

Учебный курс

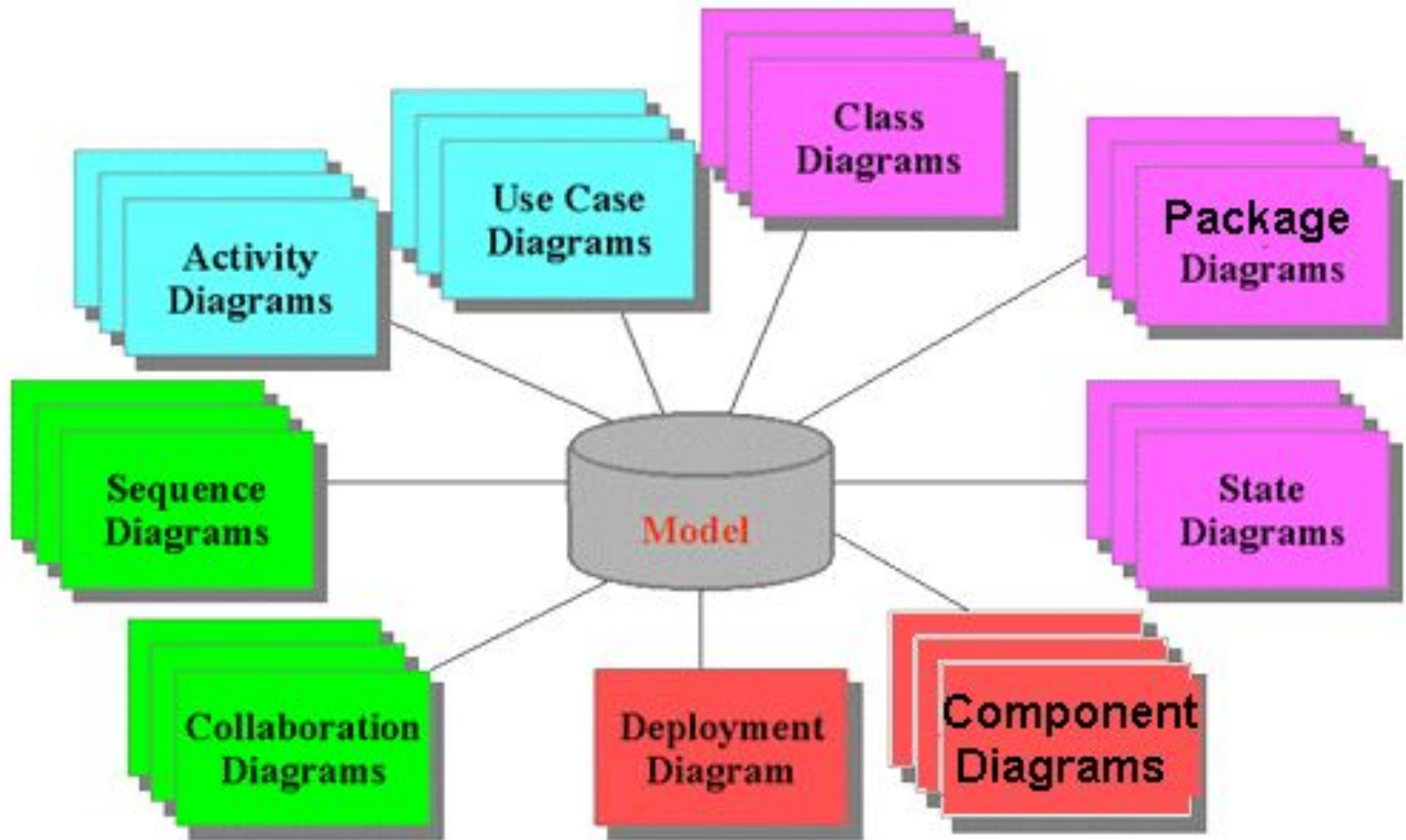
МЕТОДЫ И СРЕДСТВА ПРОЕКТИРОВАНИЯ ПО

Лекция 6

Диаграммы UML

Диаграмма вариантов использования языка UML

Канонические диаграммы языка UML 1.x



Канонические диаграммы языка UML 1.x



Диаграммы UML 1.x



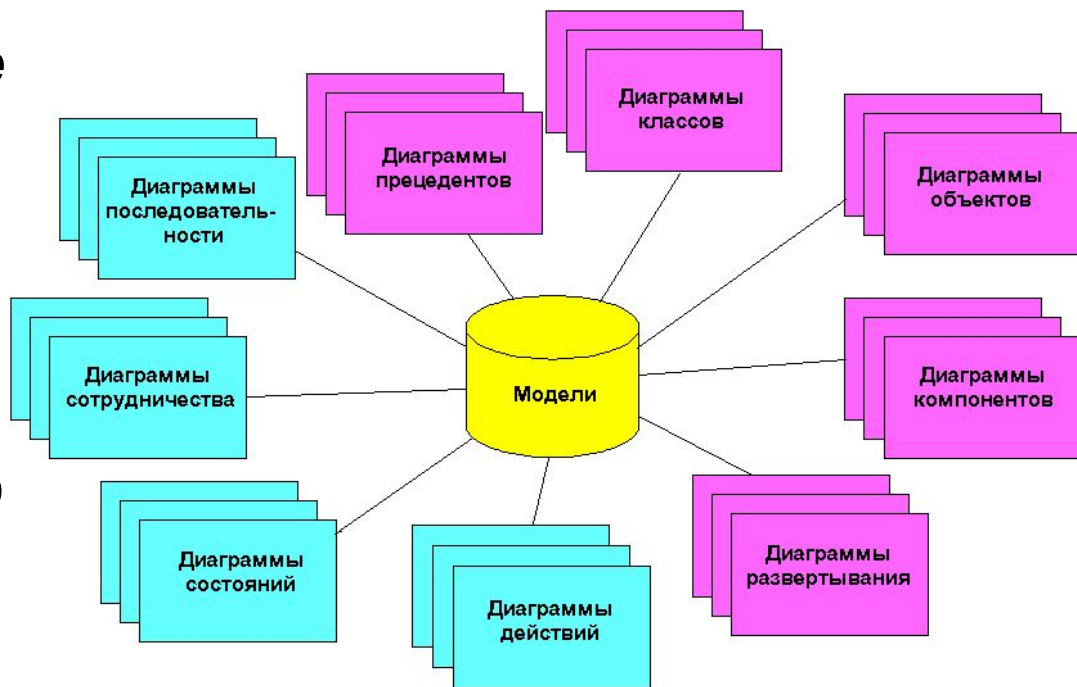
Наиболее часто используются диаграммы, помеченные голубым цветом

Терминология – проблемы с переводом отдельных терминов

source state – *исходное состояние* (правильно -состояние источник)

opaque expression – *непрозрачное выражение* (правильно – неопределенное выражение)

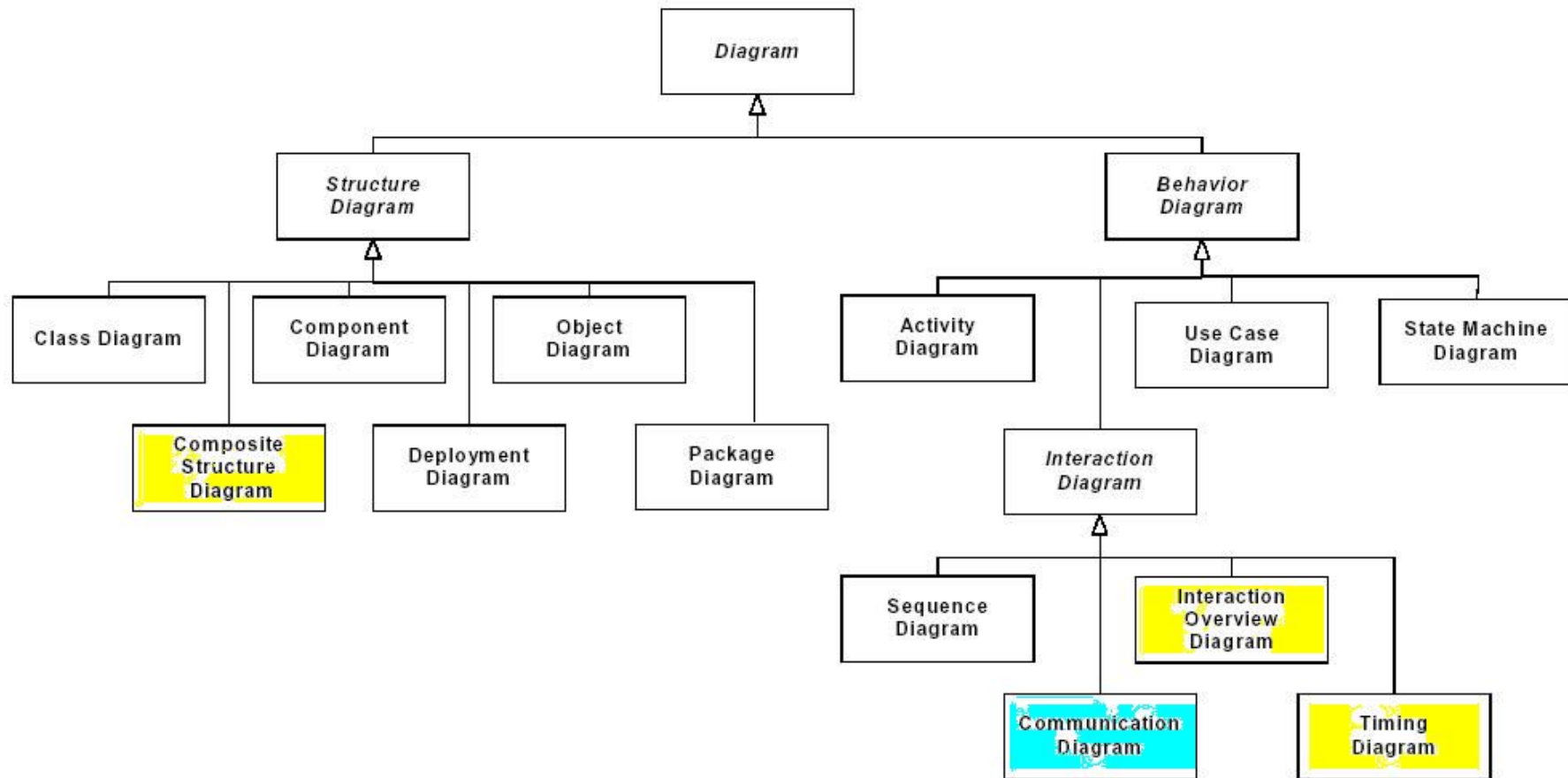
Событие – это описание группы возможных вхождений.
История (трассировка) описывает уникальные объекты и вхождения, но модели имеют дело с паттернами, применимыми ко многим объектам и вхождениям (неточный перевод англ. *occurrence*).
Линия жизни представляет собой последовательность спецификации вхождений.



Классификация моделей в языке UML

- *Структурные модели (structured models)* – модели, предназначенные для описания статической структуры сущностей или элементов некоторой системы, включая их классы, интерфейсы, атрибуты и отношения.
- *Модели поведения (behavioral models)* – модели, предназначенные для описания процесса функционирования элементов системы, включая их методы и взаимодействие между ними, а также процесс изменения состояний отдельных элементов и системы в целом.

