Министерство образования и науки

Российской Федерации

Федеральное государственное бюджетное образовательное

учреждение высшего профессионального образования

ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

СИСТЕМ УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ (ТУСУР)

Кафедра компьютерных систем в управлении и проектировании

(КСУП)

**РЕАЛИЗАЦИЯ ДИНАМИЧЕСКИ ПОДКЛЮЧАЕМОГО МОДУЛЯ**

**«ПАЛАТКА»**

**НА БАЗЕ СИСТЕМЫ КОМПАС 3D**

Пояснительная записка

на курсовой проект по дисциплине

«Разработка систем автоматизированного проектирования»

|  |  |
| --- | --- |
|  | Руководитель:  М.н.с. ЛИКС кафедры КСУП  \_\_\_\_\_\_\_\_\_\_\_ Калентьев А.А. |
|  | Задание принял к исполнению:  студент гр. 580-2  \_\_\_\_\_\_\_\_\_\_\_\_ Цымбалов В.А. |

Томск 2015

РЕФЕРАТ

Курсовой проект содержит: страниц – 23, источников – 15, рисунков – 12, таблиц – 2.

[КОМПАС-3D](http://kompas.ru/) V14, TENT, ПЛАГИН, ПАЛАТКА, ДВЕРЬ, ОКНО.

Целью данного проекта является разработка и реализация плагина «Палатка» на базе системы [КОМПАС-3D](http://kompas.ru/) V14, с использованием методов и свойств интерфейсов API 12.

В результате был создан плагин «Палатка», стоящий трехмерную модель туристической палатки в системе [КОМПАС-3D](http://kompas.ru/) V14.

Пояснительная записка выполнена в текстовом редакторе Microsoft Office 2007.

Министерство образования и науки

Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего профессионального образования

ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

СИСТЕМ УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ (ТУСУР)

Кафедра компьютерных систем в управлении и проектировании (КСУП)

Утверждаю:

Зав. кафедрой КСУП

\_\_\_\_\_\_\_\_Ю.А. Шурыгин

«\_\_\_»\_\_\_\_\_\_\_\_\_\_\_ 2014г.

ЗАДАНИЕ

по курсовому проекту по дисциплине «Разработка САПР»

Выдано: Студенту группы 580-2 Цымбалову Владимиру Алексеевичу

1. Тема проекта: Разработка плагина «Палатка» на базе системы «КОМПАС-3D V14».
2. Срок сдачи студентом проекта: 9.01.2015
3. Исходные данные

Разработать плагин «Палатка» на базе системы «КОМПАС-3D V14» согласно ГОСТ 11871-88.

1. Требования к плагину

Плагин должен обеспечивать следующую функциональность:

- выводить диалоговое окно ввода для изменения следующих параметров:

1. ширина палатки (рис. 1, поз. a);
2. высота палатки (рис. 1, поз. b);
3. длина палатки (рис. 1, поз. c);
4. ширина входа (рис. 1, поз. d);
5. высота входа (рис. 1, поз. e);
6. ширина окошка (рис. 1, поз. f);
7. высота окошка (рис. 1, поз. g);
8. ширина пола (рис. 1, поз. i);
9. длина пола (рис. 1, поз. j).

Результатом работы библиотеки является трехмерная модель палатки (Рисунок 1), построенная на основе введенных данных.

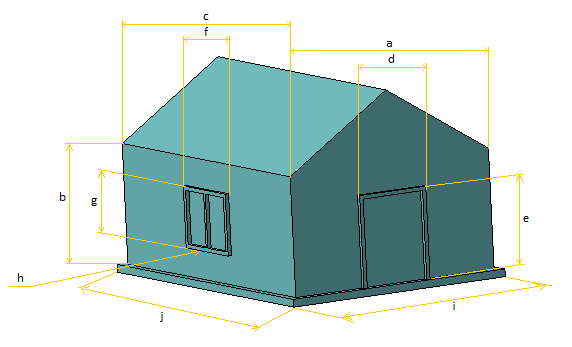


Рисунок 1 ­– Палатка

- обеспечивать построение трехмерной модели на графическом окне системы «КОМПАС-3D V14» на основе введенных значений параметров;

- обеспечить корректность ввода данных и вывод информационного сообщения при вводе некорректных данных.

