

# Варианты использования

# Документация, сопровождающая вариант использования

- \* Для пояснения варианта использования он может сопровождаться следующей **документацией**:
  - \* **комментарии и замечания** к варианту использования;
  - \* **requirements – требования** к варианту использования:
    - \* **функциональные** – т.е. тот функционал, который должен обеспечивать вариант использования;
    - \* **не функциональные** - например, производительность;

# Документация, сопровождающая вариант использования

- \* **constraints** – ограничения:
  - \* **preconditions** – предусловия, т.е. ограничения, которые должны выполняться до исполнения варианта использования;
  - \* **post conditions** – постусловия, т.е. требования, которые должны выполняться после исполнения варианта использования;
  - \* **invariants** – инварианты, т.е. ограничения, которые должны исполняться до, во время и после исполнения варианта использования;

# Документация, сопровождающая вариант использования

- \* **алгоритм исполнения** варианта использования, описанный в одной из форм:
  - \* **рассказ** или обзор (narrative);
  - \* **сценарий** (scenario);
  - \* **диалог** (conversation);

# Документация, сопровождающая вариант использования

- \* **диаграммы поведения** (activity or/and sequence diagrams) – которые представляют собой графическое отображение алгоритма исполнения варианта использования и строятся на основе словесного описания этого алгоритма.

# Документация, сопровождающая вариант использования

- \* **дополнительная информация**

- \* **version** – версия;

- \* **status** – статус или состояние варианта использования:

- \* **proposed** – предлагаемый;

- \* **approved** – одобренный, исправленный;

- \* **mandatory** – обязательный, предписанный;

- \* **validated** – действительный, имеющий силу;

- \* **implemented** – реализованный;

- \* **complexity** – сложность варианта использования:

- \* **easy** – простой;

- \* **medium** – средний;

- \* **difficult** – трудный;

- \* **другая информация.**

# Уровни абстракции описания варианта использования

- \* Словесное описание вариантов использования, принадлежащих одной диаграмме, может быть дано на следующих уровнях абстракции:
  - \* **summary** – общее описание бизнес процессов;
  - \* **core** – детальное описание бизнес процессов или задач, которые пользователь решает при взаимодействии с системой;
  - \* **supporting** – описание низкоуровневых активностей, используемых для поддержки исполнения бизнес процессов;
  - \* **internal** – описание поведения и взаимодействия внутренних компонент системы.

# Описание алгоритма исполнения варианта использования

- \* Каждый вариант использования должен иметь описание **алгоритма** своего исполнения.
- \* Существуют три вида описания алгоритма исполнения варианта использования:
  - \* **narrative (рассказ)** - представляет собой обзор исполнения варианта использования;
  - \* **scenario (сценарий)** - представляет собой описание последовательности действий и событий при исполнении варианта использования;
  - \* **conversation (разговор)** - диалог, представляет собой описание взаимодействия актера с системой в форме диалога.



# Рассказ

- \* **Рассказ** (Narrative form) – это текст в свободной форме, разбитый на параграфы.
- \* Назначение:
  - \* описать **намерения** (цели) пользователя при исполнении варианта использования;
  - \* описать на высоком уровне **действия пользователя** при исполнении варианта использования;
  - \* описать **ключевые концепции** из прикладной области, которые встречаются в варианте использования.

# Сценарий

- \* **Сценарий** (Scenario form) – это текстовое описание одного из путей исполнения варианта использования с точки зрения актера.
- \* **Назначение** сценария - описывает последовательность событий и действий при исполнении варианта использования.
- \* Каждый шаг (пункт) сценария описывается простым предложением и может включать:
  - \* **намерения** актера;
  - \* **действия** системы.

# Виды сценариев

- \* Для описания исполнения варианта использования можно использовать несколько сценариев, которые в этом случае имеют различные цели.
- \* Например, возможны следующие виды сценариев:
  - \* *основной (basic ) сценарий;*
  - \* *альтернативный (alternate) сценарий;*
  - \* *простой (simple) сценарий;*

# Диалог

- \* **Диалог** (Conversation form) – это диалог между актером и системой, в котором делается акцент на их взаимодействии.
- \* Каждое действие может быть описано одним или несколькими шагами и может включать:
  - \* **намерения и действия** актеров;
  - \* **обязанности и действия** системы.

# Сравнение трех форм описания

Форма	Достоинства	Недостатки
Рассказ	- хорошо подходит для краткого описания (summary) варианта использования;- может быть написана, не раскрывая детали реализации;	- тяжело выбрать подходящий уровень детализации описания;- не подходит для описания алгоритмически сложных вариантов использования;- описание может быть неточным;
Сценарий	- хорошо подходит для описания последовательности исполняемых действий;	- трудно описать параллелизм действий;
Диалог	- хорошо подходит для описания взаимодействия актера с системой;- можно показывать параллельные и необязательные действия (optional).	- может быть написана слишком детально;

# Шаблон для описания варианта использования

Описание варианта использования можно разбить на следующие пункты:

1. Имя варианта использования.
2. Преамбула (preamble).
3. Тело описания варианта использования (body).
4. Дополнительные детали и ограничения (supplementary details and constraints).

