные пределы (больше 20)

Если пользователь выбирает стандартное соотношение сторон (например, 16:9, 4:3 и т.д.) в выпадающем списке, то:

− Ширина или высота экрана автоматически пересчитываются в зависимости от текущих значений и выбранного соотношения сторон.

− Например, при ширине экрана 300 мм и выборе соотношения сторон 16:9, высота будет рассчитана как 168 мм (рисунок 8.3).

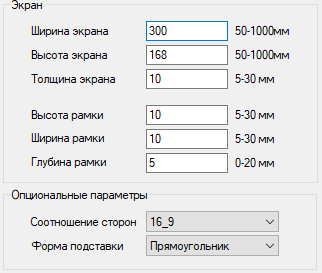


Рисунок 8.3 – Изменение соотношения сторон повлекло изменение высоты

Если пользователь вручную изменяет ширину или высоту экрана в соответствующем текстовом поле:

− Программа пересчитывает другой параметр (ширину или высоту) с учетом выбранного соотношения сторон.

− Например, если при соотношении 4:3 пользователь вводит высоту 357 мм, ширина будет автоматически рассчитана как 476 мм (рисунок 8.4).

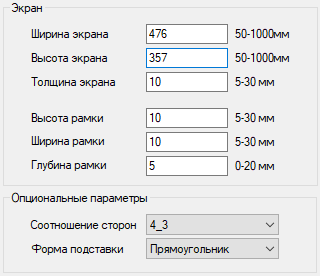


Рисунок 8.4 – При изменении высоты экрана произошел пересчет ширины в соответствии с заданным соотношением сторон

**9 ТЕСТИРОВАНИЕ ПЛАГИНА**

## 9.1 Функциональное тестирование

Далее демонстрируется обработка ошибок в связанных параметрах. На рисунке 9.1 представлен результат ввода значения 725 в параметр высоты экрана. При соотношении сторон 16 на 10, параметр ширины экрана корректируется с коэффициентом 1.6, из-за чего полученное значение выход за диапазон 50-1000мм.

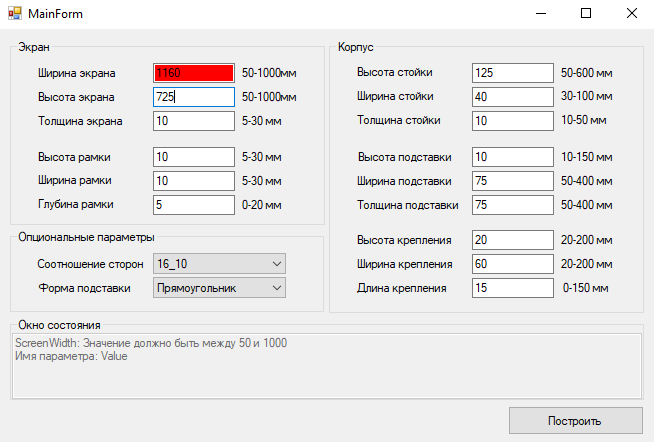


Рисунок 9.1 – Ошибка валидации зависимого параметра ширины экрана

На рисунке 9.2 – результат выставления параметра соотношения сторон на «21\_9». Для данного соотношения коэффициент высоты экрана будет равен 0.625. Это значит, что при ширине экрана, равной 100мм, механизм валидации зависимых параметров выставит значение высоты экрана на 43мм, что также выходит за диапазон 50-1000мм.

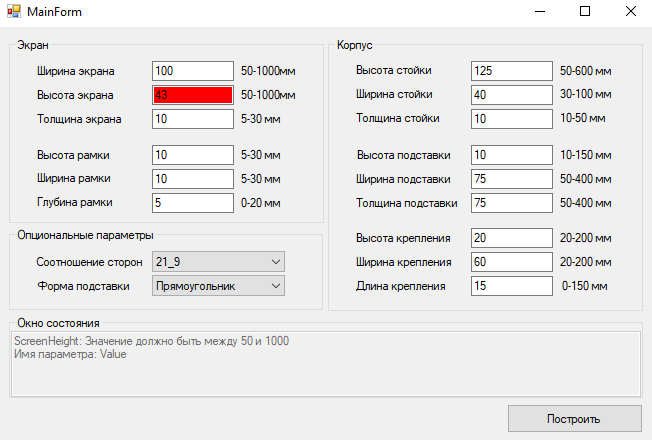


Рисунок 9.2 – Ошибка валидации зависимого параметра высоты экрана

На рисунках 9.3 и 9.4 представлено построение модели по стандартным параметрам экрана и корпуса, но с соотношением сторон выставленным на «16\_9».

