

# Проектирование DSL на основе метамодели предметной области и открытых прецедентов

---

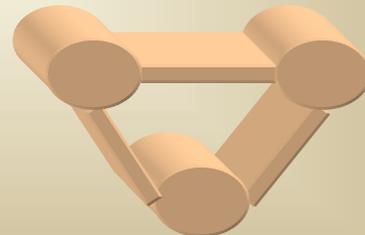
Южно-Уральский государственный университет, г. Челябинск

Цытович Павел Леонидович

✉ [pavel.tsytovich@item74.ru](mailto:pavel.tsytovich@item74.ru)

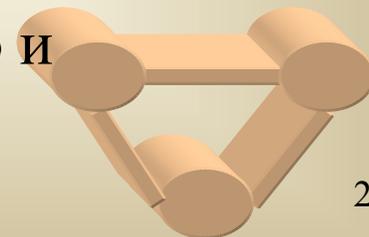
🌐 <http://www.item74.ru>

🌸 ICQ UIN: 17608677



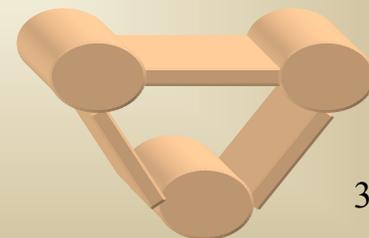
# О докладчике

- ⇒ Работаю в Южно-Уральском государственном университете (г. Челябинск) на кафедре ЭВМ
- ⇒ Читаю курсы лекций и веду практические занятия по дисциплинам «Программной инженерии» - «Программирование на языке С++», «Технология программирования», «Программирование в Internet», «Компонентные технологии»
- ⇒ В 2000 году защитил диссертацию на соискание ученой степени кандидата технических наук
- ⇒ В 2004 году проходил стажировку в Российском отделении корпорации Intel по программе «Виртуоз», проводимой при поддержке корпораций Intel, Microsoft, ЗАО «Лаборатория Касперского» и Нижегородского государственного университета
- ⇒ Сфера научных интересов: Методы и технологии проектирования программных систем прикладного и системного назначения

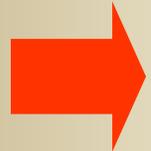


# Agenda...

- ↗ Роль DSL в проектировании
- ↗ UML vs DSL или метамодель предметной области
- ↗ UML + Open Use cases = DSL
- ↗ Практический пример
- ↗ Перспективы
- ↗ Выводы



# Agenda...



## ➤ Роль DSL в проектировании

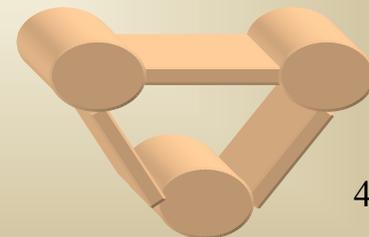
➤ UML vs DSL или метамодель предметной области

➤ UML + Open Use cases = DSL

➤ Практический пример

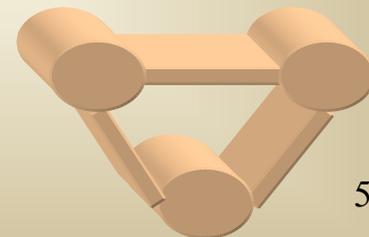
➤ Перспективы

➤ Выводы



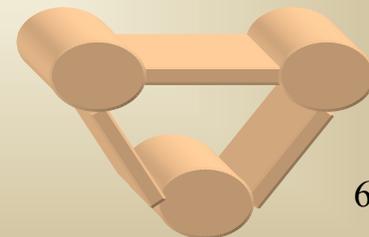
# Что такое DSL

- ↗ Язык описания предметной области
- ↗ Предназначен для описания решения задачи в терминах и концепциях предметной области
- ↗ Может использоваться приемлемая для конкретной предметной области нотация – от графических схем, до описания на естественном языке