# Преамбула

- \* **Преамбула** (предисловие, введение) содержит описание окружения (stage), в котором происходит исполнение варианта использования.
- \* Преамбула может содержать следующую информацию:
  - \* уровень варианта использования – summary, core, supporting или internal;
  - \* актеров – имена ролей, инициирующих взаимодействие с вариантом использования;
  - \* контекст (context) – текущие состояния системы и актеров;
  - \* предусловия (preconditions) – условия, которые должны выполняться перед началом исполнения варианта использования;
  - \* окна (screens) – ссылки на окна и веб-страницы, отображаемые в этом варианте использования.

# Тело описания

- \* **Тело описания** варианта использования содержит описание исполнения варианта использования в одной из трех форм:
  - \* рассказ;
  - \* сценарий;
  - \* диалог.



# Дополнительные детали и ограничения

## \* Разделы:

- \* **варианты** (variations) – различные способы исполнения варианта использования;
- \* **исключения** (exceptions) – ошибки, которые могут произойти при исполнении варианта использования;
- \* **политики** (policies) – специфические правила, которые обязательны для исполнения варианта использования;
- \* **вопросы** (issues) – проблемные вопросы по варианту использования;
- \* **постусловия** (post conditions) – условия, которые должны выполняться после исполнения варианта использования;
- \* **другие требования** – другие ограничения, которым должно удовлетворять исполнение варианта использования;
- \* **приоритет** (priority) – важность варианта использования;
- \* **частота** (frequency) – частота исполнения варианта использования.

# Что главное?

- \* Мы НЕ изучаем специфику какой-либо предметной области.
- \* Мы НЕ изучаем инструменты визуального моделирования.
- \* Главное – практическое освоение методологии моделирования

# Тяжело? Есть выход: Графическое описание

# Основные понятия

- \* Автор
- \* Вариант использования
- \* Субъект
- \* Ассоциация
- \* Отношение расширения
- \* Отношение включения
- \* Отношение обобщения

# Цели использования диаграммы вариантов использования

- \* Определить общие границы и контекст моделируемой предметной области на начальных этапах проектирования системы.
- \* Сформулировать общие требования к функциональному поведению проектируемой системы.
- \* 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.

# Вариант использования (use case)

**Вариант использования (прецедент)** – классификатор, который описывает совокупность сценариев взаимодействия актеров с системой или компонентом с целью достижения какой-либо цели, значимой для актеров.

Варианты использования могут различаться по уровню цели: высокоуровневые цели, пользовательские цели, отдельные функции системы.

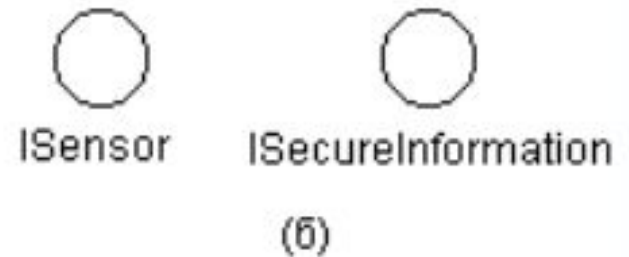


# Актер (актор)

- \* Актер – классификатор, который моделирует внешнего по отношению к моделируемой системе или компоненту пользователя или систему.
- \* Актеров, которые используют системы для достижения собственных целей, называют основными.
- \* Актеров, которых система использует для достижения целей других актеров, называют второстепенными.



# Интерфейсы





# Интерфейсы

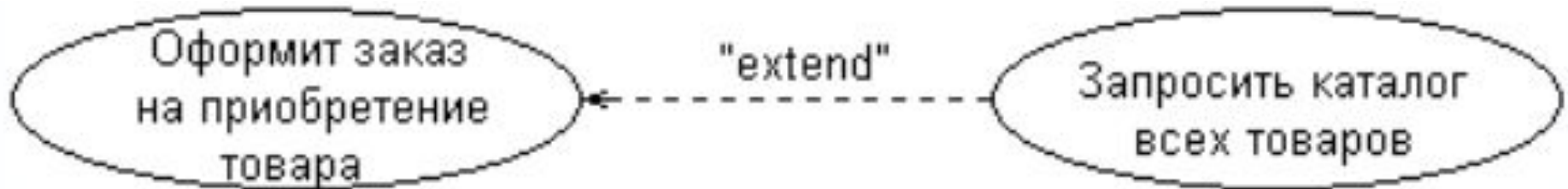
- \* Если интерфейс соединяется с вариантом использования сплошной линией, то этот вариант использования должен реализовывать все операции, необходимые для данного интерфейса, а возможно и больше.
  - \* **Приведите свой пример**
- \* Если интерфейс соединяется с вариантом использования пунктирной линией со стрелкой, это означает что вариант использования предназначен для спецификации только того сервиса, который необходим для реализации данного интерфейса.
  - \* **Приведите свой пример**

