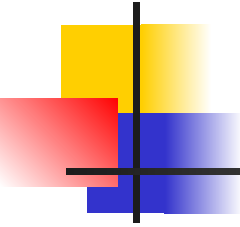


# Информационные технологии

---

- Диаграммы
- Диаграммы вариантов использования (прецедентов)



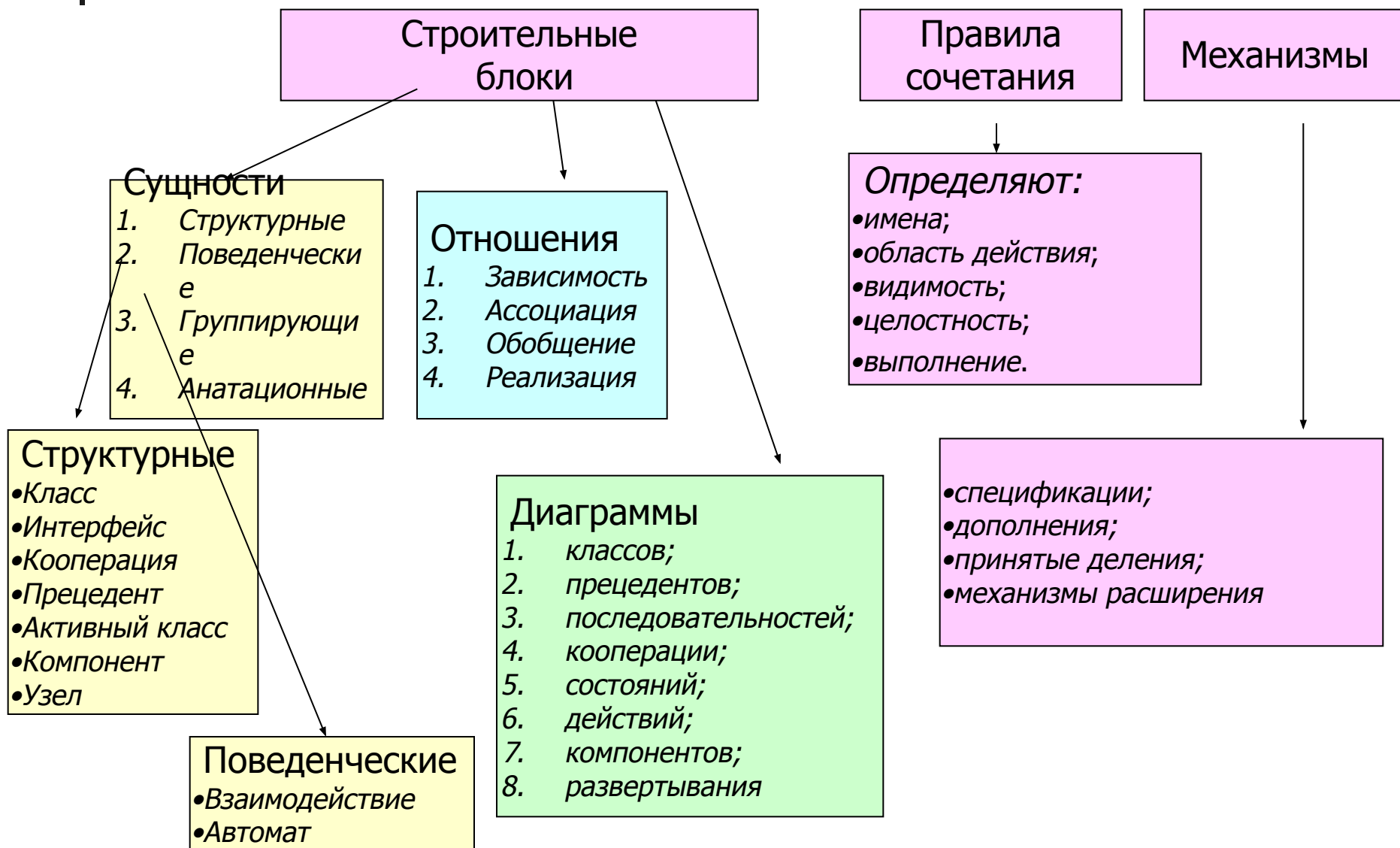
# Варианты использования

---

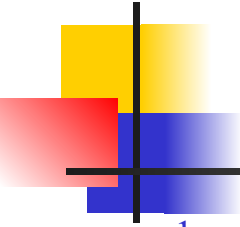
## Литература:

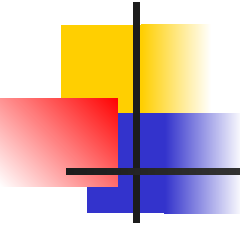
- Буч и др. «Язык UML – руководство пользователя», 1999
- Леоненков «Самоучитель UML» 2003,  
«Нотация и семантика языка UML»  
2006
- Мартин Фаулер «Введение в UML»,  
1998

