

Анализ и проектирование на UML

Направления подготовки
«Информационные системы и технологии»

Максим Валерьевич Хлопотов,
старший преподаватель кафедры ИС

11.02.2014

Темы лекционных занятий

1. Введение в UML
2. Моделирование использования
3. Моделирование структуры
4. Моделирование поведения
5. Дисциплина моделирования
6. Примеры моделей на UML

Этапы выполнения практических работ

Подготовительный этап. Выбор инструментов

1 этап. Анализ предметной области. Моделирования
использования (л.р. 1)

2 этап. Анализ требований. Моделирование поведения (л.р. 2)

3 этап. Архитектура системы. Моделирование структуры
(лабораторная работа 3)

4 этап. Реализация прототипа.

5 этап. Приёмо-сдаточные испытания (лабораторная работа 4)

Рекомендуемая литература

1. Новиков Ф. А., Иванов Д. Ю.
Моделирование на UML. Теория,
практика, видеокурс
2. Буч Г., Рамбо Д., Якобсон И. Введение
в UML от создателей языка

Введение в UML

UML

(Unified Modeling Language) –

унифицированный

язык

моделирования

UML — это язык

Язык — это знаковая система для хранения и передачи информации.

UML формальный искусственный язык.

Авторы — Гради Буч, Ивар Якобсон и Джеймс Рамбо.



Grady Booch
Грэди Буч



James Rumbaugh
Джеймс Рамбо



Ivar Jacobson
Айвар Якобсон

UML — это язык моделирования

- Моделирование – создание моделей.
- UML имеет отношение прежде всего и главным образом к созданию и применению компьютерных программ.
- В отношении разработки программного обеспечения так сложилось, что результаты фаз анализа и проектирования, оформленные средствами определенного языка, принято называть моделью.

UML — это язык моделирования

- Деятельность по составлению моделей естественно назвать моделированием. Именно в этом смысле UML является языком моделирования.
- Модель UML — это, прежде всего, основной артефакт фазы ***проектирования***.

UML — это унифицированный язык моделирования

- UML является отнюдь не первым языком моделирования.
- К моменту его появления (т.е. к 1995 году) насчитывались десятки других, различающихся системой обозначений, степенью универсальности, способами применения и т. д. Ни один из методов не дотягивал до уровня индустриального стандарта.

UML — это унифицированный язык моделирования

- Если попытаться проследить историю возникновения и развития элементов UML, то пришлось бы назвать сотни имен и десятки организаций.
- Авторы UML при поддержке и содействии всей международной программистской общественности смогли свести воедино (*унифицировать*) большую часть того, что было известно и до них.

UML

(Unified Modeling Language) –

унифицированный

язык

моделирования

Назначение UML

UML — это графический язык моделирования общего назначения, предназначенный для спецификации, визуализации, проектирования и документирования всех артефактов, создаваемых при разработке программных систем.

Назначение UML

Основное назначение UML — предоставить, с одной стороны, достаточно формальное, с другой стороны, достаточно удобное, и, с третьей стороны, достаточно универсальное средство, позволяющее до некоторой степени снизить риск расхождений в толковании спецификаций.

Назначение UML

В типичных случаях в процессе разработки приложений участвуют по меньшей мере два действующих лица: заказчик и разработчик.

Из-за того, что действующих лиц двое, очень многое зависит от степени их взаимопонимания.

Назначение UML

Одним из ключевых этапов разработки приложения является определение того, каким требованиям должно удовлетворять разрабатываемое приложение.

В результате этого этапа появляется формальный или неформальный документ (*артефакт*): постановка задачи, требования, техническое задание, внешние спецификации и др.

Аналогичные по назначению артефакты появляются и на других этапах разработки: функциональные спецификации, архитектура приложения и др.

Мы будем все такие артефакты называть спецификациями.

Спецификация — это декларативное описание того, как нечто устроено или работает.

Назначение UML

Необходимо принимать во внимание (как минимум!) три толкования спецификаций.

То, которое имеет в виду действующее лицо, являющееся источником спецификации (например, заказчик).

То, которое имеет в виду действующее лицо, являющееся потребителем спецификации (например, разработчик).

То, которое объективно обусловлено природой специфицируемого объекта.

Эти три трактовки спецификаций могут не совпадать, и, к сожалению, как показывает практика, сплошь и рядом не совпадают, причем значительно.

Назначение UML

Основное назначение UML — предоставить, с одной стороны, достаточно формальное, с другой стороны, достаточно удобное, и, с третьей стороны, достаточно универсальное средство, позволяющее до некоторой степени снизить риск расхождений в толковании спецификаций.

Назначение UML

Второе по важности назначение UML состоит в том, чтобы служить адекватным средством коммуникации между людьми.

Разумеется, наглядность визуализации моделей UML имеет значение, только если они должны составляться или восприниматься человеком — это назначение UML не имеет отношения к компьютерам.

Назначение UML

Модели UML допускают представление в форме картинок, причем эти картинки наглядны, интуитивно понятны, практически однозначно интерпретируются и легко составляются.



Назначение UML

UML предназначен не только для описания абстрактных моделей приложений, но и для непосредственного манипулирования артефактами, входящими в состав этих приложений, в том числе такими, как программный код.