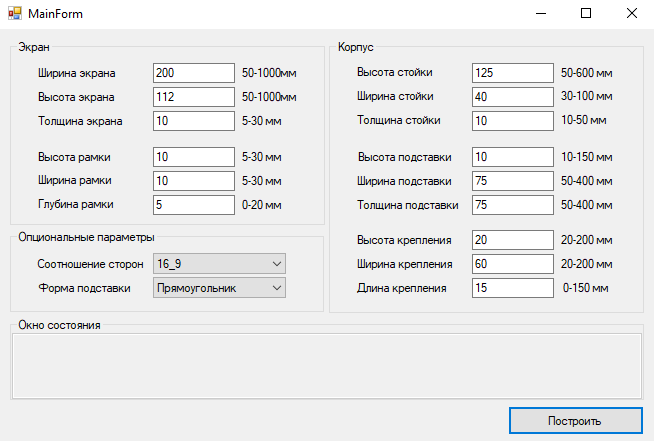


Рисунок 9.3 – Стандартные параметры, но с соотношением сторон на «16\_9»

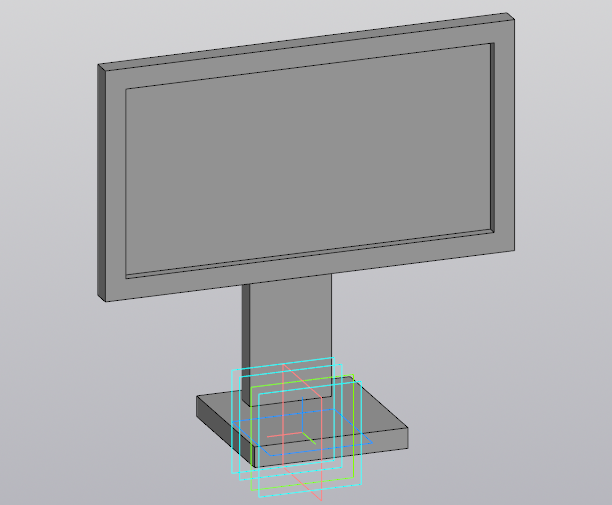


Рисунок 9.4 – Монитор по стандартным параметрам

На рисунках 9.5 и 9.6 представлено построение модели по максимальным параметрам.

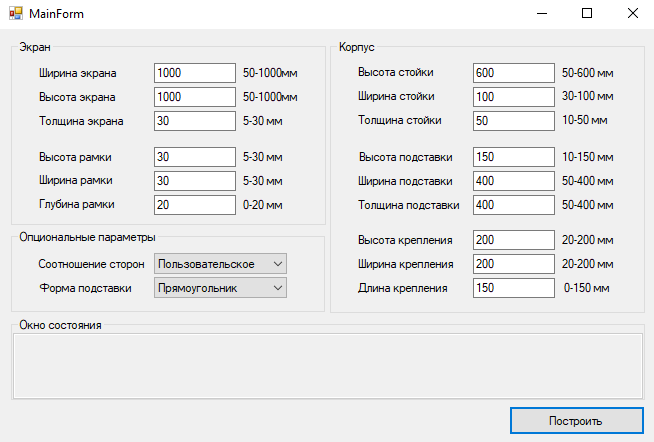


Рисунок 9.5 – Максимальные параметры

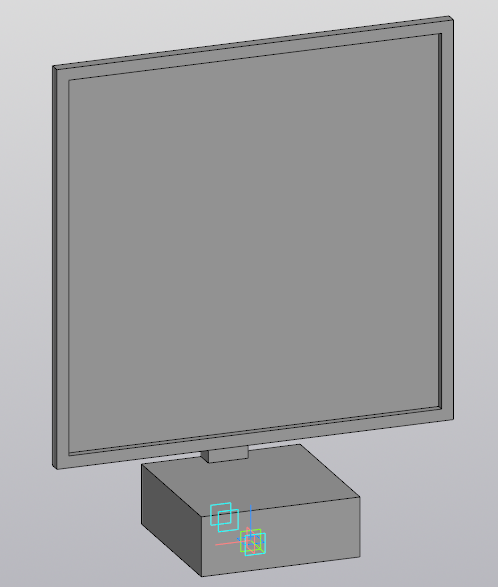


Рисунок 9.6 – Модель по максимальным параметрам

На рисунках 9.7 и 9.8 представлено построением модели по минимальным параметрам.