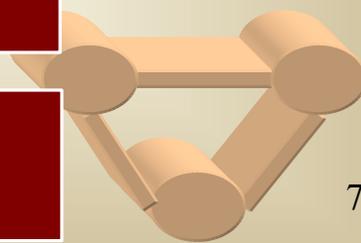
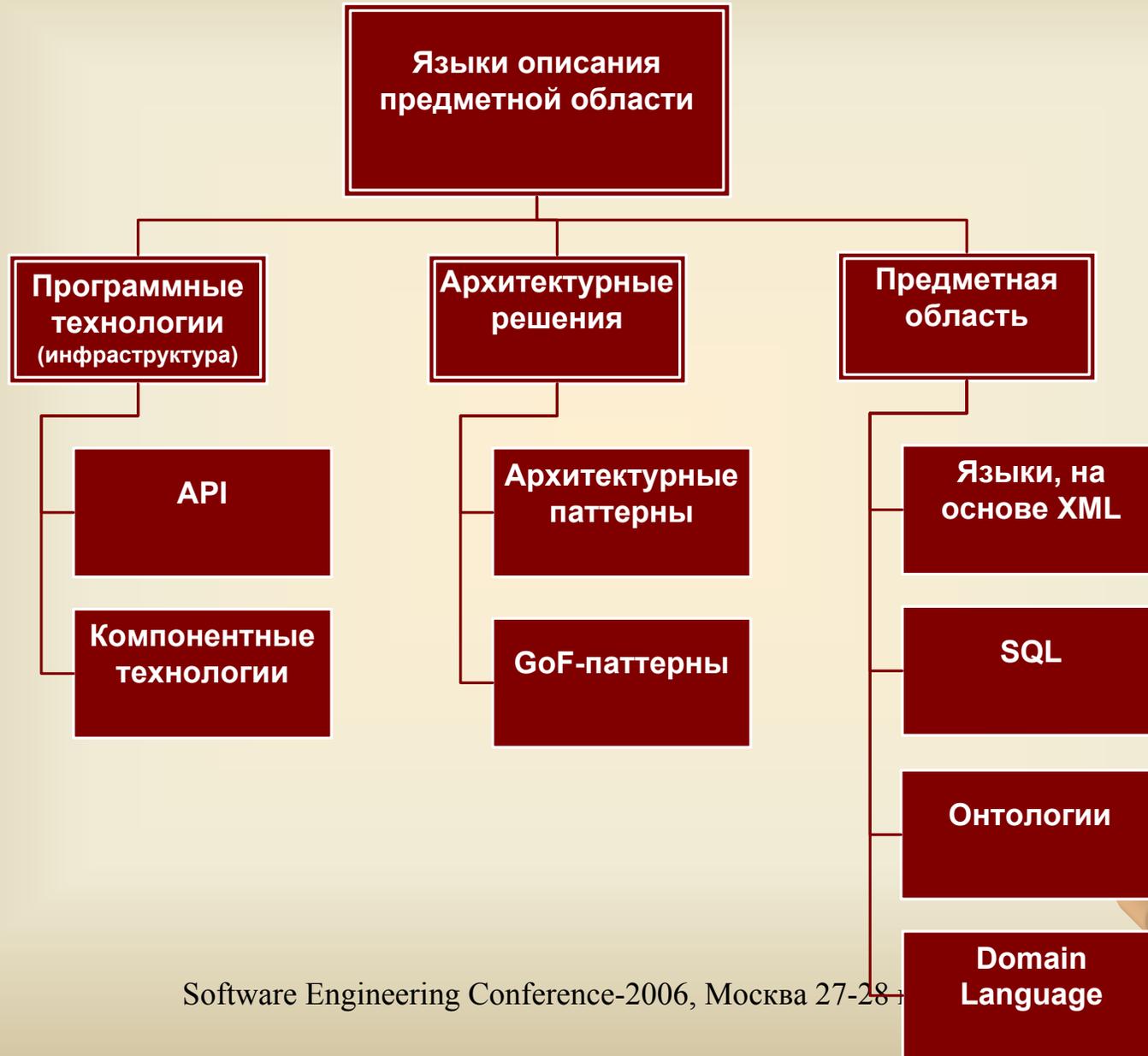


# Идея DSL

- ⇒ Объектно-ориентированный подход
- ⇒ Компонентное программирование
- ⇒ Framework для узкоспециализированных задач
- ⇒ Формализация описания предметной области

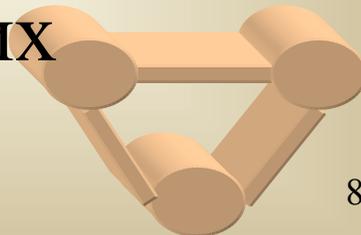


# Классификация DSL



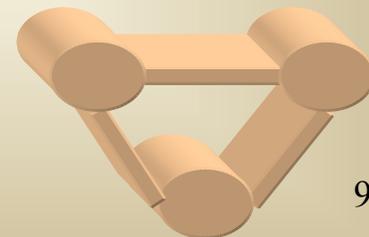
# DSL: Программные технологии (инфраструктура)

- ⇒ Application Programming Interface – группа системных сервисов, ориентированная на решение общих задач
- ⇒ Компонентные технологии – программный модуль со стандартизованным интерфейсом, ориентированный на решение общих задач



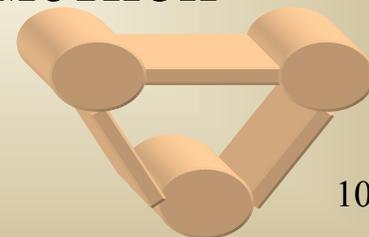
# DSL: Архитектурные решения

- ↗ Архитектурные паттерны – проектные решения, описывающие архитектуру программной системы на основе некоторой концепции
- ↗ GoF-паттерны – проектные решения, описывающие аспекты реализации программной системы для решения некоторых частных задач программирования

