Министерство науки и высшего образования Российской Федерации

ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ СИСТЕМ УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ

(ТУСУР)

Кафедра компьютерных систем в управлении и проектировании

Пояснительная записка

«**Плагин для построения заклепки в SOLIDWORKS**»

|  |  |  |  |
| --- | --- | --- | --- |
| Руководитель: |  | Выполнил: |  |
| к. т. н., доцент кафедры КСУП | | студент гр. 587-1 | |
| Калентьев А. А. | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | Шаламов А. О. | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
| Дата: | « » \_\_\_\_\_\_20\_\_ г. | Дата: | « » \_\_\_\_\_20\_\_ г. |

Содержание

Реферат 3

Термины и обозначения 4

Введение 5

1. Назначение приложения 5
2. Стек технологий разработки 6
3. Описание задачи 7
4. Проект программы-плагина и описание алгоритмов 11
5. Пользовательский интерфейс 13
6. Тестирование программы 14

Заключение 19

Список использованных источников 20

Приложение А Диаграмма классов 21

Реферат

Лабораторная работа, 21 страница, 12 рисунков, 1 таблица, 5 источников.

Ключевые слова: САПР, SOLIDWORKS, ЗАКЛЕПКА, МОДЕЛЬ, РАЗРАБОТКА, ПЛАГИН, API.

Целью данной работы является разработка плагина для создания трёхмерных моделей заклепок согласно заданным параметрам для системы автоматизированного проектирования «SolidWorks».

Отчёт по лабораторной работе выполнен в текстовом редакторе Microsoft Word.

Термины и обозначения

**Плагин** – программное обеспечение, подключаемое к другому программному обеспечению с целью расширить функциональность последней.

**Система автоматизированного проектирования** (*САПР*) – программное обеспечение, позволяющее автоматизировать процесс проектирования.

**Фреймворк** – программное обеспечение, облегчающее разработку программного обеспечения (набор библиотек и инструментов).

**Application Programming Interface** (API) – описание способа взаимодействия компьютерных программ друг с другом.

**Паттерн проектирования** – в объектно-ориентированном программировании это описание взаимодействия объектов и классов, адаптированных для решения общей задачи проектирования в конкретном контексте.

**Mock-объект** – в объектно-ориентированном программировании это объект, фиктивно реализующий определённый интерфейс для тестирования взаимодействия с ним.

**Integrated Development Environment** (IDE) – система программных средств для разработки программного обеспечения.

Введение

В настоящей пояснительной записке описывается внутренняя структура и логика разработанного плагина к САПР «SolidWorks» под названием «Конструктор заклепок».

1. Назначение приложения

Плагин «Конструктор заклепок» выполнен в виде пользовательского приложения для операционной системы Windows. Он предназначен для генерирования трёхмерной модели заклепки согласно задаваемым пользователем параметрам. Приложение должно предоставить пользователю возможность ввести параметры заклепки, а также непосредственно запускать САПР «SolidWorks» и отдавать ему команды для построения модели.

1. Стек технологий разработки

Данное приложение разработано на наборе фреймворков семейства .NET и включает в себя следующие библиотеки.

1. .NET Framework 4.7.2. Используется в качестве базового фреймворка для проектов решения.
2. WinForms. Используется для разработки пользовательского интерфейса приложения.
3. SolidWorks API. Используется для связи с САПР «SolidWorks» посредством встроенного API.
4. NUnit. Используется для юнит-тестирования бизнес-логики приложения.

Приложение разработано преимущественно на языке C# 7.3. Ресурсоёмких задач приложение не содержит, интерфейс должен соответствовать стандартам ОС Windows. Поскольку язык C# является частью .NET, он позволяет реализовать все части приложения в рамках одного базового фреймворка. Поэтому он является оптимальным выбором для данной задачи.