# Ассоциация



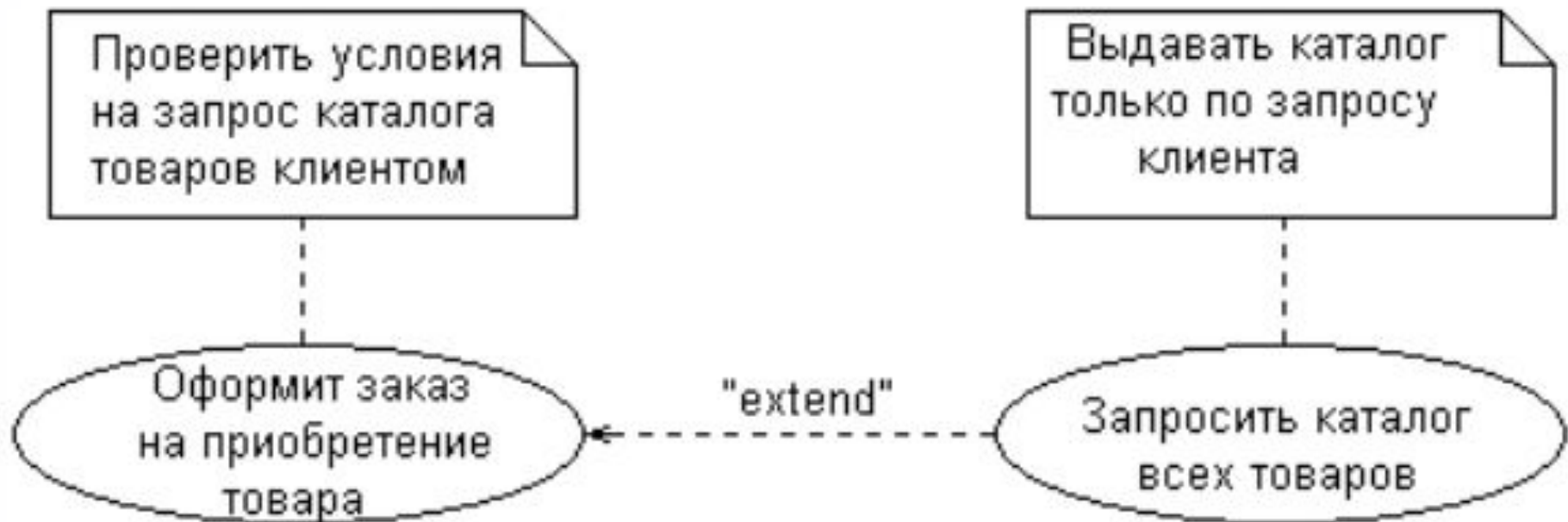
- \* Ассоциация актера с вариантом использования указывает на взаимодействие актера с субъектом в одном из сценариев данного варианта использования

# Расширение



- \* Отношение расширения между вариантами использования указывает, что при выполнении заданного в точке расширения условия сценарий расширяемого варианта использования будет приостановлен, и взаимодействие будет продолжено в рамках расширяющего варианта использования

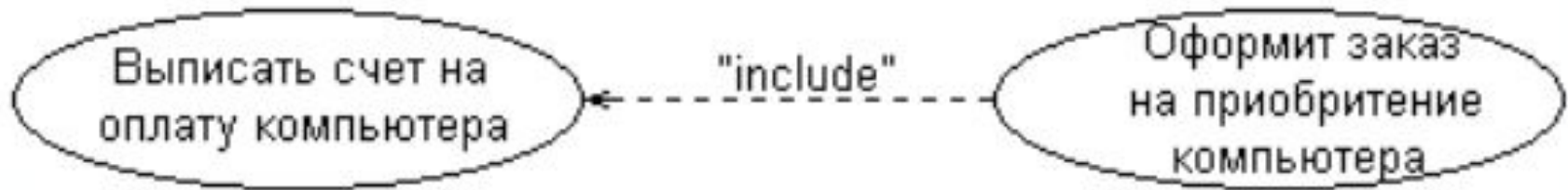
# Отношение расширения



# Когда использовать отношение расширения?

- \* Имеются дополнительные субъекты, участвующие только в случае использования расширения (например, администратор должен утвердить регистрацию клиента на веб-сайте).
- \* Отдельная подсистема обрабатывает вариант использования расширения.
- \* Это расширение будет доступно только в определенных версиях системы. Каждую версию можно показать как отдельную подсистему

# Включение



- \* Отношение включения указывает, что в процессе выполнения, сценарии базового варианта использования вызывают выполнение сценариев включаемого варианта использования.