Назначение UML

Одним из назначений UML является, например, создание таких моделей, для которых возможна автоматическая генерация программного кода (фрагментов кода) соответствующих приложений.

Автоматическое (или автоматизированное) проектирование и конструирование приложений по спецификациям дело трудное, но не безнадежное. Инструменты, поддерживающие UML, все время совершенствуются, так что в перспективе третье предназначение UML может выйти и на первое место.

Назначение UML

Наконец, четвёртое назначение – документирование.

Модели UML являются документами, которые можно использовать самыми разными способами, начиная с печати картинок и заканчивая автоматической генерацией человекочитаемых текстовых описаний.

В последних версиях UML с целью достижения более полного соответствия этому назначению сделано довольно много.

Способы использования UML

(сортировка по важности)

- Рисование картинок
- Обмен информацией
- Спецификация систем
- Повторное использование архитектурных решений
- Генерация кода
- Имитационное моделирование.
- Верификация моделей.

Стандарт UML

Искусственный язык, претендующий на массовое использование, должен быть описан так, чтобы притягивать, а не отпугивать потенциальных пользователей.

Весь текст описания UML каждой версии находится в свободно распространяемых документах, доступных по адресу <http://www.omg.org>. (Более 1000 страниц текста).

Последние версии:

2.4.1 (август 2011)

<http://www.omg.org/spec/UML/2.4.1/>

2.4 (март 2011) <http://www.omg.org/spec/UML/2.4>

2.3 (май 2010) <http://www.omg.org/spec/UML/2.3>

Рекомендуемая литература

1. Новиков Ф. А., Иванов Д. Ю.
Моделирование на UML. Теория,
практика, видеокурс
2. Буч Г., Рамбо Д., Якобсон И. Введение
в UML от создателей языка

Стандарт UML

Чтобы подчеркнуть, что UML язык графический, авторы называют правила записи (рисования) моделей не синтаксисом, а *нотацией*.

Типов элементов нотации четыре:

- фигуры;
- линии;
- значки;
- тексты.

Стандарт UML

Нотация UML довольно свободная: рисовать можно как угодно, лишь бы не возникало недоразумений. Поставщики инструментов, поддерживающих UML пользуются этой свободой кто во что горазд.

В качестве инструмента рисования диаграмм UML можно использовать различные приложения.

Например, Sun Java Studio Enterprise, Visio Professional.

В этой презентации использована программа: Visual Paradigm for UML 8.2

Модель UML

Модель UML — это конечное множество сущностей и отношений между ними.

Рассматривая модель UML с наиболее общих позиций, можно сказать, что это граф (точнее, нагруженный мульти-псевдо-гипер-орграф), в котором вершины и ребра нагружены дополнительной информацией и могут иметь сложную внутреннюю структуру. Вершины этого графа называются *сущностями*, а ребра — *отношениями*.

Сущности

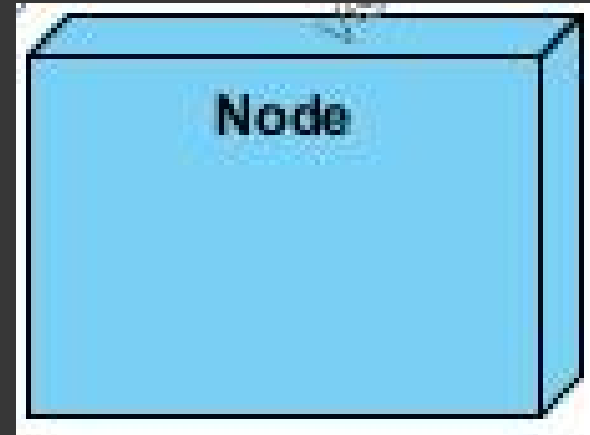
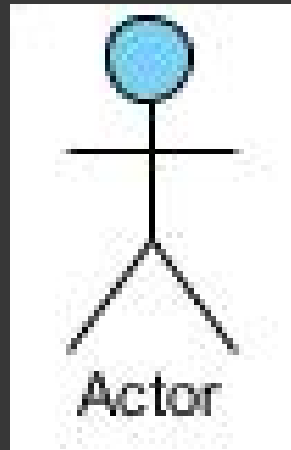
Для удобства обзора сущности в UML можно подразделить на четыре группы:

- *структурные;*
- *поведенческие;*
- *группирующие;*
- *аннотационные.*

Структурные сущности

- Класс — описание множества объектов с общими атрибутами и операциями.
- Интерфейс — множество операций, которое определяет набор услуг (службу), предоставляемых классом или компонентом.
- Действующее лицо — сущность, находящаяся вне моделируемой системы и непосредственно взаимодействующая с ней.
- Вариант использования — описание последовательности производимых системой действий, доставляющей значимый для некоторого действующего лица результат.
- Компонент — физически заменяемый артефакт, реализующий некоторый набор интерфейсов.
- Узел — физический вычислительный ресурс.

Структурные сущности



Поведенческие сущности

Состояние — период в жизненном цикле объекта, в котором объект удовлетворяет некоторому условию, выполняет деятельность или ожидает события.

Деятельность — состояние, в котором выполняется работа, а не просто пассивно ожидается наступление события.



Группирующая сущность

Пакет — группа элементов модели (в том числе пакетов).



