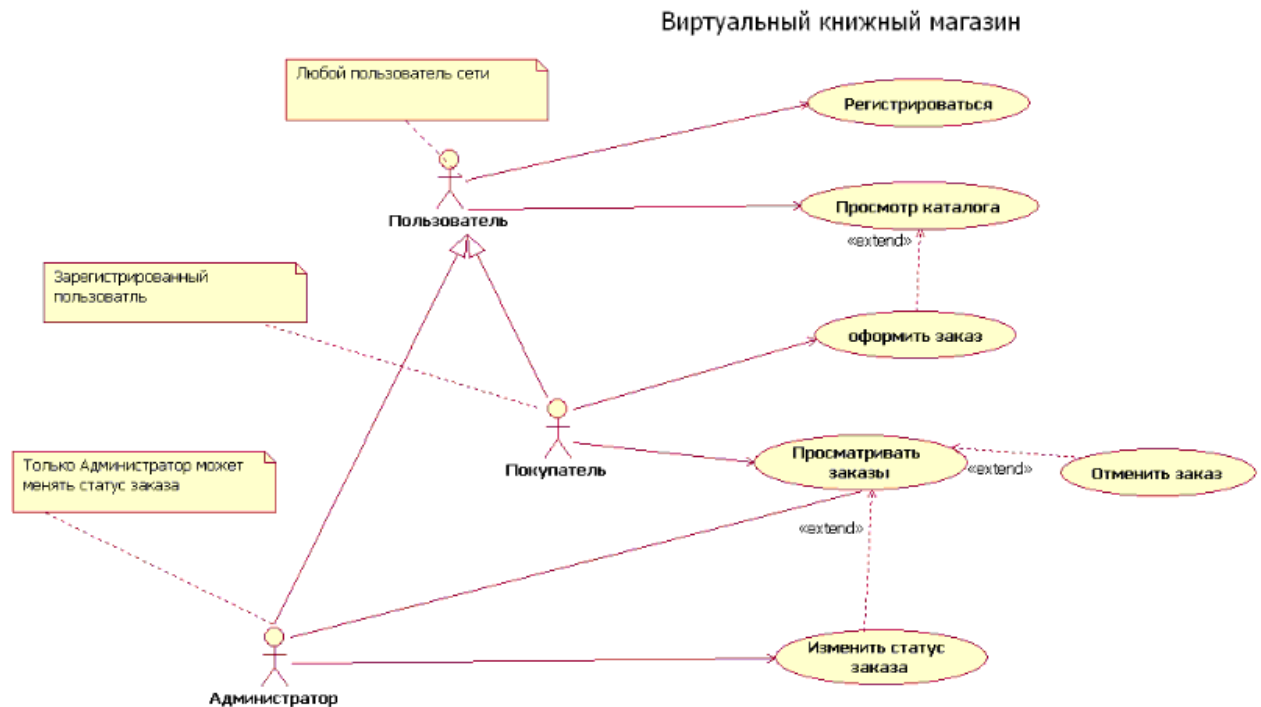


## 22. Диаграмма вариантов использования (use case diagram). Вариант использования. Актеры. Интерфейсы. Примечания. Отношения на диаграмме вариантов использования.



(это вот эта вот штука)

Диаграммы вариантов использования описывают функциональное назначение системы или то, что система должна делать. Разработка диаграммы преследует следующие цели:

- определить общие границы и контекст моделируемой предметной области;
- 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.

Важно понимать, что диаграммы вариантов использования не предназначены для отображения проекта и не могут описывать внутреннее устройство системы. Диаграммы вариантов использования предназначены для упрощения взаимодействия с будущими пользователями системы, с клиентами, и особенно пригодятся для определения необходимых характеристик системы. Другими словами, диаграммы вариантов использования говорят о том, что система должна делать, не указывая сами применяемые методы.

### Вариант использования (use case)

Отдельный **вариант использования** обозначается на диаграмме **эллипсом**, внутри которого содержится его краткое название или имя в форме глагола с пояснительными словами.

Цель варианта использования заключается в том, чтобы определить законченный аспект или фрагмент поведения некоторой сущности без раскрытия её внутренней структуры. В качестве такой сущности может выступать система или любой элемент модели, который обладает собственным поведением.