# Отношение включения

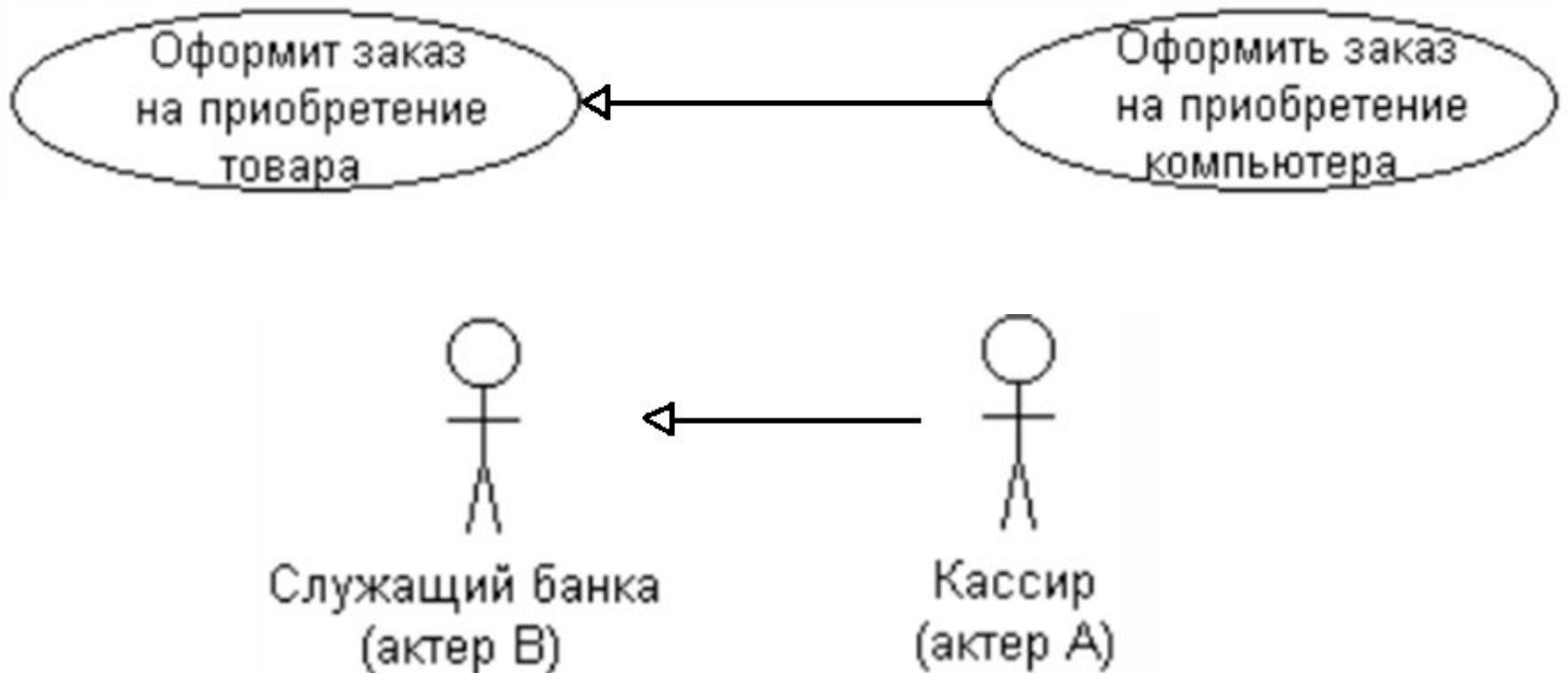
- \* Один вариант использования может быть включен в несколько других вариантов, а также включать в себя другие варианты.
- \* Включаемый вариант использования может быть независимым от базового варианта, т.к. он предоставляет последнему некоторое инкапсулированное поведение, детали реализации которого скрыты от последнего и могут быть легко перераспределены между несколькими включаемыми вариантами использования.
- \* Базовый вариант может зависеть только от результатов выполнения включаемого в него поведения, но не от структуры включаемых в него вариантов.

# Отношение включения

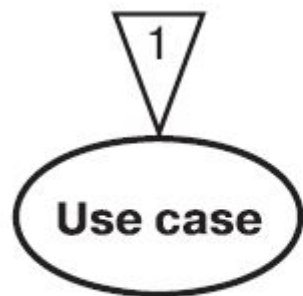
- \* Все экземпляры варианта использования выполняются лишь внутри данной сущности.
- \* Как сделать так, чтобы отношения, определенные в рамках одной сущности, были бы определены и в рамках другой сущности?