Корректность введенных данных:

1. ширина палатки не должна быть такой же, как ширина пола;
2. длина палатки не должна быть такой же, как длина пола;
3. ширина входа не должна быть меньше на одну единицу ширины палатки;
4. высота входа не должна быть меньше на семь единиц высоты палатки;
5. ширина окошка не должна быть меньше на одну единицу длины палатки;
6. высота окошка не должна быть меньше на семь единиц высоты палатки;
7. количество окон не должно превышать двух единиц;
8. Сфера применения

Модуль предназначен для использования в архитектурной сфере проектирования деталей.

1. Требования к программной части

- программа должна работать на операционных системах: Windows 7 (x32/x64), Windows 8.1 (x32/x64);

- программа должна быть выполнена на языке C# в среде Microsoft Visual Studio 2012, .NET Framework 4.5, для системы КОМПАС 3D V14.

1. Содержание пояснительной записки

* титульный лист;
* реферат;
* техническое задание;
* содержание;
* введение;
* постановка задачи;
* основная часть;
* заключение;
* список литературы;
* приложения.

1. Дата выдачи задания: 17.09.2014

Руководитель м.н.с. ЛИКС каф. КСУП:  
Калентьев A.А.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Задание принял к исполнению  
Цымбалов В.А.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Оглавление

[Введение 7](#_Toc407106854)

[1 Постановка и анализ задачи 8](#_Toc407106855)

[1.1 Выбор инструментов и средств реализации 8](#_Toc407106856)

[1.2 Назначение плагина 8](#_Toc407106857)

[2 Описание реализации 9](#_Toc407106858)

[2.1 Диаграмма вариантов использования 10](#_Toc407106859)

[3 Описание реализации 11](#_Toc407106860)

[3.1 Диаграмма классов 12](#_Toc407106861)

[4 Описание программы для пользователя 16](#_Toc407106862)

[5 Тестирование 18](#_Toc407106863)

[5.1 Функциональное тестирование 18](#_Toc407106864)

[5.2 Модульное тестирование 20](#_Toc407106865)

[5.3 Нагрузочное тестирование 21](#_Toc407106866)

[Заключение 22](#_Toc407106867)

[Список использованных источников 23](#_Toc407106868)

# Введение

В процессе проектирования в независимости от конкретного используемого программного продукта, могут возникнуть задачи, когда стандартных средств и инструментов проектирования, содержащихся в данном программном продукте, для реализации конкретной (специфичной) задачи недостаточны. При возникновении таких задач, можно перейти к другому программному продукту, который может содержать средства и функционал, позволяющий решить задачу, но не всегда такой подход эффективен, так как полный переход к новому программному продукту влечет за собой определенные трудности. Трудности, начиная с освоения нового программного продукта, влекущие временные затраты, а так же убытками на покупку и освоение.

Решением может послужить создание специализированного под конкретную задачу плагина [1], (модуля). Удобность применения плагина обусловлена тем, что плагин независимо компилируемый программный модуль, динамически подключаемый к основной программе, который можно осуществить при помощи технологии Application Programming Interface (API) [2]. API-технология предоставляет программисту набор процедур и функций для управления основной программой, но не дает прямого доступа к свойствам и методам объектов внутри основной программы.

Курсовой проект ориентирован на создание подсистемы системы автоматизированного проектирования (САПР) [3], в данном случае реализацией подсистемы будет плагин, который будет выполнять функции и задачи подсистемы на базе системы КОМПАС-3D V14 [4].

# **1** Постановка и анализ задачи

В рамках данного курсового проекта в соответствии с технически заданием требовалось разработать плагин, который на основе входных параметров, интегрируя с системой КОМПАС 3D V14, строит модель «Палатка». Более того, требовалось, чтоб плагин позволял изменять входные параметры в соответствии с выданным заданием.

## Выбор инструментов и средств реализации

В связи с требованием технического задания программа выполнена на языке C# в среде Microsoft Visual Studio 2012 [5], с использованием .NET Framework 4.5 [6], для системы КОМПАС 3D V14 [7].Инструментом тестирования и создания модульных тестов был выбран стандартный обозреватель тестов [8] среды Microsoft Visual Studio 2012.

Взаимодействие плагина с системой КОМПАС 3D V14 осуществляется посредством интерфейсов, называемых API. В КОМПАС 3D V14 на данный момент существует API двух версий API 5 и API 7. Явных преимуществ между версиями нет, поскольку обе версии реализуют различные функции системы и взаимно дополняют друг друга. Для курсового проекта была выбрана версия API 5, так как для полноценной реализации плагина достаточно методов и свойств интерфейсов API 5.