Аннотационная сущность

Примечание



Отношения

В UML используются четыре основных типа отношений:

- зависимость;
- ассоциация;
- обобщение;
- реализация.

Отношения

Зависимость — это наиболее общий тип отношения между двумя сущностями. Отношение зависимости указывает на то, что изменение независимой сущности каким-то образом влияет на зависимую сущность. Графически отношение зависимости изображается в виде пунктирной стрелки, направленной от независимой сущности к зависимой.

Ассоциация — это наиболее часто используемый тип отношения между сущностями. Отношение ассоциации имеет место, если одна сущность непосредственно связана с другой (или с другими — ассоциация может быть не только бинарной). Графически ассоциация изображается в виде сплошной линии с различными дополнениями, соединяющей связанные сущности.

Отношения

Обобщение — это отношение между двумя сущностями, одна из которых является частным (специализированным) случаем другой. Графически обобщение изображается в виде сплошной стрелки с треугольником на конце, направленной от частного к общему. Отношение наследования между классами в объектно-ориентированных языках программирования является типичным примером обобщения.

Отношение *реализации* указывает, что одна сущность является реализацией другой. Например, класс является реализацией интерфейса. Графически реализация изображается в виде пунктирной стрелки с треугольником на конце, направленной от реализующей сущности к реализуемой.

Диаграммы UML

Диаграммы UML – основная накладываемая на модель структура, которая облегчает создание и использование модели.

Диаграмма — это графическое представление некоторой части графа модели.

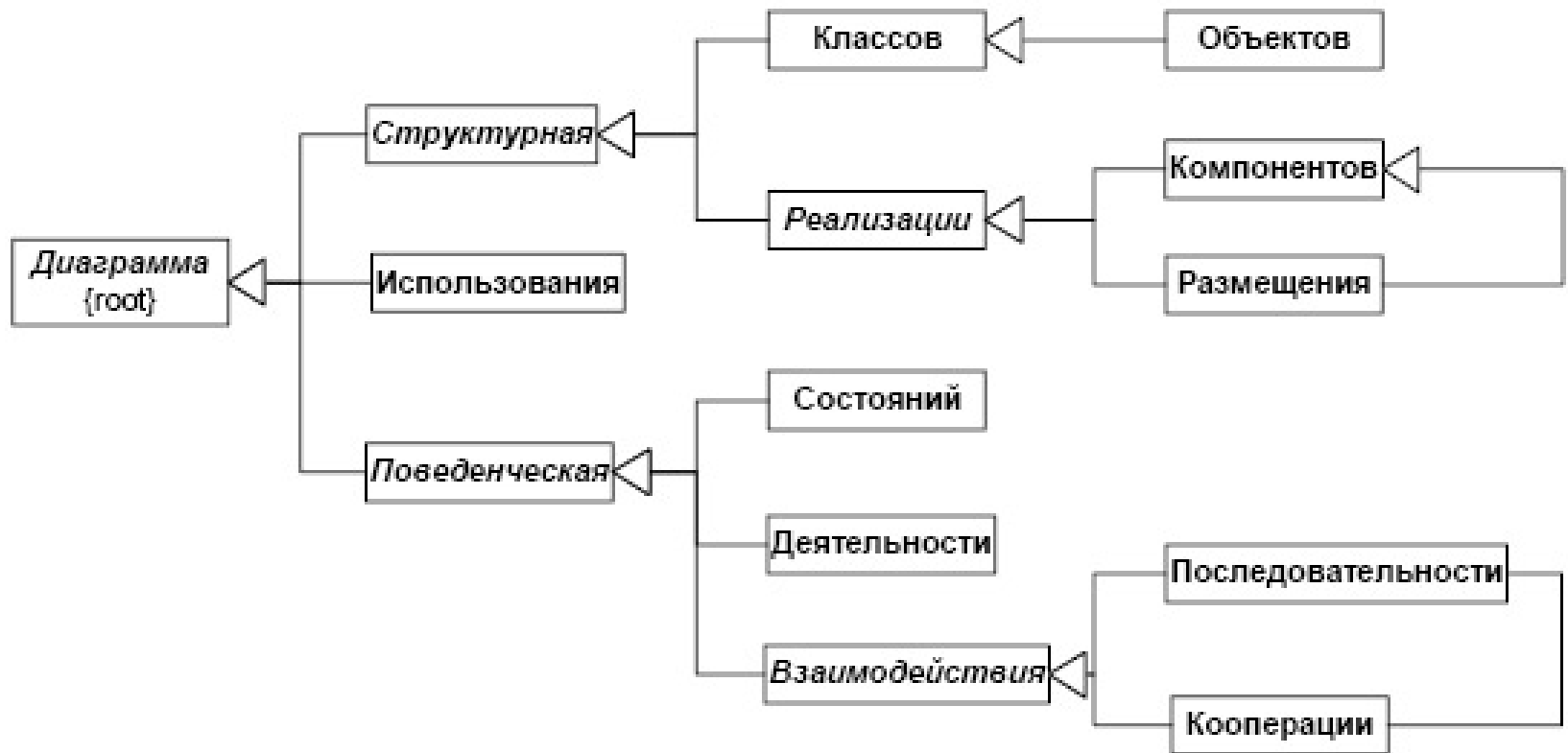
Авторы UML определили набор рекомендуемых к использованию типов диаграмм, которые получили название *канонических* типов диаграмм.

