

# Опыт использования Rational Software Architect и языка UML в разработке больших программных систем

Лепихов  
Андрей Валерьевич,  
ЮУрГУ,  
кафедра системного программирования

# Основные термины UML

- *Агрегация* – форма ассоциации, определяющая отношение “часть-целое” между агрегатом (целое) и частями
- *Ассоциация* – структурное отношение, описывающее набор связей между объектами.
- *Актёр* – внесистемный агент, который взаимодействует с системой
- *Вариант использования* – определение набора последовательностей действий, при которых система приносит отдельному актеру полезный и понятный результат
- *Граничный класс* – класс модели анализа, используемый для моделирования взаимодействия между системой и ее актерами, то есть пользователями и внешними системами
- *Класс сущности* – класс модели анализа, используемый для моделирования долгоживущей, часто персистентной информации
- *Пакет* – механизм общего назначения для организации элементов в группы
- *Подсистема* – осмысленная организация элементов в группы
- *Поток событий* – текстовое описание последовательности действий, составляющих вариант использования
- *Стереотип* – Расширение словаря UML, позволяющее создавать новые типы элементов, порождая их от уже существующих. Новые блоки предназначены для решения определенных проблем
- *Управляющий класс* – класс модели анализа, представляющий координацию, последовательность и управление другими объектами, часто используется для инкапсуляции управления варианта использования

# Унифицированный процесс

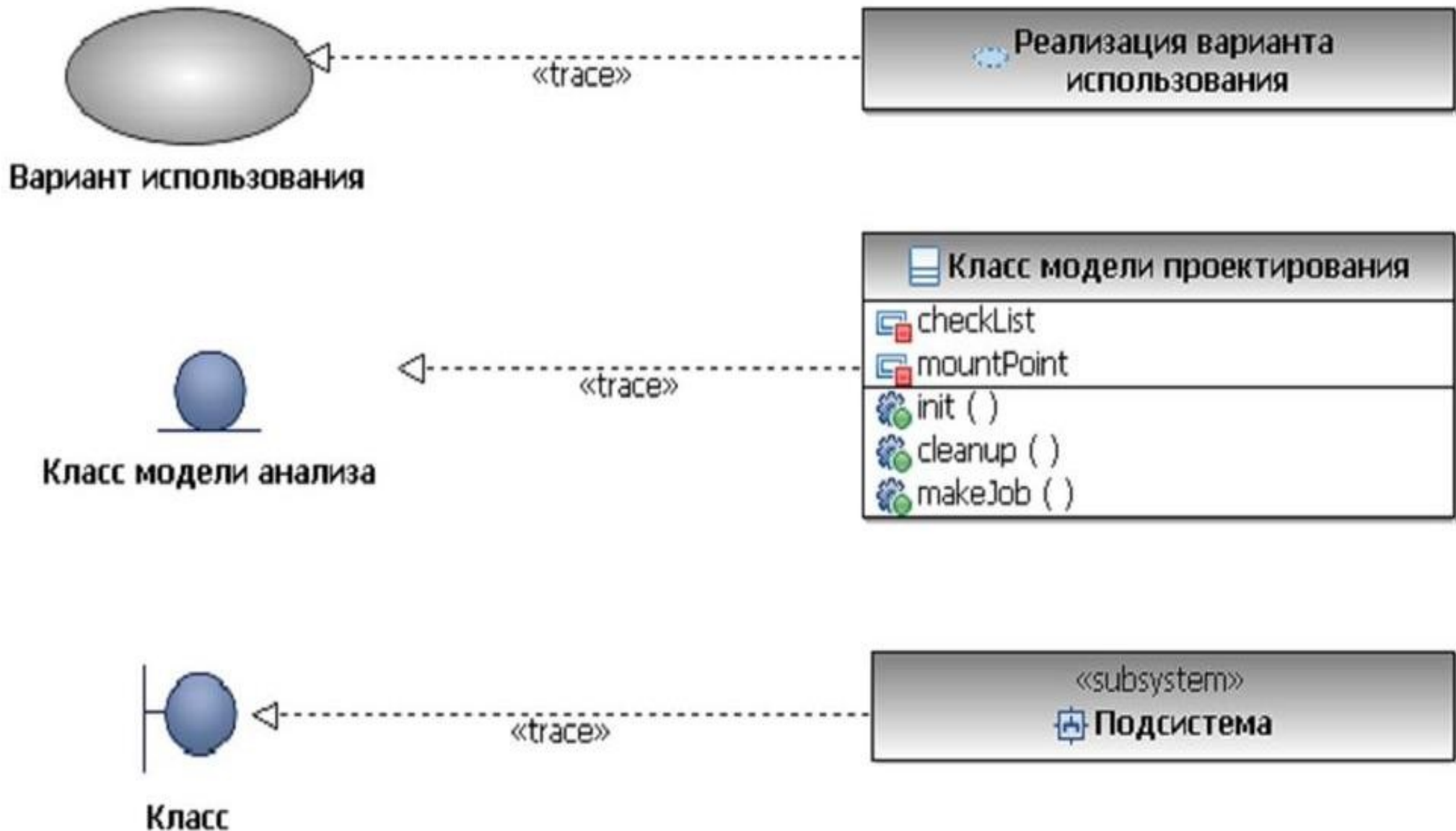
(Rational Unified Process, RUP)