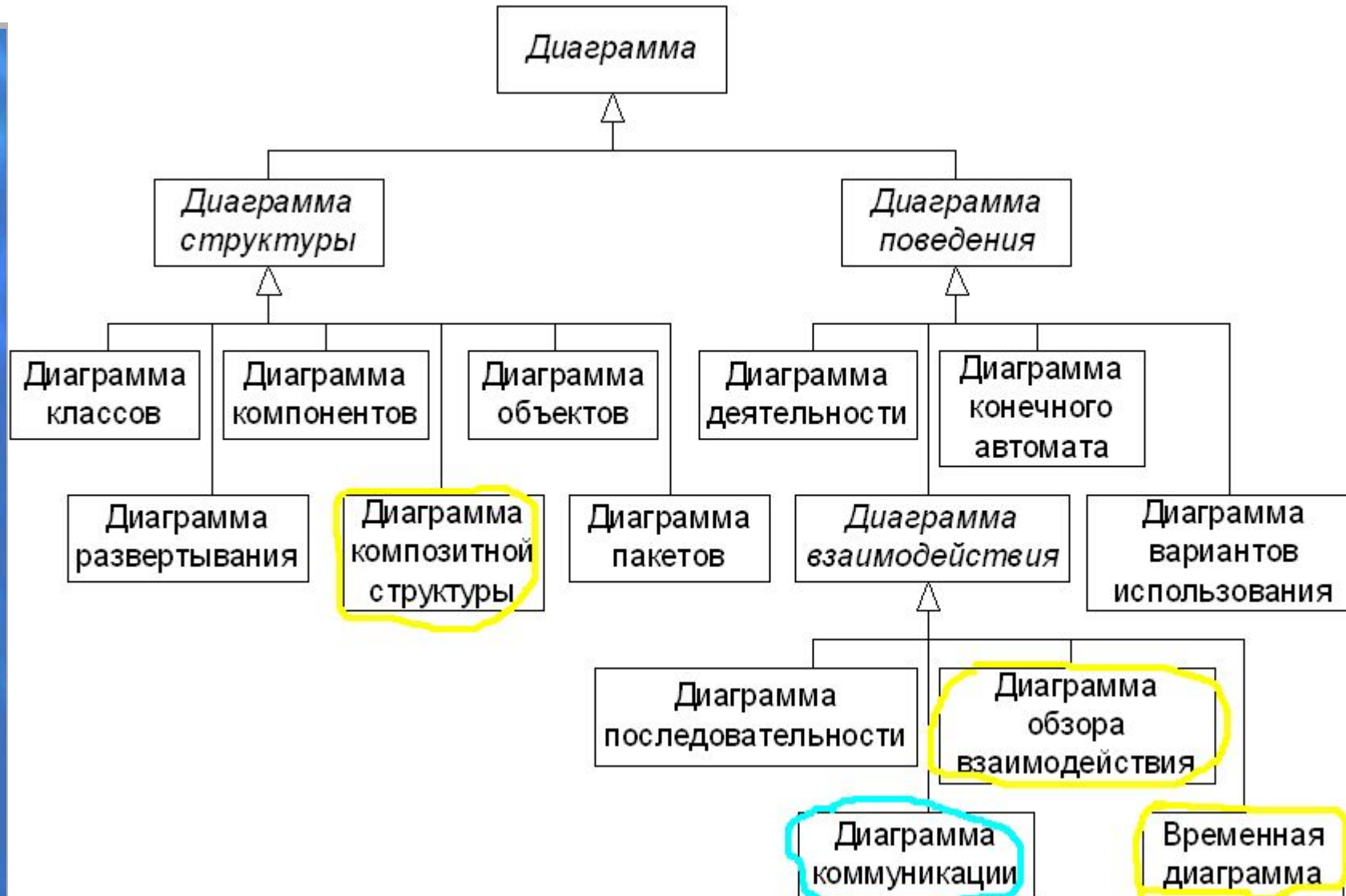
Канонические диаграммы языка UML 2.x



Новые типы диаграмм: Composite structure, interaction overview, timing

Упрощение Cooperation diagram и переименование в Communication diagram

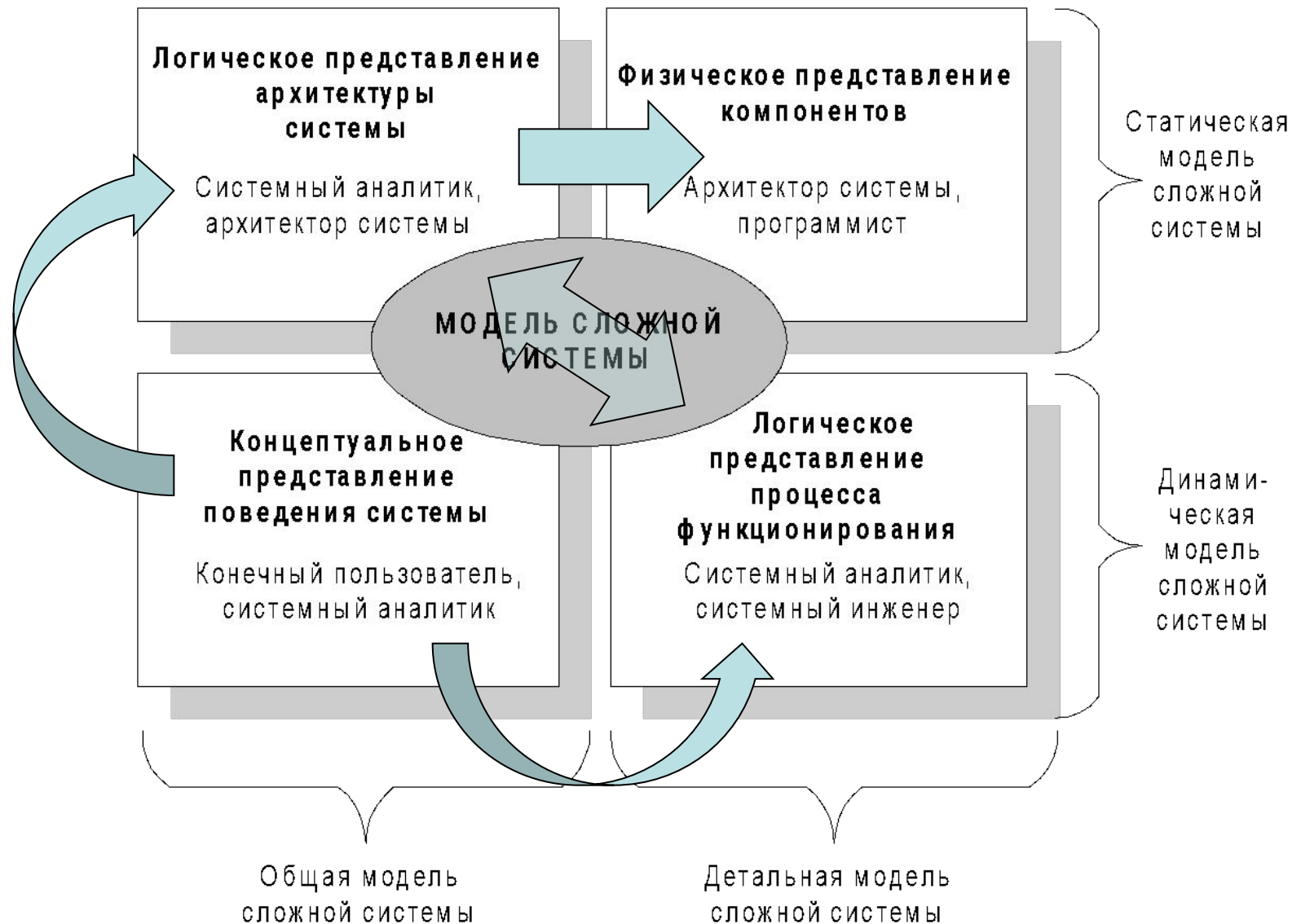
Канонические диаграммы языка UML 2.x



Канонические диаграммы языка UML 2.x



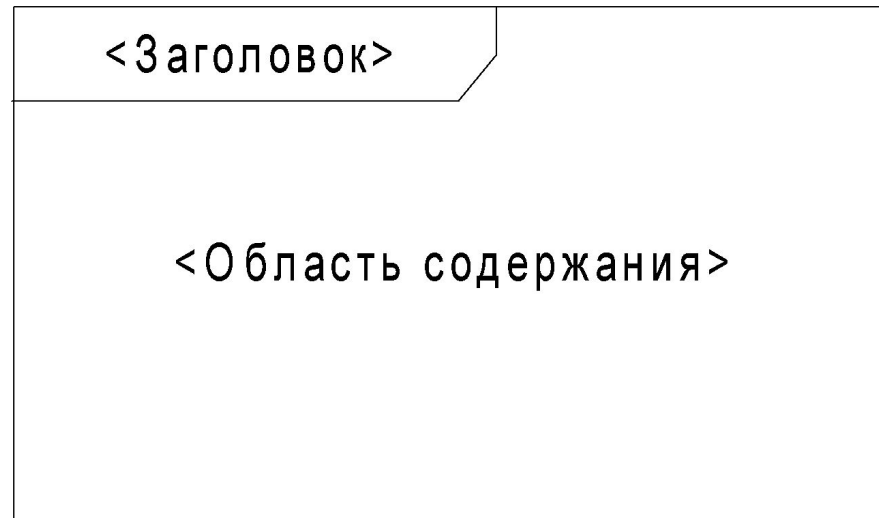
Взаимосвязь представлений сложной системы



Рекомендации по изображению диаграмм в нотации языка UML

- Количество диаграмм различных типов для модели конкретного приложения не является строго фиксированным
- Любая из моделей системы должна содержать только те элементы, которые определены в соответствующей версии языка UML
- Каждая диаграмма в нотации языка UML 2.x имеет область содержания для изображения графических узлов и путей между ними, которые представляют собой собственно элементы модели в нотации UML 2.x
- Фрейм в нотации UML 2.x используется в тех случаях, когда отдельные элементы диаграммы имеют графическую границу с другими элементами диаграммы