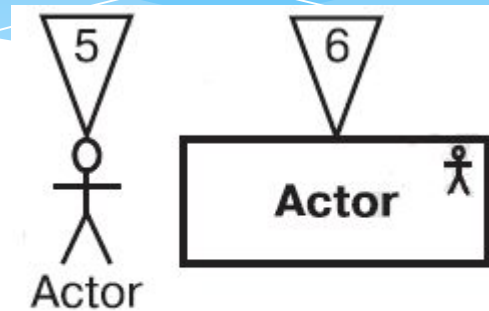
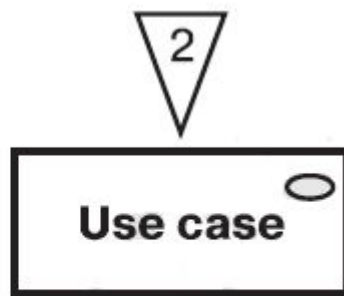
# Обобщение



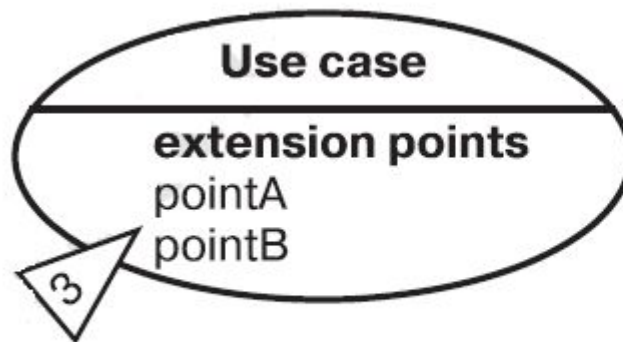
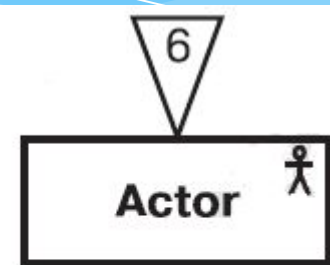
# Графическое отображение



вариант использования  
(use case)



действующее лицо  
(actor)



точка расширения (extension point)

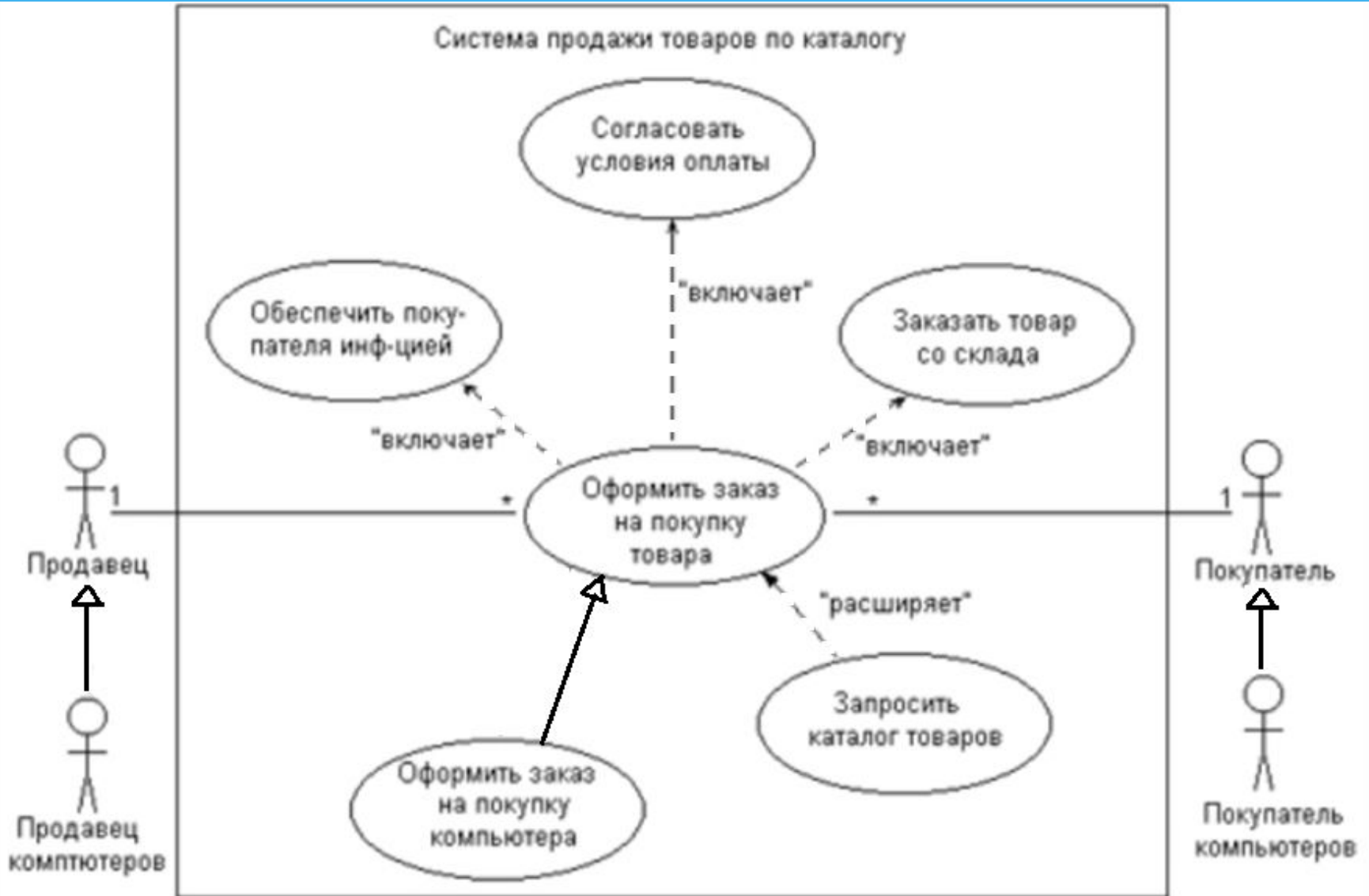
# Пример: исходная диаграмма вариантов использования