**Актёр** представляет собой любую внешнюю по отношению к моделируемой системе сущность, которая взаимодействует с системой и использует ее функциональные возможности для достижения определенных целей. При этом актёры служат для обозначения согласованного множества ролей, которые могут играть пользователи в процессе взаимодействия с проектируемой системой. Каждый актёр может рассматриваться как некая отдельная роль относительно конкретного варианта использования. Стандартным графическим обозначением актёра на диаграммах является фигурка человечка, под которой записывается имя актёра.

Каждый вариант использования соответствует отдельному сервису, который предоставляет моделируемая сущность по запросу актёра, то есть определяет способ применения этой сущности. Сервис, который инициализируется по запросу актёра, представляет собой законченную неделимую последовательность действий. Это означает, что после того как система закончит обработку запроса, она должна возвратиться в исходное состояние, чтобы быть готовой к выполнению следующих запросов.

### Интерфейс

**Интерфейс (interface)** служит для спецификации параметров модели, которые видимы извне, без указания их внутренней структуры. Применительно к диаграммам вариантов использования, интерфейсы определяют совокупность операций, которые обеспечивают необходимый набор сервисов для актёров.

На диаграмме вариантов использования интерфейс изображается в виде маленького круга, рядом с которым записывается его имя. В качестве имени может быть существительное или строка текста

Графический символ отдельного интерфейса соединяется на диаграмме сплошной линией или пунктирной линией со стрелкой с тем вариантом использования, который его поддерживает. Сплошная линия указывает, что связанный с интерфейсом вариант использования должен реализовывать все необходимые для него сервисы. Пунктирная линия со стрелкой означает, что вариант использования предназначен для спецификации только того сервиса, который необходим для реализации данного интерфейса.

Таким образом, интерфейс отделяет спецификацию операций системы от их реализации и определяет общие границы проектируемой системы.

### Примечания

Примечания (notes) в языке UML предназначены для включения в модель произвольной текстовой информации, имеющей непосредственное отношение к контексту разрабатываемого проекта. В качестве такой информации могут быть комментарии разработчика (например, дата и версия разработки диаграммы или ее отдельных компонентов), ограничения (например, на значения отдельных связей или экземпляры сущностей) и помеченные значения.

Графически примечания обозначаются прямоугольником с загнутым верхним правым углом. Внутри прямоугольника содержится текст примечания.

Если в примечании указывается ключевое слово «constraint», то оно является **ограничением, налагаемым на соответствующий элемент модели.**

### Отношения

Отношение (relationship) — семантическая связь между отдельными элементами модели.

Один актёр может взаимодействовать с несколькими вариантами использования. В этом случае этот актёр обращается к нескольким сервисам данной системы. В свою очередь один вариант использования может взаимодействовать с несколькими актёрами, предоставляя для всех них свой сервис.

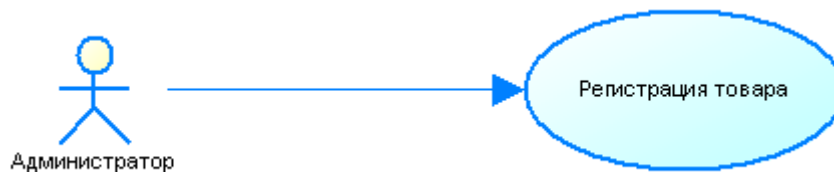
Примеры отношений:

**Ассоциация (association)** – отношение между актером и вариантом использования.

Обозначается сплошной линией между актером и вариантом использования.