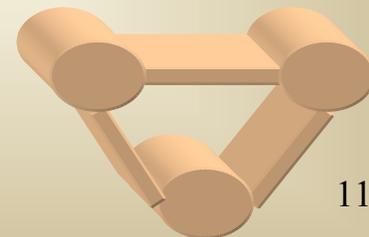
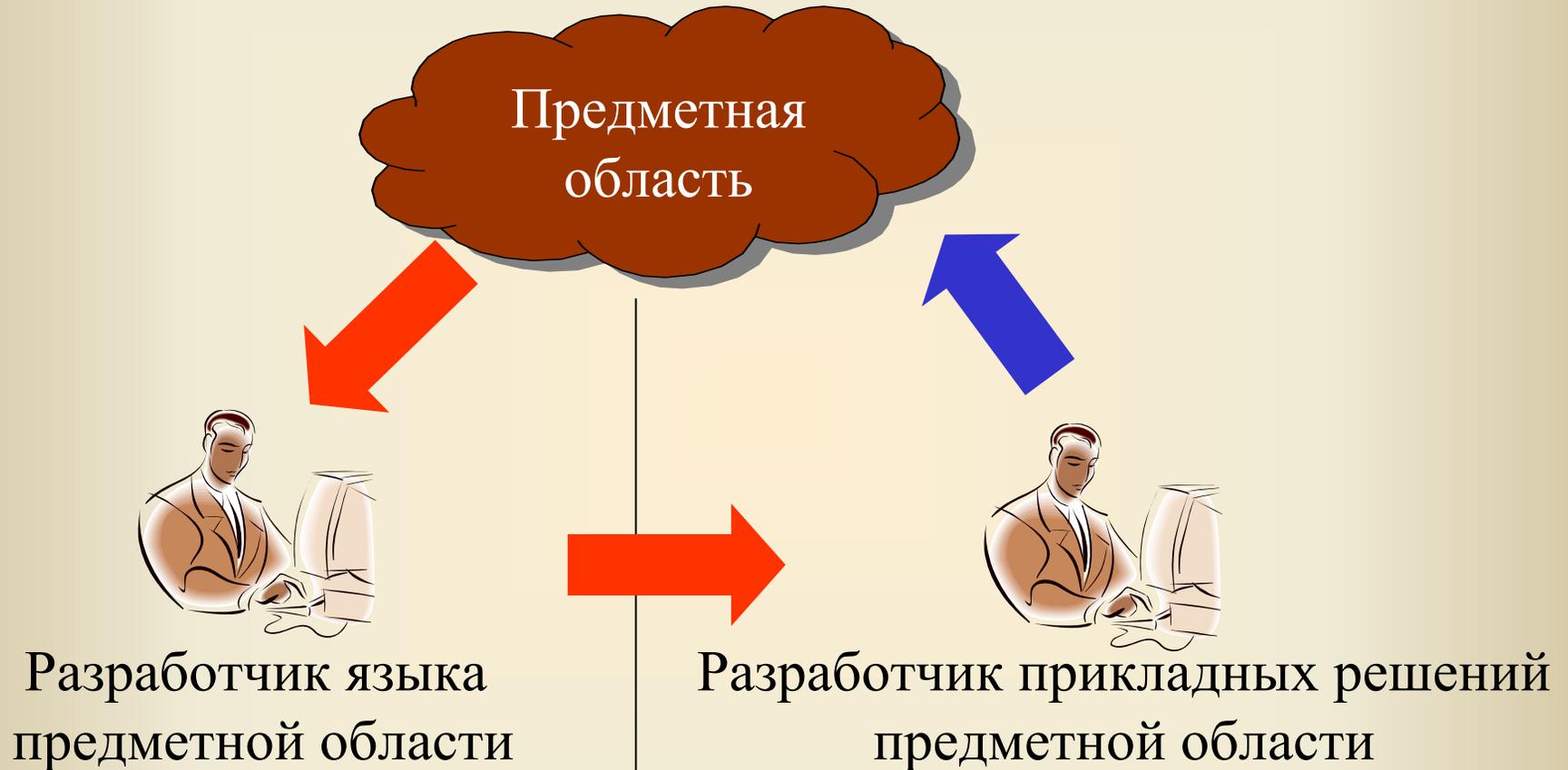


# DSL: предметная область

- ↗ Языки на основе XML – структурированное описание некоторых данных и механизмов преобразований (XSLT, XAML и т.д.)
- ↗ SQL – язык структурированных запросов к СУБД
- ↗ Онтологии - представление какой-либо области знаний или части реального мира, которое используется для семантического анализа текстов
- ↗ Domain Language – описание задач предметной области на специализированном языке



# DSL: группы разработчиков



# Agenda...

↗ Роль DSL в проектировании

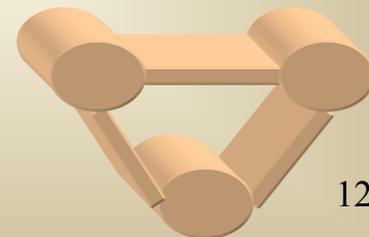
↗ UML vs DSL или метамодель предметной области

↗ UML + Open Use cases = DSL

↗ Практический пример

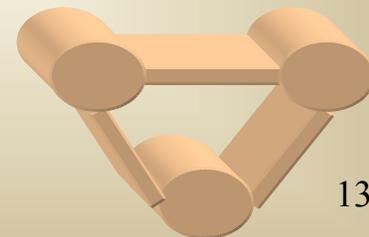
↗ Перспективы

↗ Выводы

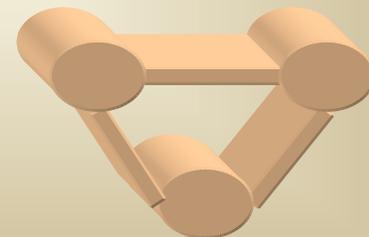
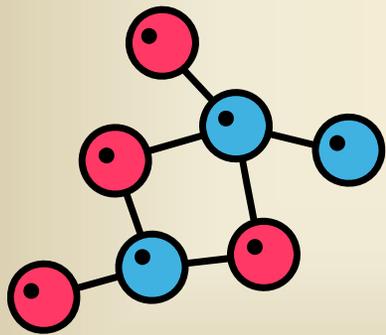