## Назначение плагина

Назначение, разрабатываемого плагина обусловлено выбором сферы его применения. Областью применения данного проекта является производство туристического оборудования, которая стала наиболее популярна в настоящее время. На сегодняшний день существует множество программных продуктов для проектирования различных элементов, деталей, конструкций.

# Описание реализации

Для формального [7] описания архитектурной особенности, пользовательского сценария системы был выбран унифицированный язык моделирования (UML) [10]. На основе UML построены: диаграммы вариантов использования [11], диаграммы классов [12].

В процессе реализации диаграммы классов и вариантов были дополнены и модифицированы.

Модификация, прежде всего, была связана с унификацией [13] архитектуры программы.

Дополнения были обусловлены в связи нарастанием функционала в реализации мелких детальных особенностей.

### Диаграмма вариантов использования

Диаграмма вариантов использования отражает возможный выбор действий (выбора состояния) пользователя внутри системы (Рисунок 2.1).

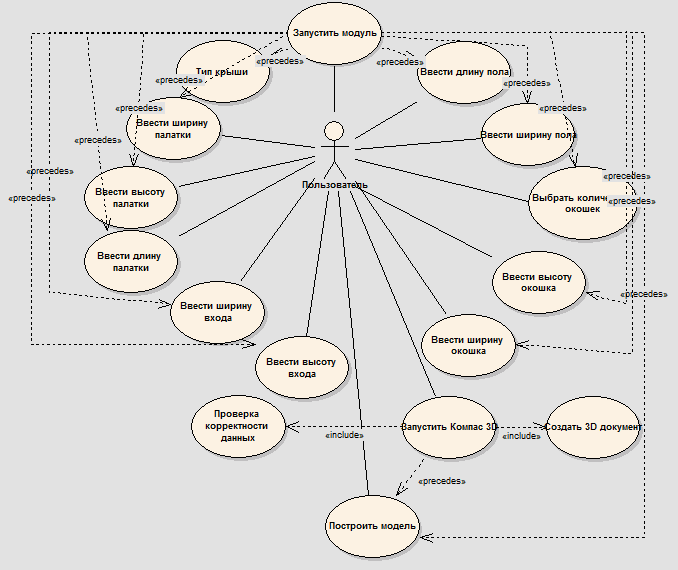


Рисунок 2.1 — Диаграмма вариантов использования

Установлено шестнадцать прецедентов (вариантов использования).

# Описание реализации

На этапе разработки проекта программы для формального [12] описания архитектурной особенности, пользовательского сценария системы был выбран унифицированный язык моделирования (UML) [13]. На основе UML построены: диаграммы вариантов использования [14] диаграммы классов [15].

В процессе реализации диаграммы классов, вариантов использования и пакетов были дополнены и модифицированы.

Дополнения были обусловлены в связи нарастанием функционала в реализации мелких детальных особенностей.

### Диаграмма классов

Диаграмма классов служит для представления статической структуры модели системы в терминологии классов объектно-ориентированного программирования. Диаграмма классов является дальнейшим развитием концептуальной модели проектируемой системы. Диаграмма классов представлена на рисунке (Рисунок 3.1).



Рисунок 3.1 – Диаграмма классов

Для реализации подсистемы были спроектированы следующие классы:

* Main – основной класс программы,
* TentCreatorWindow – базовый класс, определяющий модель палатки,
* ModelParameters − содержит параметры модели,
* ParameterData – содержит значения параметра,
* Parameter – содержит параметры модели,
* TentCreatorManager – cодержит методы для построения детали,
* ModelBuilder – cодержит методы для построения детали,
* IModelPart – cодержит интерфейс части модели,
* ModelBody – cодержит методы построения тела палатки,
* Door – содержит методы построения входа,
* Window – содержит методы построения окон,
* Floor – содержит методы построения пола,
* KompasSketch – содержит свойства эскиза.

В таблице 3.1 приведено описание классов, изображенных на диаграмме.