# Концептуальная модель UML



# Диаграммы UML

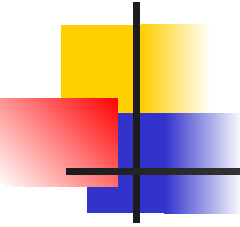
- 
- 1. Диаграмма вариантов использования (use case diagram)
  - 2. Диаграмма классов (class diagram)
  - 3. Диаграммы поведения (behavior diagrams)
    - 1. Диаграмма состояний (statechart diagram)
    - 2. Диаграмма деятельности (activity diagram)
    - 3. Диаграммы взаимодействия (interaction diagrams)
      - 1. Диаграмма последовательности (sequence diagram)
      - 2. Диаграмма кооперации (collaboration diagram)
  - 4. Диаграммы реализации
    - 1. Диаграмма компонентов
  - 5. Диаграмма развертывания
- Статические
- Динамические
- Статические



# Правила построения диаграмм UML

---

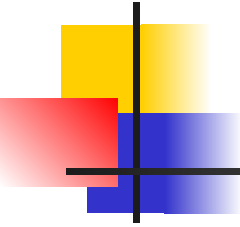
- Каждая диаграмма должна служить законченным представлением
- Все сущности на диаграмме модели должны быть одного концептуального уровня
- Вся информация о сущностях должна быть явно представлена на диаграммах
- Диаграммы не должны содержать противоречивой информации



# Правила построения диаграмм UML

---

- Диаграммы не следует перегружать текстовой информацией
- Количество типов диаграмм для конкретной модели приложения не является строго фиксированным



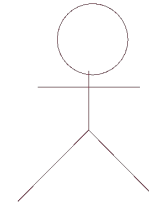
# Цели диаграмм прецедентов

---

- 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.

# Диаграммы прецедентов

- Актер



Actor

(from Use Case View)

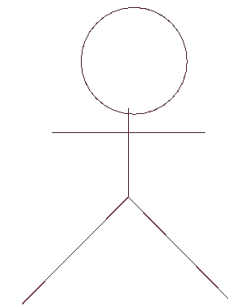
- Прецедент (вариант использования, use case)



UseCase

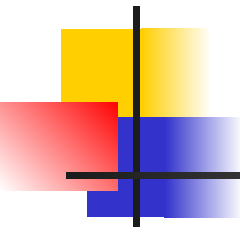
# Актер

- Актер – любая сущность, взаимодействующая с системой извне



Actor

(from Use Case View)

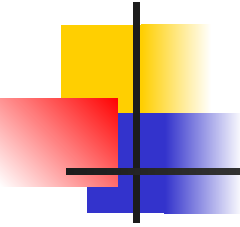


# Актер

---

## Особенности

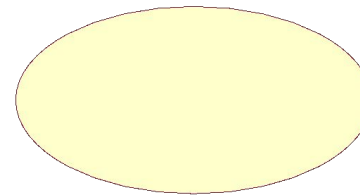
- Актер – это роль
- Может не быть реального человека
- Один человек может играть несколько ролей
- Легче пересчитать актеров...
- События могут выступать актерами..



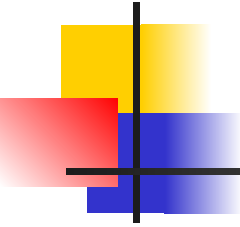
# Вариант использования

---

- Вариант использования – сервисы или некоторый набор действий, которые система предоставляет актеру



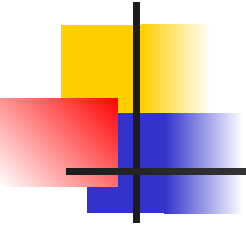
UseCase



# Вариант использования

---

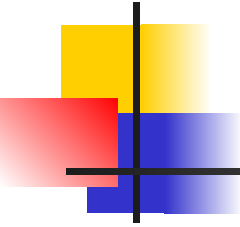
Прецедент – это набор сценариев, которые представляют собой последовательность действий, выполняемых конкретной системой для достижения ощутимого результата для конкретного исполнителя.



# Вариант использования

---

Сценарий – это специальная последовательность действий или взаимодействий между исполнителями и системой

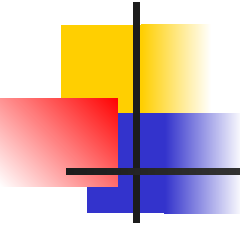


# Вариант использования

---

Сервис представляет собой законченную последовательность действий.

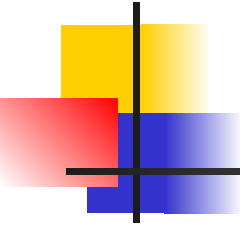
После того как система закончит обработку запроса пользователя, она должна возвратиться в исходное состояние



# Вариант использования

---

Варианты использования могут применяться как для спецификации внешних требований к проектируемой системе,  
так и для спецификации функционального поведения уже существующей системы



# Вариант использования

---

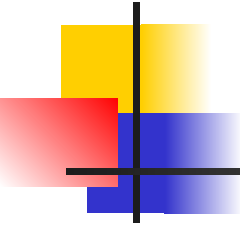
Имя прецедента

- простое «*Разместить заказ*»
- составное «*Датчики:: откалибровать положение*»

# Вариант использования

- Сценарий или примечание – пояснительный текст, который раскрывает смысл или семантику составляющих диаграмму КОМПОНЕНТОВ.