# Почему разработчики не используют UML в полном объеме?

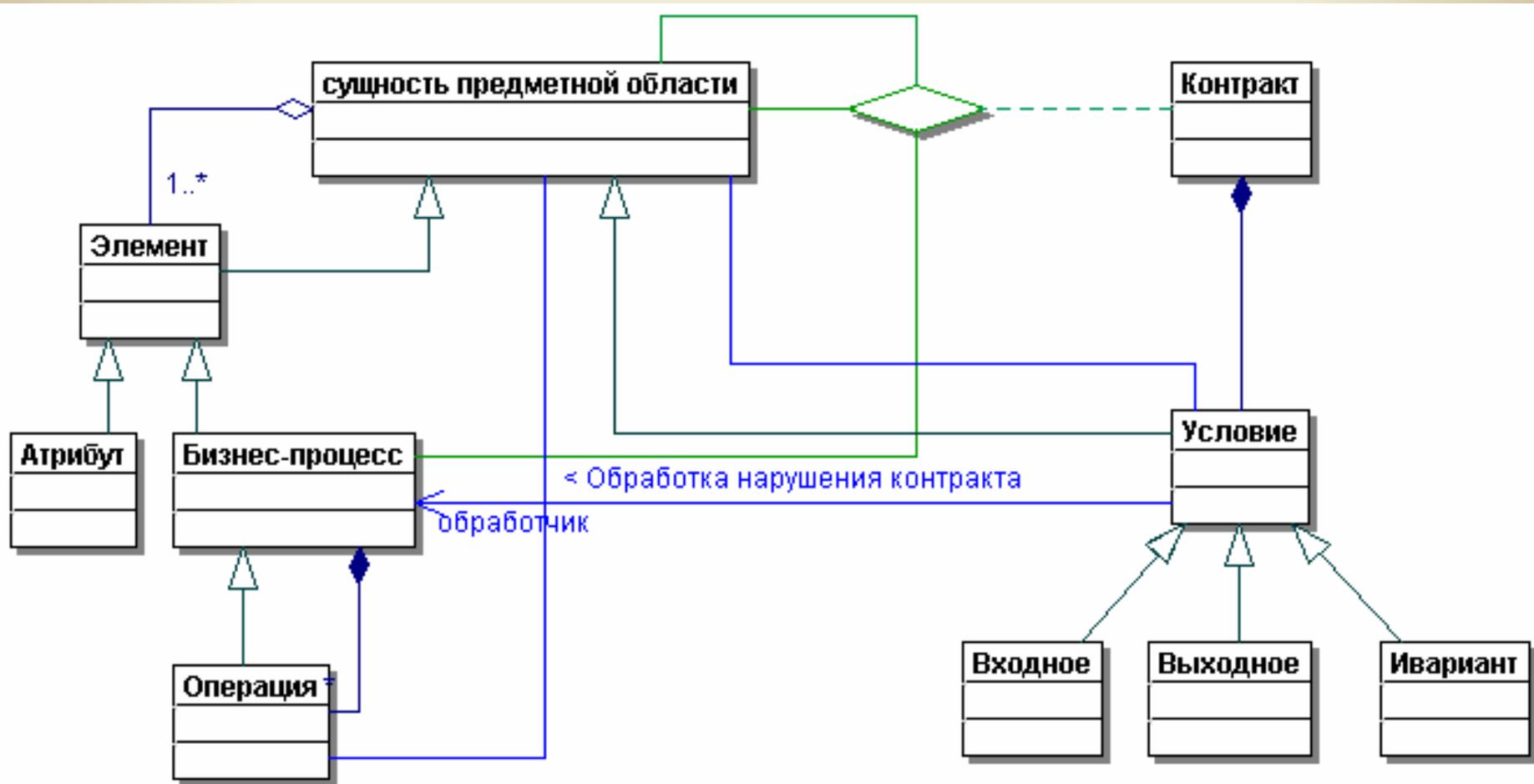
- ⇒ Сложность UML
- ⇒ Возможность неконтролируемого расширения
- ⇒ Сложность проектирования программных систем и их компонент



Основа DSL – предметная область. Можно ли это понятие описать на UML?

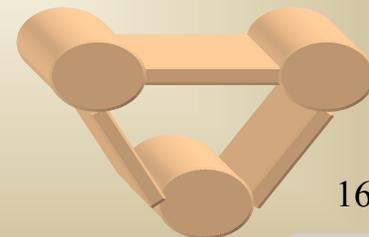


# Метамодель предметной области



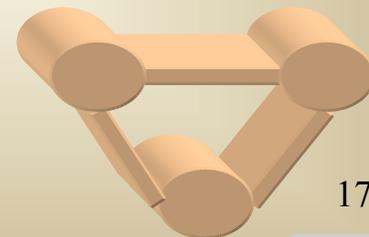
# Сущность предметной области

- ↗ Представляет собой объект хранения информации и (или) процесс обработки информации.
- ↗ Содержит элементы, которые представляют собой атрибуты (характеристики, отличающие одну бизнес-сущность от другой), процессы и операции.
- ↗ Каждая сущность предметной области, может содержать один или несколько бизнес-процессов, который содержит функционал взаимодействия с другими сущностями предметной области.