Изображение диаграмм языка UML 2 в виде фрейма

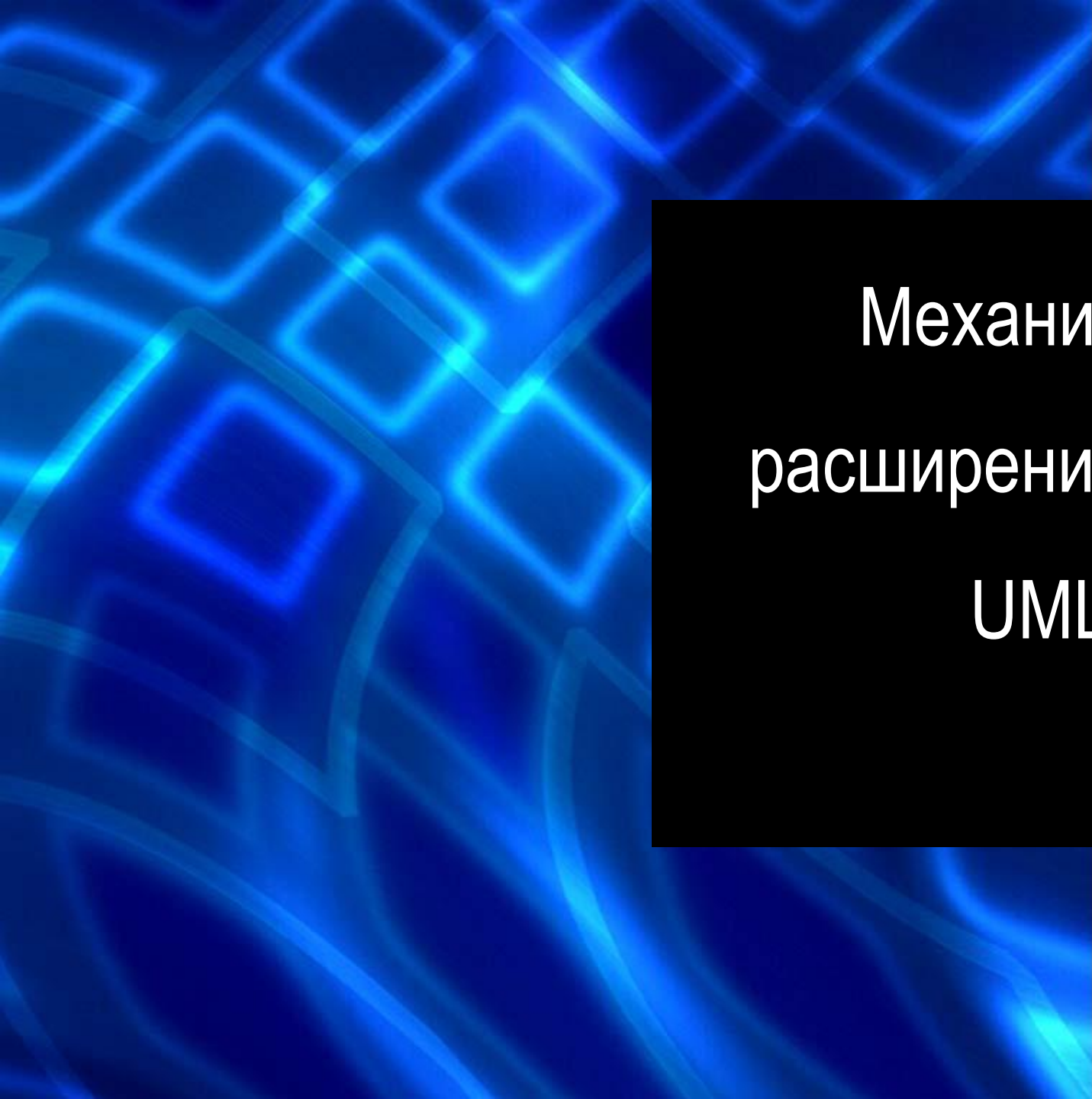


- Заголовок диаграммы является строкой текста, записанной в прямоугольнике с обрезанным углом в верхнем левом углу фрейма и имеющей следующий синтаксис:

[<тип диаграммы>]<имя>[<параметры>]

Теги заголовков и их сокращения для диаграмм UML 2.x

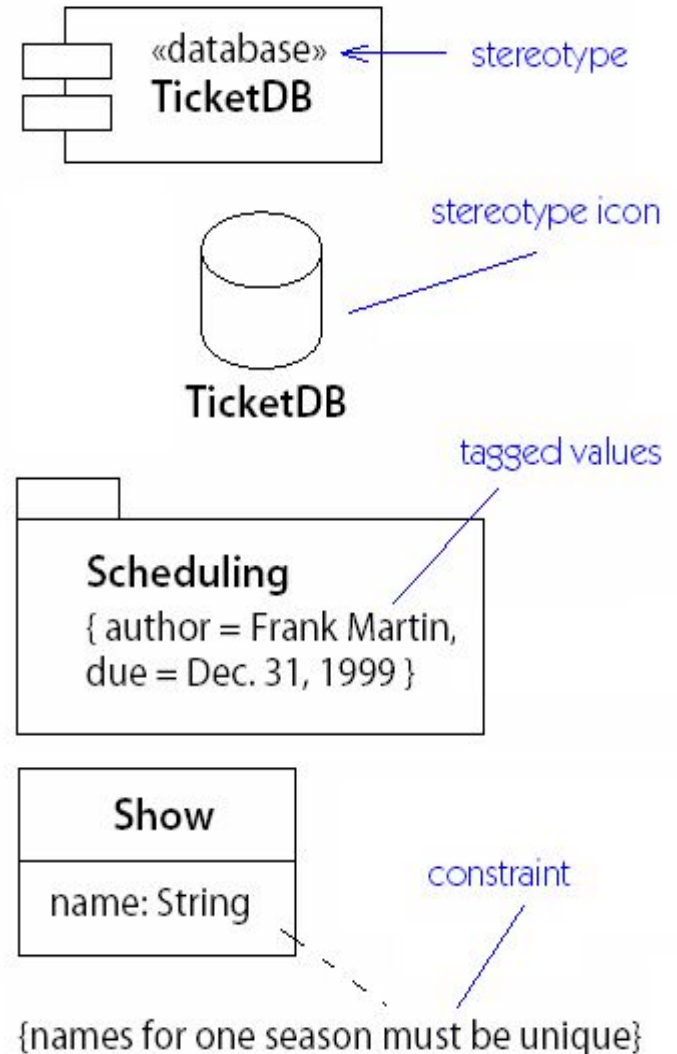
- **activity <act>** (для фреймов диаграммы деятельности)
- **class** (для фреймов диаграммы классов)
- **component <cmp>** (для фреймов диаграммы компонентов)
- **interaction <sd>** (для фреймов диаграмм взаимодействия)
- **package <pkg>** (для фреймов диаграммы пакетов)
- **state machine <stm>** (для фреймов диаграммы конечного автомата)
- **use case <uc>** (для фреймов диаграммы вариантов использования)



Механизмы
расширения языка
UML

Механизмы расширения языка UML

- **Стереотип (stereotype)** — новый тип элемента модели, который расширяет семантику базового типа метамодели языка UML
- **Помеченное значение (tagged value)** — явное определение некоторого свойства объекта как пары "имя = значение"
- **Ограничение (constraint)** — некоторое логическое условие, ограничивающее семантику выбранного элемента модели



Стереотипы в языке UML

- Стереотипы предназначены для расширения семантики отдельных элементов языка UML, но не структуры уже описанных типов или классов
- Некоторые стереотипы predefinedены в языке UML, другие могут быть определены разработчиками
- **Текстовые Стереотипы** — ключевые слова языка UML
 - Примеры: «entity», «control», «boundary»
- **Графические Стереотипы** — описываются в профилях языка UML и поддерживаются некоторыми CASE-средствами

Использование стереотипов

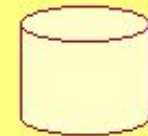
- В форме текста, заключенного в двойные угловые кавычки и размещенного выше или перед именем элемента модели
 - Первая буква текста имени стереотипа не должна быть заглавной буквой.
 - Если применяются несколько стереотипов, то имена применяемых стереотипов изображаются в форме разделенного запятыми списка с парой кавычек.
- В форме графической пиктограммы, которая заменяет текст имени соответствующего стереотипа
- В форме прямоугольника класса с уменьшенной пиктограммой стереотипа внутри этого прямоугольника, расположенного, как правило, в верхнем правом углу

Графические стереотипы компонентов в IBM Rational Rose

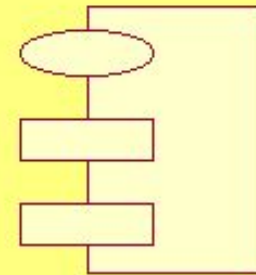
Обычный компонент



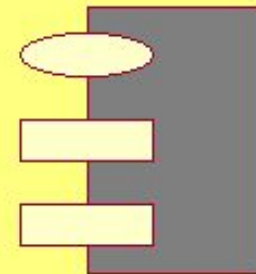
База данных



Спецификация пакета (заголовочный файл в C++ с расширением «.h»)

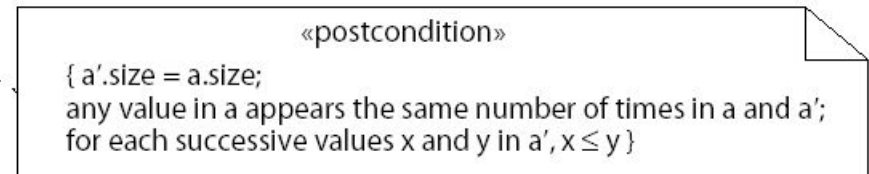
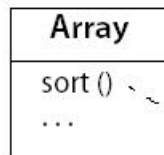
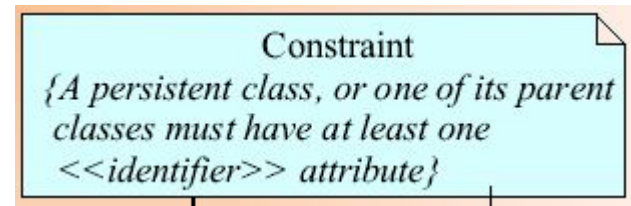
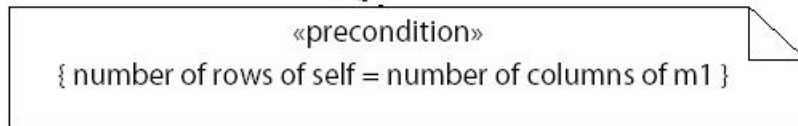
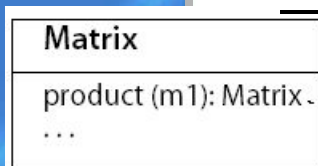


Тело пакета (файл с текстом программы в C++ с расширением «.cpp», в Java – «.java»)



Ограничения в языке UML

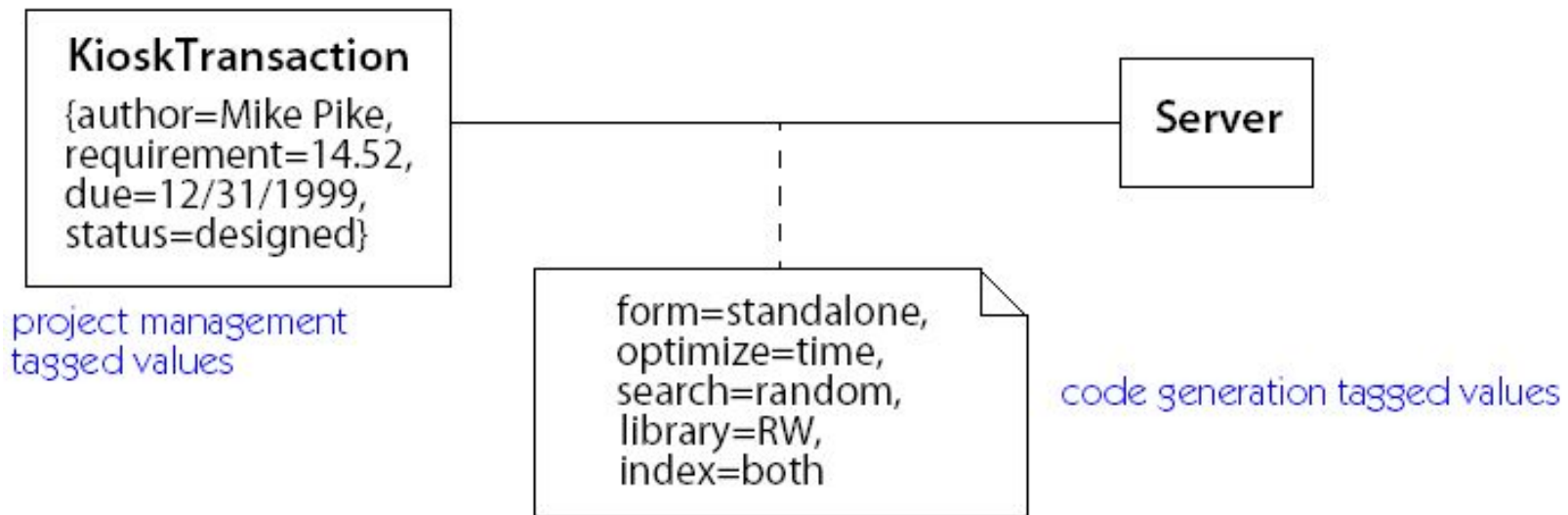
- Некоторые ограничения predefinedены в языке UML, другие могут быть специфицированы самим разработчиком
- Для корректной записи ограничений предназначен специальный язык — язык объектных ограничений (Object Constraint Language - **OCL**)



