[Определение 2](#_Toc149054903)

[Классификация UML диаграмм 3](#_Toc149054904)

[Структурные диаграммы UML. 4](#_Toc149054905)

[Диаграмма класса 4](#_Toc149054906)

[Диаграмма объектов – 5](#_Toc149054907)

[Диаграмма развертывания – 6](#_Toc149054908)

[Диаграмма пакетов – 7](#_Toc149054909)

[Диаграмма кооперации – 8](#_Toc149054910)

[Диаграмма профилей – 10](#_Toc149054911)

[Диаграмма компонент – 11](#_Toc149054912)

[Диаграммы поведения UML. 12](#_Toc149054913)

[Диаграмма прецендентов – 12](#_Toc149054914)

[Диаграмма состояний – 12](#_Toc149054915)

[Диаграмма деятельности – 13](#_Toc149054916)

[Диаграмма взаимодействия – 13](#_Toc149054917)

[Диаграмма последовательности 13](#_Toc149054918)

[Диаграмма синхронизации 14](#_Toc149054919)

[Диаграмма коммуникации 14](#_Toc149054920)

[Диаграмма обзора взаимодействия 14](#_Toc149054921)

# Определение

UML (Unified Modeling Language, Унифицированный Язык Моделирования) — язык графического описания для объектного моделирования в области разработки программного обеспечения, который также можно применять для моделирования бизнес-процессов, системного проектирования и отображения организационных структур.

# Классификация UML диаграмм

Структурные диаграммы UML. Описывают систему статически.

Диаграмма класса –

**Определение**: "Класс описывает набор объектов, которые объединяют одинаковые описания свойств, ограничений и семантики."

**Что это означает**: Класс – это категория. Все объекты, которые принадлежат этой категории имеют одинаковые атрибуты и операции (но значения атрибутов может меняться от объекта к объекту). Это означает, значение атрибутов может меняться от объекта к объекту в рамках класса, например, цвет одной ручка может быть синим, а цвет другой зеленым.

Все объекты класса имеют одинаковые операции и методы. Другими

словами, все объекты класса могут делать одно и тоже, и одинаковым

способом.

Несмотря на название «Объектно Ориентированный Анализ», вы

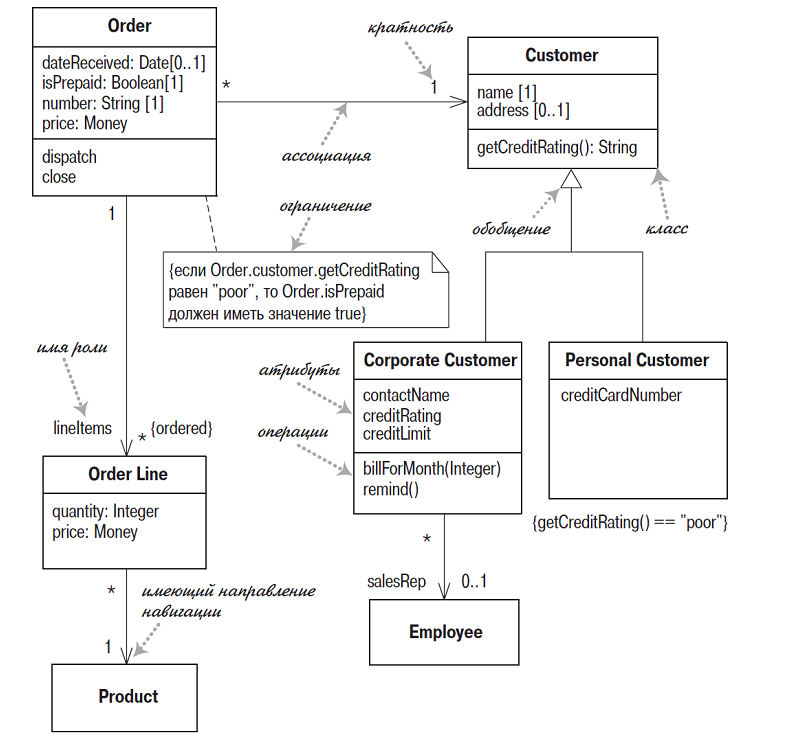
будете тратить большую часть своего времени на определение

классов (их называют классы- сущности (entity classes), а не объектов.

Например, в банковской системе, нужно

определить характеристики класса - сущности «Счет» и класса -

сущности «Клиент» организации.



## Диаграмма объектов –

Мы начинаем анализировать бизнес, прося опрашиваемых описать

бизнес-объекты, которые существуют в рамках данного бизнеса.