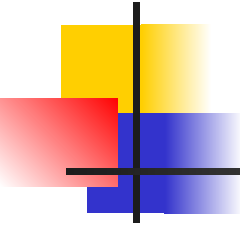


# Вариант использования

---

## Особенности

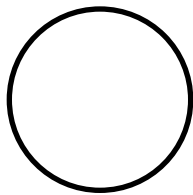
- Use Case – требование к системе
- Нет необходимости рисовать
- 10 человеко-лет – 12 – 100 вариантов использования



# Интерфейс (*Interface*)

---

- интерфейс (*Interface*) – модель поведения системы без указания способа реализации этого поведения

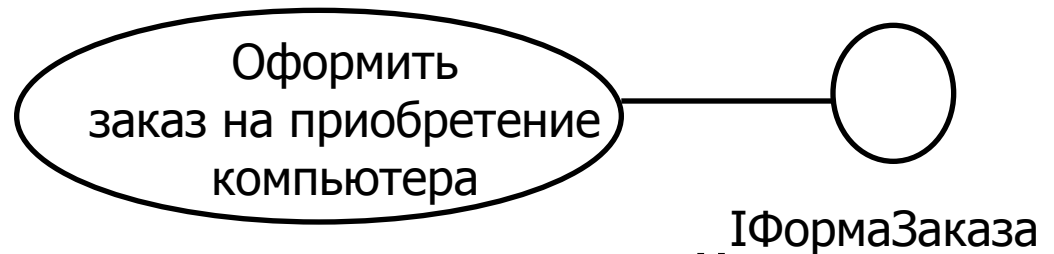


IDocument

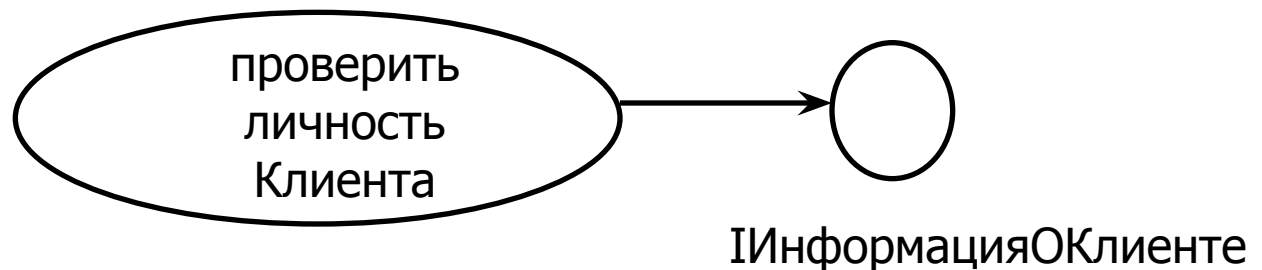
# Вариант использования

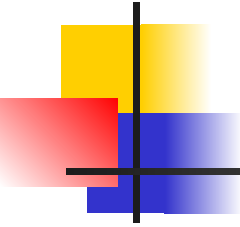
## Вариант использования

- реализует ВСЕ операции



- УС реализует ЧАСТЬ операций





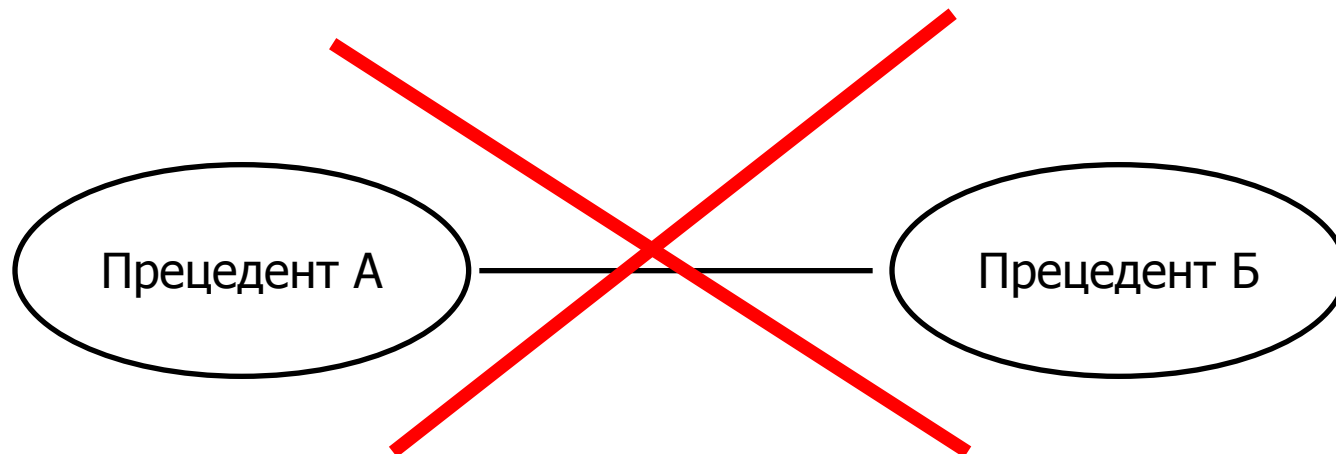
# Отношения прецедентов

---