- Управляется вариантами использования
- Ориентирован на архитектуру
- Итеративный и инкрементный

# Основные модели процесса разработки

- Модель вариантов использования
- Модель анализа
- Модель проектирования
- Модель развертывания
- Модель реализации

# Связи между моделями



# Общие элементы шаблонов моделей

Диаграмма *Overview* – обзорная диаграмма по содержанию модели

- Пакет *overviews* со стереотипом «perspective»
- Пакет *building blocks* – содержит стандартный шаблон структуры соответствующей модели и стандартный набор элементов, которые могут быть использованы при разработке в точном соответствии с идеологией Унифицированного процесса

# Роль пакета в модели

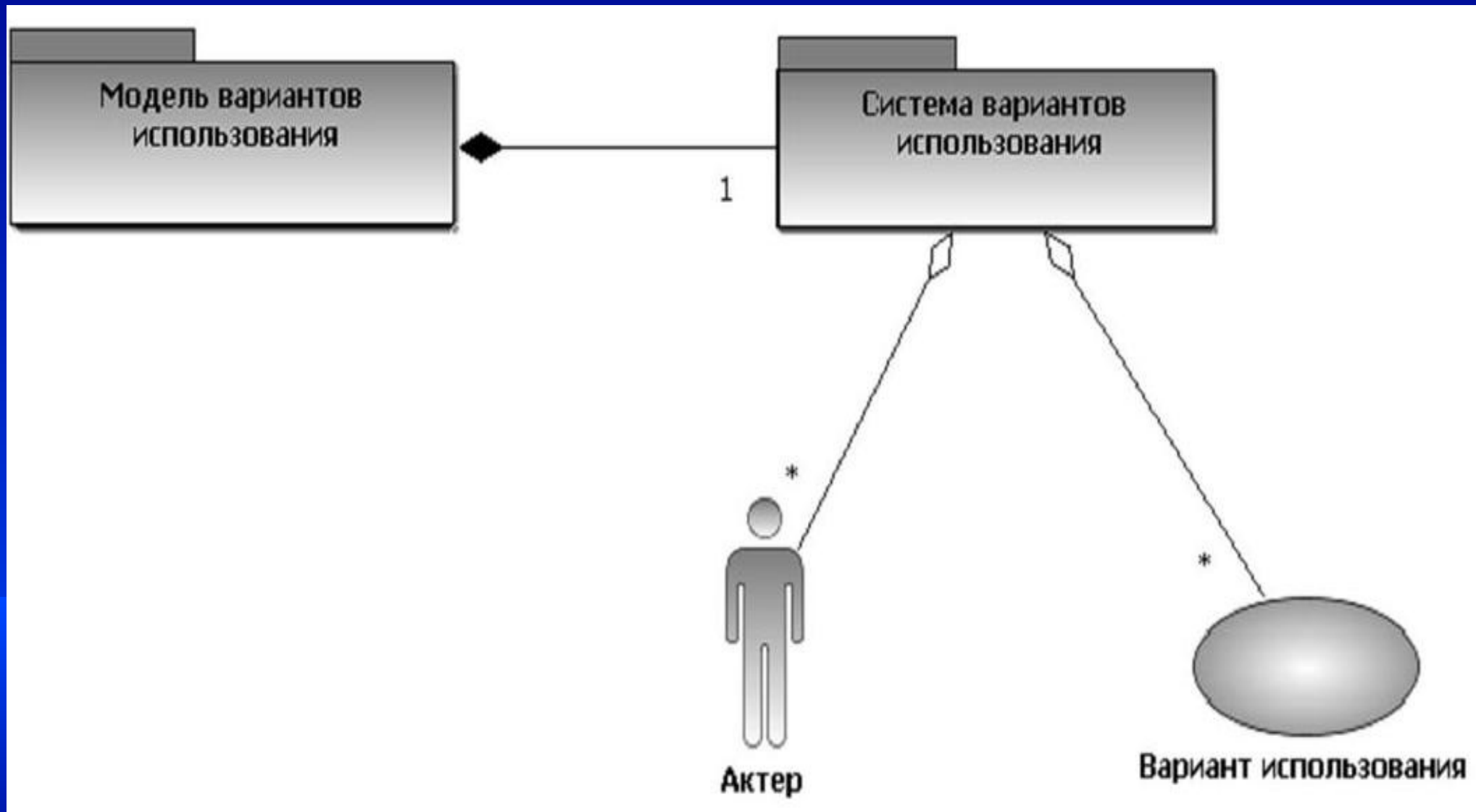
Пакет – основной инструмент структурирования UML-модели. Используется для:

