

ПЛАН ЛАБОРАТОРНОЙ РАБОТЫ № 1

По дисциплине:	Программная инженерия
Тема занятия:	Методология Rational Unified Process. Определение общей идеи проекта.
Цель занятия:	описать организационные аспекты создания программного обеспечения на стадии определения требований к системе по методологии Rational Unified Process.
Количество часов:	2

Содержание работы:

1. Предметная область проекта
2. Постановка задачи, определение требований
3. Создание прецедентов
4. Диаграмма прецедентов
5. Диаграмма действий

Методические указания по выполнению:

Создание диаграммы вариантов использования

Постановка задачи. Разрабатываемая система должна обеспечивать возможность добавления новых заказов, изменения старых, выполнения заказов, проверки и возобновления инвентарных описей. При получении заказа система должна также послать сообщение бухгалтерской системе, которая выписывает счет. Если требуемого товара нет на складе, заказ должен быть отклонен.

Поведение системы – её функционал – описывается с помощью модели прецедентов, которая отображает:

- Прецедентов – возможность моделируемой системы (use cases)
- Актёров – действующие лица (actor)
- Связи между актёрами и прецедентами (use cases diagram)

Прецеденты говорят, **что** делает система, но не говорят, как. Об этом говорят *сценарии*, но в текстовой форме, что делает их довольно сложными для восприятия. На помощь приходят *диаграммы взаимодействия*, которые визуализируют сценарии.

Актёры

Актёры не являются частью системы, они представляют собой объект, который должен взаимодействовать с системой.

Актёры могут:

- Снабжать информацию систему
- Получать информацию из системы.

В языке UML актер изображается в виде фигуры человечка (рис.1).

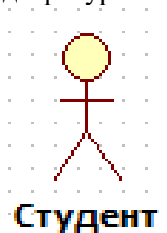


Рис.1 Нотация UML для изображения актера

С помощью прецедентов (use cases) моделируется диалог между актером и системой. Прецеденты определяют возможности, обеспечиваемые системой для актера.

В языке UML прецедент изображается в виде фигуры овала (рис.2).

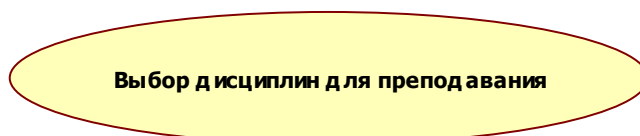


Рис.2 Нотация UML для изображения прецедента

Создание диаграммы Вариантов Использования

Создайте диаграмму Вариантов Использования для системы обработки заказов. Требуемые для этого действия подробно перечислены далее. Готовая диаграмма Вариантов Использования должна выглядеть как на рисунке 1.

Этапы выполнения упражнения

Создать диаграммы Вариантов Использования, вариантов использования и Действующих лиц

1. Дважды щелкните на Главной диаграмме Вариантов Использования (Main) в браузере, чтобы открыть ее.
2. С помощью кнопки Use Case (Вариант Использования) панели инструментов поместите на диаграмму новый вариант использования.
3. Назовите этот новый вариант использования "Ввести новый заказ".
4. Повторите этапы 2 и 3, чтобы поместить на диаграмму остальные варианты использования: Изменить существующий заказ, Напечатать инвентарную опись, Обновить инвентарную опись, Оформить заказ, Отклонить заказ
5. С помощью кнопки Actor (Действующее лицо) панели инструментов поместите на диаграмму новое действующее лицо.
6. Назовите его "Продавец"
7. Повторите шаги 5 и 6, поместив на диаграмму остальных действующих лиц: Управляющий магазином, Клерк магазина, Бухгалтерская система