Диаграммы UML

В UML 1.x всего определено 9 канонических типов диаграмм.

- Диаграмма использования
- Диаграмма классов
- Диаграмма объектов
- Диаграмма состояний
- Диаграмма деятельности
- Диаграмма последовательности
- Диаграмма кооперации
- Диаграмма компонентов
- Диаграмма размещения

Иерархия диаграмм UML

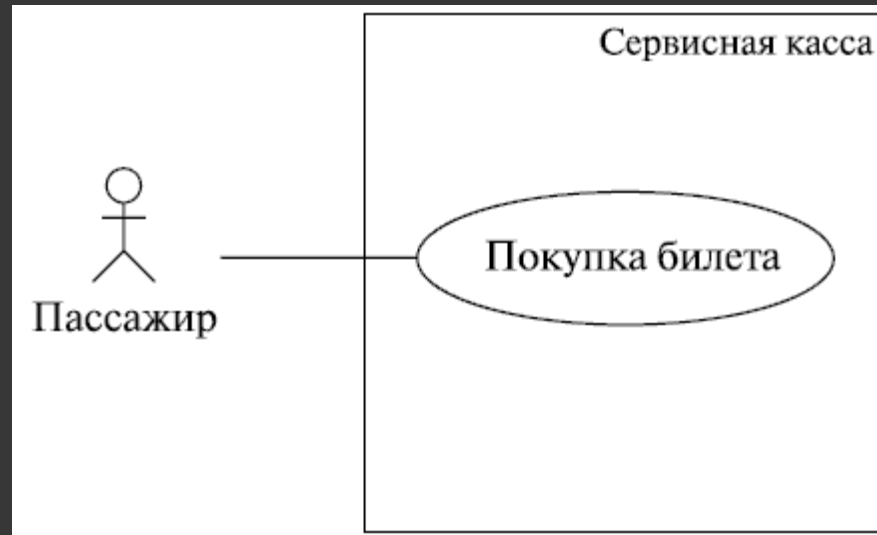


Диаграммы UML

Диаграмма использования — это наиболее общее представление функционального назначения системы. Диаграмма использования призвана ответить на главный вопрос моделирования: что делает система во внешнем мире?

Пример

действующее лицо (эктор), ассоциация, вариант использования (прецедент), рамки системы



Диаграммы UML

Диаграмма классов — основной способ описания структуры системы. Это не удивительно, поскольку UML сильно объектно-ориентированный язык, и классы являются основным "строительным материалом" системы.