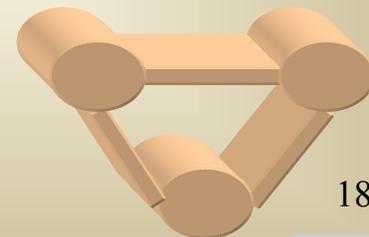
# Атрибут

- ↗ Выражает некоторую характеристику объекта в предметной области
- ↗ Имеет определенное множество значений (Инварианты)
- ↗ Механизм инвариантов обеспечивается контрактом

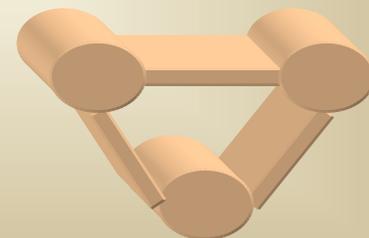


# Бизнес-процесс

- ↗ содержит функционал взаимодействия с другими сущностями предметной области
- ↗ можно представить, согласно определению процесса, как последовательность элементарных операций, которые могут быть тоже процессом
- ↗ процесс может иметь различные метрики (атрибуты), отражающие его состояние, что, по сути, позволяет рассматривать бизнес-процесс как бизнес-сущность исходя из ее определения



Объекты предметной области  
взаимодействуют на основе  
контракта. Если контракт  
нарушается, это не может быть  
безнаказанным...



# Описание взаимодействия на основе контракта

↪ Любое взаимодействие есть контракт между двумя объектами на выполнение некоторой операции

↪ Контракт описывается триплетом Хоара:

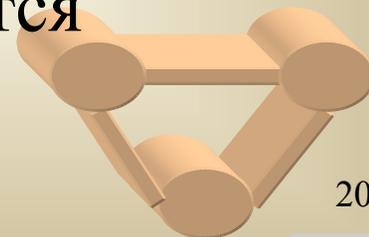
$\{P\} A \{Q\}$ , где

↪  $P$  – исходное множество выполняющихся утверждений, характеризующих состояние свойств различных сущностей

↪  $A$  – это выполняемое действие

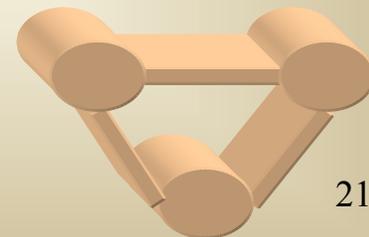
↪  $Q$  – и результирующее выполняющихся утверждений, характеризующих состояние свойств различных сущностей

↪ Средствами UML контракт представляется классом-ассоциацией



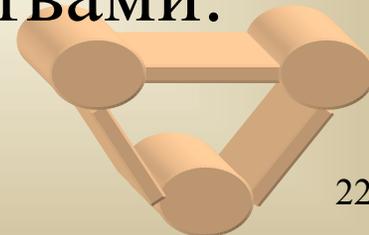
# Agenda...

- ↗ Роль DSL в проектировании
- ↗ UML vs DSL или метамодель предметной области
- ➔ ↗ UML + Open Use cases = DSL
- ↗ Практический пример
- ↗ Перспективы
- ↗ Выводы



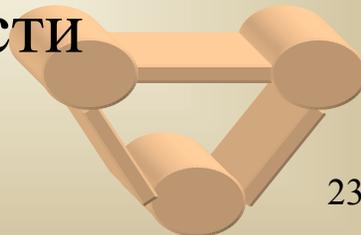
# Шаги разработки DSL

- Определение цели создания языка описания предметной области.
- Выбор средств представления языка описания предметной области.
- Разработка элементов языка описания предметной области.
- Реализация языка описания предметной области инструментальными средствами.



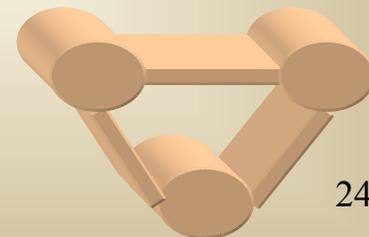
# Определение цели создания языка описания предметной области

- ↗ Описание отдельных алгоритмов, которые не могут быть однозначно определены во время проектирования программной системы
- ↗ Создание инфраструктуры решений, позволяющей на понятном пользователю языке описать решение задачи, например, описать документооборот или технический процесс
- ↗ Описание структурированных данных, для последующего создания средств автоматизированного анализа и обработки массивов информации предметной области



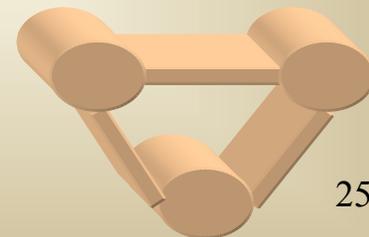
# Выбор средств представления языка описания предметной области

- ↗ Графические, в виде различных диаграмм
- ↗ Текстовые, в виде некоторого языка программирования высокого уровня
- ↗ Текстовые, в виде формализованного формата текста на естественном языке