Для работы с приложением потребуются .NET Framework 4.8 Runtime, требующий, в свою очередь, ОС Windows 7 SP1 или новее. Минимальные требования Windows 7: 32- или 64-битный x86-совместимый CPU 1 ГГц, 1 ГиБ (для 32-битной системы) или 2 ГиБ (для 64-битной системы) ОЗУ, 16 ГиБ (для 32-битной системы) или 20 ГиБ (для 64-битной системы) дискового пространства, видеокарта с поддержкой DirectX 9. В системных требованиях к САПР «КОМПАС-3D» нет упоминаний минимальных требований к аппаратной конфигурации компьютера, за исключением обязательного наличия инструкций SSE2 у CPU, а также 1,6 ГиБ дискового пространства для установки базового пакета. Допускается работа на более современных версиях Windows для x86-совместимых ПК.

1. Описание задачи

Система автоматизированного проектирования «SolidWorks» от компании « Dassault Systèmes» представляет собой семейство программ, объединённых целью автоматизации процесса инженерного проектирования. Данная САПР позволяет создавать трёхмерные модели и чертежи технических объектов и готовить техническую документацию к ним в соответствии с единой системой конструкторской документацией, а также другими стандартами. Несмотря на то, что этим функциональность программы не ограничивается, в рамках данного проекта будет задействована лишь малая доля функциональности данной САПР, а именно трёхмерное моделирование.



Рисунок 3.1 – SolidWokrs2020 в режиме трёхмерного моделирования

Программа «SolidWorks» обладает стандартным для ОС Windows интерфейсом. Для трёхмерного моделирования используется множество операций: построение двумерных эскизов, выдавливание, вырезание, создание фасок, скруглений, оболочек и так далее. Все операции, совершаемые одна за другой, в итоге формируют дерево операций (на рисунке выше оно находится в левой части окна). Модель же формируется за счёт последовательного выполнения этих операций. Кроме того, можно задать свойства модели, такие как цвет, оптические свойства, масса и так далее.

Несмотря на очень высокую степень автоматизации процесса конструирования технических объектов, «SolidWorks» всё ещё предоставляет простор для дальнейшей автоматизации. В частности, создание однотипных технических объектов можно свести к генерации на основе входных параметров, дабы избежать множественного копирования модели-шаблона с ручным изменением параметров или создания их каждый раз с нуля. Осуществить такую автоматизацию «SolidWorks» позволяет с помощью своего API, включающего в себя существенную часть функциональности данной САПР. В частности, с помощью данного API можно строить чертежи и трёхмерные модели. С API «SolidWorks» можно работать на языках VBA, C++ и C#. В дистрибутиве «SolidWorks» также поставляется справочник по API данной программы.

В рамках данного проекта будет разработан плагин, позволяющий автоматизировать создание трёхмерных моделей заклепок. Технический объект – заклепка– показан на рисунке ниже.



Рисунок 3.2 – Заклепка

У данной заклепки пять изменяемых параметров.

1. Толщина шапки болта X (от 10 мм до 100 мм);
2. Радиус шапки болта Y (от 30 мм до 100 мм);
3. Радиус вырезки шапки болта SX (от 10 до X-10 мм);
4. Радиус стержня болта SY (от 15 мм до Y-10 мм);
5. Длина стержня болта CX (от 100 мм до 500мм);

Параметр *SY* зависит от параметра*Y*. Параметр *SX* не должен превышать

На рисунке ниже приведён вид чертежа данной заклепки. На нём можно легко видеть изменяемые параметры.

  
Рисунок 3*.3* – Изменяемые параметры

Для создания данной модели потребуется задействовать ряд операций моделирования, в числе которых создание эскизов и плоскостей, выдавливание (в том числе с вырезанием).

Взаимодействие с «SolidWorks» можно обеспечить несколькими способами, в частности, получив объект его экземпляра посредством COM. Если «SolidWorks» установлен на компьютере, то получить его экземпляр можно с помощью маршалинга. Если получить объект работающего экземпляра «SolidWorks» получилось, то необходимо его привести к интерфейсу *SldWorks* для дальнейшей работы. Если нет, то необходимо попытаться запустить «SolidWorks», после чего получить его экземпляр. Если и это не удалось, то это означает, что «SolidWorks» не установлен на компьютере должным образом.