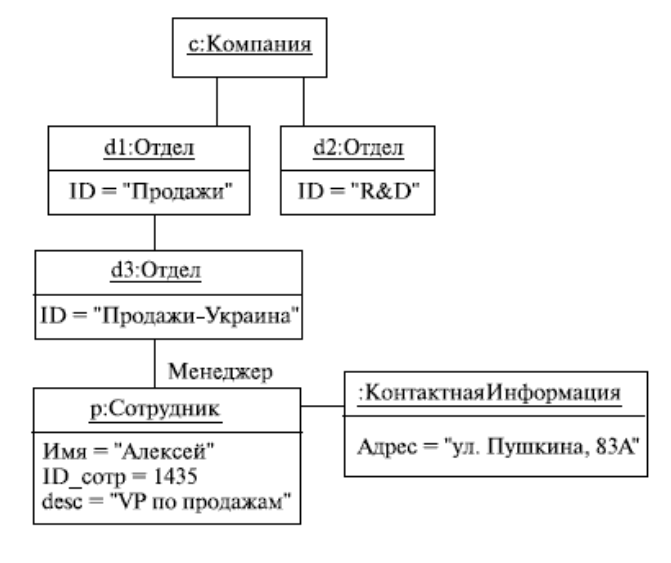
Бизнес-объекты — это сущности или понятия, которые бизнес (и IT

система должна автоматизировать это) должен контролировать или

которые участвуют в бизнес-процессах. Примерами таких объектов

могут включать счета, представителей службы поддержки клиентов,

или, собственно, вызов.



## Диаграмма развертывания –

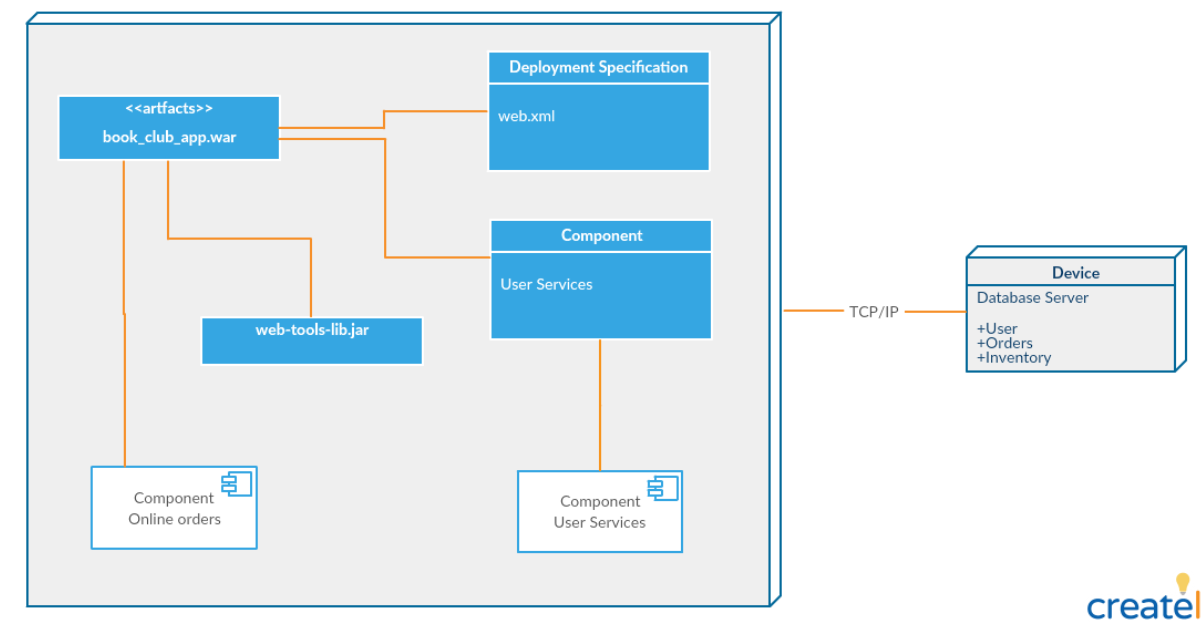
физическое развертывание информации, выработанной программой на аппаратных компонентах. В отношении информации, которую генерирует программа.

Диаграммы развертывания используются в разных целях. Например, с их помощью можно:

* наглядно показать, какие программные элементы развертываются на тех или иных аппаратных компонентах
* проиллюстрировать обработку процессов исполнения аппаратными компонентами
* провести обзор топологии аппаратного комплекса

Что бы создать диаграмму UML надо рассмотреть следующие вопросы:

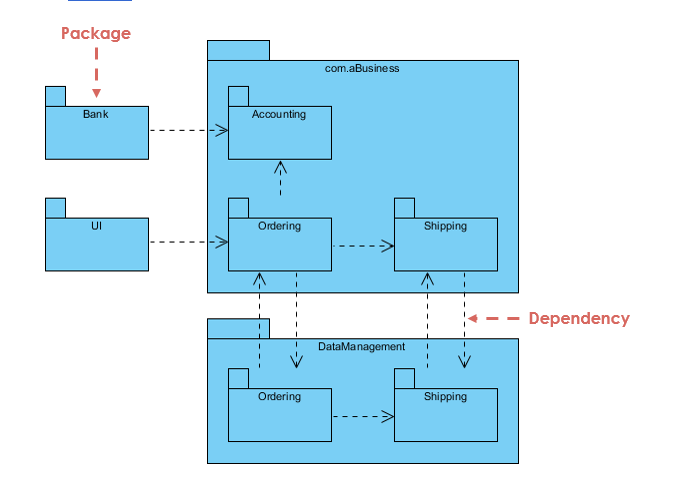
* Каковы масштабы системы? К примеру, важно заранее определиться, с каким видом развертывания мы имеем дело, — одного приложения или целой компьютерной сети.
* Каковы ограничения физического оборудования? С какими устаревшими системами вам придется столкнуться? Обязательно изучите операционное ПО и протоколы, с которыми вам предстоит работать, и подберите необходимые методы мониторинга.
* Какой архитектурой распределения вы пользуетесь? Важно знать, сколько уровней будет в вашем приложении и где будет осуществляться развертывание.
* Все ли необходимые узлы в наличии? Как они связаны друг с другом?
* Какие компоненты будут находиться на каждом узле?