# Разработка элементов языка описания предметной области

- ↪ Составление словаря предметной области.
- ↪ Определение объектов информационного анализа и обработки.
- ↪ Определение бизнес-процессов предметной области:
  - ↪ Определить общие, а не частные (в конкретной задаче) способы и формы взаимодействия объектов в предметной области.
  - ↪ Построить модель прецедентов предметной области
- ↪ Определение бизнес-процессов, которые зависят от конкретной задачи в рамках предметной области.  
Используем открытые прецеденты //SEC(R)2005
- ↪ Определение условий корректности и непротиворечивости взаимодействия между сущностями предметной области на основе контракта

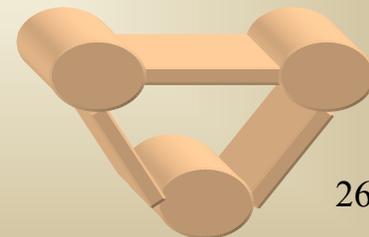


# Реализация языка описания предметной области инструментальными средствами

↗ Microsoft Visual Studio .NET SDK  
(September, 2006)

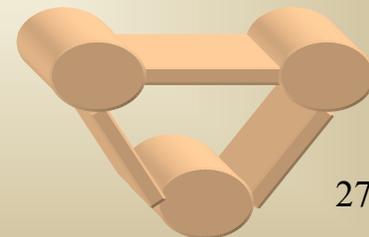
↗ .NET Framework 3.0 (Windows  
Presentation Foundation)

↗ ...



# Agenda...

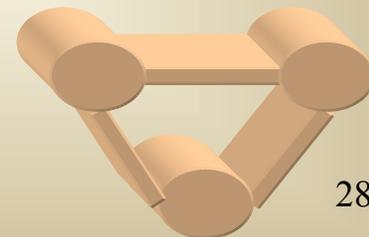
- ↗ Роль DSL в проектировании
- ↗ UML vs DSL или метамодель предметной области
- ↗ UML + Open Use cases = DSL
- ↗ Практический пример
- ↗ Перспективы
- ↗ Выводы





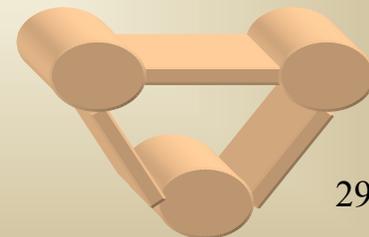
# Постановка задачи

- ↗ В системе организации поддержки учебного процесса TeachWizard 3.0 необходимо предоставить средства описания алгоритмов тестирования
- ↗ Алгоритмы тестирования зависят от методики преподавания и не могут быть однозначно определены на этапе анализа требований

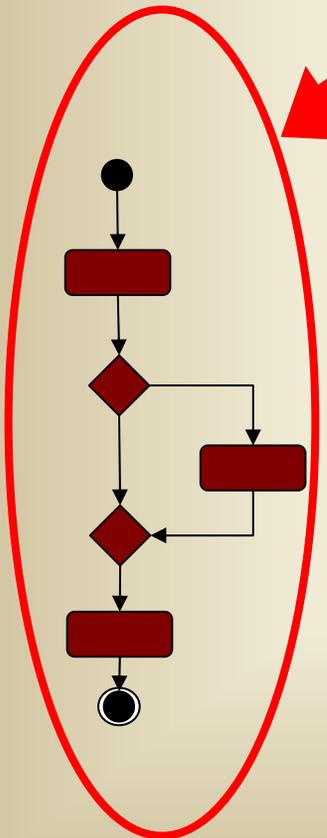


# Шаг 1. Определяем цель создания DSL

⇒ Целью разработки языка является описание алгоритмов тестирования, так как в зависимости от методики предлагаемой преподавателем требуются различные способы предъявления вопросов

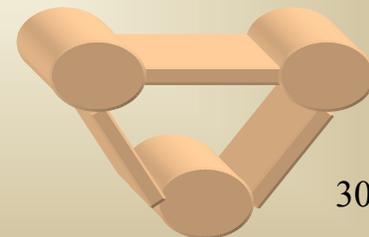


# Шаг 2. Выбор средств представления языка описания предметной области



**Если вопрос.открытый  
То диалог.открыть  
Иначе вопрос.показать**

*Необходимо, чтобы все открытые вопросы отображались в диалоге ввода ответа с учетом регистра*



# Шаг 3. Разработка элементов языка описания предметной области – Словарь предметной области

↗ Вопрос

↗ Тип вопроса

↗ Тест

↗ Ответ

↗ Оценка

↗ Балл

↗ ...

↗ История обучения

↗ Показ вопроса

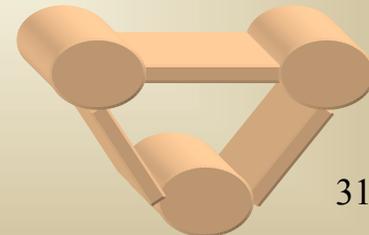
↗ Получение ответа

↗ Анализ ответа

↗ Переход к вопросу

↗ Подсчет баллов

↗ ...