Пример

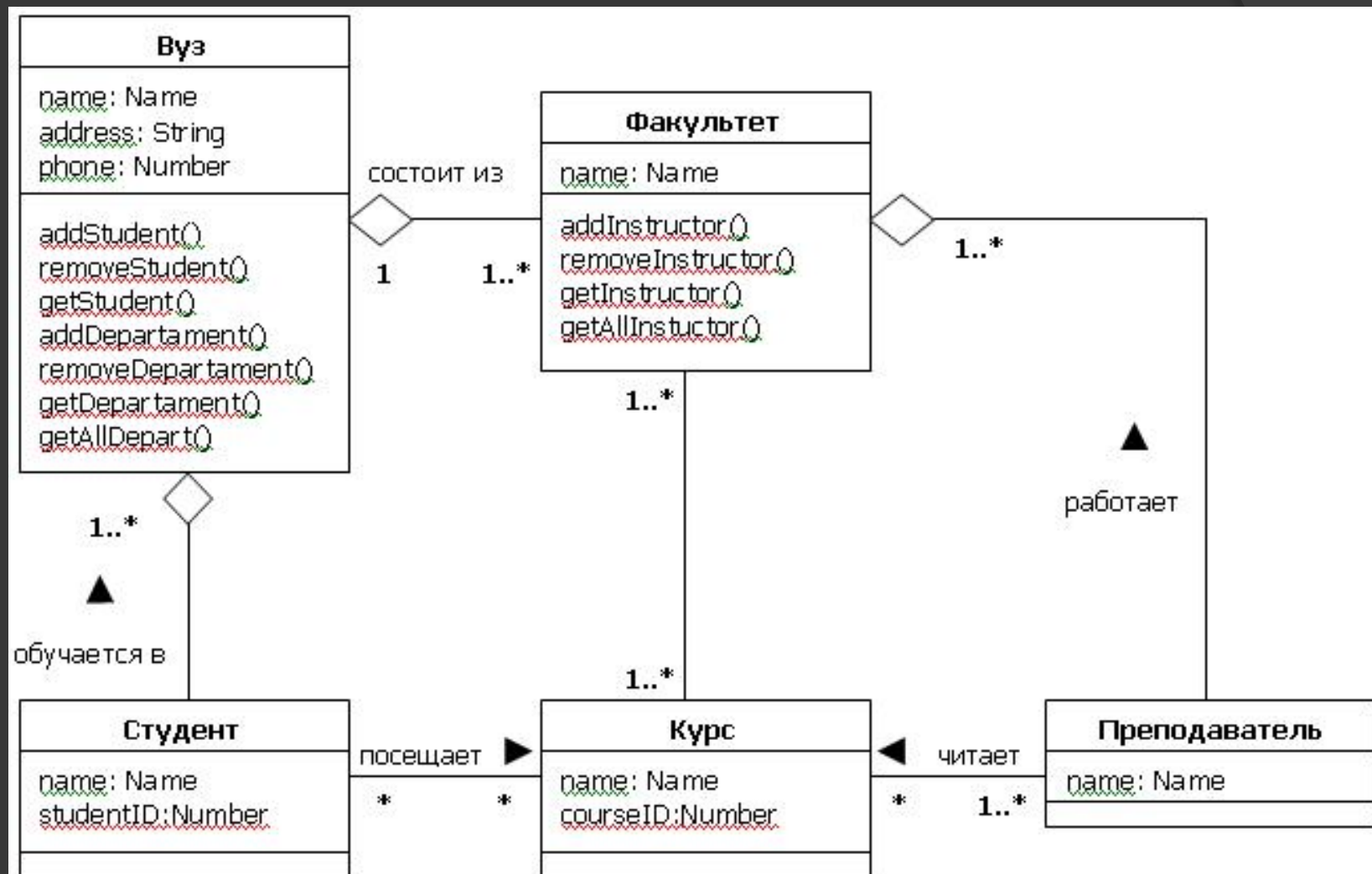


РИС 3.9

Диаграммы UML

Диаграмма объектов — это частный случай диаграммы классов. Диаграммы объектов имеют вспомогательный характер — по сути это примеры, показывающие, какие имеются объекты и связи между ними в некоторый конкретный момент функционирования системы.

Диаграммы UML

Диаграмма состояний — это основной способ детального описания поведения в UML. В сущности, диаграммы состояний представляют собой граф состояний и переходов конечного автомата, нагруженный множеством дополнительных деталей и подробностей.

Пример

(изменение состояний банкомата при проверке ПИН-кода)



Диаграммы UML

Диаграмма деятельности — это, фактически, блок-схема алгоритма, в которой модернизированы обозначения, а семантика согласована с современным объектно-ориентированным подходом.