## Диаграмма пакетов –

используется для упрощения сложных диаграмм классов, вы можете группировать классы в пакеты. Пакет — это набор логически связанных элементов UML.

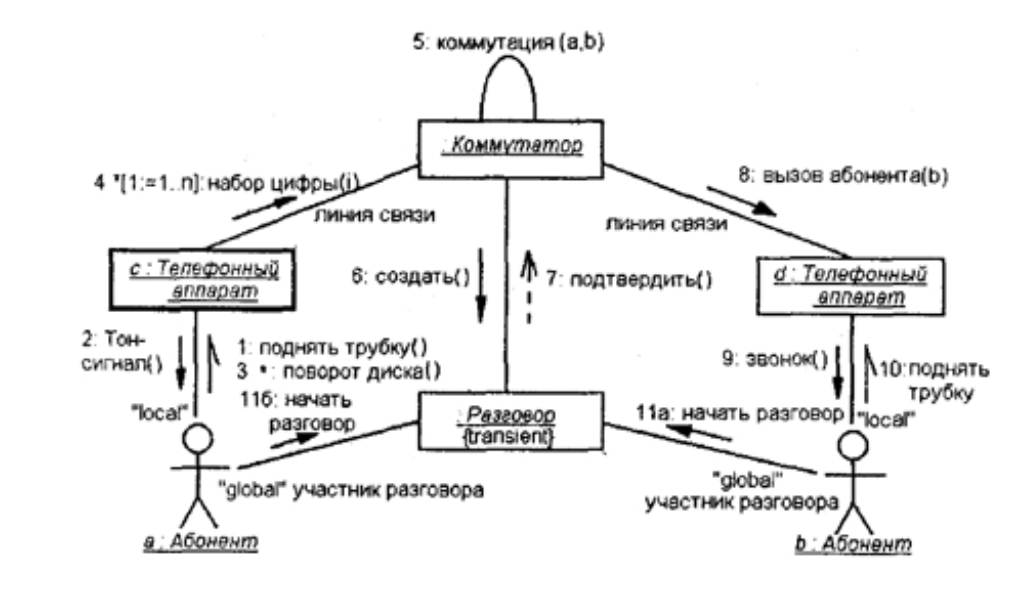
На диаграмме ниже представлена ​​бизнес-модель, в которой классы сгруппированы в пакеты:



* Пакеты отображаются в виде прямоугольников с небольшими вкладками вверху.
* Имя пакета находится на вкладке или внутри прямоугольника.
* Пунктирные стрелки — зависимости.
* Один пакет зависит от другого, если изменения в другом могут вызвать изменения в первом.

## Диаграмма кооперации –

предназначена для спецификации структурных аспектов взаимодействия. Главная особенность диаграммы кооперации заключается в возможности графически представить не только последовательность взаимодействия, но и все структурные отношения между объектами, участвующими в этом взаимодействии.



На диаграмме

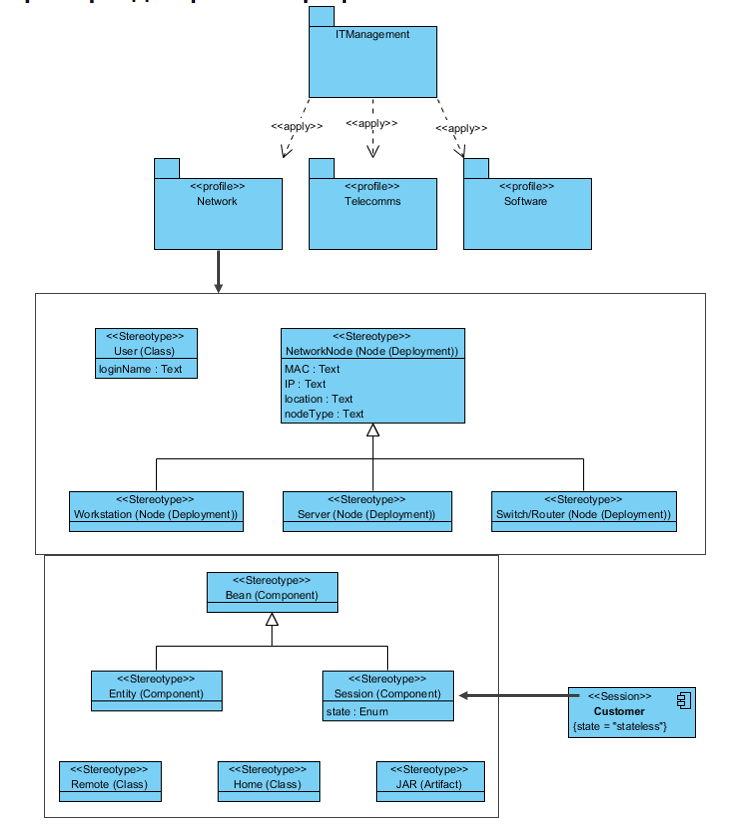
* в виде прямоугольников изображаются участвующие во взаимодействии объекты, содержащие имя объекта, его класс и, возможно, значения атрибутов
* указываются ассоциации (можно указать имена ассоциации и ролей) между объектами в виде различных соединительных линий.
* изображены динамические связи – потоки сообщений (они представляются также в виде соединительных линий между объектами, над которыми располагается стрелка с указанием направления, имени сообщения и порядкового номера в общей последовательности инициализации сообщений).