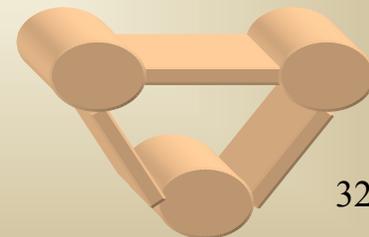
# Шаг 3. Разработка элементов языка описания предметной области – Определение объектов информационного анализа и обработки

↗ Вопрос

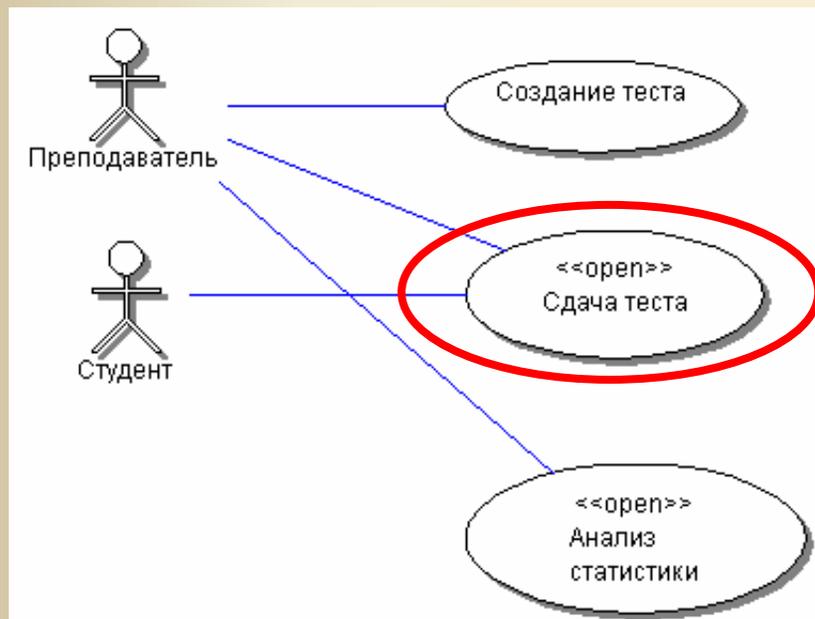
↗ Ответ

↗ Тест

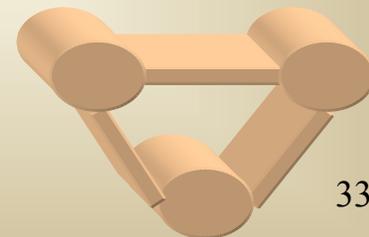
↗ Балл



# Шаг 3. Разработка элементов языка описания предметной области – Определение бизнес-процессов

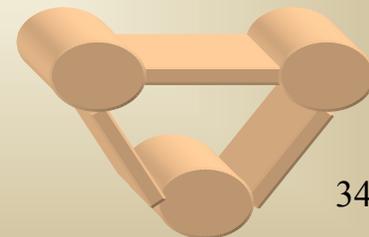


- ⇒ Предъявить вопрос
- ⇒ Получить ответ
- ⇒ Принять ответ
- ⇒ Увеличить Балл
- ⇒ Уменьшить Балл
- ⇒ Верный Ответ?
- ⇒ ...