# Пример: уточненный вариант диаграммы



# Пример: последующее уточнение диаграммы



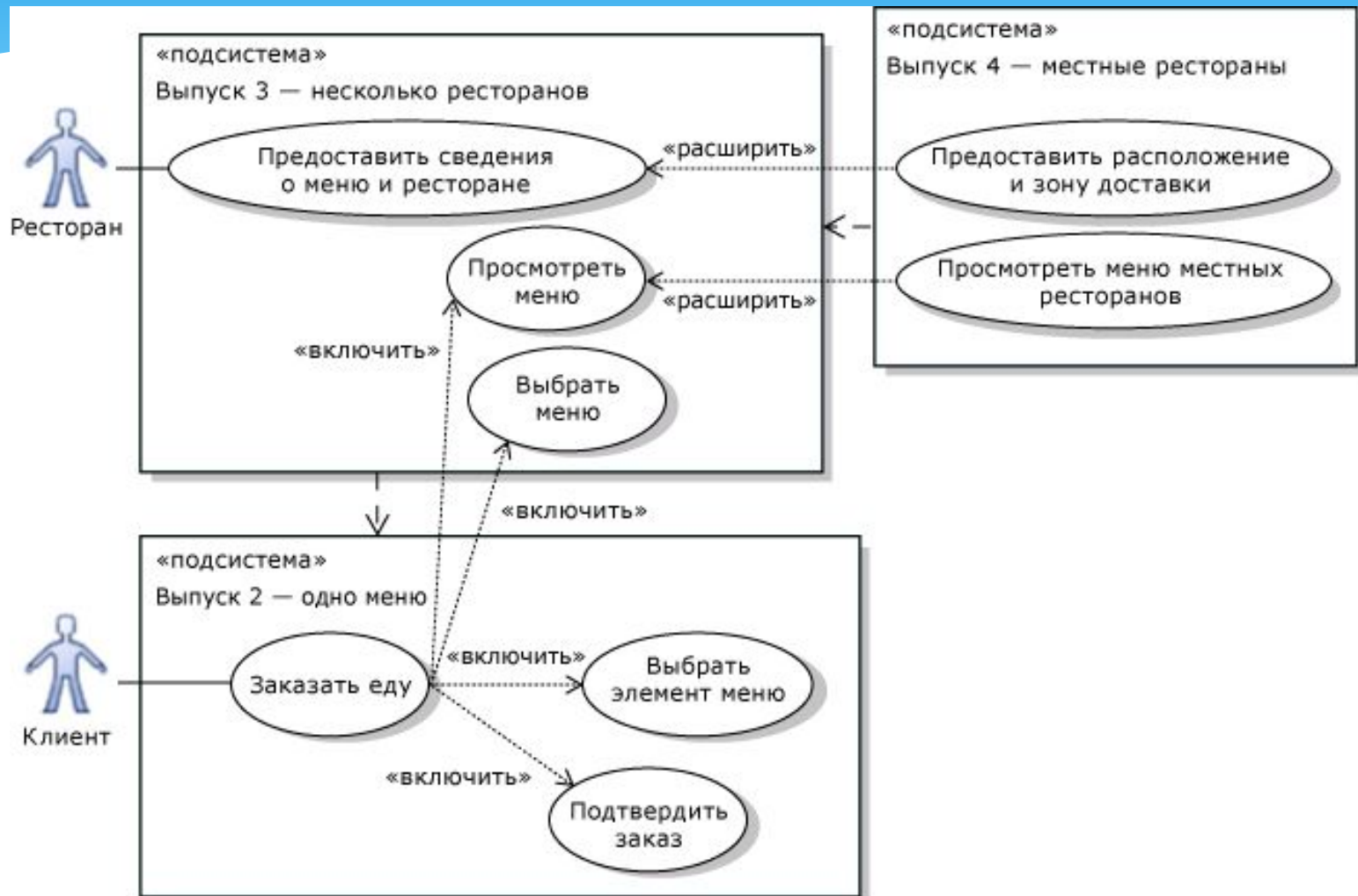
# Пример: последующее уточнение диаграммы