Отличие диаграммы последовательности и диаграммы кооперации

|  |  |
| --- | --- |
| Диаграмма последовательности | Диаграмма кооперации |
| изображаются все отношения между объектами | изображаются только отношения между объектами, играющими определенные роли во взаимодействии |
| Может явно специфицировать взаимосвязи между объектами в реальном времени | не указывается время в виде отдельного измерения, поэтому последовательность взаимодействий и параллельных потоков может быть определена с помощью порядковых номеров |

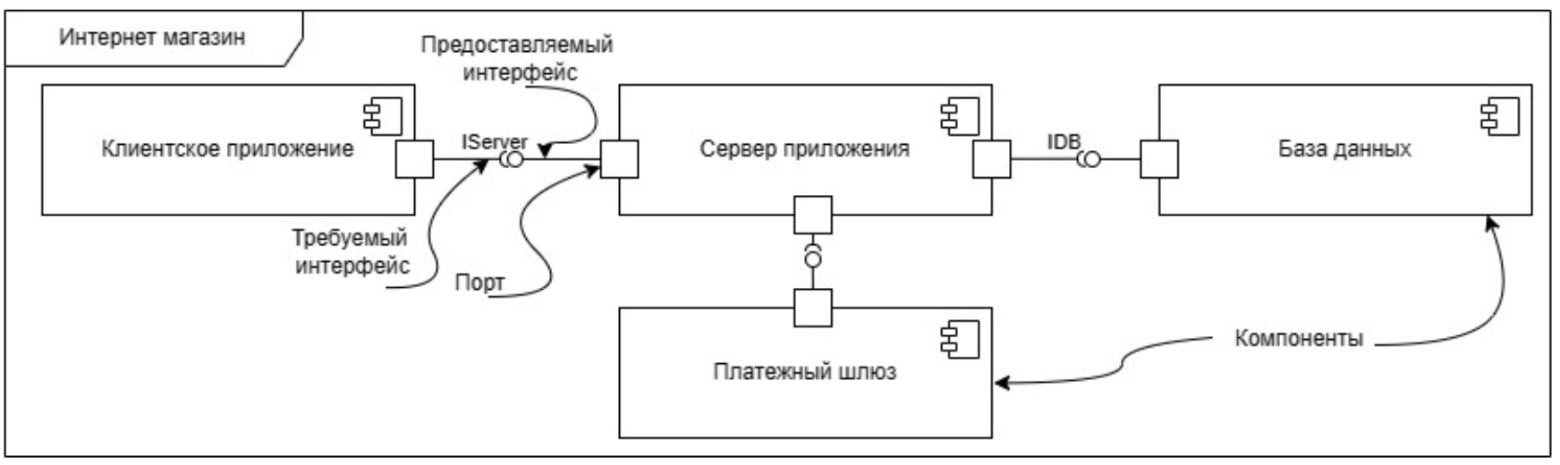
## Диаграмма профилей –

это структурная диаграмма UML, которая используется для расширения стандартной нотации UML, чтобы адаптировать ее под конкретные потребности и предметные области проекта.

Диаграмма профиля позволяет расширить стандартную нотацию UML путем создания новых элементов и отношений, которые отражают специфические требования и особенности предметной области проекта. Это особенно полезно, когда нотация UML не обладает достаточным набором символов, чтобы описать все аспекты системы.

Стереотипы — это особые метки, которые мы присваиваем элементам модели, чтобы указать их особую роль или значение в системе. Они позволяют расширить словарь UML. Стереотипы позволяют добавлять и создавать новые элементы модели, производные от существующих, но имеющие специфические свойства, которые соответствуют вашей предметной области.

## Диаграмма компонент –

это структурная диаграмма языка унифицированного моделирования, она описывает особенности физического представления системы. Диаграмма компонентов позволяет определить архитектуру разрабатываемой системы, установив зависимости между программными компонентами, а также является полезным инструментом для коммуникации и документирования архитектурных решений.

Диаграммы поведения UML. Описывают систему динамически.