- Разделения внутри модели разных типов информации (интерфейсы, реализации, диаграммы)
- Группировка элементов с целью управления зависимостями.
- Группировка диаграмм, которые представляют собой разные точки зрения на одну и ту же модель.

*Модель – это тоже пакет, представляющий собой верхний уровень иерархии пакетов в модели.*



# Модель вариантов использования





# Содержание модели вариантов использования

- Набор пакетов, структурирующих описание предметной области
- Каждый вариант использования должен содержать описание основного потока событий и альтернативных потоков
- При наличии сложной модели и большом количестве вариантов использования можно структурировать модель, разбивая множество вариантов использования на отдельные функциональные области

*Компания IBM в руководстве к Rational Architect рекомендует снабжать каждый пакет (включая корневой пакет – модель) диаграммой, описывающей его структуру.*

# Критерии качества вариантов ИСПОЛЬЗОВАНИЯ

- Результирующая ценность

Вариант использования должен предоставлять актеру явную пользу

- Завершенность

Вариант использования представляет собой некоторую законченную функциональность, которую инициирует один актер

- Атомарность

Вариант использования выполняется целиком, или не выполняется вообще, без какой-либо интерференции с другими вариантами использования

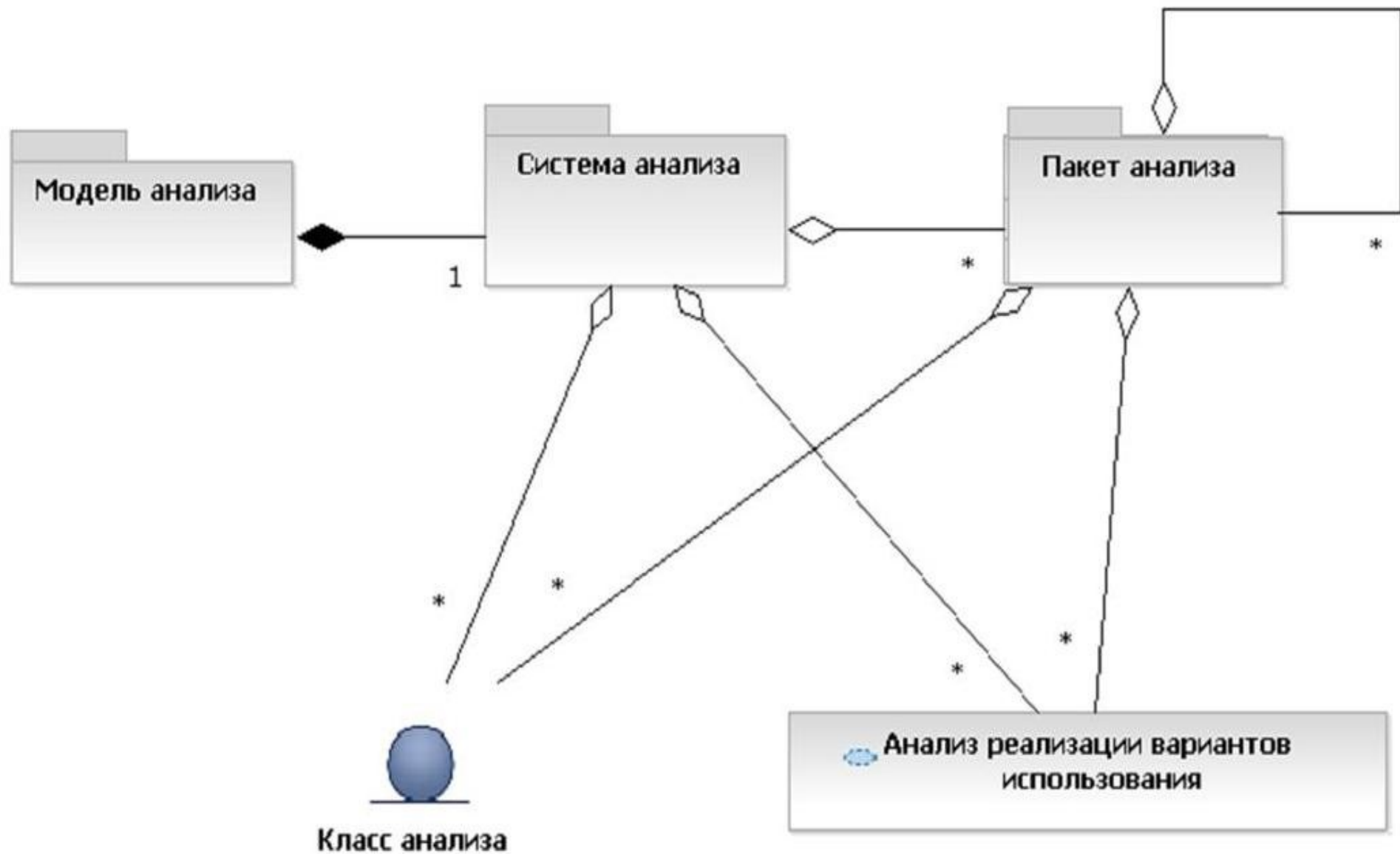
# Поток событий варианта использования

- Название варианта использования
- Краткое описание
- Поток событий:
  - Основной поток событий
  - Альтернативные потоки событий
- Предусловия
- Постусловия

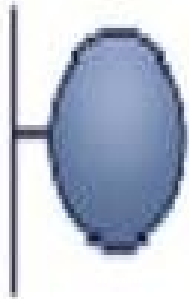
# Распределение вариантов использования по пакетам

- Обозначить подсистемы верхнего уровня в системе
- Распределить ответственность и ресурсы между независимыми командами разработчиков
- Обозначить типы пользователей
- В целях безопасности. Ограничить область доступа пользователей