Таблица 3.1 – Описание классов

|  |  |
| --- | --- |
| Поле/метод | Описание |
| Main | |
| \_kompas: KompasObject | Интерфейс объекта КОМПАС |
| \_tentCreatorWindow: TentCreatorWindow | Форма для ввода параметров |
| \_mainWindow | Форма для ввода параметров |
| GetLibraryName(): string | Возвращает название библиотеки |
| ExternalRunCommand([In] short command, [In] short mode, [In, MarshalAs(UnmanagedType.IDispatch)] object kompas): void | Срабатывает при выборе пункта меню |
| ExternalMenuItem(short number, ref short itemType, ref short command): string | Формируем пункты меню |
| ExecuteCommand(short command): void | Выполняет команду |
| ShowMainWindow(): void | Показывает основное окно программы |
| RegisterKompasLib(Type type): void | Функция выполняется при регистрации класса для COM |
| UnregisterKompasLib(Type type): void | Отменяет регистрацию библиотеки |
| TentCreatorWindow | |
| \_modelParameters: ModelParameters | Содержит методы для работы с параметрами модели |
| \_tentCreatorManager: TentCreatorManager | Содержит методы для построения модели |
| \_controlsDictionary: Dictionary<Parameter, Control> | Список Control'ов |
| TentCreatorWindow(KompasObject kompas) | Конструктор с параметром, интерфейс объекта КОМПАС |
| InitializeVariables(): void | Инициализирует переменные |
| BuildModel(): void | Строит модель |
| GetModelParameters():Dictionary<Parameter, ParameterData> | Получает параметры модели |
| GetParameterValue(string text): double? | Преобразует текстовое значение поля в числовое значение |
| BuildButton\_Click(object sender, EventArgs e): void | Возникает в момент нажатия на кнопку |
| \_nonNumberEntered: bool | Переменная для ограничения ввода |
| TextBox\_KeyDown(object sender, KeyEventArgs e): void | Переменная для ограничения ввода |
| TextBox\_KeyPress(object sender, KeyPressEventArgs e): void | Переменная для ограничения ввода |
| Продолжение таблицы 3.1 | Про |
| TextBox\_Leave(object sender, EventArgs e): void | Возникает в момент потери фокуса ввода |
| ModelParameters | |
| Parameters: Dictionary<Parameter, ParameterData> | Словарь параметров |
| ModelParameters | Содержит параметры модели |
|  |  |
| ModelParameters() | Конструктор по умолчанию |
| Initialize(): void | Инициализирует переменные |
| CheckData(Dictionary<Parameter, ParameterData>parameters): List<string> | Проверяет корректность введенных данных, словарь параметров для проверки, список ошибок |
| GetValidValue(Parameter parameter): ParameterData | Возвращает допустимые значения |
| SetMaxValue(Parameter parameter, float maxValue): void | Задает новое максимальное значение параметра |
| SetMinValue(Parameter parameter, float minValue): void | Задает новое минимальное значение параметра |
| SetRange(Parameter parameter, float minValue, float maxValue): void | Задает новый диапазон значений параметра |
| ModelBuilder | |
| Build (ksDocument3D document3D, Dictionary<Parameter, ParameterData> parameters): void | Строит модель |
| HideAllGeom(ksDocument3D document3D): void | Скрывает все оси и геометрические обозначения |
| SetViewProjection(ksDocument3D document3D, int index):void | Задает ориентацию |
| IModelPart | |
| Create(ksDocument3D document3D, Dictionary<Parameter, ParameterData> parameters): void | Строит часть модели |
| ModelBody | |
| Create(ksDocument3D document3D, Dictionary<Parameter, ParameterData> parameters): void | Строит часть модели |
| Window | |
| Create(ksDocument3D document3D, Dictionary<Parameter, ParameterData> parameters): void | Строит часть модели |
| Floor | |
| Create(ksDocument3D document3D, Dictionary<Parameter, ParameterData> parameters):void | Строит часть модели |
| Door | |
| Create(ksDocument3D document3D, Dictionary<Parameter, ParameterData> parameters): void | Строит часть модели |

Продолжение таблицы 3.1

|  |  |
| --- | --- |
| KompasSketch | |
| \_breakPointsList: List<int> | Список разделителей |
| KompasSketch() | Конструктор по умолчанию |
| Initialize(): void | Инициализирует переменные |
| SketchName: string | Название эскиза |
| NormalValue: double | Значение свойства |
| ReverseValue: double | Количество копий в массиве |
| Shape: ShapeType | Примитив |
| Plane: ShapeType | Плоскость для рисования |
| OperationColor: Color | Цвет операции |
| DirectionType: Direction\_Type | Направление команды |
| EndType: End\_Type | Тип выдавливания |
| PointsList: List<PointF> | Список координат фигуры |
| Operation: OperationType | Список операций |
| AddBreakPoint(): void | Добавляет разделитель линий |
| ClearBreakPoint(): void | Очищает список разделителей линий |
| CreateNewSketch(ksPart part): void | Создает новый эскиз |
| CreateNewOffsetSketch(ksPart part, double distance, bool direction): void | Создает новый эскиз |
| DrawLine(ksDocument2D sketchEdit): void | Рисует линию |
| DrawArc (ksDocument2D sketchEdit): void | Рисует дугу |
| BaseExtrusion (ksPart part, ksEntity entitySketch): void | Базовая операция выдавливания |
| CutExtrusion (ksPart part, ksEntity entitySketch): void | Базовая операция выреза выдавливанием |
| GetKompasColor(Color color): int | Преобразует цвет модели в понятный для Компаса |
| ActivePlane: short | Возвращает текущую плоскость для рисования |