## Диаграмма прецендентов (Use case) –

**Определение:** " Use case А — это спецификация набора действий, выполняемых

системой, дающих наблюдаемый результат, который, как правило,

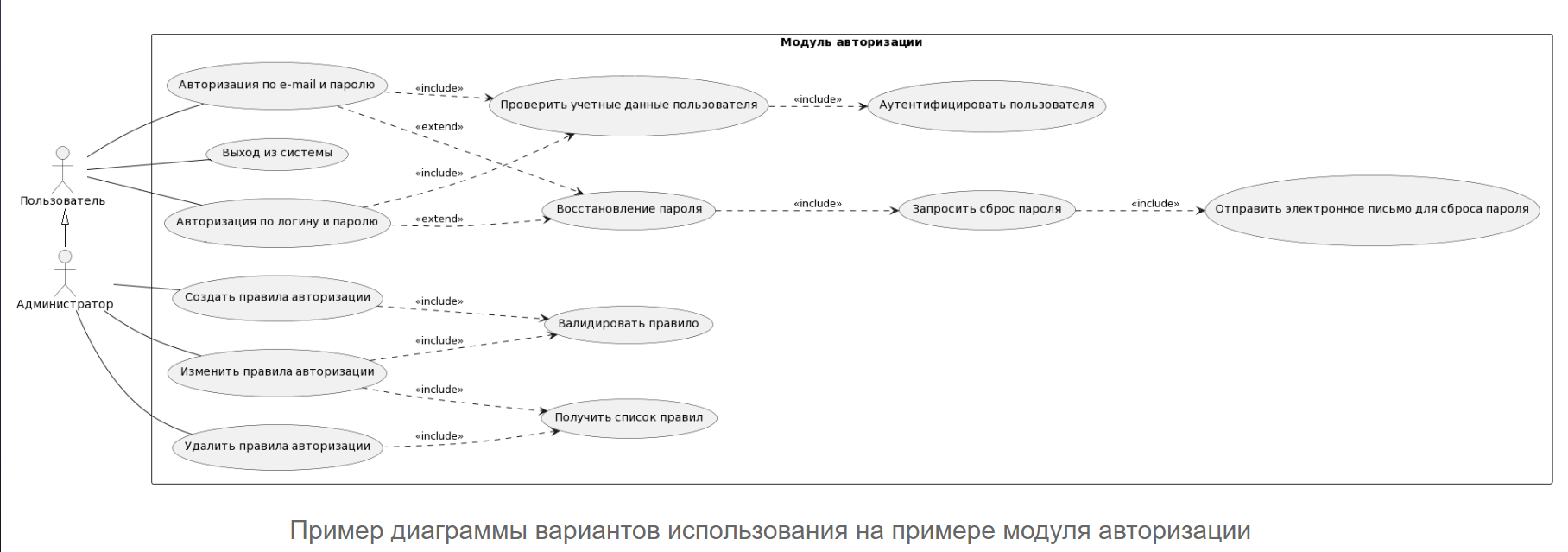
важны для одного или нескольких Actors или других заинтересованных

лиц системы. "

**Что это означает: Use** case – это использование системы, которое обеспечиваетвидимый и (обычно) значимый результат. Документация по Use case(схемы и / или текст) должен описывать последовательность шагов,которые выполняются во время взаимодействия с системой и включаютв себя различные способы (пути) этого взаимодействия.

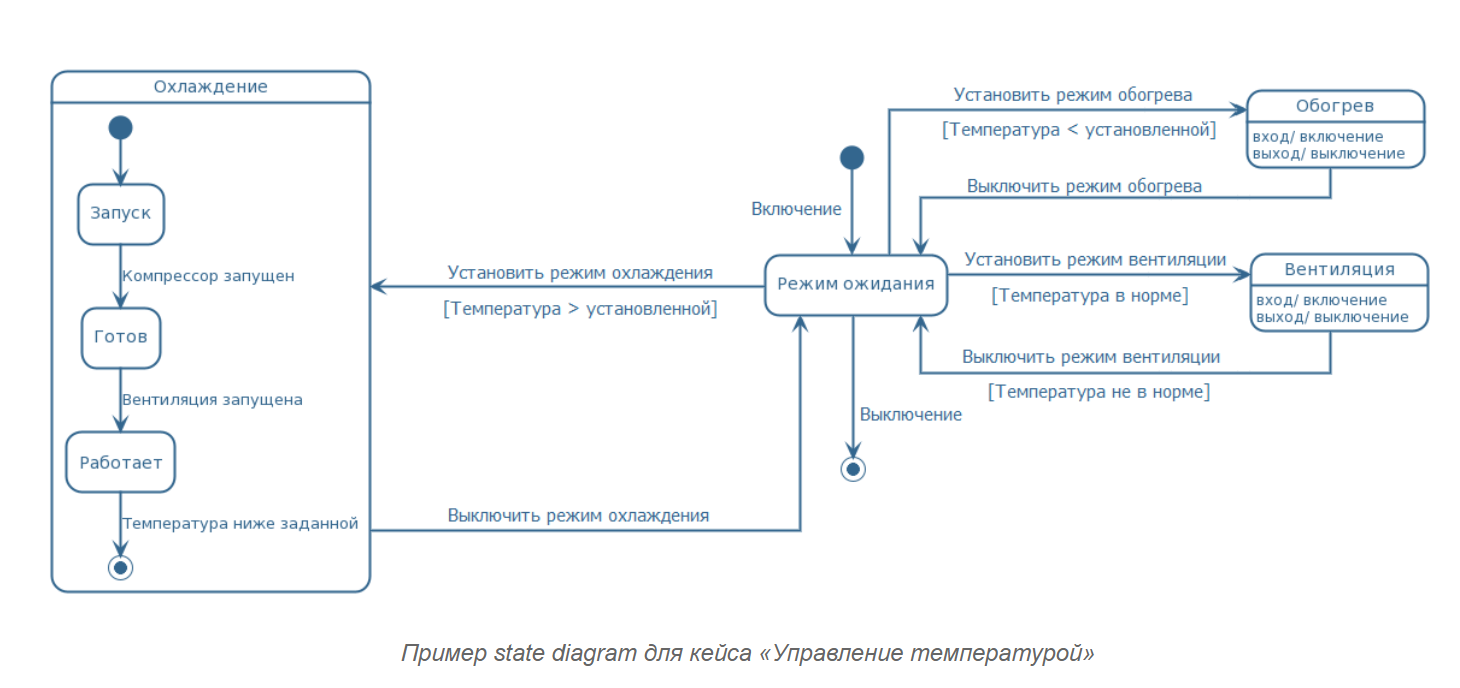
* Use case (общий) описывает взаимодействие с некоторым типом системы. Вопрос