Объект *SldWorks* позволяет производить множество операций, из которых потребуются, в частности, создание двумерных и трёхмерных документов, а также геометрические расчёты.Для создания детали необходимо вызвать метод *CreateNewDoc()*.

Также через *SldWorks* необходимо создавать и получать объекты различных структур. Например, через него можно получить структуры геометрических объектов (прямоугольников, линий, окружностей) для их дальнейшего построения, создать динамические массивы и коллекции тех или иных объектов или же получить специальный объект для геометрических расчётов.

1. Проект программы-плагина и описание алгоритмов

Исходя из назначения программы, логично выделить в ней несколько архитектурных единиц: графический интерфейс пользователя, бизнес-логику и внешние библиотеки. В число последних входят фреймворки, используемые при разработке, а также библиотеки API «SolidWOrks». Кроме того, необходимо покрыть бизнес-логику юнит-тестами.

Проект программы-плагина включает в себя три пакета, реализующих бизнес-логику, интерфейс и юнит-тесты соответственно. Диаграмма пакетов выглядит следующим образом.

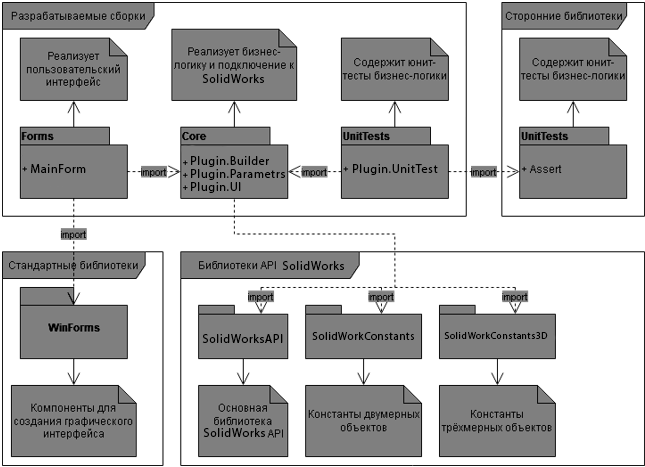


Рисунок *4.1* – Диаграмма пакетов

Пакет бизнес-логики включает в себя следующие классы.

1. Plugin.Builder – класс, реализующий модель заклепки. Данный класс будет включать в себя набор изменяемых свойств.
2. Plugin.Parametrs – класс, хранящий параметры заклепки.
3. Plugin.UI – класс, отвечающий за интерфейс программы.

Общий алгоритм работы программы заключается в следующем.

1. Пользователь вводит параметры розетки в соответствующие поля для ввода, меняя тем самым значения свойств модели Plugin.Parametrs. Если какое-либо из значений оказывается недопустимым, то программа покажет диалоговое окно с описанием ошибки.
2. При нажатии кнопки построения модели инициируется процесс построения модели, состоящий из нескольких шагов. За весь процесс построения отвечает класс Plugin.Builder.
3. Первым делом программа словит окно SoildWorks, затем полученные параметры приведутся в значения SolidWorks и будет произведено построение.

Алгоритмы работы программы реализованы согласно описанию. Стоит отметить, что в архитектуре программы имеются допущения в виде достаточно высокой степени связности и обращений бизнес-логики к API конкретному пакету САПР, а не к абстрактному, являющимся прослойкой между бизнес-логикой и API САПР. Применение паттернов проектирования «Стратегия» или «Мост» (подробнее о них в книге [2]) позволило бы решить эту проблему.

После выполнения основной части работы в проект были внесены правки с требованием включить следующие изменения.

1. Под кнопкой «Создать новый документ» расположен CheckBox, который отвечает за наличие вырезки.
2. Новый параметр, отвечающий за глубину вырезки.