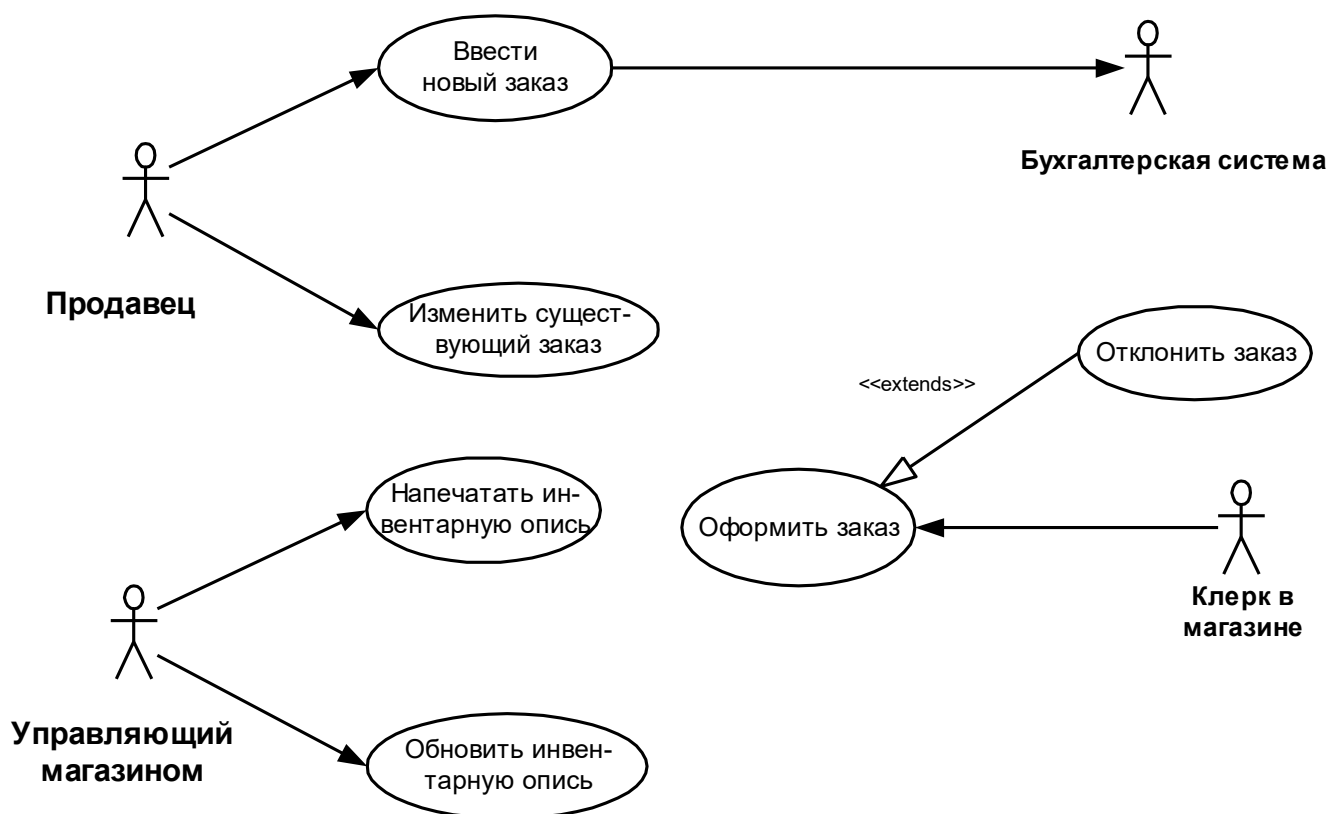


Рис. 1. Диаграмма Вариантов Использования для системы обработки заказов.

Указать абстрактные варианты использования

1. Щелкните правой кнопкой мыши на варианте использования "Отклонить заказ" на диаграмме.
2. В открывшемся меню выберите пункт Open Specification (Открыть спецификацию).
3. Пометьте контрольный переключатель Abstract (Абстрактный), чтобы сделать этот вариант использования абстрактным.

Добавить ассоциации

1. С помощью кнопки Unidirectional Association (Однонаправленная ассоциация) панели инструментов нарисуйте ассоциацию между действующим лицом Продавец и вариантом использования "Ввести новый заказ".
2. Повторите этот этап, чтобы поместить на диаграмму остальные ассоциации.

Добавить связь расширения

1. С помощью кнопки Generalization панели инструментов нарисуйте связь между вариантом использования "Отклонить заказ" и вариантом использования "Оформить заказ". Стрелка должна протянуться от первого варианта использования ко второму. Связь расширения означает, что вариант использования "Отклонить заказ" при необходимости дополняет функциональные возможности варианта использования "Оформить заказ".
2. Щелкните правой кнопкой мыши на новой связи между вариантами использования "Отклонить заказ" и "Оформить заказ".
3. В открывшемся меню выберите пункт Open Specification (Открыть спецификацию).
4. В раскрывающемся списке стереотипов введите слово extends (расширение), затем нажмите ОК.
5. Слово <<extends>> появится на линии данной связи.