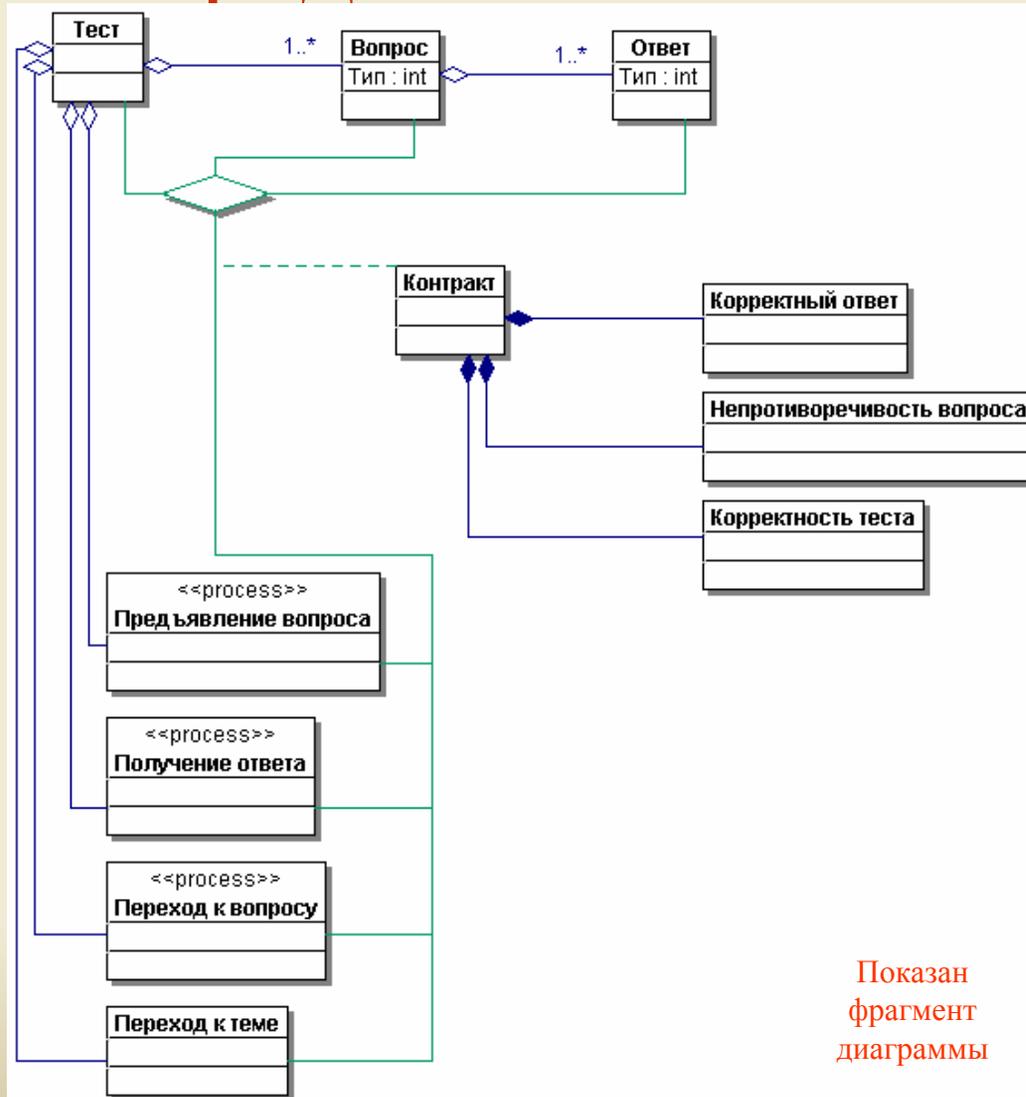


# Шаг 3. Разработка элементов языка описания предметной области – Определение контракта

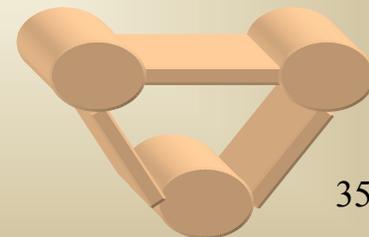
- ↗ Корректный ответ
- ↗ Непротиворечивый вопрос
- ↗ Корректность теста
- ↗ ...



# Проект DSL: Метамодель предметной области

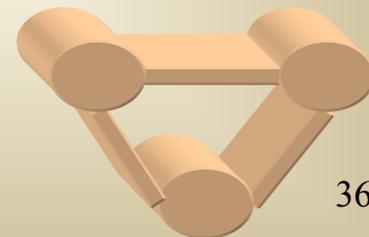


Показан  
фрагмент  
диаграммы



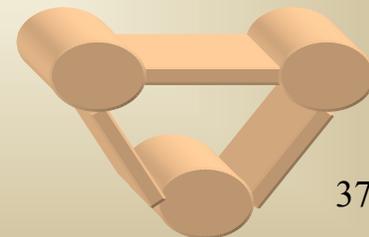
# Шаг 4. Реализация DSL

- ↗ В настоящее время выполняется практическая реализация проекта языка DSL для Teach Wizard 3.0
- ↗ В рамках осуществления проекта планируется разработка курса лекций по проектированию и разработке языков предметной области средствами Microsoft Visual Studio 2005 и .NET Framework 3.0



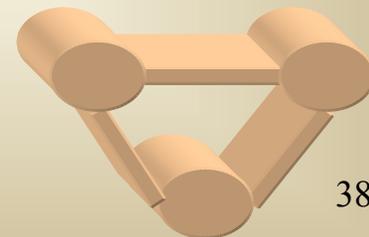
# Agenda...

- ↗ Роль DSL в проектировании
- ↗ UML vs DSL или метамодель предметной области
- ↗ UML + Open Use cases = DSL
- ↗ Практический пример
- ↗ Перспективы
- ↗ Выводы



# Перспективы

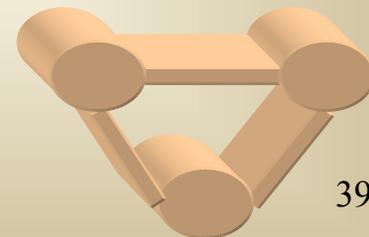
- ⇒ Разработка методики проектирования ПО на основе использования DSL
- ⇒ Формализация проектирования DSL для естественных языков
- ⇒ Создание собственной платформы разработки DSL и исполнения программ, написанных на DSL



# Agenda...

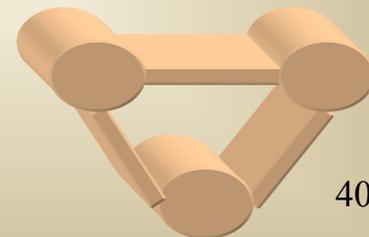
- ↗ Роль DSL в проектировании
- ↗ UML vs DSL или метамодель предметной области
- ↗ UML + Open Use cases = DSL
- ↗ Практический пример
- ↗ Перспективы

 ↗ Выводы



# Выводы

- ↪ С помощью языка UML мы можем проектировать DSL
- ↪ Основу DSL составляет метамодель предметной области
- ↪ Элементы метамодели состояются на основе словаря предметной области и открытых прецедентов
- ↪ DSL, получаемый на основе предлагаемой методики может быть реализован средствами .NET Framework 3.0 (Windows Workflow Foundation)



# Вопросы

