

2.7.9. Примечания и пакеты

На диаграммах UML можно размещать примечания, содержащие дополнительную информацию об элементах диаграммы. Хотя примечания не влияют на генерируемый код, они помогают разработчикам и другим участникам проекта лучше понять модель.

Для добавления примечания на диаграмму можно использовать два инструмента. К элементу диаграммы можно прикрепить примечание-комментарий (note). Если же требуется прокомментировать целую диаграмму, применяется текстовая область (text box). В частности, в такой области указывается заголовок диаграммы.

На рис. 2.68 подпись «Пограничный класс является примечанием».

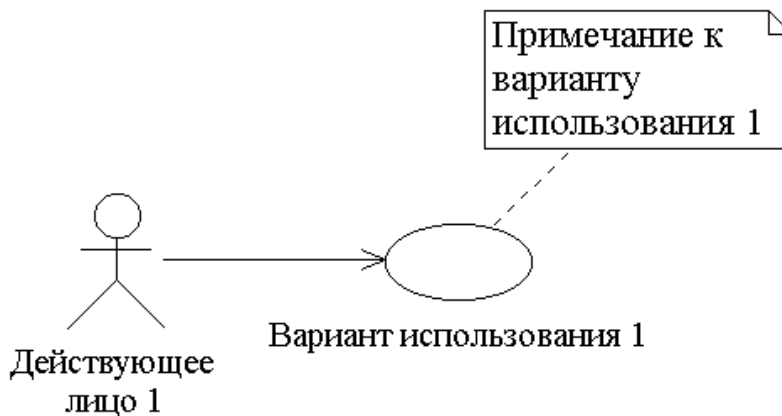


Рис. 2.68. Прикрепление примечания к варианту использования

На языке UML такие элементы, как действующие лица, варианты использования, классы и компоненты, можно сгруппировать в **пакеты** (packages). Это позволяет упорядочить элементы модели. Пакет изображается с помощью нотации показанной на рис. 2.69.

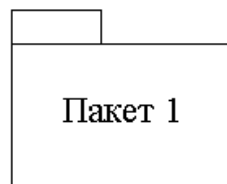


Рис. 2.69. Графическая нотация для изображения пакетов

2.8. Модели и ракурсы

Невозможно охватить все тонкие детали сложной программной системы одним взглядом. Необходимо понять как функциональные, так и структурные свойства системы. Следует уяснить также таксономическую структуру классов, используемые механизмы наследования, индивидуальное поведение объектов и динамическое поведение системы в целом. На рис. 2.70 представлены различные типы моделей, которые считаются главными в объектно-ориентированном подходе. Через них будут выражаться результаты анализа и проектирования, выполненные в рамках любого проекта. Эти модели в совокупности семантически достаточно богаты и универсальны, чтобы разработчик мог выразить все заслуживающие внимания стратегические и тактические решения, которые он должен принять при анализе системы и формировании ее архитектуры. Кроме того, эти модели достаточно полны, чтобы служить техническим проектом реализации на любом объектно-ориентированном языке программирования.



Рис. 2.70. Объектные модели

При принятии решений в анализе и проектировании полезно рассмотреть взаимодействие классов и объектов в двух измерениях: логическом/физическом и статическом/динамическом (рис. 2.70). Оба аспекта

необходимы для определения структуры и поведения объектной системы.

В каждом из двух измерений строят несколько диаграмм, которые представляют систему в различных ракурсах. Диаграммы содержат информацию о ключевых абстракциях системы, их связях и поведении. В установившемся состоянии проекта все диаграммы должны быть согласованы между собой и со всей моделью.

1. Логическая и физическая модели. *Логическое представление* описывает перечень и смысл ключевых абстракций и механизмов, которые формируют предметную область или определяют архитектуру системы.

Физическая модель определяет конкретную программно-аппаратную платформу, на которой реализована система.

При анализе мы должны задать следующие вопросы. Каково требуемое поведение системы? Каковы роли и обязанности объектов по поддержанию этого поведения?

При проектировании необходимо задать следующие вопросы относительно архитектуры системы.

1. Какие существуют классы, и какие есть между ними связи?
2. Какие механизмы регулируют взаимодействие классов?
3. Где должен быть объявлен каждый класс?
4. Как распределить процессы по процессорам и как организовать работу каждого процессора, если требуется обработка нескольких процессов?

Чтобы ответить на эти вопросы, используются следующие диаграммы: Диаграммы Вариантов Исползования; Кооперативные диаграммы; Диаграммы Классов; Диаграммы Компонентов; Диаграммы Размещения.

2. Статическая и динамическая модели. Перечисленные в предыдущем пункте типы диаграмм являются в большей части *статическими*. Но практически во всех системах происходят события: объекты рождаются и уничтожаются, посылают друг другу сообщения (причем в определенном порядке), внешние события вызывают операции объектов. Описание *динамических* событий на статическом носителе, например, на листе бумаги, будет трудной задачей, но эта же трудность встречается фактически во всех областях науки. В объектно-ориентированном проектировании мы отражаем динамическую семантику двумя дополнительными диаграммами: Диаграммы Последовательности; Диаграммы Состояний. Каждый класс может иметь собственную Диаграмму Состояний, которая показывает, как объект класса переходит из состояния в состояние под воздействием событий. На Диаграмме Последовательности можно показать порядок передачи сообщений.