# 4 Описание программы для пользователя

Плагин состоит из одного диалогового окна с одиннадцатью пунктами для ввода параметров и кнопкой «Построить модель» (Рисунок 4.1).

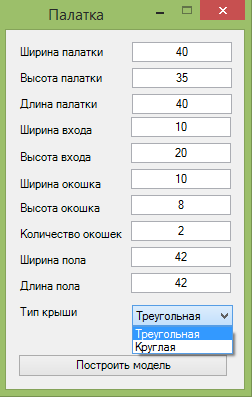


Рисунок 4.1 — Диалоговое окно плагина перед выбором параметров

После ввода всех параметров и нажатия на кнопку «Построить модель», выводится диалоговое окно (Рисунок 4.2), сообщающее о том, что нет активного документа, который предлагает два варианта выбора.

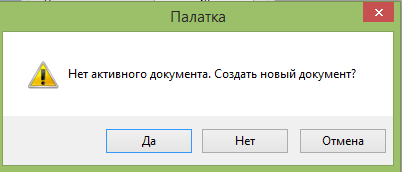


Рисунок 4.2 — Диалоговое окно плагина после нажатия кнопки «Построить модель»

После нажатия кнопки «Да» в диалоговом окне, открывается диалоговое окно документа системы КОМПАС 3D V14 и строится деталь (Рисунок 4.3).