в том, какой тип системы имеется в виду?

* Бизнес Use case описывает взаимодействие с бизнес-системой. Например, процесс выставления претензии является бизнес Use case, описывающим взаимодействие со страховой компанией.
* Системный Use case описывает взаимодействие с ИТ системой. Например, системные Use cases, которые описывают реализацию вышеупомянутого бизнес Use case — это «Зарегистрировать претензию», «Проверить случай», «Назначить ответственного» и так далее. Каждый из них описывает взаимодействие пользователя и компьютерной системы. Системные Use case обычно включает в себя одного активного (primary) пользователя (actor) и, как правило, выполняется в течение одной сессии. В конце исполнения системного Use case, пользователь должен чувствовать, что он или она достигли какой-то значимой, полезной с точки зрения бизнеса, цели.

## Диаграмма состояний –

это один из видов диаграмм UML, используемых в разработке программного обеспечения, чтобы визуализировать и моделировать поведение объекта или системы в различных состояниях. Она позволяет описать все возможные состояния объекта, а также переходы между ними в ответ на определенные события.

Диаграмма состояний описывает все возможные состояния, и при каких условиях происходит переход между состояниями. События являются внешними сигналами или действиями, которые вызывают переходы между состояниями.

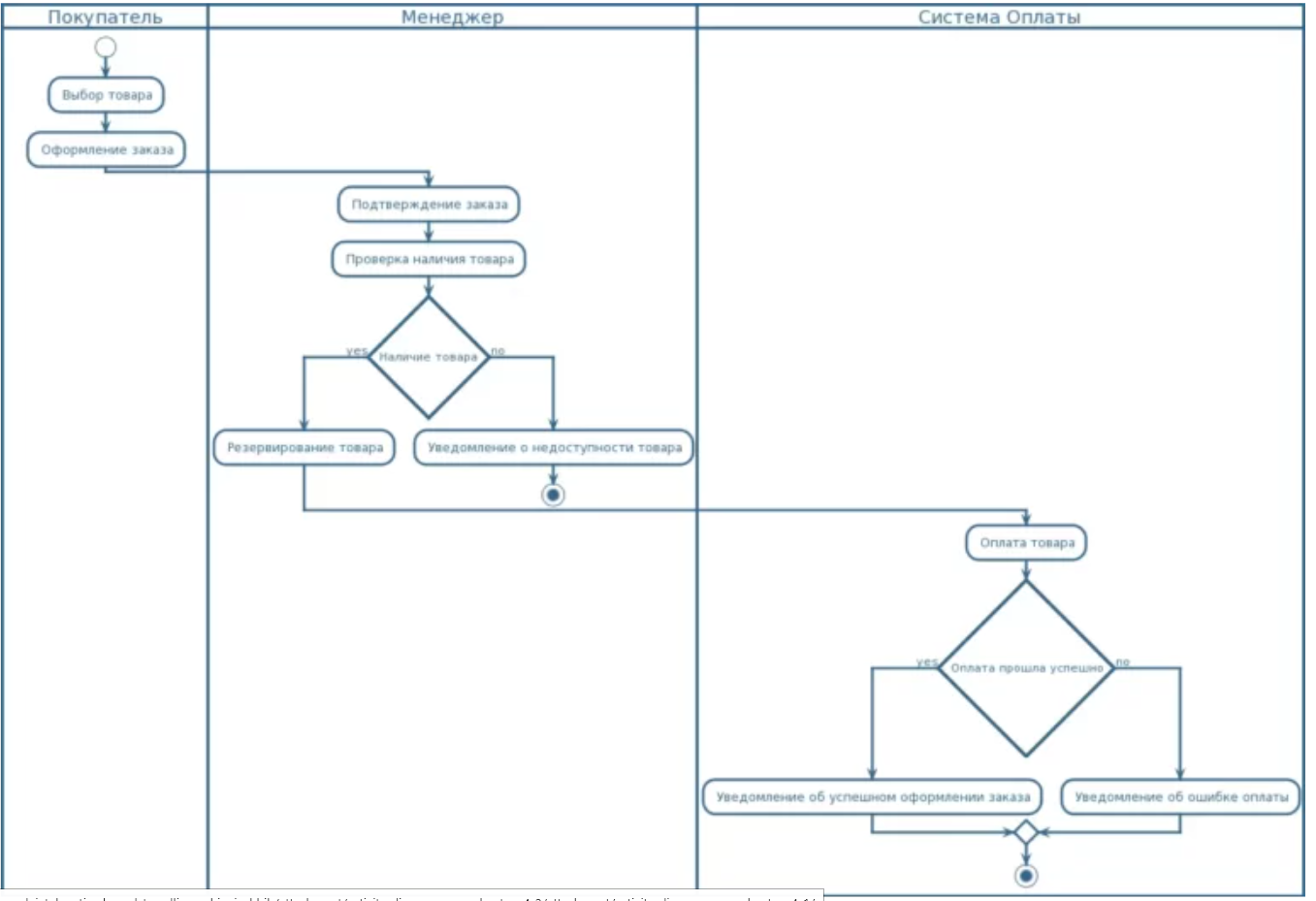
состояния обычно представляются в виде прямоугольников с названиями