Данная дополнительная функциональность также была реализована. Диаграмма классов приведена в приложении А.

1. Пользовательский интерфейс

В пакете пользовательского интерфейса реализован класс окна: MainForm.

Представляет собой главное окно программы, в котором пользователь должен будет указать параметры розетки и инициировать построение.

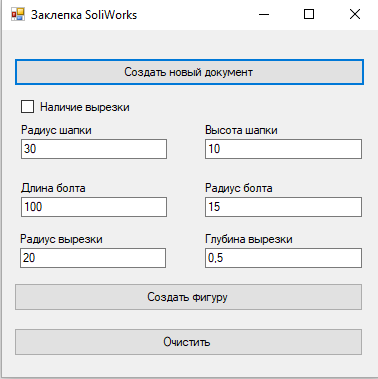


Рисунок *5.1* – Главное окно

1. **Тестирование программы**

Тестирование программы состоит из нескольких этапов.

1. Функциональное тестирование. Тестируется заявленная функциональность программы.
2. Модульное тестирование. Тестируется бизнес-логика на корректность работы с помощью инструментов юнит-тестирования.
3. Нагрузочное тестирование. Тестируется влияние большого числа построений за короткий промежуток времени на рост использования ресурсов компьютера.

Ввод неправильных значений пресекается программой: при попытке ввода неверных значений, программа выдаст окно с описанием ошибки.

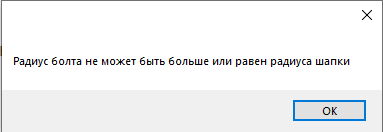


Рисунок *6.1* – Защита от ввода некорректных значений

Изначально все поля будут иметь значения по умолчанию. При нажатии кнопки «Создать модель» происходит переключение на «SolidWorks».



Рисунок *6.2* – Готовая модель

Юнит-тестирование произведено с помощью библиотеки NUnit. Условия задачи предполагают тестирование бизнес-логики.

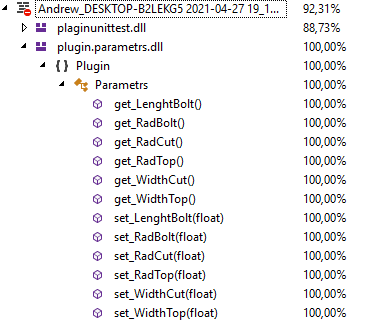


Рисунок 6.3 – Тестовые случаи

В таблице ниже приведены методы, покрывающие тестовые случаи.

Таблица 6.1 – Методы юнит-тестирования

|  |  |
| --- | --- |
| **Метод** | **Описание** |
| CorrectParametrs() | Проверяет корректность вносимых параметров |
| UncorrectParametrs() | Проверяет поведение программы на некорректный ввод параметров |
| Dependence() | Проверяет зависимости модели |

Отчёт VisualStudio о выполнении тестов приведён на рисунке ниже. Легко видеть, что все тесты завершены успешно.

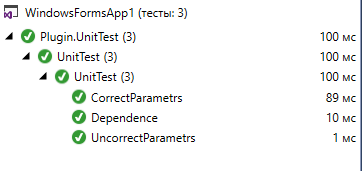


Рисунок *6.4* – *Отчёт о выполнении тестов*

Поскольку потребление памяти приложением статично, а скорость построения зависит от скорости работы «SolidWorks» и от конфигурации компьютера, измерению в стресс-тесте будет подвергаться потребление памяти при многократных построениях. Стоит заранее отметить, что программой не допускается построение нескольких моделей одновременно, а только по очереди.