# Модель анализа



# Классы анализа



Граничный класс

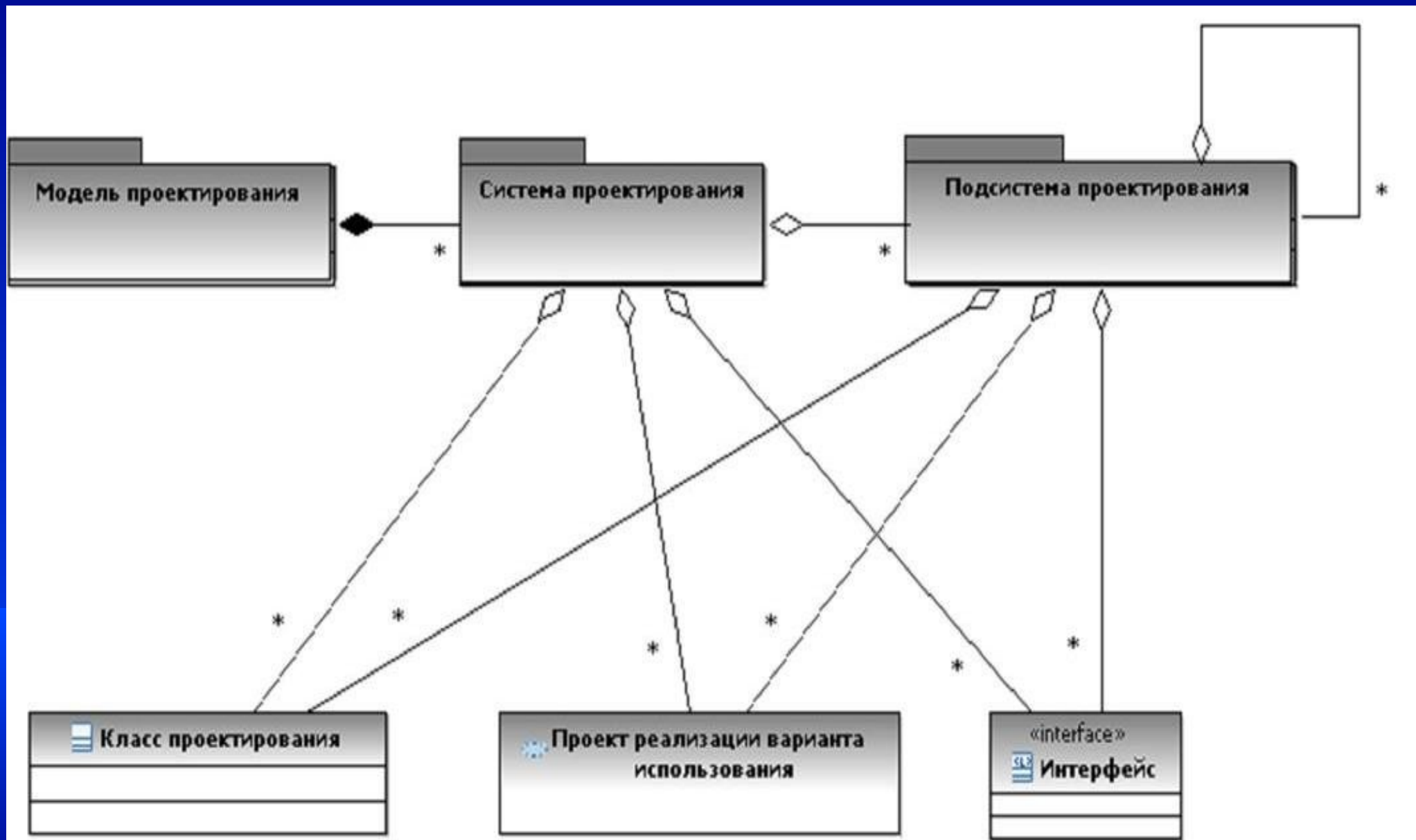


Класс управления



Класс сущности

# Модель проектирования





# Литература

- Кватрани Т., Палистрант Дж. Визуальное моделирование с помощью IBM Rational Software Architect и UML. Пер. с англ. - М.: КУДИЦ-ПРЕСС. - 2007. - 192 с.
- Якобсон А., Буч Г., Рамбо Дж. Унифицированный процесс разработки программного обеспечения. - Спб.: Питер, 2002. - 496 с.
- Ларман К. Применение UML и шаблонов проектирования. 2-е издание. М.: Изд. дом «Вильямс», 2004. - 624 с.
- Буч Г., Рамбо Дж., Якобсон И. Язык UML. Руководство пользователя. 2-е изд. М.: ДМК Пресс, 2007. - 496 с.
- Справка Rational Software Architect. Process Advisor.
- Smith B. White Paper: Model Structure Guidelines For Rational Software Modeler, Rational System Developer, and Rational Software Architect ("Traditional RUP" orientation)
- Брауде Э. Технология разработки программного обеспечения. - Спб.: Питер, 2004. - 655 с.