переходы — стрелками

аннотации могут использоваться для указания условий переходов или действий, выполняемых в определенных состояниях.

## Диаграмма деятельности –

это графическое представление процессов и задач, выполняемых в рамках определенного проекта или деятельности. Эта диаграмма используется для анализа и оптимизации бизнес-процессов, планирования работы, принятия решений и управления проектами. Диаграмма деятельности состоит из набора символов и линий, каждый из которых имеет свое значение и назначение.



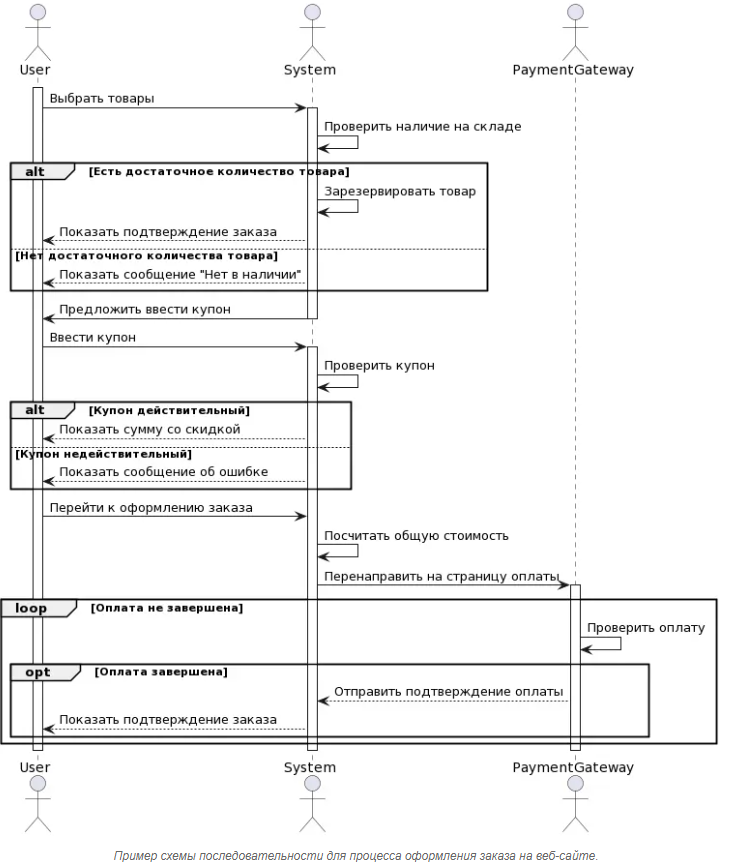
Суть диаграммы деятельности заключается в графическом отображении последовательности действий. Для каждой задачи на диаграмме деятельности указывается,

* кто выполняет задачу
* какие ресурсы необходимы для ее выполнения
* какие результаты ожидаются на каждом этапе процесса
* какая последовательность задач и операций необходима для успешного завершения проекта.

## Диаграмма взаимодействия –

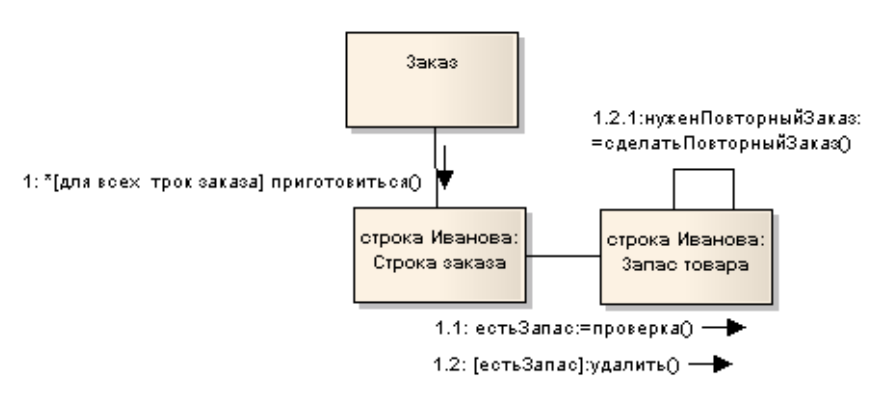
### Диаграмма последовательности

это подвид диаграмм взаимодействия, который позволяет описать взаимодействие между объектами в системе в виде последовательности сообщений, действий и операций, отображая порядок выполнения действий и обмена информацией между объектами во времени.



