# Задание на семестр

RUP -> UP

Выполнить разработку проекта в рамках первой итерации унифицированного процесса.

Каждая команда получит от преподавателя предметную область: мини-проект, который является частью одного большого проекта, другие части достанутся другим командам, взаимодействие команд не подразумевается.

## Лабораторные

Лаба 1

Определение прецедентов (include, extend)

Лаба 2

Создание модели предметной области(классы должны быть похожими; представление понятий, выраженных в терминах реального мира)

Лаба 3

Создание *системных* диаграмм последовательностей(коммуникаций-маленький брат, не такая выразительная как послед., показывает какие есть задачи) – основана на лабе 1

Лаба 4

Создание диаграмм коммуникаций (понять какие классы и какие обязанности у этих классов, показывает как (кем) реализуется) – описывает функции, которые идут из предыдущей лабы

(Паттерн Грасп?)

Лаба 5

Создание диаграмм классов проектирования (прямое отображение: диаграмма классов проектирования может не до конца соответствовать модели предметной области)

Лаба 6

Создание схемы данных (в любой нотации, например, UML)

Лаба 7

Реализовать программное обеспечение (проектирование на C#)

Анализ - 1-2

Проектирование 3-6

Реализация - 7

Что будет в отчете?

Отчет: введение, основная часть, заключение; каждый может реализовывать отчет отдельно

Курсовая аттестуется независимо от предмета, в форме зачета

## Лекция 1: Описание прецедентов

Прецеденты – это не диаграммы, а текстовые описания. Диаграммы имеют второстепенное значение.

* Исполнитель – actor – сущность обладающая поведением, например, человек (идентифицируемый по **роли** -> кассир), компьютерная система, организация.
* Сценарий – специальная последовательность действий или взаимодействий между исполнителем и системой (max – 3-4). Т.е. покупка товара – прецедент один, а сценариев несколько.
* Прецедент – набор взаимосвязанных сценариев использования.
* Три типа исполнителя
  + **Основной исполнитель**
  + Вспомогательный исполнитель
  + Закулисный исполнитель

Форматы прецедентов:

* Сжатый
* Свободный
* **Развернутый**

Не более 4 сценариев и не более 10 действий

### Предусловия и постусловия

Предусловия – это те предпосылки, на выполнение которых разработчик прецедента хочет обратить свое внимание. (Условия, которые всегда считаются истинными)

Постусловия – описывают, какие условия обязательно должны выполнятся в случае успешного завершения сценария.

Пример:

Постусловие – данные о продаже сохранены.

### Сценарии

Основной успешный сценарий

Имена исполнителей начинают с заглавной буквы. Повторяющиеся действия *выделяют курсивом.*

Альтернативный сценарий

Здесь указываются все основные сценарии или ветви, приводящие к и успешному и к неудачному

Альтернативные сценарии – ответвления от основного сценария. Сначала определяются условия, а затем реакция на них. (п.3, альтернативный 3а, 3б и т.д.)

По окончанию обработки альт сценария по умолчанию происходит возврат к основному сценарию, если не предусмотрен другой ход событий.

Если нужно описать условия, которые могут возникнуть в любой момент, обозначается - \*.

### Формат описания сценариев

2 колонки: действия исполнителя, отклик системы. Для каждого сценария своя таблица.

### Базовый стиль описания

Не уделяйте внимания вопросам интерфейса пользователя, сосредоточьте внимание на содержательной стороне вопроса. При выполнении задания использовать *базовый стиль*. Описание намерений.

### Конкретный стиль (не используем)

## Описание прецедентов

### Описание прецедентов в формате «черный ящик»

Можно указать, что делает система

|  |  |
| --- | --- |
| Базовый | Другие |
| Система регистрирует покупку | Система записывает сведения о покупке в базу данных |

1. Определение рамок системы

Определить, что не относится к системе, т.е. определить внешний основных и вспомогательных исполнителей

1. Определение основных исполнителей
2. Определение основных задач
3. Определение прецедентов

Одна задача – один прецедент. Имя прецедента начинается с существительного, описывающего действия. (Использовать единую часть речи)

### Критерии выделения прецедентов

* Критерий одобрения руководством
* Критерий EBP (элементарный бизнес-процесс)

Прецедент – это не один маленький шаг, основное внимание уделяется получению ощутимого результата и переходу системы и данных в устойчивой состояние

* Критерий размера (подозрительно - слишком описание короткое)

Исключения (подзадача (include) (например, проверка прав), расширение (extend)(фильтрация и сортировка)). include – нельзя исключить, это обязательно, а без extend можно обойтись.

## Лекция 2: Модель предметной области

Модель предметной области – визуальное представление концептуальных классов или объектов реального мира в терминах предметной области.

Модель предметной области может отражать:

* объекты предметной области или концептуальные классы,
* ассоциации между концептуальными классами,
* атрибуты,

**не используются:**

* артефакты программирования наподобие окон или базы данных, если только разрабатывается система не является моделью программного средства, например моделью графического интерфейса пользователя,
* обязанности или методы.

НИКАИХ МЕТОДОВ, КРОМЕ СИСТЕМНЫХ ДО 4-5 ЛАБ.

НИКАКИХ ТАБЛИЦ И БАЗ ДАННЫХ ПРИ ПРОЕКТИРОВАНИИ.

Модель предметной области не является **моделью данных.**

### Как создать модель предметной области?

\_\_\_\_\_\_\_\_\_\_\_\_\_таам шото было\_\_\_\_\_\_\_\_\_

### Стратегии идентификации

1. **Повторное использование или модификация существующих моделей**
2. Использование списка категорий концептуальных классов
3. Выделение существительных

### Класс или атрибут

Если некоторый объект Х в реальном мире не является числом или текстом, то это скорее класс чем атрибут.

### Классы описаний

– используются для элементов, которые многократно повторяются, поэтому описание можно хранить отдельно

### Сокращение количества ассоциаций

Ассоциация – линия между классами, пишется с большой буквы

Однонаправленная ассоциация это линия со стрелкой (A -> B – А знает о B, но B не знает о A) – рисуем БЕЗ стрелочке на линии

### Роли

* Кратность определяет, сколько экземпляров класса А может быть ассоциировано с одним экземпляром B.
* Имя
* Направление связи

Если указываем роль, то не надо писать имя на ассоциации.

Между двумя классами может быть установлено несколько ассоциаций.

### Атрибуты

Используем атрибуты, которые выполняются в конкретном прецеденте.

Вычисляемый атрибут показывается с помощью «/».

Не использовать атрибут в качестве внешнего ключа, связывайте типы с помощью ассоциаций, а не атрибутов.

## Системные диаграммы последовательностей

Только системные функции (прямой аналог API)

Сколько сценариев – столько диаграмм последовательностей

Норма – один исполнитель