Спецификация UML 2: Only binary associations can be aggregations
self.memberEnd->exists(aggregation <> Aggregation::none) implies
self.memberEnd->size() = 2

Помеченные значения в языке UML

- В помеченном значении само имя называют *тегом* (tag). Отдельные теги predeterminedены в языке UML, другие могут быть определены пользователем
- Определяющим символом помеченного значения является знак равенства, который отделяет тег от его значения



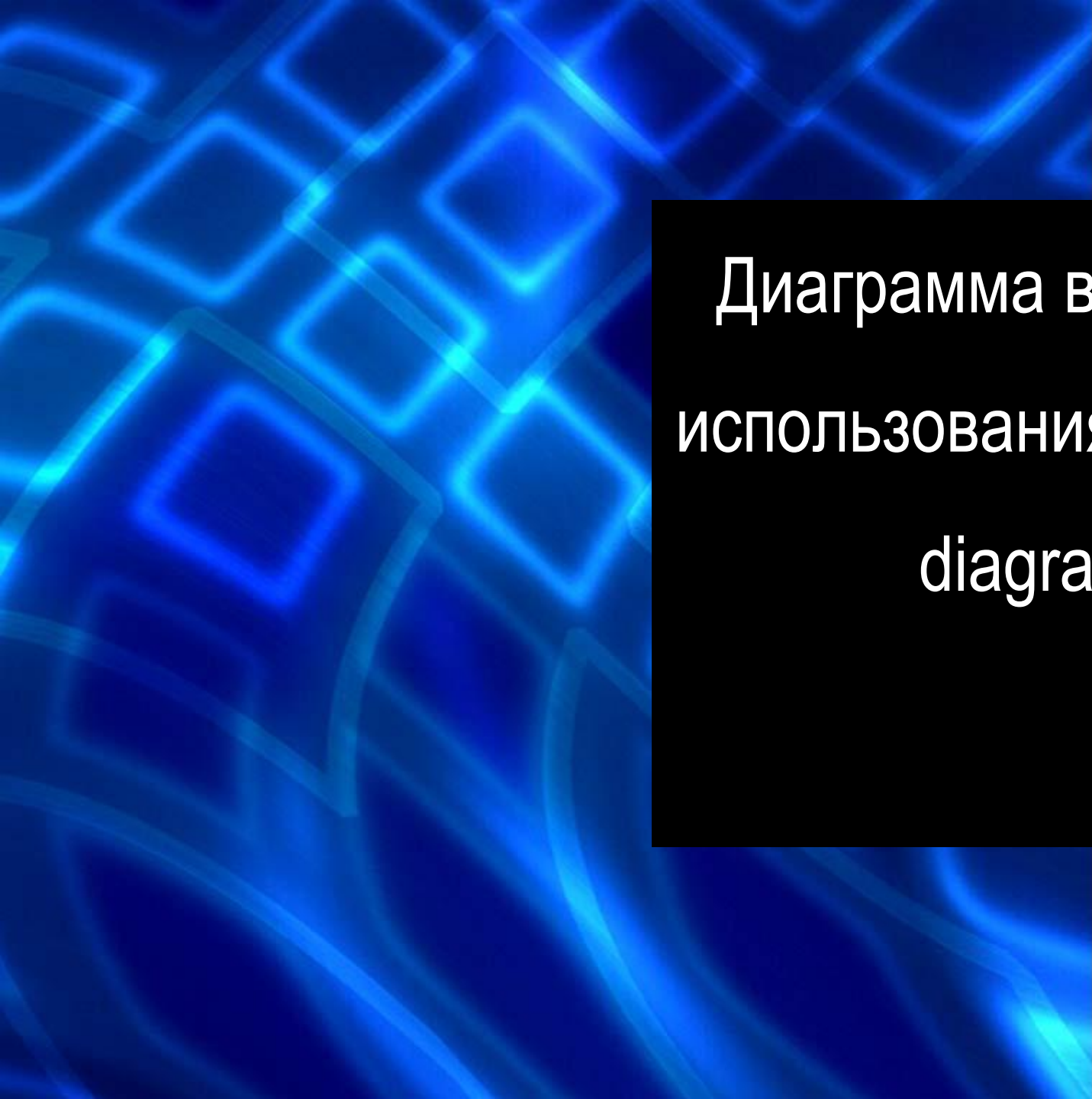


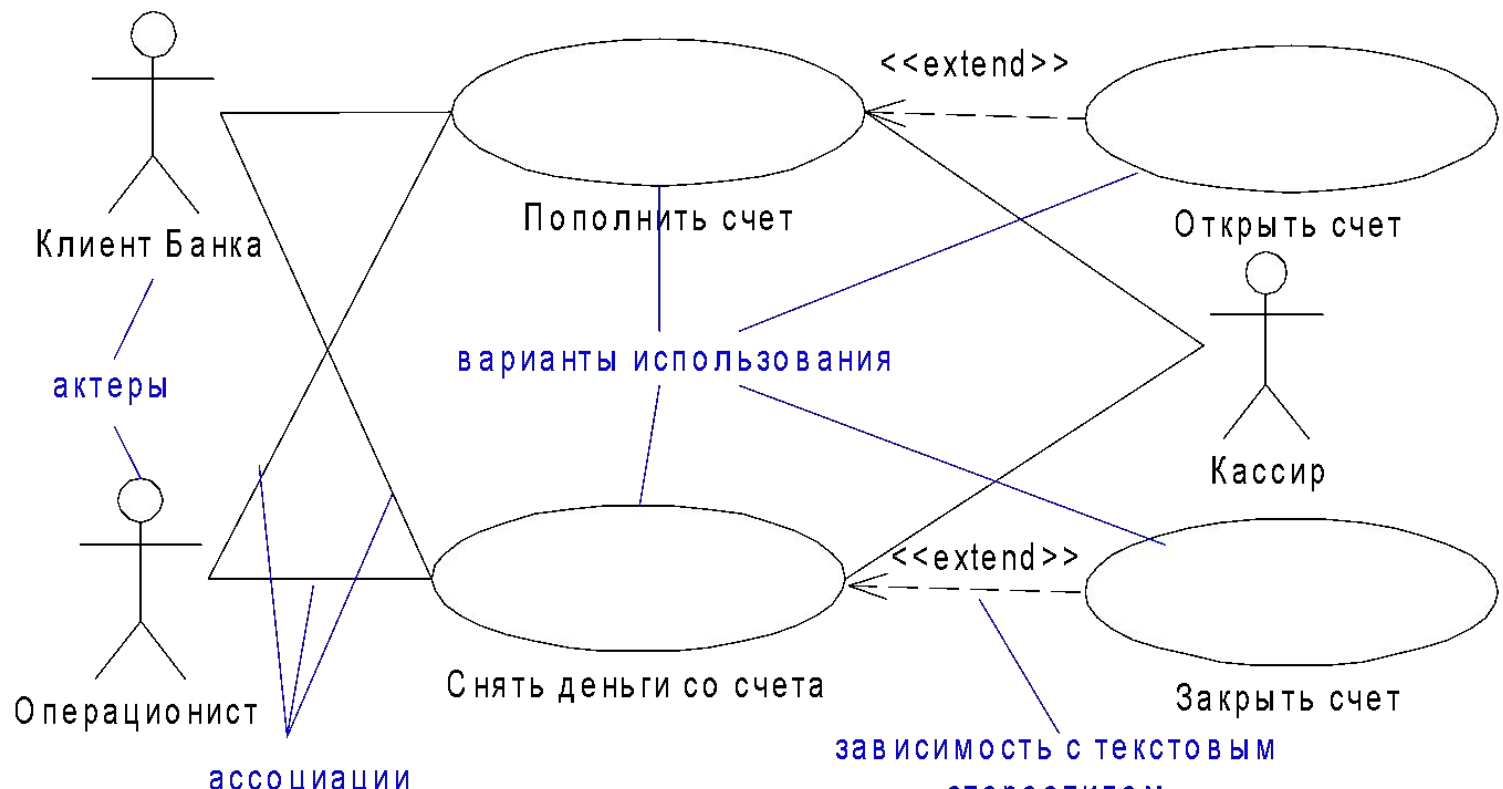
Диаграмма вариантов
использования (use case
diagram)

Диаграмма вариантов использования (UC)

- Диаграммы вариантов использования применяют для моделирования статического вида системы с точки зрения вариантов использования. Этот вид охватывает поведение системы, т.е. видимые извне сервисы, предоставляемые системой в контексте ее окружения
- **Концепция** (от [лат. *conceptio*](#) — понимание, система) — определённый способ [понимания](#) — понимание, система) — определённый способ понимания, трактовки каких-либо явлений, основная точка зрения, руководящая [идея](#) — понимание, система) — определённый способ понимания, трактовки каких-либо явлений, основная точка зрения, руководящая идея для их освещения; система взглядов на явления в мире, в природе, в обществе; ведущий замысел, конструктивный [принцип](#) — понимание, система) — определённый способ понимания, трактовки каких-либо явлений, основная точка зрения, руководящая идея для их освещения; система взглядов на явления в мире, в природе, в обществе; ведущий замысел, конструктивный принцип в [научной](#) — понимание, система) — определённый способ понимания, трактовки каких-либо явлений, основная точка зрения, руководящая идея для их освещения; система

Диаграмма вариантов использования (*use case diagram*)

- диаграмма, на которой изображаются варианты использования проектируемой системы, заключенные в границу системы,
- внешние актеры,
- а также определенные отношения между актерами и вариантами использования



Назначение диаграммы вариантов использования

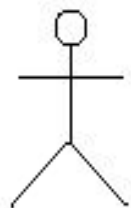
- Определить общие границы функциональности проектируемой системы в контексте моделируемой предметной области.
- Специфицировать требования к функциональному поведению проектируемой системы в форме вариантов использования.
- Разработать исходную концептуальную модель системы для ее последующей детализации в форме логических и физических моделей.
- Подготовить исходную документацию для взаимодействия разработчиков системы с ее заказчиками и пользователями

Проектируемая система и ее окружение



- *Субъект (subject)* – любой элемент модели, который обладает функциональным поведением

Основные обозначения на диаграмме вариантов использования



actor



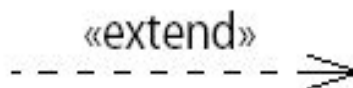
communication association

Ассоциация в контексте коммуникации



generalization

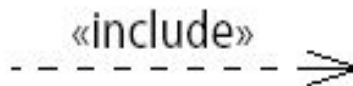
Обобщение



«extend»

extend

Расширение



«include»

include

Включение,
Как 2
разновидности
отношения
зависимости



use case

system boundary



Вариант использования (use case)

- – представляет собой общую спецификацию совокупности выполняемых системой действий с целью предоставления некоторого наблюдаемого результата, который имеет значение для одного или нескольких актеров
- Отвечает на вопрос «Что должна выполнять система?», не отвечая на вопрос «Как она должна выполнять это?»
- Имена – отглагольное существительное или глагол в неопределенной форме