- ассоциации  
(*association relationship*)  
 $\frac{1..10}{*}$
- расширения  
(*extend relationship*)  
← «extend»
- обобщения  
(*generalization relationship*)  
←
- включения  
(*include relationship*)  
→ «include»

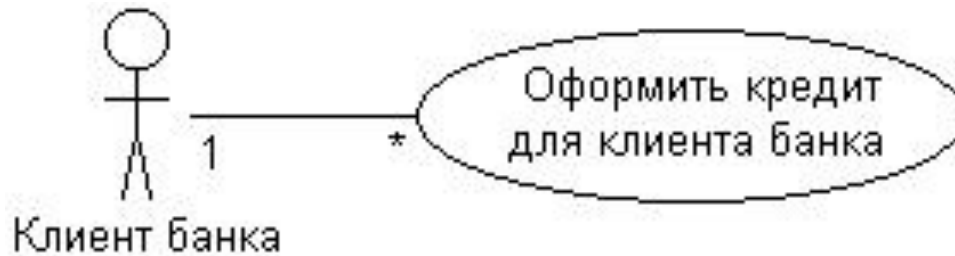
# Отношения ассоциации

общие свойства вариантов  
использования могут быть  
представлены тремя различными  
способами, а именно с помощью  
отношений расширения, обобщения  
и включения



# Отношения ассоциации

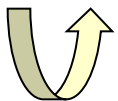
определяет семантические (смысловые) особенности взаимодействия актеров



# Отношения ассоциации

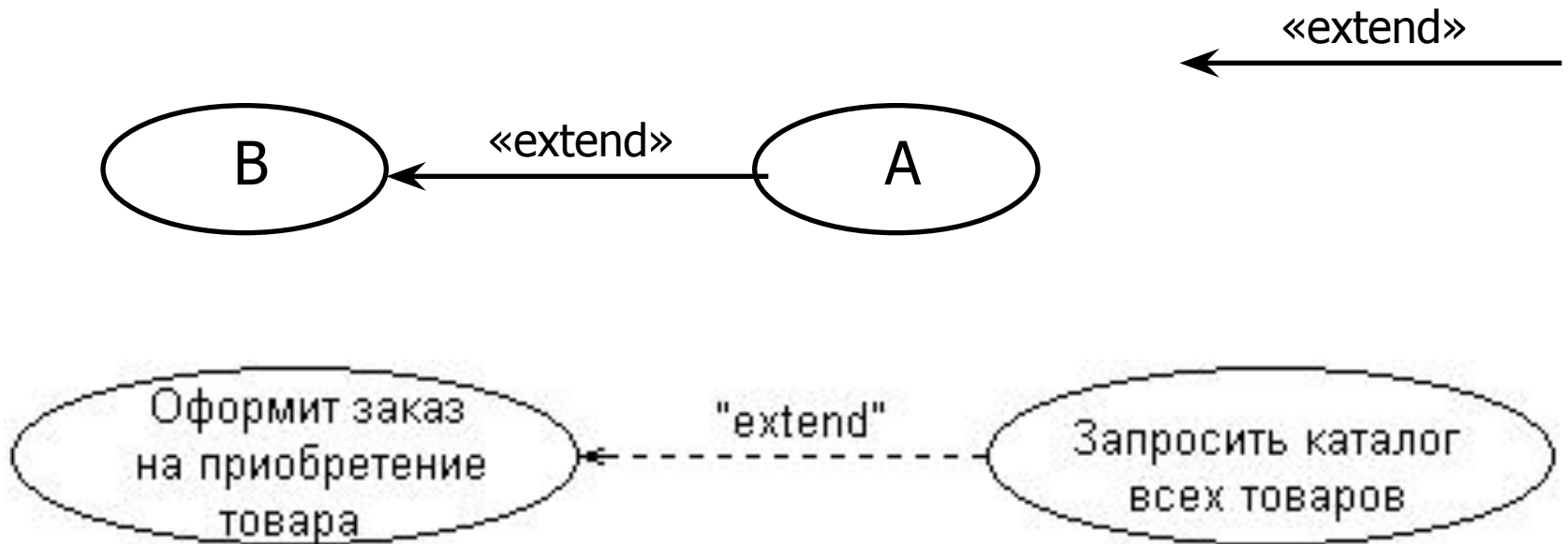
## Кратность (multiplicity)