### Диаграмма синхронизации

### Диаграмма коммуникации

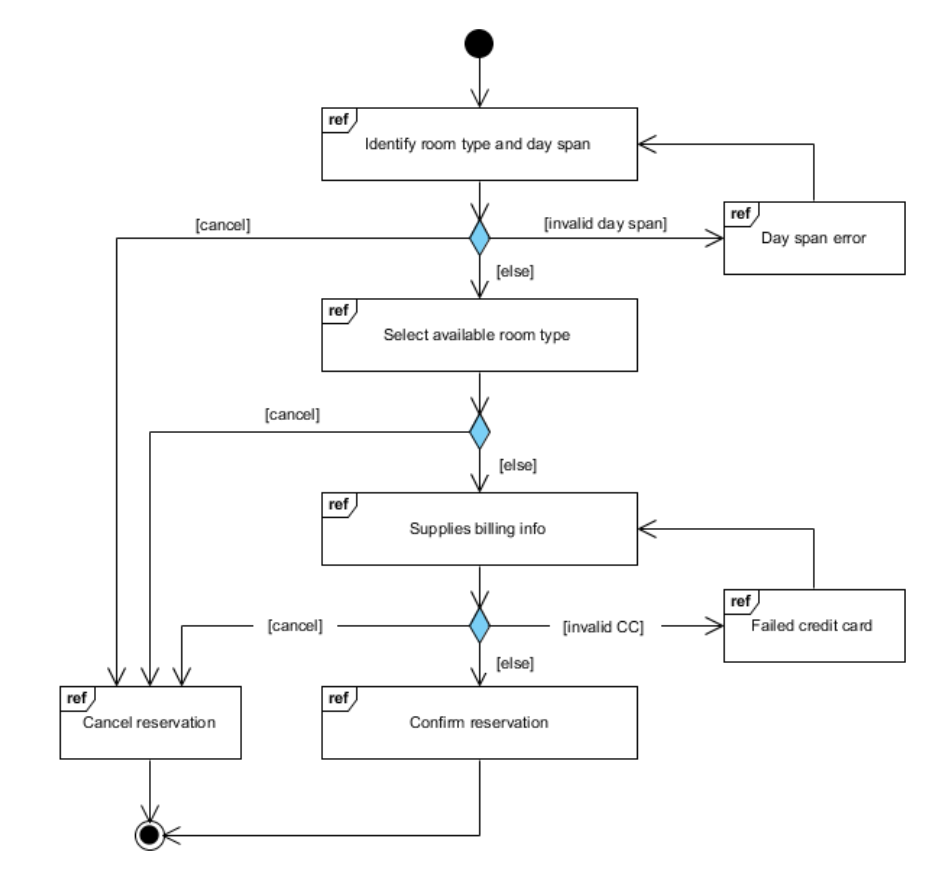
Диаграмма коммуникации (communication diagram) отображает ту же информацию, что и диаграмма последовательностей, но на диаграмме коммуникации зависимость от времени указывается посредством нумерации сообщений. На диаграмме коммуникации отражается распределение процессов между объектами и их зависимости друг от друга, что очень полезно при разработке различных проектов. Основной целью построения данной диаграммы является понимание структурной организации занятых в системе объектов, принимающих и передающих сообщения.

### Диаграмма обзора взаимодействия

Что такое обзорная диаграмма взаимодействия?

Обзорная диаграмма взаимодействия является одним из типов диаграмм взаимодействия UML и представляет собой графическое представление взаимодействия между различными элементами системы. Эта диаграмма позволяет визуализировать и описать поток выполнения событий в системе, объединяя другие диаграммы взаимодействия, такие как диаграммы последовательности и диаграммы коммуникации.

Обзорная диаграмма взаимодействия используется для представления сложных взаимодействий между различными объектами или компонентами системы. Она позволяет объединить несколько диаграмм взаимодействия в одну общую структуру, что помогает упростить и улучшить понимание системы и ее поведения. Обзорные диаграммы взаимодействия фокусируются на общем представлении потока управления, где узлами являются взаимодействия (sd) или ссылки на взаимодействия (ref). Взаимодействия — это взаимодействия между объектами, которые могут включать передачу сообщений или выполнение определенных действий. Ссылки на взаимодействия — это ссылки на диаграммы взаимодействия, которые используются повторно для упрощения моделирования.



* Элементы системы представлены в виде прямоугольников, называемых фреймами. Фреймы содержат другие диаграммы взаимодействия, которые отображают поток выполнения событий в системе.
* Поток выполнения определяется взаимодействием между различными объектами или компонентами системы и может отображаться в виде последовательности или параллельных ветвей.