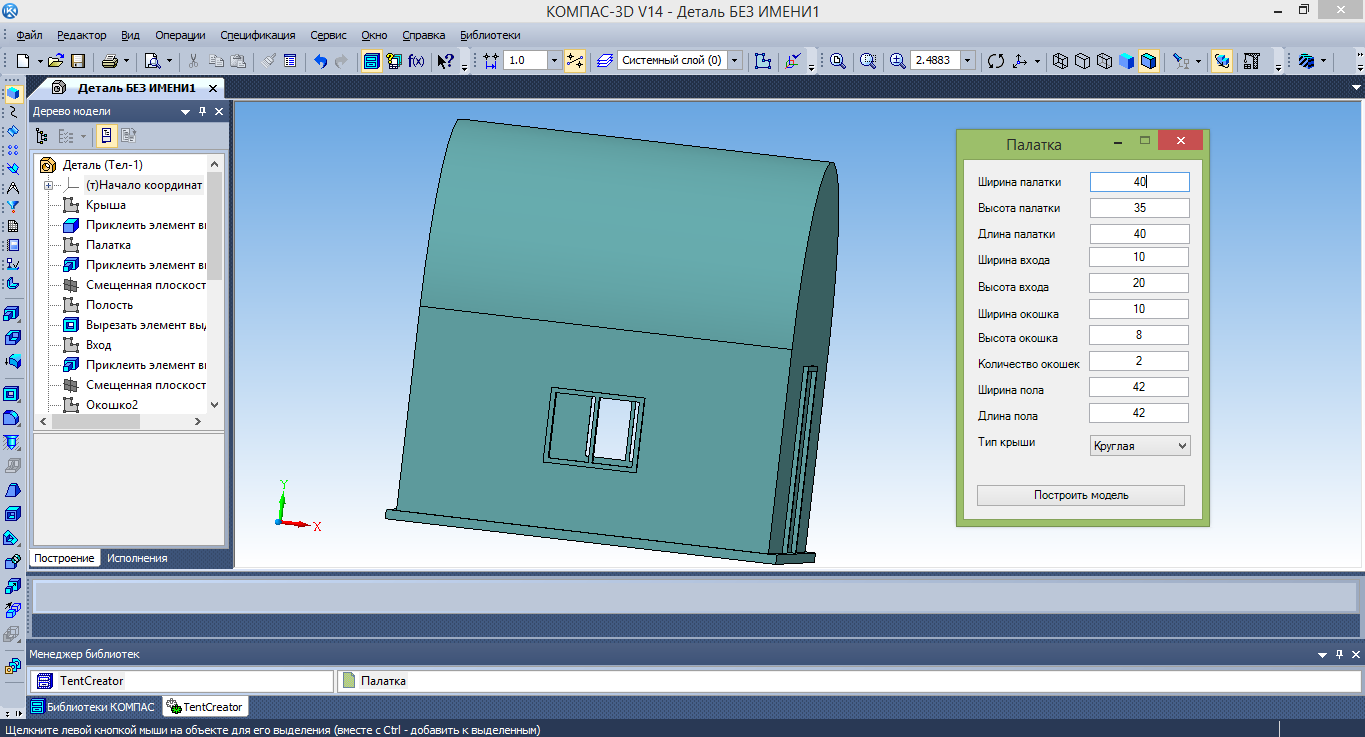


Рисунок 4.3 — Диалоговое окно документа системы КОМПАС 3D V14 после построения модели

При последующем нажатии кнопки «Построить деталь», выводится диалоговое окно (Рисунок 4.4), сообщающий о том, что документ не пуст, который предлагает два варианта выбора.

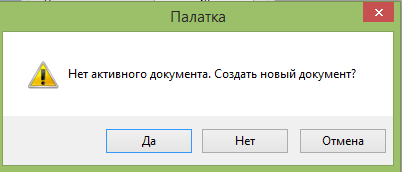


Рисунок 4.4 — Диалоговое окно плагина системы КОМПАС 3D V14

при повторном нажатии кнопки «Построить деталь»

Если пользователь выберет «Да», то построенная деталь удаляется, и на ее месте строится новая, а в случае, когда выбрано «Нет» − текущий документ не закрывается, а новая деталь создается в новом документе. Если оставить поле не заполненным, выводится сообщение (Рисунок 4.5), сообщающее о неполном вводе. В этом случае следует заполнить все поля и запустить плагин снова.

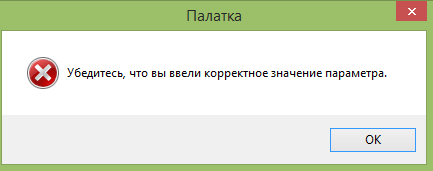


Рисунок 4.5 — Диалоговое окно программы при оставлении пустого поля

# Тестирование

## Функциональное тестирование

При функциональном тестировании проверялась корректность работы плагина «Палатка», а именно, соответствие полученного результата в виде трехмерной модели с моделью, соответствующей в задании, поведение плагина на различные входные данные.

При запуске плагина с не заполненными полностью данными, плагин выдает сообщение о том, что построение в таком случае невозможно. (Рисунок 4.5).

Предусмотрен случай, повторного построения детали, то есть при повторном нажатии кнопки построить деталь, выводится сообщение (Рисунок 4.4) о том, что документ не пуст.

Проверялась возможность, не закрыв текущий документ, создать новую деталь в новом документе (Рисунок 5.1, Рисунок 5.2).