Проверка состояния
текущего счета клиента

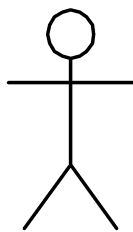
<<use case>>

Формирование отчета по
выполненным заказам

Формирование отчета по
выполненным заказам

Актер (actor)

- – любая внешняя по отношению к проектируемой системе сущность, которая взаимодействует с системой и использует ее функциональные возможности для достижения определенных целей или решения частных задач
- *Примеры актеров:* кассир, клиент банка, банковский служащий, президент, продавец магазина, менеджер отдела продаж, пассажир авиарейса, водитель автомобиля, администратор гостиницы, сотовый телефон



Клиент банка

<<actor>>
Посетитель
Интернет-магазина







Удаленный
пользователь

Вопросы для идентификации актеров в системе

- Какие организации или лица будут использовать систему
- Кто будет получать пользу от использования системы
- Кто будет использовать информацию от системы
- Будет ли использовать система внешние ресурсы
- Может ли один пользователь играть несколько ролей при взаимодействии с системой
- Могут ли различные пользователи играть одну роль при взаимодействии с системой
- Будет ли система взаимодействовать с законодательными, исполнительными, налоговыми или другими органами

Отношения на диаграмме вариантов использования

<i>Relationship</i>	<i>Function</i>	<i>Notation</i>
association	The communication path between an actor and a use case that it participates in	
include	The insertion of additional behavior into a base use case that explicitly describes the insertion	«include» 
extend	The insertion of additional behavior into a base use case that does not know about it	«extend» 
use case generalization	A relationship between a general use case and a more specific use case that inherits and adds features to it	

Отношения на диаграмме UC

Графические примитивы:



Актор (Актор)



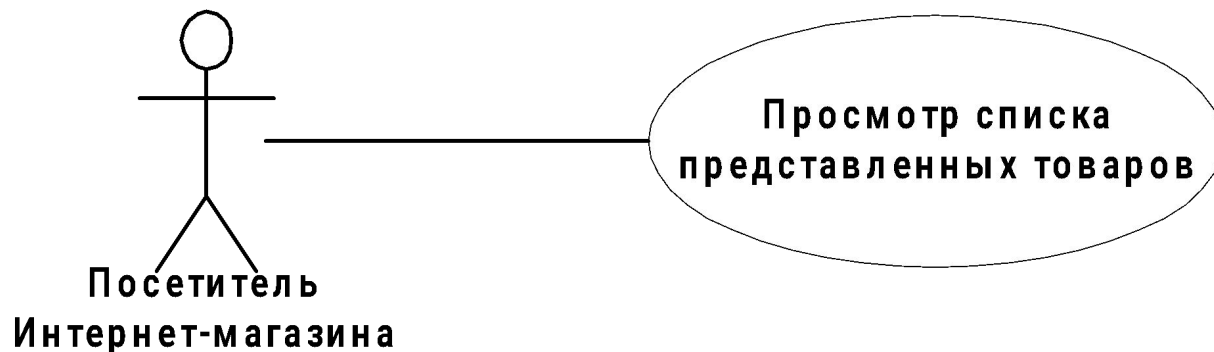
Вариант
использования

Виды отношений:

	Ассоциация	
	Включение	
	Расширение	
	Обобщение	

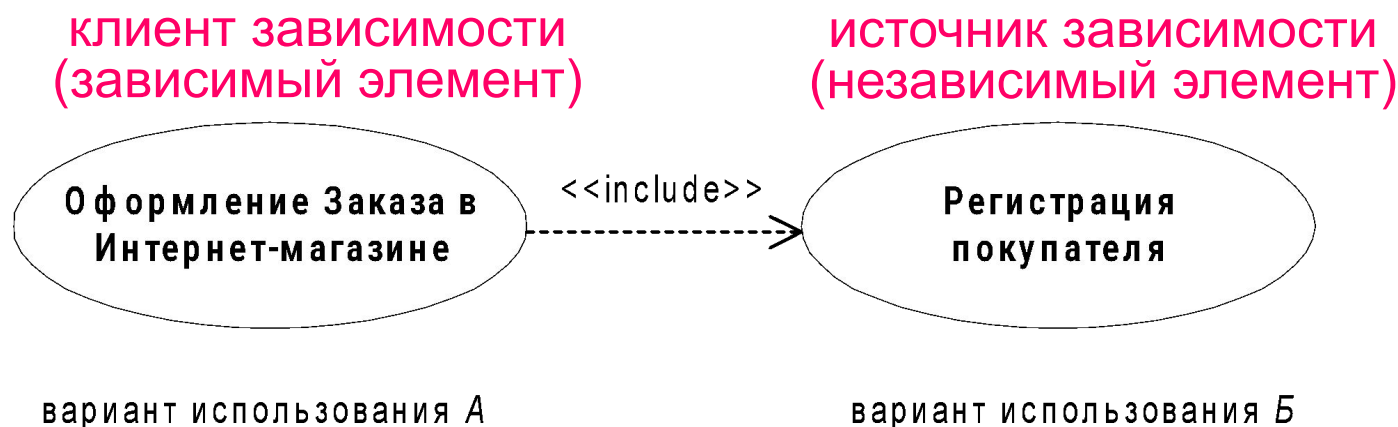
Отношение ассоциации

- Ассоциация (association) является одним из фундаментальных понятий в языке UML 2.x и может использоваться на различных канонических диаграммах при построении визуальных моделей
- Применительно к диаграммам вариантов использования отношение ассоциации может служить только для обозначения взаимодействия актера с вариантом использования.



Отношение включения

- Отношение *зависимости* (*dependency*) определяется как форма взаимосвязи между двумя элементами модели, предназначенная для спецификации того обстоятельства, что изменение одного элемента модели приводит к изменению некоторого другого элемента
- Отношение *включения* (*include*) специфицирует тот факт, что некоторый вариант использования содержит поведение, определенное в другом варианте использования

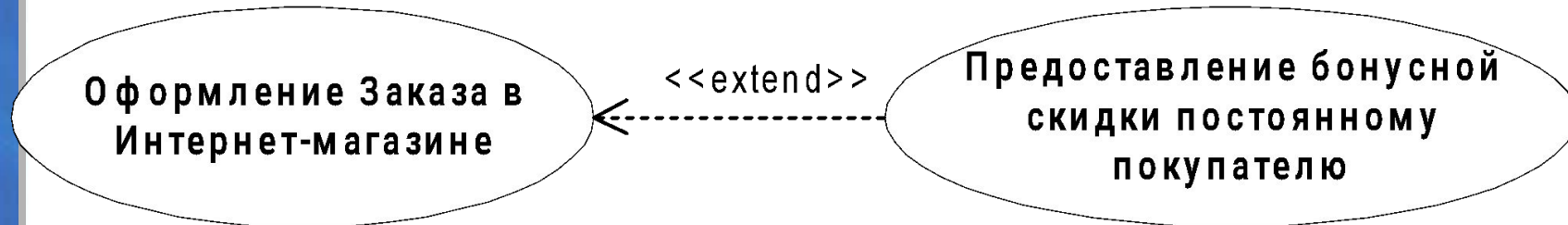


Отношение расширения

- Отношение *расширения* (*extend*) определяет взаимосвязь одного варианта использования с некоторым другим вариантом использования, функциональность или поведение которого задействуется первым не всегда, а только при выполнении некоторых **ДОПОЛНИТЕЛЬНЫХ** условий.

ИСТОЧНИК ЗАВИСИМОСТИ
(независимый элемент)

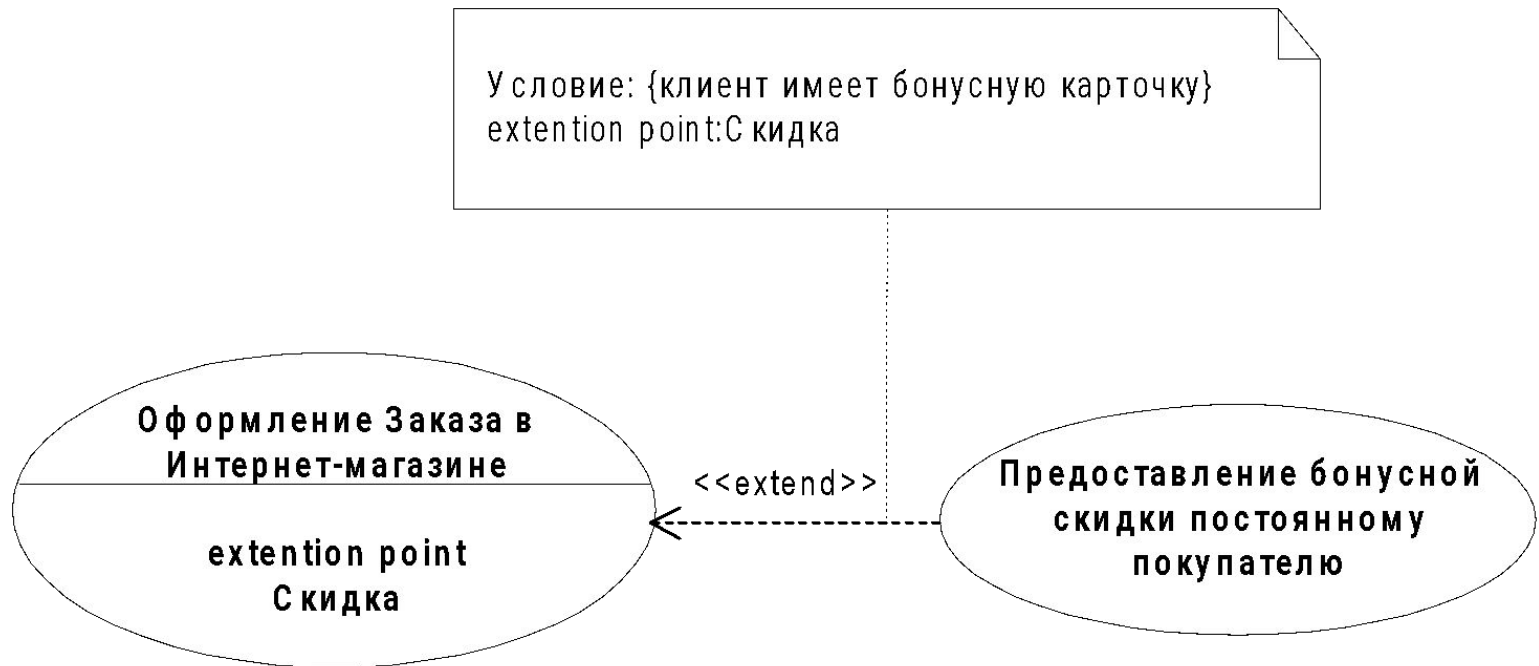
КЛИЕНТ ЗАВИСИМОСТИ
(зависимый элемент)