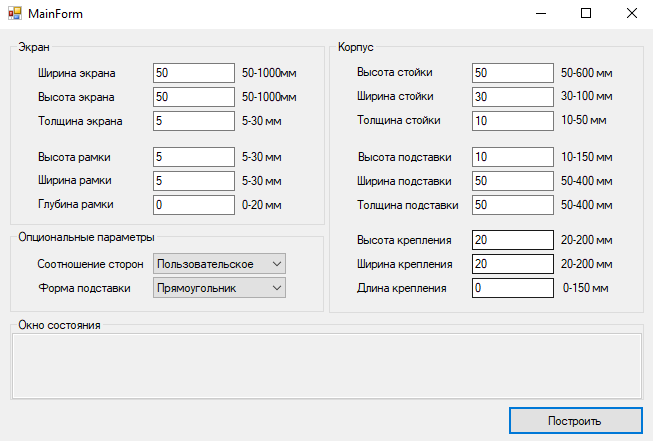


Рисунок 9.7 – Минимальные параметры

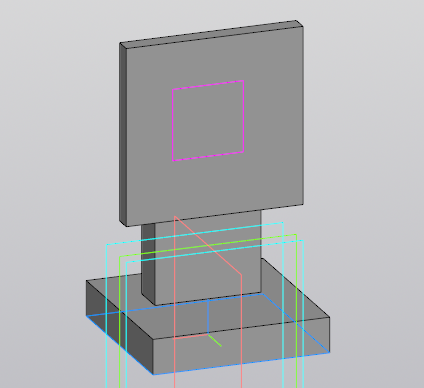


Рисунок 9.8 – Модель по минимальным параметрам

## 9.2 Модульное тестирование

На рисунке 9.9 представлено количество написанных Unit-тестов, а также что их выполнение происходит корректно.

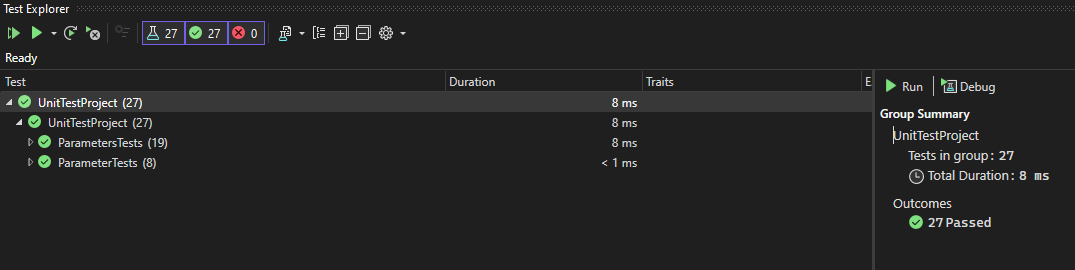


Рисунок 9.9 – Количество написанных Unit-тестов

В таблицах 9.1 и 9.2 представлены тесты для Parameters и Parameter.

Таблица 9.1 – Unit-тесты для Parameters

|  |  |
| --- | --- |
| Название теста | Краткое описание |
| Validate\_ScreenWidth\_WithinRange\_ShouldPass | Проверяет, что ширина экрана в пределах допустимого диапазона сохраняется корректно. |
| Validate\_ScreenWidth\_OutOfRange\_ShouldThrowException | Проверяет, что ширина экрана вне диапазона вызывает исключение. |
| AspectRatio\_ChangeAspectRatio\_ShouldRecalculateHeight | Убедиться, что изменение соотношения сторон пересчитывает высоту экрана. |
| CustomAspectRatio\_ShouldNotRecalculateHeight | Проверяет, что пользовательское соотношение сторон не влияет на пересчет высоты. |
| JointHeight\_ValidValue\_ShouldPass | Проверяет, что высота стойки в пределах диапазона сохраняется корректно. |
| JointHeight\_InvalidValue\_ShouldThrowException | Проверяет, что высота стойки вне диапазона вызывает исключение. |
| GetAspectRatioFactor\_FourThree\_ShouldReturnCorrectValue | Проверяет коэффициент для соотношения сторон 4:3. |
| GetAspectRatioFactor\_SixteenTen\_ShouldReturnCorrectValue | Проверяет коэффициент для соотношения сторон 16:10. |
| GetAspectRatioFactor\_SixteenNine\_ShouldReturnCorrectValue | Проверяет коэффициент для соотношения сторон 16:9. |
| GetAspectRatioFactor\_TwentyOneNine\_ShouldReturnCorrectValue | Проверяет коэффициент для соотношения сторон 21:9. |
| GetAspectRatioFactor\_Custom\_ShouldReturnDefaultValue | Проверяет, что пользовательское соотношение сторон возвращает значение по умолчанию. |
| GetAspectRatioFactor\_InvalidAspectRatio\_ShouldReturnDefaultValue | Проверяет, что для неверного соотношения сторон возвращается значение по умолчанию. |
| AddValueToParameter\_SetScreenWidth\_ShouldRecalculateScreenHeight | Убедиться, что изменение ширины экрана пересчитывает высоту. |
| AddValueToParameter\_SetScreenHeight\_ShouldRecalculateScreenWidth | Убедиться, что изменение высоты экрана пересчитывает ширину. |
| AddValueToParameter\_InvalidKey\_ShouldThrowKeyNotFoundException | Проверяет, что добавление значения с некорректным ключом вызывает исключение. |
| AddValueToParameter\_ScreenWidth\_CausesException\_ShouldCatchIt | Проверяет поведение при удалении параметра высоты экрана и задании ширины. |
| AddValueToParameter\_ScreenHeight\_CausesException\_ShouldCatchIt | Проверяет поведение при удалении параметра ширины экрана и задании высоты. |
| BaseShape\_SetAndGet\_ShouldWorkCorrectly | Проверяет корректность установки и получения формы подставки. |
| Constructor\_WithParameters\_ShouldInitializeCorrectly | Проверяет, что конструктор корректно инициализирует объект с параметрами. |