- \* Чего не хватает на предыдущей диаграмме?
- \* При затруднении ответьте на вопросы:
  - Какой актер имеет возможность оформить заказ на покупку компьютера?
  - Любой ли продавец имеет право оформить покупку компьютера?
  - Кто и когда запрашивает каталог товаров?

# Пример 2. Начальная диаграмма



# Пример 2. Подсистемы на диаграмме вариантов использования





# Цели, которые преследуют варианты использования

- \* Диаграмма вариантов использования может не только преследовать цели пользователей, но также может включать описание способов реализации сервиса и различных исключительных ситуаций (например, корректная обработка ошибок).

# Цели, которые преследуют варианты использования

- \* Множество вариантов использования в целом должно определять все возможные стороны ожидаемого поведения системы.
- \* Для удобства множество вариантов использования может рассматриваться как отдельный пакет.

# Особенности

- \* Каждый выполняемый вариантом использования метод реализуется как неделимая транзакция, т.е. выполнение сервиса не может быть прервано никаким другим экземпляром варианта использования.

# Сценарии использования

- \* Сценарий - это конкретная последовательность действий, иллюстрирующая поведение.
- \* Написание сценария напоминает написание художественного рассказа, использование сценариев широко распространено среди аналитиков.

...

Получить от пользователя код

Проверить его правильность

Показать подтверждение

...

# Сценарии и прецеденты

- \* Прецеденты рождаются из требований к системе. Они говорят о том, что делает система.
- \* Сценарии говорят, как система это делает.
- \* *Сценарии специфицируют прецеденты*
- \* *Диаграммы прецедентов, дополненные сценариями, являются отличным средством общения между разработчиками и заказчиком.*

# Шаблон описания сценария

Главный раздел	Типичный ход события	Исключения	Примечания
Имя варианта использования	Типичный ход событий, приводящий к успешному выполнению варианта использования (текстовое описание)	Исключение № 1 Текстовое описание хода событий	Примечание № 1 Текстовое описание
Актеры, его использующие		Исключение № 2 Текстовое описание хода событий	Примечание № 2 Текстовое описание
Цель		...	...
Краткое описание			
Тип			
Ссылки на другие варианты использования		Исключение № N Текстовое описание хода событий	Примечание № N Текстовое описание

# Прецедент и кооперация

- \* Прецедент и кооперация находятся в отношении реализации.
- \* Каждый прецедент реализуется одной или несколькими кооперациями.

# Выводы, рекомендации

- \* Количество актеров – не более 20
- \* Количество вариантов использования – не более 50
- \* Не давать актерам имена собственные, т.к. даже конкретный объект может играть различные роли и использовать различные варианты использования
- \* Все сервисы системы должны быть явно определены в диаграмме вариантов использования
- \* Любые другие сервисы, которые не отражены на диаграмме вариантов использования, системы исполнять не может



# Типичные ошибки при разработке диаграммы вариантов использования

- \* Превращение диаграммы прецедентов в диаграмму деятельности за счет желания отразить все функциональные действия
- \* Инициатором действия является разрабатываемая система
- \* Спецификация атрибутов и операций классов до того, как сформулированы все варианты использования
- \* Задание слишком кратких имен прецедентов
- \* Отсутствие описаний альтернативных действий

# Немного практики

# 1. Найдите пользователя и цель, опишите сценарий

Для кого сделан функционал? Системы не делают ради систем. Мы хотим, чтобы пользователь:

- \* сделал заказ;
- \* заплатил налоги;
- \* посмотрел на котика;
- \* прочитал блог-пост;

Всегда есть цель, которую пользователь должен достичь. И основной путь, заложенный в коде. Вот его и опишите.

Чтобы написать вариант, надо подумать о будущем пользователе. Зачем он к нам придет? Что будет делать и как?