Пример

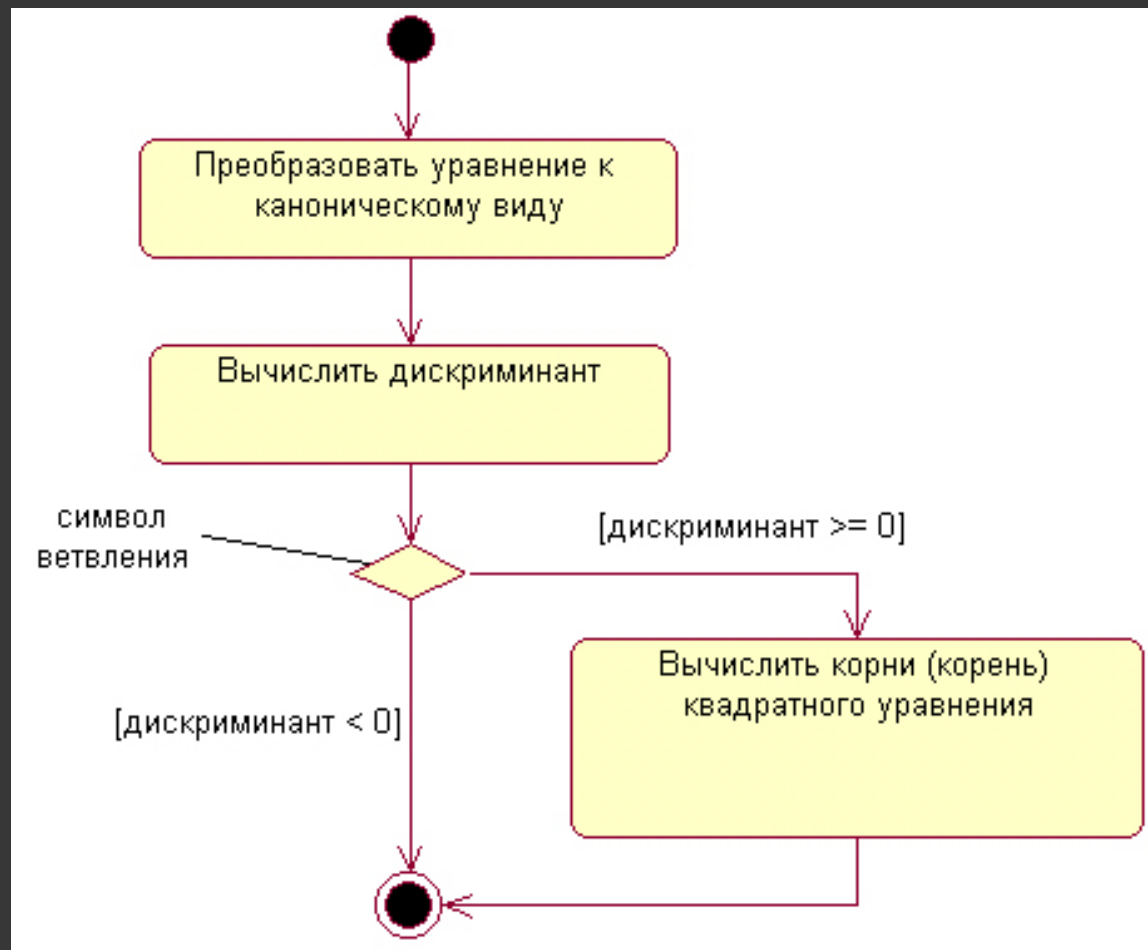
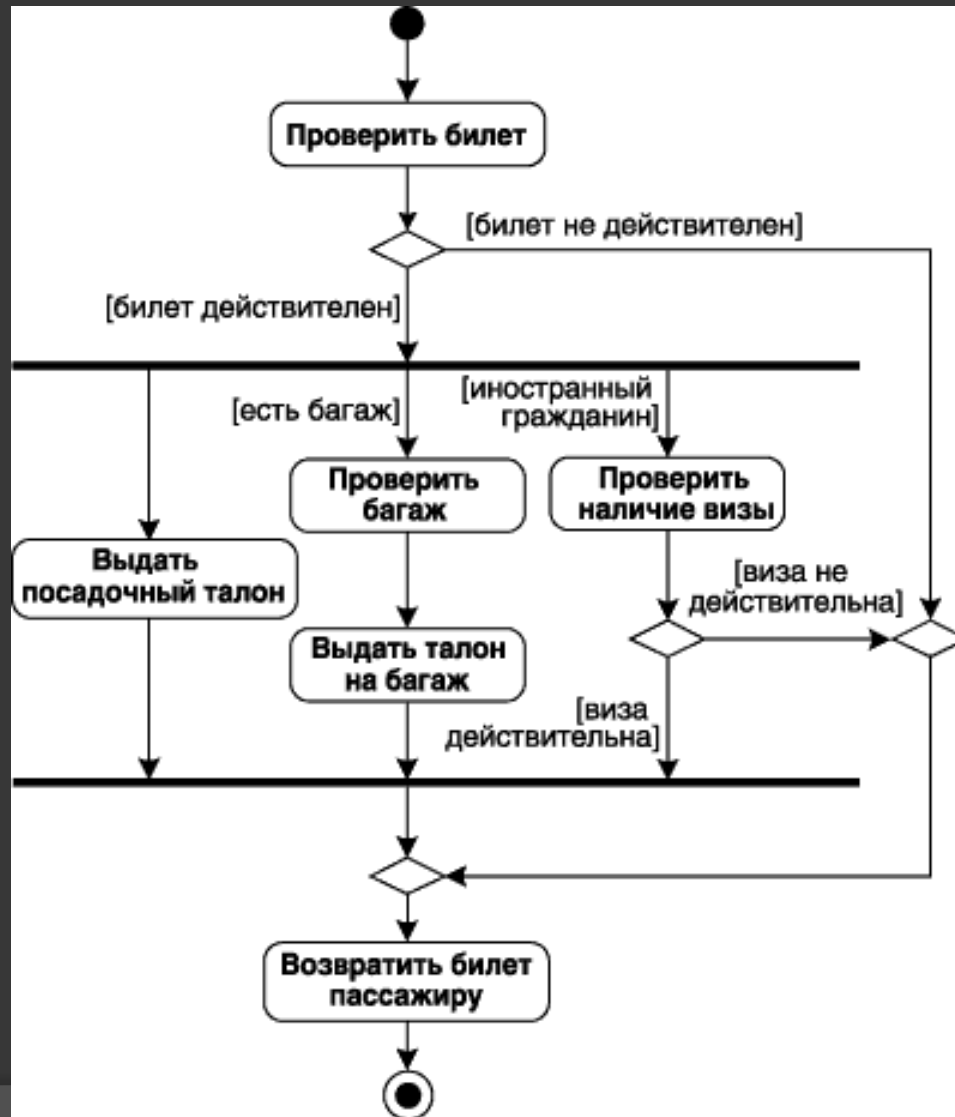