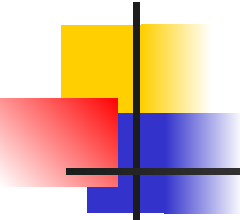
- количество конкретных экземпляров данного компонента, которые могут выступать в качестве элементов данной ассоциации
- 1 (включая 0)
- 1..8
- 2..\*
- \* = 0..\*



# Отношения расширения

свойства варианта использования В  
могут быть дополнены свойствами  
расширенного варианта  
использования А





# Отношения расширения

---

Отношение включает в себя некоторое условие и ссылки на точки расширения в базовом варианте использования

условие отношения расширения проверяется лишь один раз - при первой ссылке на точку расширения

← «extend»

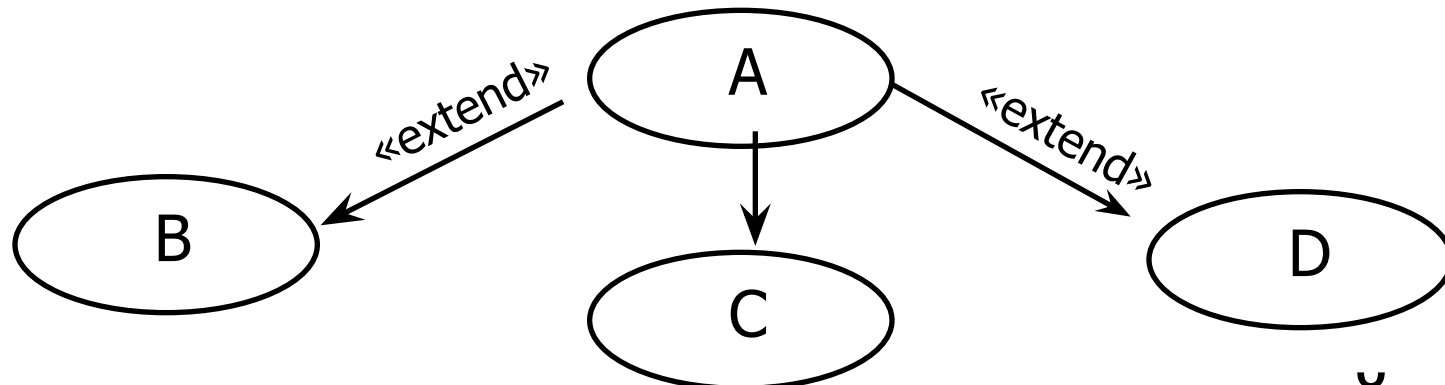
# Отношения расширения

«extend»



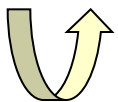
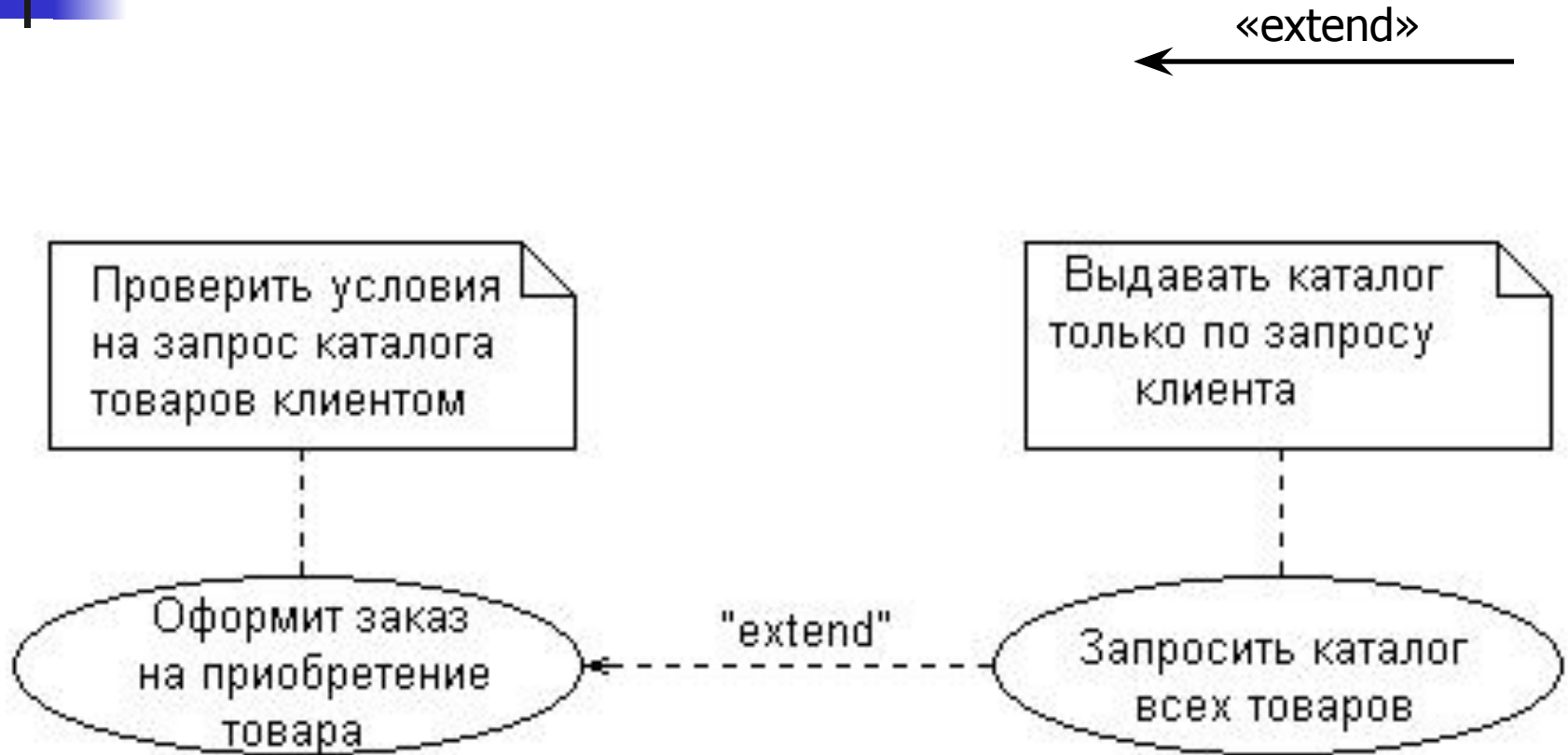
вариант использования может быть