вариант использования А

вариант использования Б

Изображение отношения расширения с условием выполнения



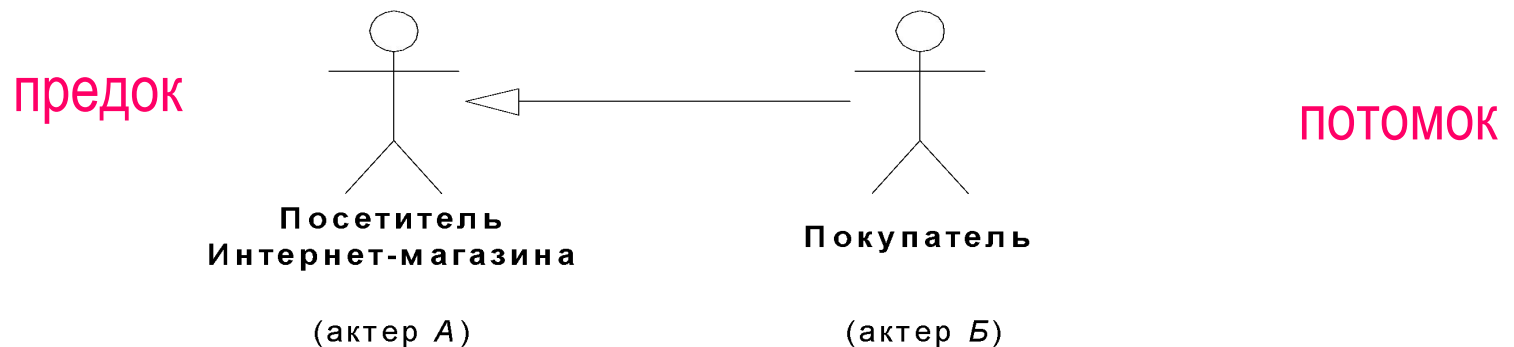
Отношение обобщения

- *Отношение обобщения (generalization relationship)* предназначено для спецификации того факта, что один элемент модели является специальным или частным случаем другого элемента модели



вариант использования А

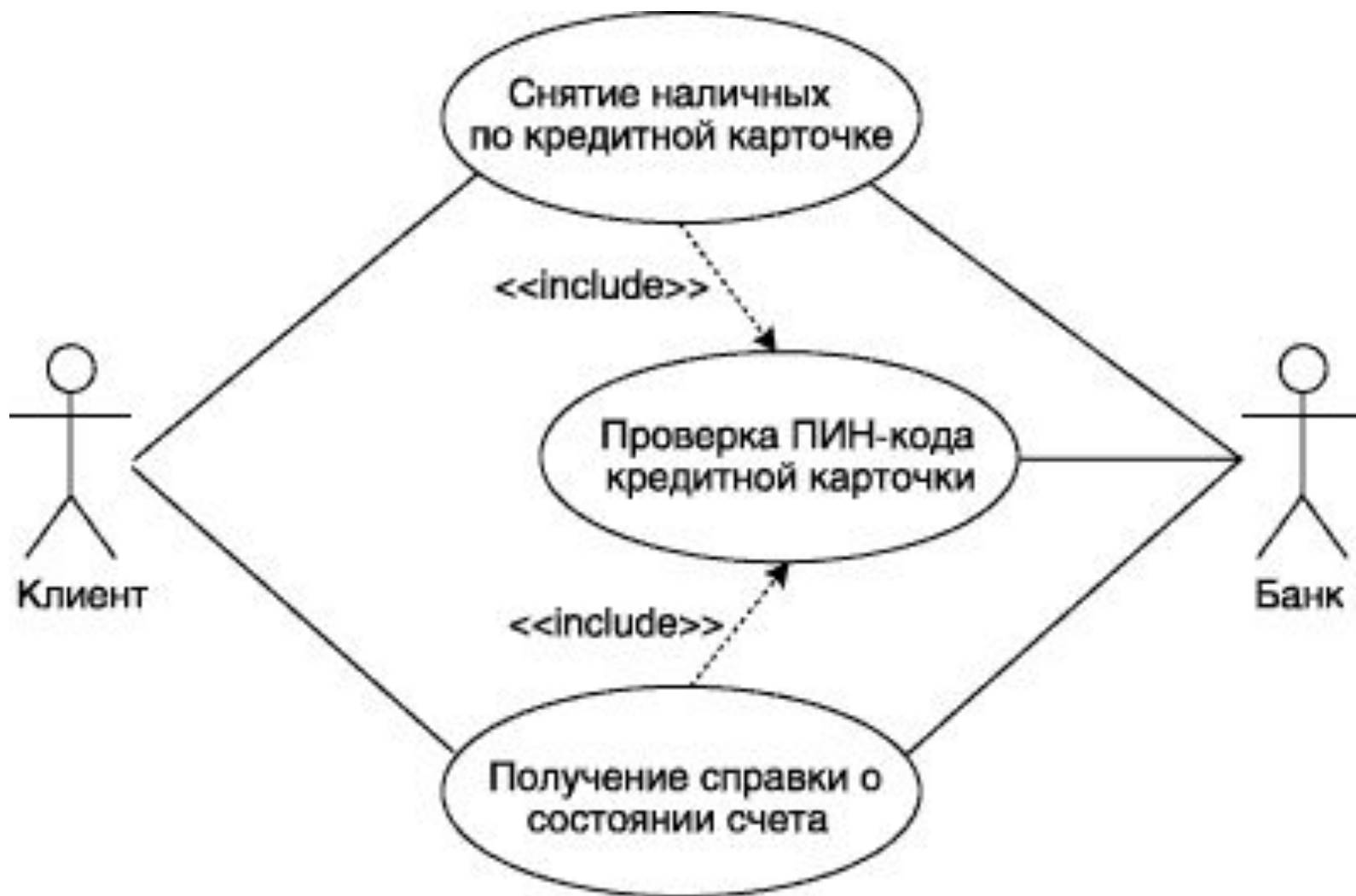
вариант использования Б



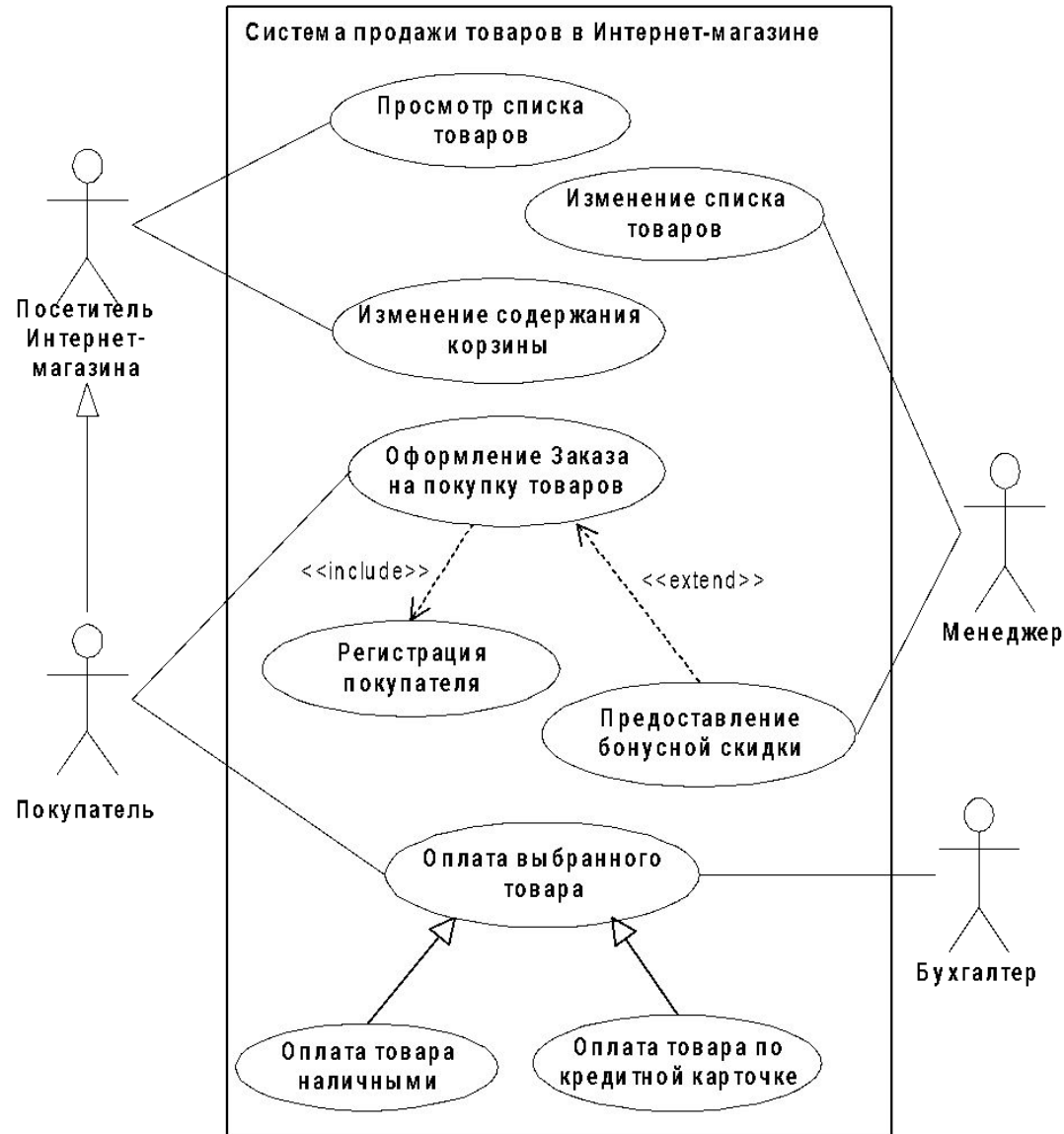
Что внутри УС?



Диаграмма вариантов использования для модели банкомата



Пример диаграммы ВИ для системы продажи товаров в Интернет-магазине



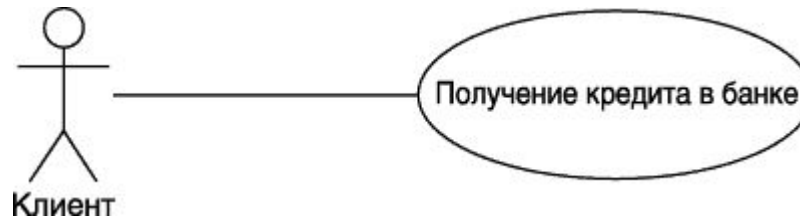
Типичные ошибки при разработке диаграмм вариантов использования

- Превращение диаграммы вариантов использования в диаграмму деятельности за счет желания отразить все функциональные действия
- Инициатором действий является разрабатываемая система
- Спецификация атрибутов и операций классов до того, как сформулированы все варианты использования
- Задание слишком кратких имен вариантам использования
- Описание вариантов использования в терминологии, непонятной пользователям системы или заказчику
- Отсутствие описаний альтернативных последовательностей действий
- Тратится слишком много времени на решение вопросов о том, какие стереотипы и ассоциации использовать на диаграмме

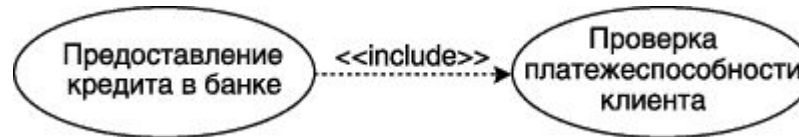
Контрольный опрос:

Прокомментировать смысл графического изображения на рисунке

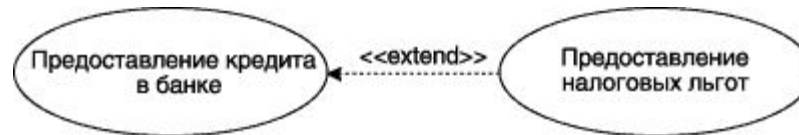
• Рис 1



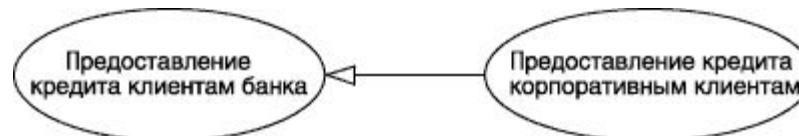
• Рис 2



• Рис 3



• Рис 4



Формализация функциональных требований с помощью диаграммы ВИ

- *Требование (requirement)* – желательное свойство, характеристика или условие, которым должна удовлетворять система в процессе своей эксплуатации
- *Требование к ПО* – некоторое свойство ПО, которым должна обладать система или ее компонент, чтобы удовлетворять условиям контракта, положениям стандартов, формальной спецификации или технической документации
- *Управление требованиями* – это систематический подход к выявлению, организации и документированию требований к системе, а также процесс, в ходе которого вырабатывается и обеспечивается соглашение между заказчиком и разработчиком по поводу меняющихся требований к системе