Добавить описания к вариантам использования

1. Выделите в браузере вариант использования "Ввести новый заказ".
2. В окне документации введите следующее описание к этому варианту использования: Этот вариант использования дает клиенту возможность ввести новый заказ в систему.
3. С помощью окна документации введите описания ко всем остальным вариантам использования.

Добавить описания к действующему лицу

1. Выделите в браузере действующее лицо Продавец
2. В окне документации введите для этого действующего лица следующее описание: Продавец - это служащий, доставляющий и старающийся продать продукцию.
3. С помощью окна документации введите описания к оставшимся действующим лицам.

Прикрепление файла к варианту использования

1. Для описания главного потока событий варианта использования "Ввести новый заказ" создайте файл OrderFlow.doc, содержащий следующий текст:
 1. *Продавец выбирает пункт "Создать новый заказ" из имеющегося меню.*
 2. *Система выводит форму "Подробности заказа".*
 3. *Продавец вводит номер заказа, заказчика и то, что заказано.*
 4. *Продавец сохраняет заказ.*
 5. *Система создает новый заказ и сохраняет его в базе данных.*
2. Щелкните правой кнопкой мыши на варианте использования "Ввести новый заказ".
3. В открывшемся меню выберите пункт Open Specification (Открыть спецификацию)
4. Перейдите на вкладку файлов.
5. Щелкните правой кнопкой мыши на белом поле и из открывшегося меню выберите пункт Insert File (Ввести файл).
6. Укажите файл OpenFlow.doc и нажмите на кнопку Open (Открыть), чтобы прикрепить файл к варианту использования.

Задание для самостоятельного выполнения. После выполнения всех указаний по рассматриваемому проекту сделать те же действия по теме, указанной ниже.

Предметная область. Крымский инженерно-педагогический университет – высшее учебное заведение, предоставляющее образовательные услуги в традиционной форме. В связи с растущим интересом к онлайн обучению, на кафедре прикладной информатики принято решение развернуть тестовую версию компьютерной системы, позволяющую вести регистрацию дисциплин в формате e-learning.

Преподаватели кафедры решают, какие дисциплины они будут вести в течение семестра, а служба регистрации дисциплин вносит информацию в систему. В результате преподаватели получают отчет по дисциплинам, которые они ведут, а студенты – перечень дисциплин на семестр.

Студенты для службы регистрации заполняют регистрационную форму, где указывают выбранные дисциплины (ограниченное количество). После успешного распределения студентов по курсам, им высылается расписание, а преподавателям – список студентов по каждой дисциплине.

1. Постановка задачи, определение требований

В начале каждого семестра студенты могут запросить перечень дисциплин, предлагаемых в данном семестре. Информация о дисциплинах должна содержать ФИО преподавателя, факультет, краткое описание. Студент должен выбрать 4 дисциплины + 2 дисциплины про запас, если курс будет переполнен. На курс, читаемый по дисциплине, должно быть записано не более 10, но не менее 3 студентов (менее 3-х студентов – дисциплина не читается).

После регистрации система регистрации направляет информацию в систему оплаты для выставления счетов студентам.

Преподаватели должны иметь доступ к системе для указания дисциплин и для просмотра списка записавшихся студентов.

Создание прецедентов

Прецеденты и Актеры в системе регистрации дисциплин

1. Студент может зарегистрироваться на курсы (дисциплины).
2. Преподаватель может выбрать дисциплины, которые будет читать.
3. Регистратор должен создать учебный план и составить перечень дисциплин на семестр.
4. Регистратор должен хранить информацию о дисциплинах, преподавателях, студентах.
5. Система оплаты должна получать информацию из системы регистрации.

Выделим актеров:

- Студент,
- Преподаватель,
- Регистратор,
- Система оплаты.

На основании перечисленных потребностей можно выделить прецеденты:

- регистрация на курсы;
- выбор курсов для преподавания;
- запрос расписания курсов;
- управление информацией о курсах;
- управление информацией о преподавателях;
- управление информацией о студентах;
- создание каталога курсов.

Инструменты:

1. Rational Software Architect – средство моделирования.
2. Rational Rose Enterprise Edition
3. Свободно-распространяемый программный продукт StarUML – Режим доступа:
<http://www.staruml.com/>
4. Microsoft Visual Studio 2010
5. Microsoft Visual Studio 2012