- расширением нескольких других ВИ



- содержать несколько расширений

# Отношения расширения



# Отношения обобщения

служит для указания, что некоторый прецедент А может быть обобщен до прецедент В.

- А – потомок В
- В – предок А



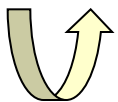
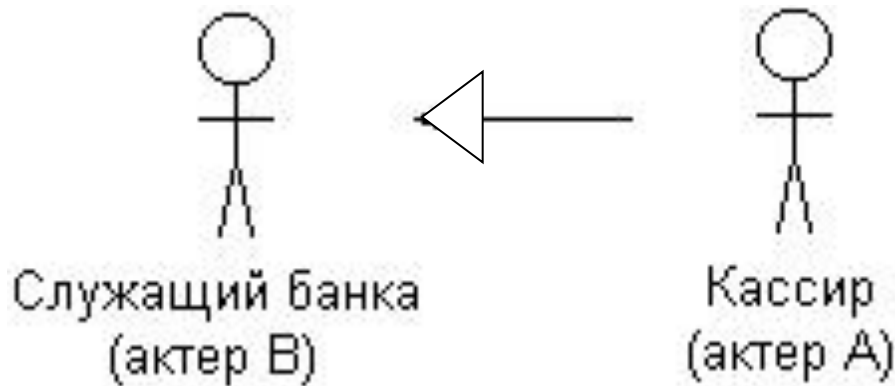
# Отношения обобщения

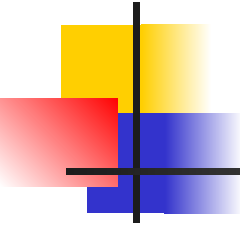
- дочерние прецеденты обладают всеми свойствами предков
- может быть несколько дочерних
- может быть несколько родителей (множественное наследование)



# Отношения обобщения

- отношение обобщения может возникать между актерами





# Отношения включения

---

- поведение одного прецедента включается в качестве составного компонента в последовательность поведения другого прецедента

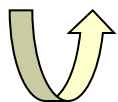
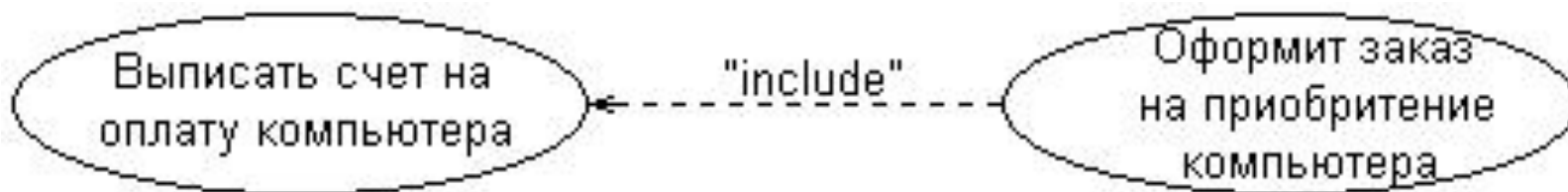
«include»