Классификация требований – модель FURPS+

- **Functionality**
 - **функциональные требования**
- Usability (требования практичности)
- Reliability (требования надежности)
- Performance (требования производительности)
- Supportability (требования обслуживания и сопровождения)

Дополнительно + IEEE 610.12.1990

- Проектные ограничения
- Требования выполнения
- Требования к GUI
- Физические требования



Functionality – функциональные требования

- **Функциональные требования** определяют действия, которые должна быть способна выполнить система, без рассмотрения физических особенностей их реализации
- Тем самым функциональные требования определяют внешнее поведение системы
- Лучше всего они описываются в форме *модели вариантов использования*
- Каждому функциональному требованию в этом случае будет соответствовать отдельный *вариант использования*

Текстовые сценарии в UML

- Центральное место занимают функциональные требования, специфицирующие особенности реализации отдельных бизнес-процессов моделируемой системы. Они служат исходной информацией для построения диаграмм ВИ. Однако графических средств языка UML на практике оказывается недостаточно для спецификации функциональных требований.
- Одним из требований языка UML является самодостаточность диаграмм для представления информации о моделях проектируемых систем. Однако большинство разработчиков и экспертов согласны с тем, что изобразительных средств языка UML явно не хватает для того, чтобы учесть на диаграммах варианты использования особенности функционального поведения сложной системы. С этой целью рекомендуется дополнять этот тип диаграмм текстовыми сценариями, которые уточняют или детализируют последовательность действий, совершаемых системой при выполнении ее вариантов использования.

Спецификация ВИ с помощью текстовых сценариев

- *Сценарий (scenario)* – специально написанный текст, который описывает поведение моделируемой системы в форме последовательности выполняемых действий актеров и самой системы.
- В контексте языка UML сценарий используется для дополнительной иллюстрации взаимодействия актеров и вариантов использования. Предлагаются различные способы представления или написания подобных сценариев.

Шаблон для написания сценария отдельного варианта использования

Главный раздел	Раздел "Типичный ход событий"	Раздел "Исключения"	Раздел "Примечания"
Имя варианта использования	Типичный ход событий, приводящий к успешному выполнению варианта использования	Исключение № 1	Примечания № 1
Актеры		Исключение № 2	Примечания № 2
Цель	
Краткое описание			
Тип			
Ссылки на другие варианты использования			Исключение № N

Шаблон для написания сценария отдельного варианта использования

Актер	Цель № 1	Успех	Исключение № 1	Примечания
			Исключение № 2	
			Исключение № 3	
	Цель № 2	Успех	Исключение № 1	
			Исключение № 2	
			Исключение № 3	

ГЛАВНЫЙ РАЗДЕЛ сценария выполнения варианта использования "Снятие наличных по кредитной карточке"

Вариант использования	Снятие наличных по кредитной карточке
Актеры	Клиент, Банк
Цель	Получение требуемой суммы наличными
Краткое описание	Клиент запрашивает требуемую сумму. Банкомат обеспечивает доступ к счету клиента. Банкомат выдает клиенту наличные.
Тип	Базовый
Ссылки на другие варианты использования	Включает в себя ВИ: <ul style="list-style-type: none">• Проверка ПИН-кода кредитной карточки• Идентифицировать кредитную карточку

Раздел ТИПИЧНЫЙ ход событий сценария выполнения варианта использования "Снятие наличных по кредитной карточке"

Действия актеров	Отклик системы
1. Клиент вставляет кредитную карточку в устройство чтения банкомата Исключение №1: Кредитная карточка недействительна	2. Банкомат проверяет кредитную карточку 3. Банкомат предлагает ввести ПИН-код
4. Клиент вводит персональный PIN-код Исключение №2: Клиент вводит неверный ПИН-код	5. Банкомат проверяет ПИН-код 6. Банкомат отображает опции меню
7. Клиент выбирает снятие наличных со своего счета	8. Система делает запрос в Банк и выясняет текущее состояние счета клиента 9. Банкомат предлагает ввести требуемую сумму
10. Клиент вводит требуемую сумму 11. Банк проверяет введенную сумму Исключение №3: Требуемая сумма превышает сумму на счете клиента	12. Банкомат изменяет состояние счета клиента, выдает наличные и чек
13. Клиент получает наличные и чек	14. Банкомат предлагает клиенту забрать кредитную карточку
15. Клиент получает свою кредитную карточку	16. Банкомат отображает сообщение о готовности к работе

Раздел ИСКЛЮЧЕНИЯ сценария выполнения варианта использования "Снятие наличных по кредитной карточке"

Исключение №1. Кредитная карточка недействительна или неверно вставлена

Действия актера	Отклик системы
	3. Банкомат отображает информацию о неверно вставленной кредитной карточке
	14. Банкомат возвращает клиенту его кредитную карточку
15. Клиент получает свою кредитную карточку	

Исключение №2. Клиент вводит неверный ПИН-код

	6. Банкомат отображает информацию о неверном <u>ПИН-коде</u>
4. Клиент вводит новый ПИН-код	

Исключение №3. Требуемая сумма превышает сумму на счете клиента

	12. Банкомат отображает информацию о превышении кредита
10. Клиент вводит новую требуемую сумму	

Последовательность разработки вариантов использования

- Определить главных (первичных) актеров и определить их цели по отношению к системе
- Специфицировать все базовые (основные) варианты ИСПОЛЬЗОВАНИЯ (они соответствуют высокоуровневым требованиям, предъявляемым к системе)
- Выделить цели базовых ВИ, интересы актеров в контексте этих ВИ, предусловия и постусловия ВИ
- Написать успешный сценарий выполнения базовых ВИ
- Определить исключения (неуспех) в сценариях ВИ и написать сценарии для всех исключений (может быть изображено на отдельном UC)
- Выделить ВИ исключений и изобразить их со стереотипом «extend» (может быть изображено на отдельном UC)
- Выделить общие фрагменты функциональности ВИ (для нескольких UC) и изобразить их отдельными ВИ со стереотипом «include»

Показатели качества для модели вариантов использования

- Все ли функциональные требования описываются вариантами использования?
- Не содержит ли модель вариантов использования ненужное поведение, которое отсутствует в требованиях?
- Действительно ли в модели необходимы все выявленные связи включения, расширения и обобщения?
- Правильно ли произведено деление модели на пакеты вариантов использования?
- Стала ли модель в результате деления на пакеты проще и удобнее для восприятия и сопровождения?
- Можно ли на основе модели вариантов использования составить четкое представление о функционировании системы в контексте ее пользователей?

UML Profile for Business Modeling



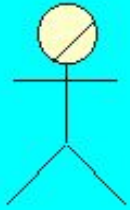
Business use case



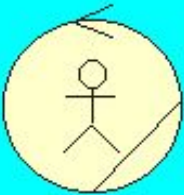
Business use-case realization

- Бизнес вариант использования – элемент модели, предназначенный для представления отдельного бизнес процесса
- Реализация бизнес варианта использования – описывает реализацию отдельного бизнес варианта использования в терминах кооперации объектов, экземпляров сотрудников и бизнес сущностей

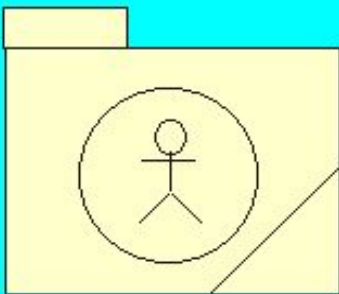
UML Profile for Business Modeling



Business actor



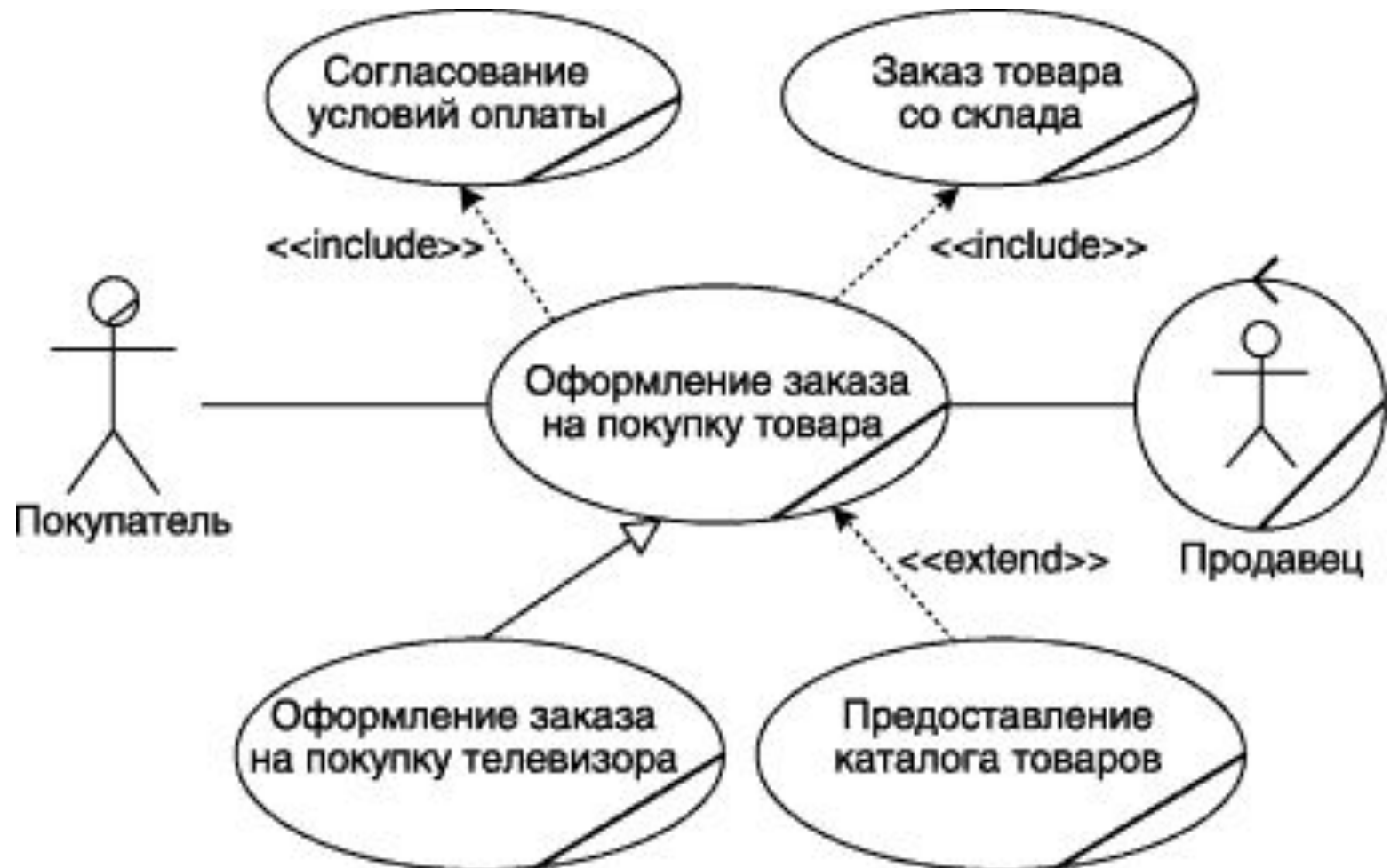
Business worker



Organization unit

- Бизнес актер – индивидуум, группа, организация, компания или система, которые взаимодействуют с моделируемой системой (компанией), но не входят в нее. Примеры – клиенты, поставщики, партнеры.
- Сотрудник – индивидуум, который действует внутри моделируемой системы (компании), взаимодействует с другими сотрудниками и манипулирует бизнес сущностями.
- Организационная единица – пакет, в состав которого могут входить сотрудники, бизнес сущности, реализации бизнес вариантов использования, диаграммы языка UML и другие организационные единицы