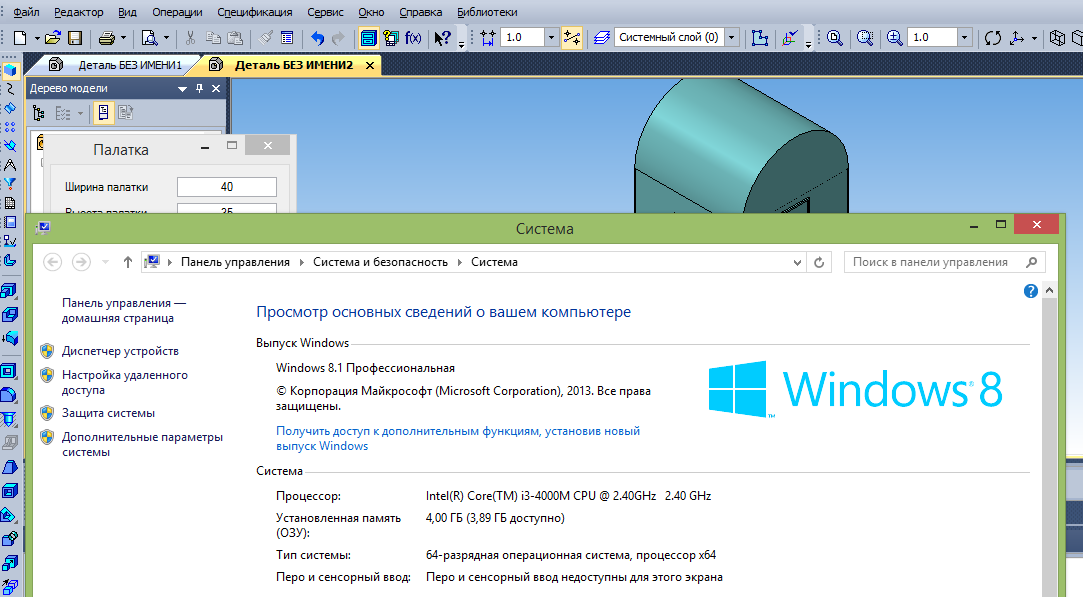


Рисунок 5.1 — Диалоговое окно системы КОМПАС 3D V14 в случае отказа закрытия текущего документа

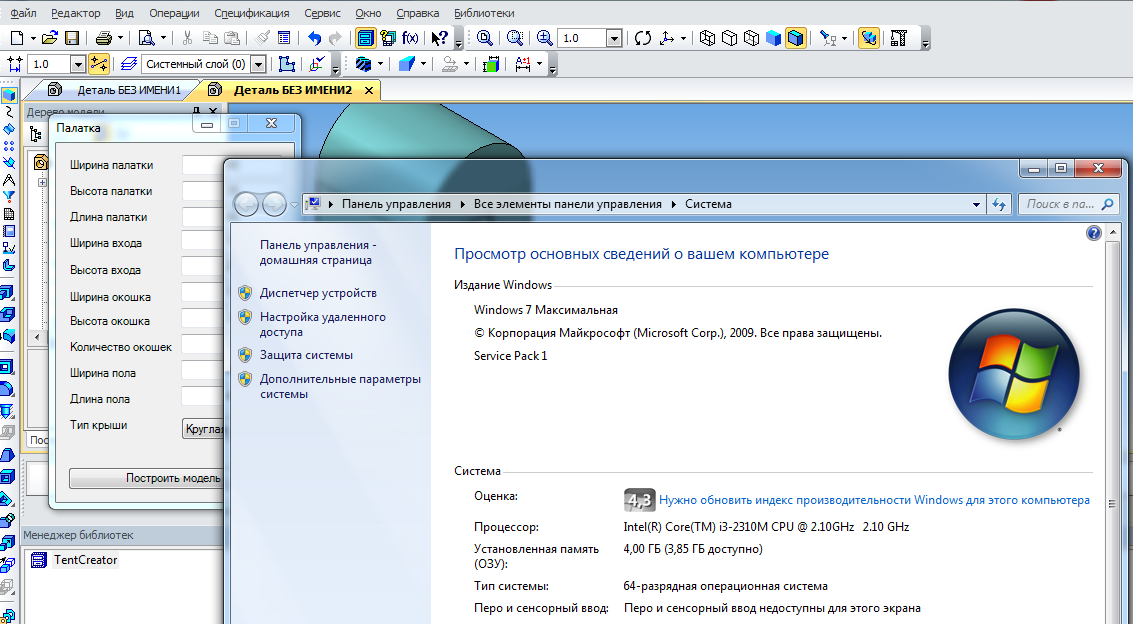


Рисунок 5.2 — Диалоговое окно системы КОМПАС 3D V14 в случае отказа закрытия текущего документа

## Модульное тестирование

В целях проверки корректности работы методов и свойств классов при помощи обозревателя тестов Visual Studio 12 (Рисунок 5.3) было проведено модульное тестирование, проверялись открытые члены поля и методы, для этого был создан тестовый класс:

* TestTentCreator – содержит тестовые методы

В таблице 5.1 приведено описание класса.