# Основной сценарий

## Сценарий использования

- \* Юзер открывает список товаров и фильтрует по категории.
- \* Система отображает товары выбранной категории.
- \* Юзер видит интересный товар и переходит на его карточку.
- \* Система отображает карточку товара, оценку покупателей и отзывы.
- \* Юзер изучает товар и кладет его в корзину.
- \* Система добавляет товар в корзину.
- \* Юзер переходит в корзину и оформляет заказ.
- \* Система сохраняет заказ, отправляет уведомление по email.

## 2. Продумайте альтернативы

Альтернативные сценарии - это отклонения от основной ветки.

- \* отменил все на этапе фильтрации товаров;
- \* отменил все, находясь в корзине;
- \* открыл в соседней вкладке `bash.org`;
- \* пошел налить чай;
- \* наклонился погладить кота;
- \* обожрался вкусняшек;

Не забываем о системе и внешних условиях:

- \* Внутренний сбой программы на любом этапе.
- \* Компьютер сдох. Сценарий завершился.
- \* Кот уронил сладкий кофе на клавиатуру.
- \* Уборщица шваброй выдернула сервер из розетки.
- \* На сервер упал метеорит.

# 3. Выделите параметры

Параметры — это когда одну операцию можно выполнить разными способами, но сам вариант от этого не меняется.

Параметры выглядят так:

название: ... значение 1, значение 2, значение 3 - **без** привязки к конкретному пункту сценария.

**Запомните:**

- \* **Альтернатива — это когда ВМЕСТО исходного события происходит другое.**
- \* **Параметры — это когда В ОДНОМ И ТОМ ЖЕ событии есть несколько вариаций, как его совершить.**

# 4. Соберите все вместе

Читать удобнее именно в формате:

- \* основной вариант;
- \* альтернативы со ссылками на него;
- \* возможные параметры и особенности реализации.

# Пример

## Легенда

- \* П — пользователь
- \* С — система

## Сценарий использования

1. П открывает список товаров и фильтрует по категории.
2. С отображает товары выбранной категории.
3. П видит интересный товар и переходит на его карточку.
4. С отображает карточку товара, оценку покупателей и отзывы.
5. П изучает товар и кладет его в корзину.
6. С добавляет товар в корзину.
7. П переходит в корзину и оформляет заказ.
8. С сохраняет заказ, отправляет уведомление по email.



# Пример

## Альтернативные варианты

- 1а. П фильтрует список по несуществующей категории. Система выдает ошибку. Завершение сценария.
- 2а. Товаров не найдено. Вывод сообщения об ошибке. Завершение сценария.
- 2б. Товаров слишком много. Система выводит первые 100 и предлагает сузить поиск.
- 5а. П возвращается к покупкам. Переход к шагу 1.

## Параметры

- \* Категории товаров: джинсы, свитера.
- \* Время хранения товара в резерве: 2 часа с момента добавления в корзину.

# Пример описания «по науке»

## Вариант использования 1.

- \* Открытие текстового файла.
- \* Основное действующее лицо: пользователь приложения NotePad, желающий открыть файл
- \* Область действия: текстовый редактор (NotePad)
- \* Уровень: цель пользователя
- \* Участники и интересы:
- \* Пользователь - хочет открыть текстовый файл в выбранном приложении.
- \* Предусловие: программа NotePad у пользователя уже открыта.
- \* Минимальные гарантии: файл открылся в приложении NotePad
- \* Гарантия успеха: файл без ошибок открылся в приложении, все данные файла корректно отражены и имеют удобный для чтения и понимания вид.
- \* Триггер: пользователю необходимо увидеть содержание файла

# Пример описания «по науке»

\* Основной сценарий:

1. Пользователь нажимает на меню "Файл" в верхнем левом углу приложения
2. Пользователь выбирает "Открыть" пункт-меню из предложенных вариантов
3. Приложение открывает диалоговое окно и запрашивает ввести полное имя файла, который пользователь желает открыть для чтения
4. Пользователь вводит полное имя файла, который он хочет открыть или выбирает путь в ручную в навигаторе компьютера
5. Нажимает на кнопку "Открыть" в диалоговом окне
6. Приложение открывает файл. Вся информация из файла отражена корректно и удобна для чтения.

# Вариант описания, чтобы читали



- \* Открытие файла в Notepad.

Легенда:

- \* П — пользователь
- \* Б — блокнот

Вариант.

- \* П инициирует открытие файла.
- \* Б открывает файл
- \* Альтернативы...
- \* Параметры..

# Типовые ошибки

1. **Вариант начинает система** - Иницирует сценарий всегда человек
2. **Система — ванга-терминатор**
3. **Действие по устранению вместо самой альтернативы**

# Итого:

Как написать вариант:

- \* Найти пользователя и цель.
- \* Записать основной сценарий.
- \* Продумать альтернативы.
- \* Выделить параметры.
- \* Отрисовать диаграмму.