Конфигурация компьютера, на котором выполнялся тест:

1. X86-совместимый центральный процессор AMD Ryzen 1600. Частота 3,2 ГГц, 6 ядер, 12 потоков
2. 8 ГБ DDR4 ОЗУ на частоте 2800 МГц в двухканальном режиме
3. Графический процессор AMD Radeon R9 270 с 2 ГБ видеопамяти
4. ОС Windows 10 x64

Нагрузочное тестирование изначально предполагало построение 1000 моделей и удержание их открытыми в «SolidWorks». Тест завершился досрочно на построении 76-ой модели, так как «SolidWorks» аварийно завершил свою работу без вывода информации об ошибке.

Потребление ОЗУ самой программой составляет в среднем 20 МБ, в то время как потребление памяти пакетом «SolidWorks» сильно увеличивается со временем.



Рисунок 6.5 – Зависимость потребления ОЗУ от количества моделей

Скорость построения напрямую зависит от свободного времени CPU и пропускной способности памяти. Косвенная зависимость от количества памяти заключается в том, что при её нехватке операционная система будет вынуждена предоставлять пакету «SolidWorks» области файла подкачки, что существенно снизит его производительность. Такой ситуации не удалось достичь на вышеуказанной конфигурации по причине избытка памяти и преждевременного завершения работы «SolidWorks».

Тем не менее, на нижеприведённом графике легко видеть, что количество времени, затрачиваемого на построение, растёт с количеством открытых моделей. Отсюда можно предположить наличие операций при работе с деревом, выполняющихся более чем за константное время.

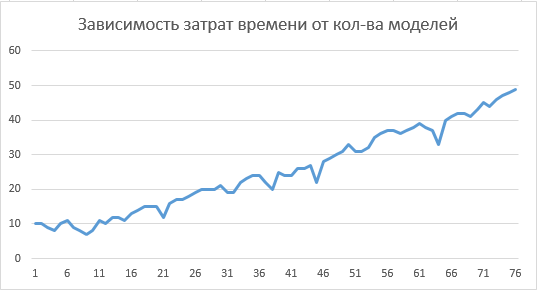


Рисунок *6.7* – Зависимость затрат времени от количества моделей

Пиковое потребление ОЗУ было при удержании 76 моделей, оно составило 2,6 ГБ.

Исходя из результатов тестирования, можно сделать вывод, что программа выполняет заявленную функциональность в полном объёме. Использовать её для построения огромного количества моделей и удержания их в памяти не представляется возможным, так как среда «SolidWorks» версии 2020 не выдерживает нагрузки подобного рода. Рекомендуется строить модель заклепки для разового применения.

Заключение

В результате выполнения работы в рамках курса «Основы разработки систем автоматизированного проектирования» была разработана программа-плагин для системы «SolidWorks», выполняющая построение трёхмерной модели заклепки.

Плагин выполнен в виде отдельного приложения, подключающегося к системе «SolidWorks» и отдающего ему команды на построение модели. Заявленная функциональность реализована полностью, включая дополнительную функциональность. Тестирование программы проведено в три этапа: функциональное, модульное и нагрузочное, по итогам которого были проверены основные функции программы и защита от ввода некорректных данных, протестирована математическая модель розетки и измерено влияние количества одновременно открытых построенных моделей на расход оперативной памяти компьютера.

Список использованных источников

Книги

1. Новые технологии в программировании: учебное пособие / А. А. Калентьев, Д. В. Гарайс, А. Е. Горяинов – Томск: Эль Контент, 2014. – 176 с.
2. Паттерны объектно-ориентированного программирования / Э. Гамма, Р. Хелм, Р. Джонсон, Дж. Влиссидес – Санкт-Петербург: Питер, 2020. – 448 с.

Электронные ресурсы

1. Плагин – Википедия. [Электронный ресурс]. – Режим доступа: https://ru.wikipedia.org/wiki/API (дата обращения 24.03.2021)
2. API – Википедия. [Электронный ресурс]. – Режим доступа: https://ru.wikipedia.org/wiki/API (дата обращения 24.03.2021)
3. Справочная система КОМПАС-МАСТЕР. [Электронный ресурс]. 2014. 11 декабря.

Приложение А

Диаграмма классов