Диаграмма вариантов использования для системы продажи товаров по каталогу в UML Profile for Business Modeling



Примеры оформления сценариев

Сценарий №1 выполнения ВИ "Снятие наличных по кредитной карточке"

Главный раздел

- *Вариант использования:* Снятие наличных по кредитной карточке
- *Актеры:* Клиент Банкомата, Банк
- *Цель:* Получение требуемой суммы наличными
- *Краткое описание:* Клиент использует свою карточку для снятия наличных. Клиент запрашивает требуемую сумму. Банкомат обеспечивает доступ к счету клиента. Банкомат выдает клиенту наличные.
- *Тип:* Базовый
- Ссылки на другие варианты использования: Включает в себя ВИ:
 - Проверка ПИН-кода кредитной карточки

Раздел Типичный ход событий

- 1. Клиент вставляет кредитную карточку в устройство чтения банкомата
- 2. Банкомат передает информацию о кредитной карточке в Банк
- 3. Банк проверяет информацию о кредитной карточке
- *Исключение №1:* Кредитная карточка недействительна (утрачена)
- *Исключение №2:* Кредитная карточка просрочена
- 4. Банкомат предлагает ввести ПИН-код
- 5. Клиент вводит PIN-код
- 6. Банкомат проверяет ПИН-код
- *Исключение №3:* Введенный ПИН-код неверный
- *Исключение №4:* Клиент ввел неверный ПИН-код 3 раза
- 7. Банкомат отображает опции меню
- 8. Клиент выбирает снятие наличных со своего счета
- 9. Банкомат предлагает ввести требуемую сумму

Раздел Типичный ход событий

- 10. Клиент вводит требуемую сумму
- 11. Банкомат делает соответствующий запрос в Банк
- 12. Банк проверяет введенную сумму
- Исключение №5: Требуемая сумма превышает сумму на счете клиента
- 13. Банк изменяет состояние счета клиента
- 15. Клиент получает свою кредитную карточку
- Исключение №6: Клиент выбрал печать чека
- 14. Банкомат предлагает клиенту забрать его кредитную карточку
- 16. Банкомат выдает наличные и предлагает забрать их клиенту
- 17. Клиент получает наличные
- 18. Банкомат отображает сообщение о готовности к дальнейшей работе

Раздел Исключений

- Исключение №1. Кредитная карточка недействительна (утрачена)
- 4. Банкомат блокирует кредитную карточку
- 18. Банкомат отображает сообщение о готовности к дальнейшей работе

- Исключение №2: Кредитная карточка просрочена
- 4. Банкомат предлагает клиенту забрать его кредитную карточку
- 15. Клиент получает свою кредитную карточку
- 18. Банкомат отображает сообщение о готовности к дальнейшей работе

- Исключение №3. Введенный ПИН-код неверный
- 4. Банкомат предлагает ввести ПИН-код
- 5. Клиент вводит ПИН-код

Сценарий №2 "Получение справки о состоянии счета"

Главный раздел

- *Вариант использования:* Получение справки о состоянии счета
- *Актеры:* Клиент Банкомата, Банк
- *Цель:* Получение информации о балансе
- *Краткое описание:* Клиент использует свою карточку для получения справки о состоянии счета. Банкомат обеспечивает доступ к счету клиента. Банкомат выдает клиенту справку в форме чека.
- *Тип:* Базовый
- *Ссылки на другие варианты использования:* Включает в себя ВИ:
 - Проверка ПИН-кода кредитной карточки

Типичный ход событий

- 1. Клиент вставляет кредитную карточку в устройство чтения банкомата
- 2. Банкомат передает информацию о кредитной карточке в Банк
- 3. Банк проверяет информацию о кредитной карточке
- Исключение №1: Кредитная карточка недействительна (утрачена)
- Исключение №2: Кредитная карточка просрочена
- 4. Банкомат предлагает ввести ПИН-код
- 5. Клиент вводит PIN-код
- 6. Банкомат проверяет ПИН-код
- Исключение №3: Введенный ПИН-код неверный
- Исключение №4: Клиент ввел неверный ПИН-код 3 раза

Типичный ход событий

- 7. Банкомат отображает опции меню
- 8. Клиент выбирает получение справки о состоянии счета
- 9. Банкомат делает соответствующий запрос в Банк
- 10. Банкомат предлагает клиенту забрать его кредитную карточку
- 11. Клиент получает свою кредитную карточку
- 12. Банкомат выдает справку о состоянии счета и предлагает забрать ее клиенту
- 13. Клиент получает справку о состоянии своего счета
- 14. Банкомат отображает сообщение о готовности к дальнейшей работе

Раздел Исключений

- Исключение №1. Кредитная карточка недействительна (утрачена)
- 4. Банкомат блокирует кредитную карточку
- 14. Банкомат отображает сообщение о готовности к дальнейшей работе
- Исключение №2: Кредитная карточка просрочена
- 4. Банкомат предлагает клиенту забрать его кредитную карточку
- 11. Клиент получает свою кредитную карточку
- 14. Банкомат отображает сообщение о готовности к дальнейшей работе
- Исключение №3. Введенный ПИН-код неверный
- 4. Банкомат предлагает ввести ПИН-код
- 5. Клиент вводит ПИН-код
- Исключение №4: Клиент вводит неверный ПИН-код 3 раза
- 4. Банкомат блокирует кредитную карточку
- 18. Банкомат отображает сообщение о готовности к дальнейшей работе

Раздел Исключений

- Исключение №4: Клиент вводит неверный ПИН-код 3 раза
- 4. Банкомат блокирует кредитную карточку
- 18. Банкомат отображает сообщение о готовности к дальнейшей работе
- Исключение №5. Требуемая сумма превышает сумму на счете клиента
- 9. Банкомат предлагает ввести новую сумму
- 10. Клиент вводит новую требуемую сумму
- Исключение №6: Клиент выбрал печать чека
- 16.1. Банкомат предлагает клиенту забрать чек
- Примечание. Клиент может отказаться от выполнения транзакции "Снятие наличных по кредитной карточке" при введении ПИН-кода, при выборе типа транзакции и при вводе суммы.

Примеры использования диаграммы УС

- Существует два вида принципиально разных диаграмм случаев использования - для ПО и для всей системы в целом. Ведь, как правило, ПО является частью более крупной системы. Последняя может включать другое ПО, а также некоторый бизнес-процесс. Пользователями такой системы будут различные клиенты системы (бизнес-актеры), поскольку система создается именно для них. А сама система будет предоставлять для них бизнес-случаи использования. Пример диаграммы бизнес-случаев использования для системы обработки телефонных заявок

Пример диаграммы бизнес-случаев использования для системы обработки телефонных заявок



Диаграмма вариантов использования. Бизнес УС.

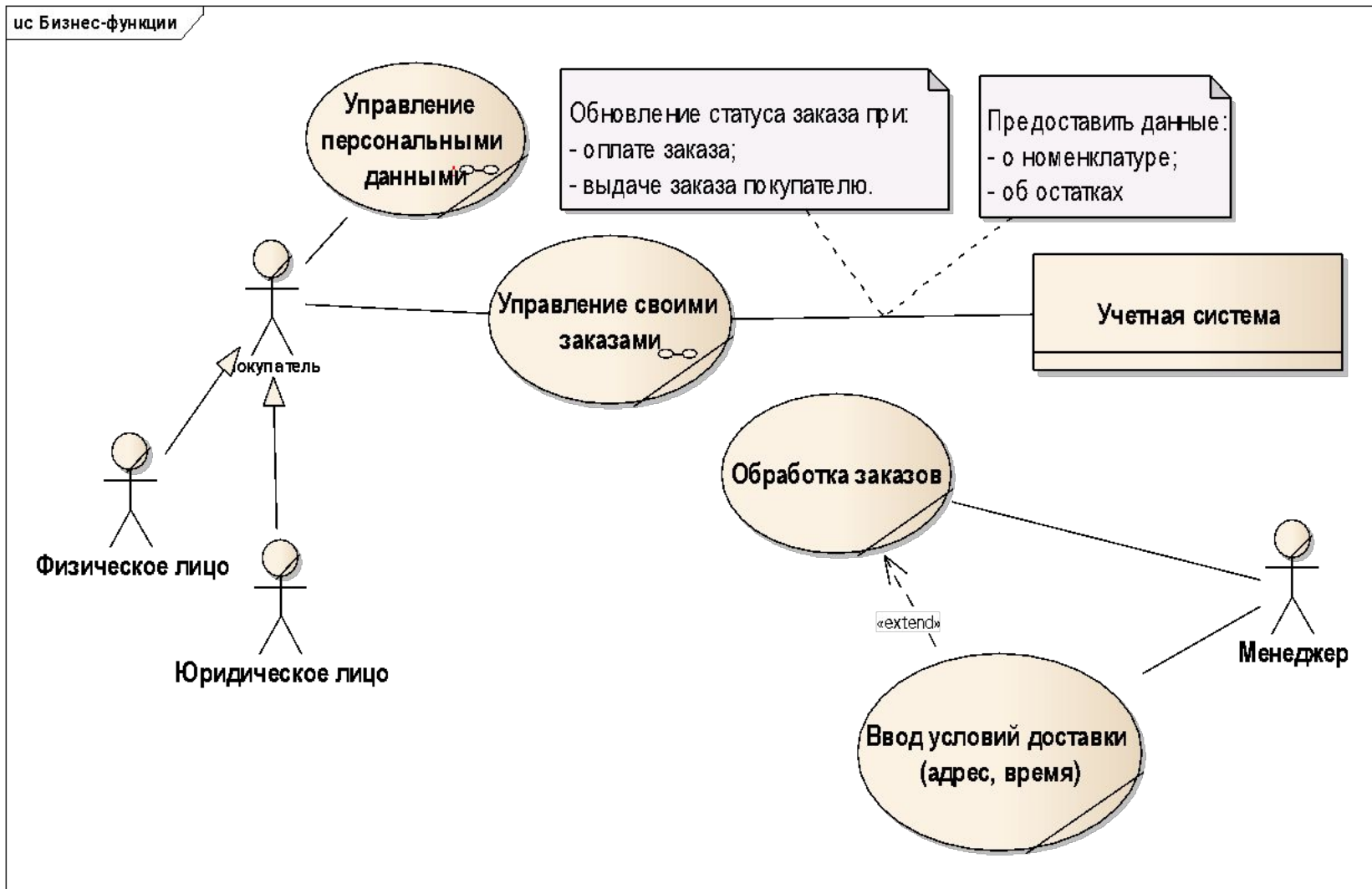


Диаграмма УС.

Моделирование контекста системы

Диаграмма УС. Моделирование требований к системе