Таблица 9.2 – Unit-тесты для Parameters

|  |  |
| --- | --- |
| Название теста | Краткое описание |
| Constructor\_ShouldSetInitialValuesCorrectly | Проверяет, что конструктор корректно задаёт начальные значения параметра. |
| Value\_SetWithinRange\_ShouldSetValueSuccessfully | Проверяет, что ширина экрана вне диапазона вызывает исключение. |
| Value\_SetBelowMinValue\_ShouldThrowArgumentOutOfRangeException | Проверяет, что установка значения ниже минимального вызывает исключение. |
| Value\_SetAboveMaxValue\_ShouldThrowArgumentOutOfRangeException | Проверяет, что установка значения выше максимального вызывает исключение. |
| MinValue\_SetBelowCurrentValue\_ShouldUpdateMinValueSuccessfully | Проверяет, что минимальное значение успешно обновляется ниже текущего значения. |
| MinValue\_SetBelowValue\_ShouldSucceed | Проверяет, что минимальное значение можно установить ниже текущего значения параметра. |
| MaxValue\_SetAboveValue\_ShouldSucceed | Проверяет, что максимальное значение можно установить выше текущего значения параметра. |
| MaxValue\_SetAboveCurrentValue\_ShouldUpdateMaxValueSuccessfully | Проверяет, что максимальное значение успешно обновляется выше текущего значения. |

На рисунке 9.10 также представлен скриншот плагина ReSharper dotCover, измеряющего процент покрытия модульными тестами



Рисунок 9.10 – Результаты плагина

## 9.3 Нагрузочное тестирование

На рисунке 9.11 представлен график зависимости памяти ОЗУ от построения модели, а на рисунке 9.12 представлен график зависимости времени от построения модели.

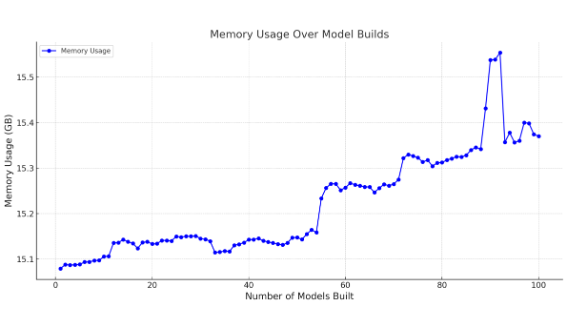


Рисунок 9.11 – График зависимости памяти ОЗУ от количества построенных моделей

Из графика 9.11 можно сделать вывод, что потребление памяти увеличивается с ростом количества построенных моделей. Это свидетельствует о том, что каждая новая модель добавляет нагрузку на оперативную память. При этом можно заметить, что рост потребления памяти носит нелинейный характер: в некоторых участках графика видны резкие скачки, которые могут быть вызваны особенностями реализации программы, такими как накопление промежуточных данных, неэффективная очистка временных объектов или утечки памяти.

После достижения определенного количества построенных моделей, которое зависит от объема доступной оперативной памяти, могут возникнуть проблемы, например, увеличение времени обработки из-за использования виртуальной памяти или даже сбой программы при нехватке памяти.