Таблица 5.1 – Описание класса

|  |  |
| --- | --- |
| TestTentCreator | |
| CheckPatametrConstructor(): void | Метод тестирующий ввод допустимых значений параметров |
| CheckPatametrConstructorThree(): void | Метод тестирующий ввод допустимых значений параметров |
| CheckNormalValueProperty(): void | Метод тестирующий свойства и методы класса KompasSketch |
| CheckValidParametrs(): void | Метод тестирующий ввод недопустимых значений параметров |

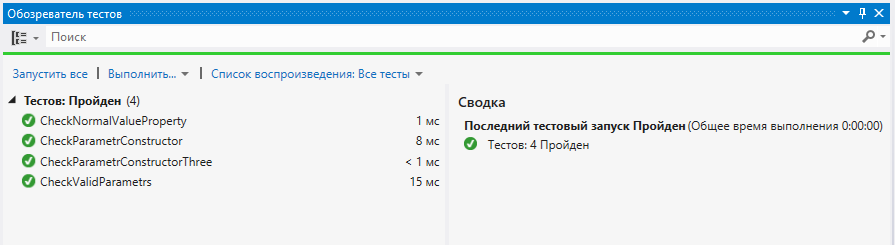


Рисунок 5.3 — Диалоговое окно состояний запущенных тестов

# Нагрузочное тестирование

В целях проверки производительности работы плагина, было проведено нагрузочное тестирование. Для проведения нагрузочного тестирования был добавлен таймер, который засекал время от начала до конца построения. Был произведен эксперимент на десять построений, который реализовывался в цикле, после каждого построения, текущий интервал записывался в список интервалов. В конце эксперимента был изучен список из экспериментальных данных, на основе которых был построен график зависимости времени от номера построения модуля. (Рисунок 5.4)



### Рисунок 5.4 – График зависимости

Из данного графика видно, что наиболее длительное время было потрачено на последнюю операцию построения.

Тестирование проводилось на компьютере, со следующими характеристиками: Процессор Intel® Core™ i3-4000M CPU @ 2.40 GHz, с частотой 2.4 ГГц, 4 Гб оперативной памяти, чипсет графического контроллера NVIDIA GeForce 840M, с размером видеопамяти 2048 Мб.

# Заключение

В ходе реализации плагина были изучены предметная область проектирования, предмет проектирования, аналоги предмета проектирования, API и на основании полученных данных была спроектированы архитектура системы и макет системы, создан плагин «Палатка», проведены функциональные, модульные и нагрузочные тесты.

# Список использованных источников

1. Плагин [Электронный ресурс]. − Режим доступа: http://ohostinge.com/dictionary/plug-in (дата обращения 14.11.2014)
2. API [Электронный ресурс]. − Режим доступа: http://habrahabr.ru/sandbox/52599/ (дата обращения 14.11.2014)
3. САПР [Электронный ресурс]. − Режим доступа: http://www.tadviser.ru/index.php/Система\_автоматизированного\_проектирования (дата обращения 14.11.14)
4. Компас (САПР) [Электронный ресурс]. − Режим доступа: http://machinery.ascon.ru/software/tasks/items/?prcid=6&prpid=7 (дата обращения 14.11.14)
5. Visual Studio 2012 [Электронный ресурс]. − Режим доступа: http://msdn.microsoft.com/ru-ru/hh916378.aspx (дата обращения 14.11.2014)
6. .NET Framework 4.5 [Электронный ресурс]. − Режим доступа: http://msdn.microsoft.com/ru-ru/library/w0x726c2(v=vs.110).aspx (дата обращения 14.11.2014)
7. КОМПАС 3D V14 [Электронный ресурс]. − Режим доступа: http://kompas.ru/news/items/?news=1911 (дата обращения 14.11.2014)
8. Модульные тесты [Электронный ресурс]. − Режим доступа: http://msdn.microsoft.com/ru-ru/library/ms182532.aspx (дата обращения 14.11.2014)
9. DWG [Электронный ресурс]. − Режим доступа: http://www.autodesk.ru/products/dwg (дата обращения 14.11.14)
10. CAM. [Электронный ресурс]. − Режим доступа: http://rucadcam.ru/ (дата обращения 14.11.14)
11. Формальный язык [Электронный ресурс]. − Режим доступа: http://inf1.info/language (дата обращения 14.11.14)
12. UML. [Электронный ресурс]. - Режим доступа: http://www.uml.org/ (дата обращения 14.11.2014)
13. Диаграмма вариантов использования [Электронный ресурс]. − Режим доступа: http://www.infosystem.ru/designing/methodology/uml/theory/use\_case\_diagram\_theory.html (дата обращения 14.11.14)
14. Диаграмма классов [Электронный ресурс]. − Режим доступа: http://www.informicus.ru/Default.aspx?SECTION=6&id=73&subdivisionid=3 (дата обращения 14.11.14)
15. Унификация [Электронный ресурс]. − Режим доступа: http://www.nuru.ru/delo/004.htm 3 (дата обращения 14.11.14)