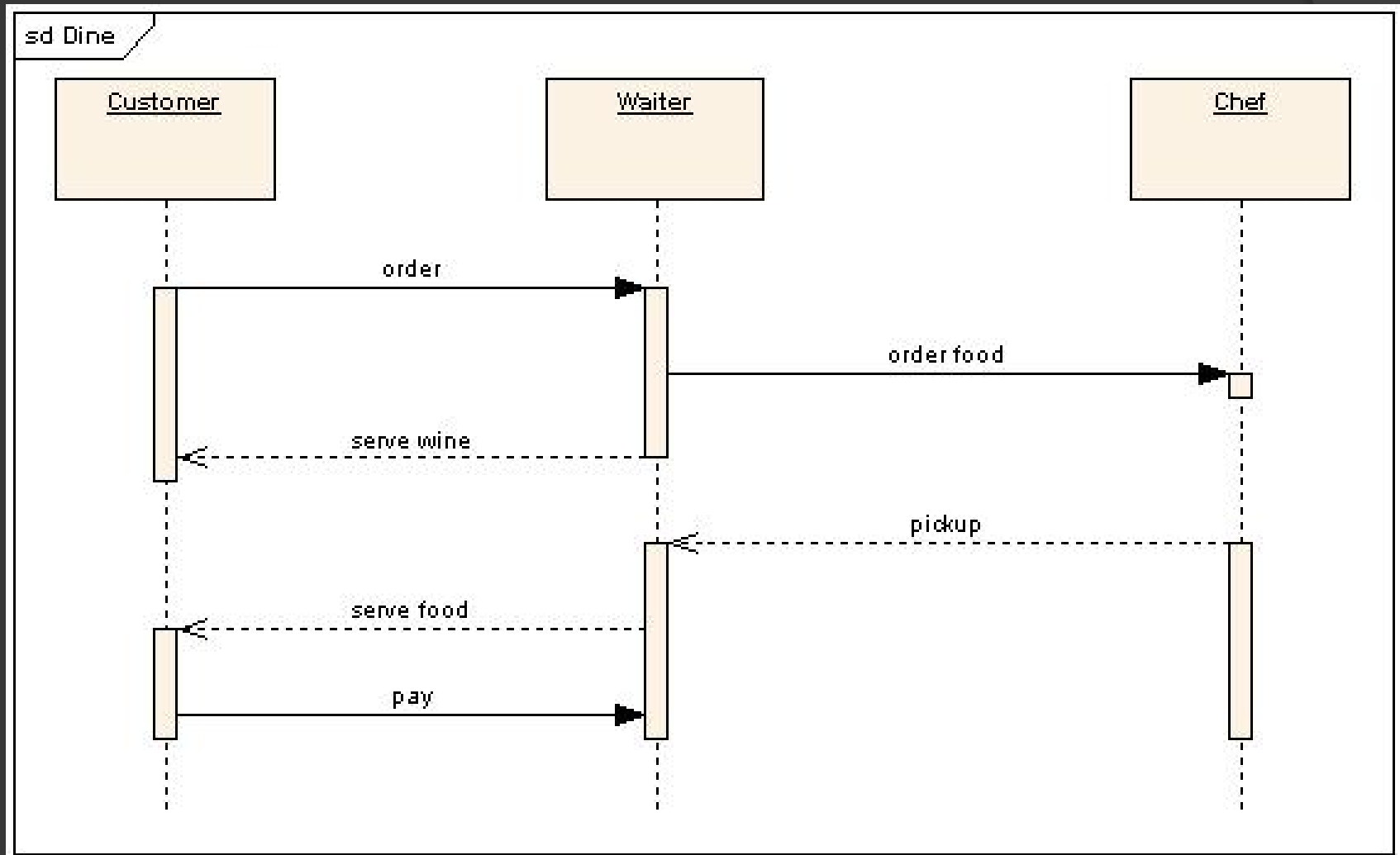
Диаграмма деятельности (пример)



Диаграммы UML

Диаграмма последовательности — это способ описать поведение системы "на примерах". Фактически, диаграмма последовательности — это запись протокола конкретного сеанса работы системы (или фрагмента такого протокола). В объектно-ориентированном программировании самым существенным во время выполнения является посылка сообщений взаимодействующими объектами.

Пример



Диаграммы UML

Диаграмма кооперации (в UML 2 – диаграмма коммуникации) семантически эквивалентна диаграмме последовательности.

Фактически, это такое же описание последовательности обмена сообщениями взаимодействующих объектов, только выраженное другими графическими средствами.