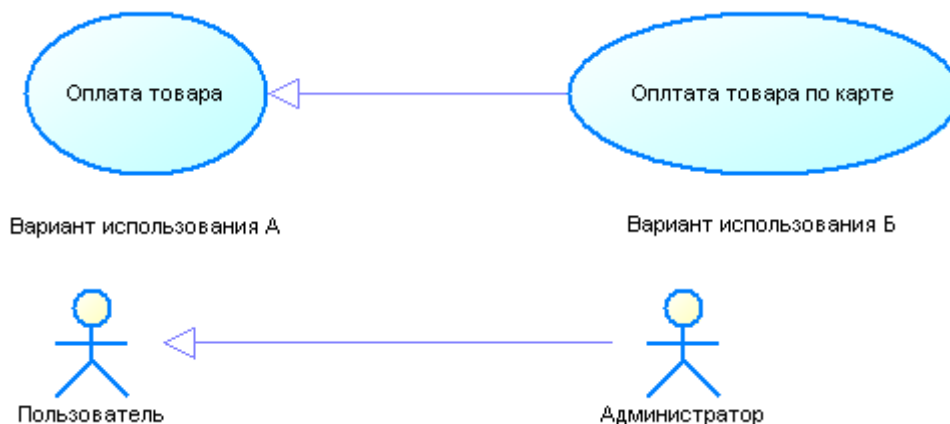


**Направленная ассоциация (directed association)** – то же что и простая ассоциация, но показывает, что вариант использования инициализируется актером. Обозначается стрелкой.



**Отношение обобщение (generalization)** – показывает, что потомок наследует атрибуты и поведение своего прямого предка, т.е. один элемент модели является специальным или частным случаем другого элемента модели. Может применяться как для актеров, так для вариантов использования.

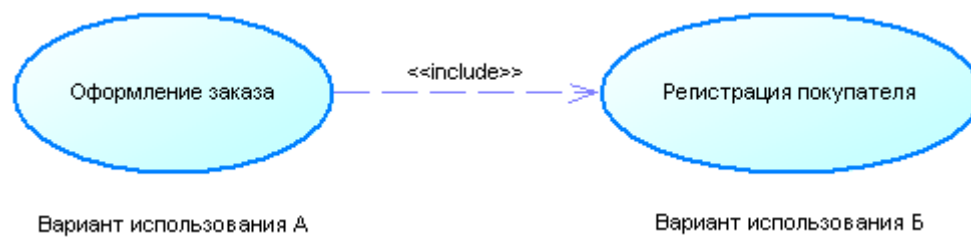
Графически отношение обобщения обозначается сплошной линией со стрелкой в форме не закрашенного треугольника, которая указывает на родительский вариант использования.



Потомки участвуют во всех отношениях родительских вариантов использования. В свою очередь, потомки могут наделяться новыми свойствами поведения, которые отсутствуют у родительских вариантов использования, а также уточнять или модифицировать наследуемые от родителей свойства поведения.

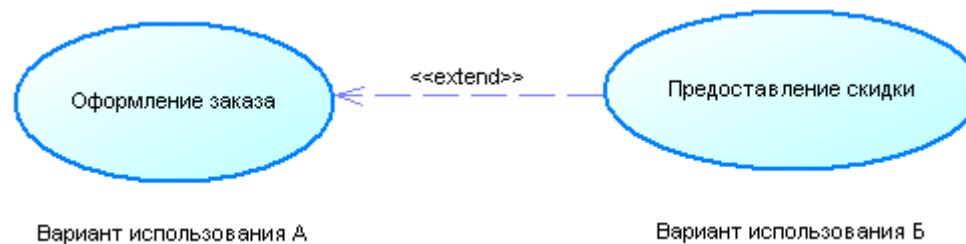
**Отношение зависимости (dependency)** определяется как форма взаимосвязи между двумя элементами модели, предназначенная для спецификации того обстоятельства, что изменение одного элемента модели приводит к изменению некоторого другого элемента. В общем случае зависимость является направленным бинарным отношением, которое связывает между собой два элемента модели: независимый и зависимый

**Отношение включения (include)** - частный случай общего отношения зависимости между двумя вариантами использования, при котором некоторый вариант использования содержит поведение, определенное в другом варианте использования. Графическое изображение - пунктирная стрелка с ключевым словом <<include>>



**Отношение расширение (extend)** – показывает, что вариант использования расширяет базовую последовательность действий и вставляет собственную последовательность. При этом в отличие от типа отношений “включение” расширенная последовательность может осуществляться в зависимости от определенных условий.

Графически изображение - пунктирная стрелка направленная от зависимого варианта (расширяющего) к независимому варианту (базовому) с ключевым словом `<<extend>>`.



Дополнительно: UML-Specification-1.4.2-(ISO-IEC-19501).pdf (249 - 253)