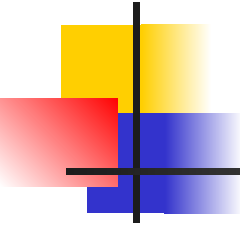


# Отношения включения

Оформить заказ

- заполнить «корзину»
- внести данные покупателя
- выписать счет





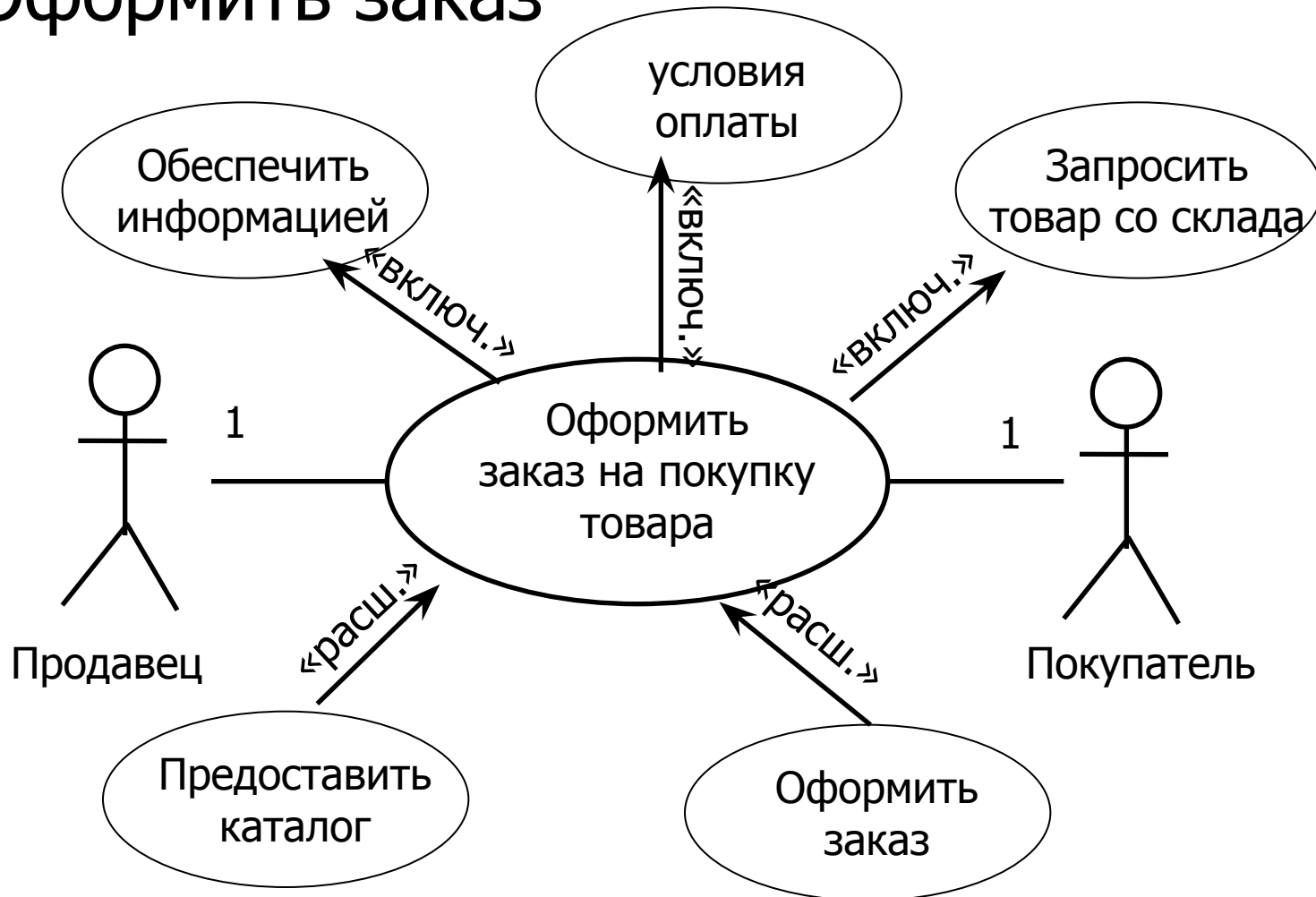
# Пример прецедентов

---

Один вариант использования может быть включен в несколько других вариантов, а также включать в себя другие варианты

# Пример прецедентов

## Оформить заказ



# Расширения

Дополнительные обозначения языка UML для бизнес-моделирования:

***Бизнес-актер (business actor)*** –  
индивидуум, группа, организация,  
компания или система, которые  
взаимодействуют с моделируемой бизнес-  
системой, но не входят в нее



Бизнес-актер

(a)

# Расширения

**Сотрудник (*business worker*)** –  
индивидуум, который действует внутри  
моделируемой бизнес-системы,  
взаимодействует с другими *сотрудниками*  
и является участником бизнес-процесса  
моделируемой системы