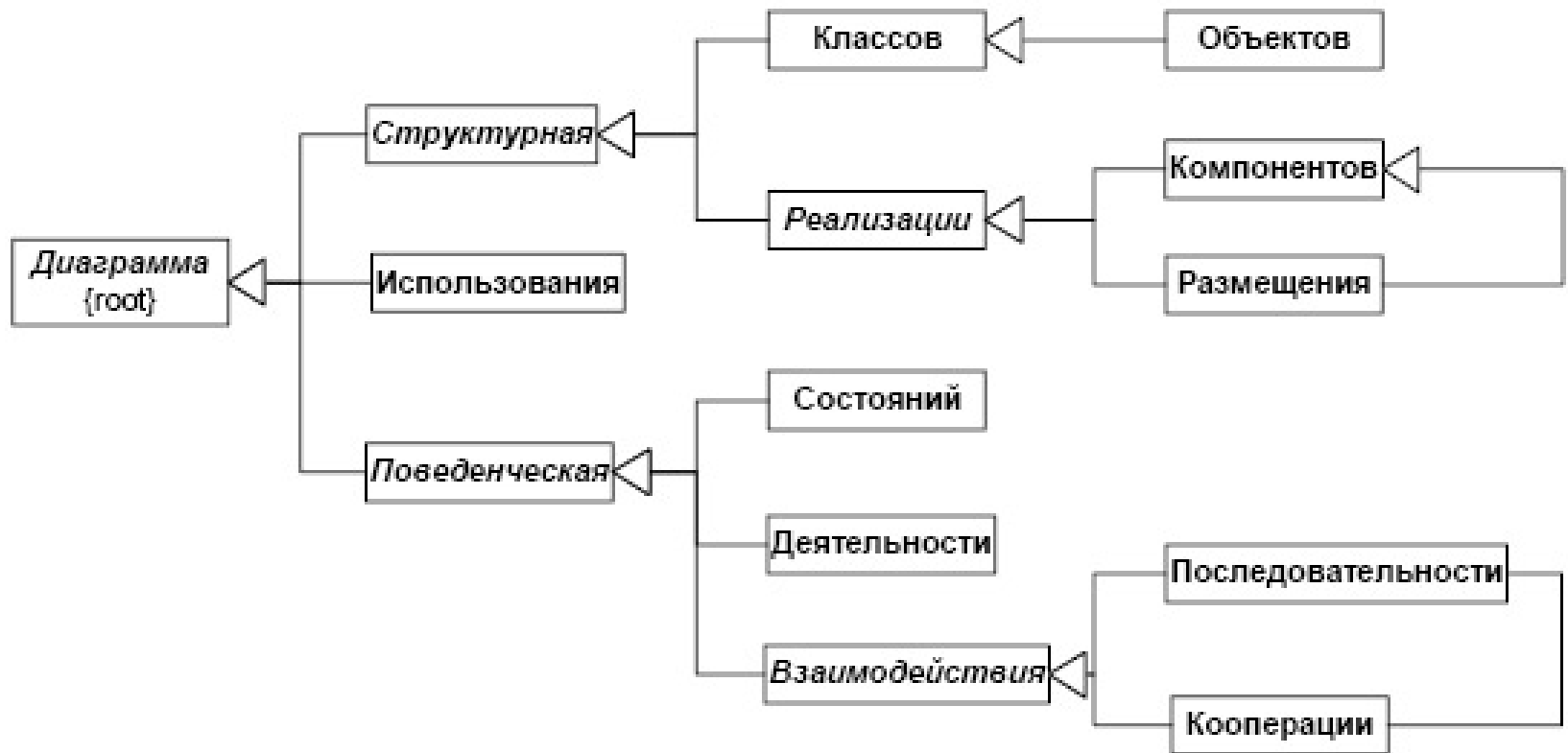
Диаграммы UML

Диаграмма компонентов — это, фактически, список артефактов, из которых состоит моделируемая система, с указанием некоторых отношений между артефактами. Наиболее существенным типом артефактов программных систем являются программы. Таким образом, на диаграмме компонентов основной тип сущностей — это компоненты (как исполнимые модули, так и другие артефакты), а также интерфейсы (чтобы указывать взаимосвязь между компонентами) и объекты (входящие в состав компонентов).

Диаграммы UML

Диаграмма размещения (диаграмма развёртывания) немногим отличается от диаграммы компонентов. Фактически, наряду с отображением состава и связей компонентов здесь показывается, как физически размещены компоненты на вычислительных ресурсах во время выполнения.

Иерархия диаграмм UML



Представления

Все аспекты моделируемой системы не удастся описать с единой точки зрения.

Моделировать сложную систему следует с нескольких различных точек зрения, каждый раз принимая во внимание один аспект моделируемой системы и абстрагируясь от остальных.

Этот тезис является одним из основополагающих принципов UML.

Представления

Выделим три представления:

- представление использования (что делает система полезного?);
- представление структуры (из чего состоит система?);
- представление поведения (как работает система?).

Представления

Выделим три представления:

- представление использования;
- представление структуры;
- представление поведения.

Представления

Представление использования призвано отвечать на вопрос, что делает система полезного.

Определяющим признаком для отнесения элементов модели к представлению использования является, по нашему мнению, явное сосредоточение внимание на факте наличия у системы внешних границ, то есть выделение внешних действующих лиц, взаимодействующих с системой, и внутренних вариантов использования, описывающих различные сценарии такого взаимодействия.

Описывается диаграммой использования.

Представления

Представление структуры призвано отвечать на вопрос: из чего состоит система.

Определяющим признаком для отнесения элементов модели к представлению структуры является явное выделение структурных элементов — составных частей системы — и описания взаимосвязей между ними. Принципиальным является чисто статический характер описания, то есть отсутствие понятия времени в любой форме, в частности, в форме последовательности событий и/или действий.

Описывается диаграммами классов, а также, если нужно, диаграммами компонентов и размещения и, в редких случаях, диаграммами объектов.

Представления

Представление поведения призвано отвечать на вопрос: как работает система.

Определяющим признаком для отнесения элементов модели к представлению поведения является явное использования понятия времени, в частности, в форме описания последовательности событий/действий, то есть в форме алгоритма.

Описывается диаграммами состояний и деятельности, а также диаграммами взаимодействия в форме диаграмм кооперации и/или последовательности.

Выводы

- Знание UML является необходимым, но не является достаточным условием построения разумных моделей программных систем.
- UML имеет синтаксис, семантику и прагматику, которые нужно знать и использовать с учетом особенностей реальной задачи и инструмента.
- Модель UML состоит из описания сущностей и отношений между ними.
- Элементы модели группируются в диаграммы и представления для наилучшего описания моделируемой системы с различных точек зрения.

Выводы

- Модель UML состоит из описания сущностей и отношений между ними.
- Диаграмма — это графическое представление некоторой части графа модели.
- Для удобства обзора сущности в UML можно подразделить на четыре группы: *структурные; поведенческие; группирующие; аннотационные.*
- В UML используются четыре основных типа отношений: *зависимость; ассоциация; обобщение; реализация.*
- Элементы модели группируются в диаграммы и представления для наилучшего описания моделируемой системы с различных точек зрения.