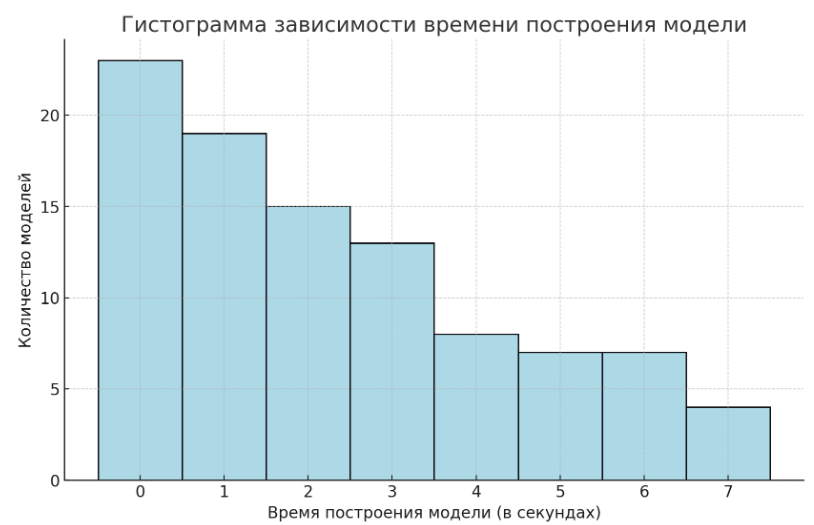


Рисунок 9.12 – График зависимости времени от построения модели в виде гистограммы

По графику можно сделать вывод, что время, затрачиваемое на построение каждой модели, остаётся относительно стабильным на начальных этапах, но с ростом количества построенных моделей появляются выбросы и тенденция к увеличению времени. Это может свидетельствовать о том, что программа постепенно начинает испытывать трудности с производительностью по мере увеличения нагрузки (возможно, из-за сторонних программ).

**10 ЗАКЛЮЧЕНИЕ**

В процессе лабораторных работ был создан плагин для КОМПАС-3D, который автоматически генерирует модель монитора на основе заданных параметров. Каждая работа была направлена на достижение этой цели: выбор объекта моделирования и САПР расширял знания о связанных технологиях, техническое задание определяло чёткий вектор действий, а проектирование системы требовало внимательного продумывания деталей, чтобы избежать сложных правок на поздних этапах. Реализация кода включала интеграцию с API и разработку удобного пользовательского интерфейса.

**11 СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

1. ГОСТ 23501.101-87 «Системы автоматизированного проектирования. Основные положения» (дата обращения 15.12.2024)
2. API [Электронный ресурс]. − Режим доступа <https://itglobal.com/ru-ru/company/glossary/api/> (дата обращения 25.09.2024)
3. КОМПАС-3D [Электронный ресурс]. − Режим доступа <https://kompas.ru/kompas-3d/about/> (дата обращения 26.09.2024)
4. Монитор [Электронный ресурс]. − Режим доступа <https://gostassistent.ru/doc/33b0a053-a9ee-4dfe-ae64-f93f6147cbd0> (дата обращения 20.09.2024)
5. Windows Forms [Электронный ресурс]. − Режим доступа <https://learn.microsoft.com/ru-ru/dotnet/desktop/winforms/overview/?view=netdesktop-9.0> (дата обращения 15.12.2024)
6. Github [Электронный ресурс]. − Режим доступа <https://docs.github.com/ru/get-started/start-your-journey/about-github-and-git> (дата обращения 15.12.2024)
7. ReSharper [Электронный ресурс]. − Режим доступа <https://www.jetbrains.com/ru-ru/resharper/> (дата обращения 13.12.2024)
8. Fine Code Coverage [Электронный ресурс]. − Режим доступа <https://marketplace.visualstudio.com/items?itemName=FortuneNgwenya.FineCodeCoverage> (дата обращения 17.12.2024)
9. NUnit [Электронный ресурс]. − Режим доступа <https://nunit.org/> (дата обращения 16.12.2024)
10. **AutoCAD Plant 3D Toolset** [Электронный ресурс]. − Режим доступа [https://www.autodesk.com/au/products/autocad/overview?plc=ACDIST&term=1-YEAR&support=ADVANCED&quantity=1&tab=subscription](https://www.autodesk.com/au/products/autocad/overview?plc=ACDIST&term=1-YEAR&support=ADVANCED&quantity=1&tab=subscription%20) (дата обращения 05.10.2024)
11. UML [Электронный ресурс]. − Режим доступа <https://www.uml-diagrams.org/> (дата обращения 07.10.2024)