Бизнес-сотрудник

# Расширения

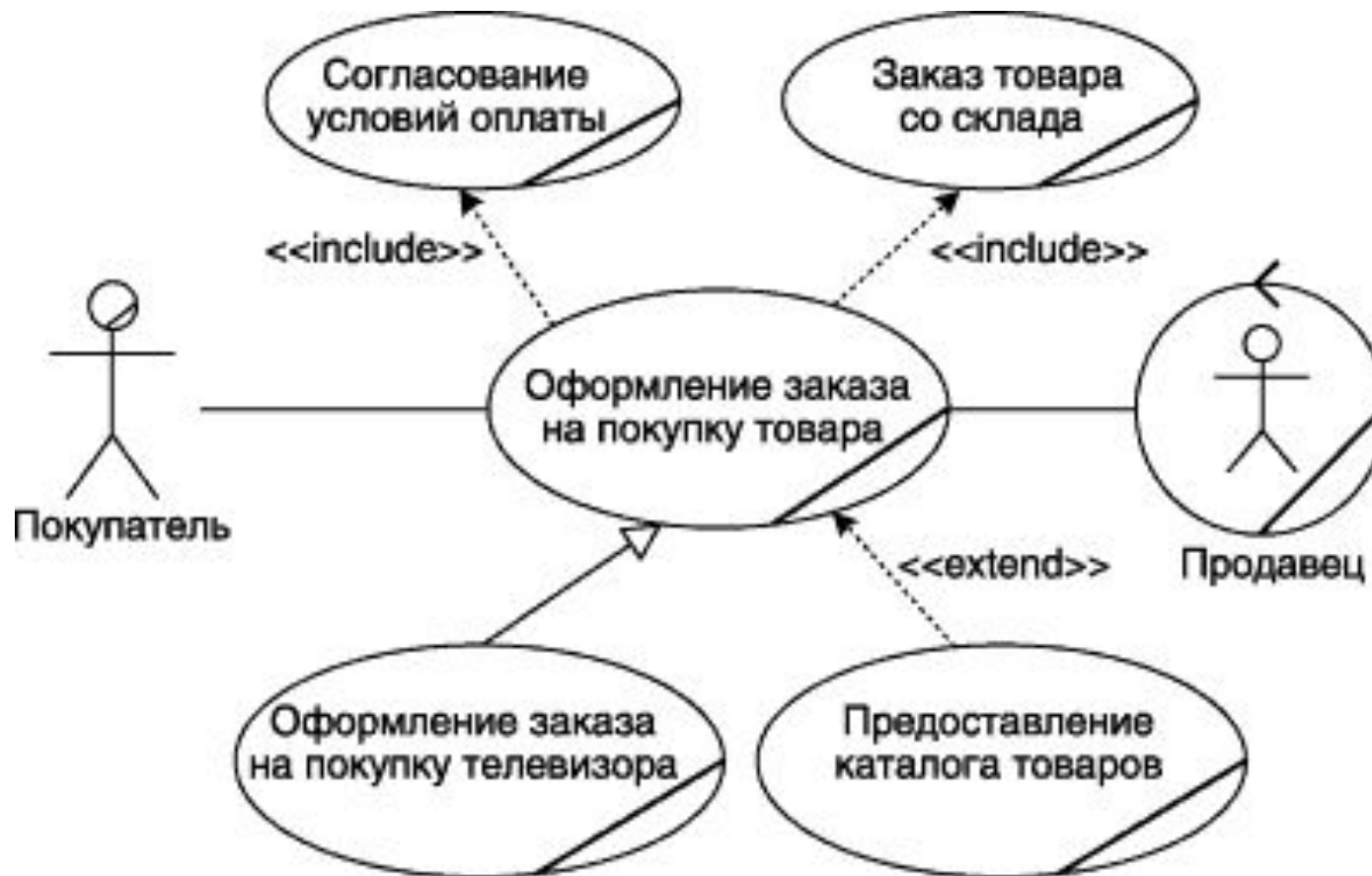
**Бизнес-вариант использования** .  
**(business use case)** — вариант использования, определяющий последовательность действий моделируемой системы, направленных на выполнение отдельного бизнес-процесса

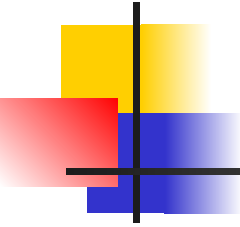


Бизнес вариант использования

# Расширения

## Покупка телевизора

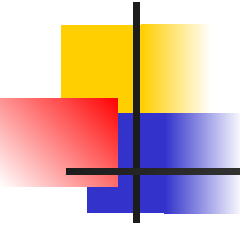




# Рекомендации

---

- Определить главных или первичных и второстепенных актеров
- Определить цели главных актеров по отношению к системе
- Сформулировать основные варианты использования, которые специфицируют *функциональные требования* к системе
- Упорядочить варианты использования по степени убывания риска их реализации



# Рекомендации

---

- Выделить общие варианты использования и изобразить их взаимосвязи с базовыми со стереотипом <<include>>
- Выделить варианты использования для исключений и изобразить их взаимосвязи с базовыми со стереотипом <<extend>>
- Проверить диаграмму на отсутствие дублирования вариантов использования и актеров