Федеральное государственное бюджетное образовательное учреждение высшего

образования

«Чувашский государственный университет им. И.Н. Ульянова»

Кафедра вычислительной техники

Лабораторная работа № 6

**Модель анализа**

Вариант 9

Выполнил: Иванов В.С.

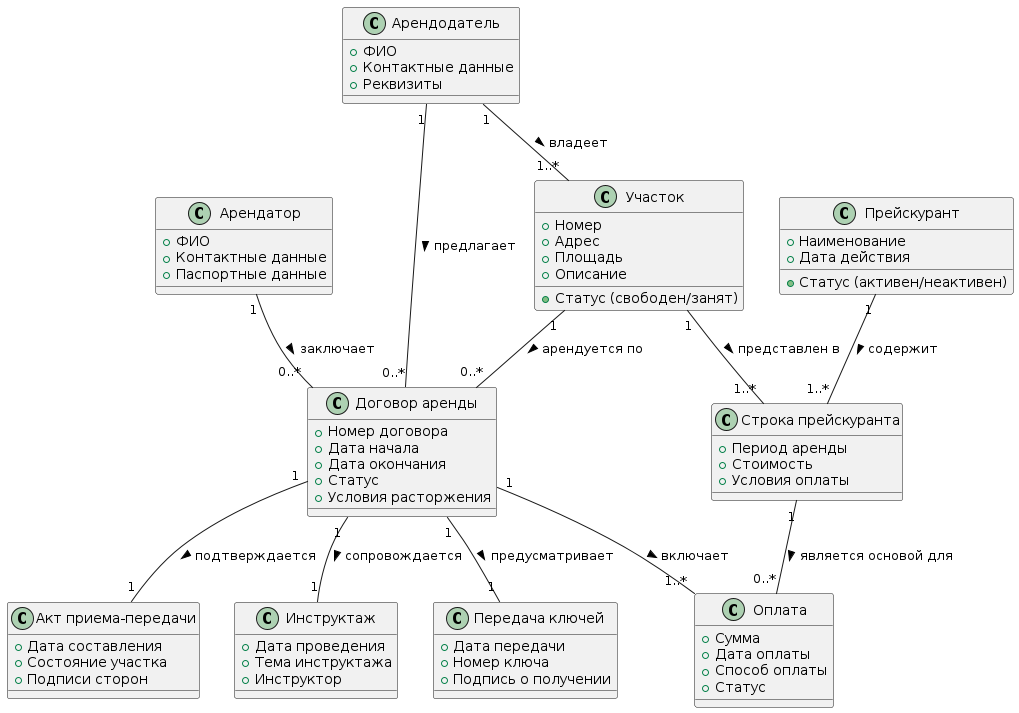
студент группы ИВТ-41-22

Проверил: кандидат технических наук

Ржавин Вячеслав Валентинович

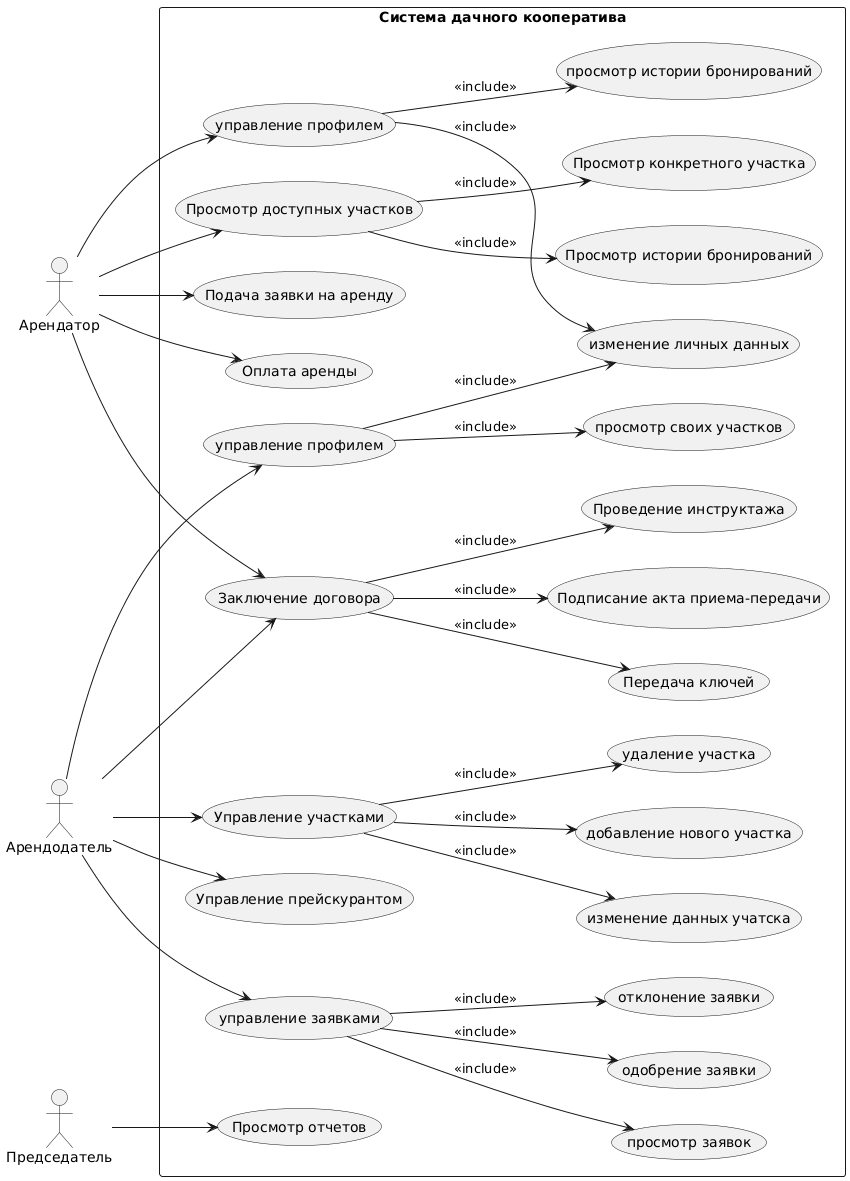
Чебоксары, 2025

Модель предметной области

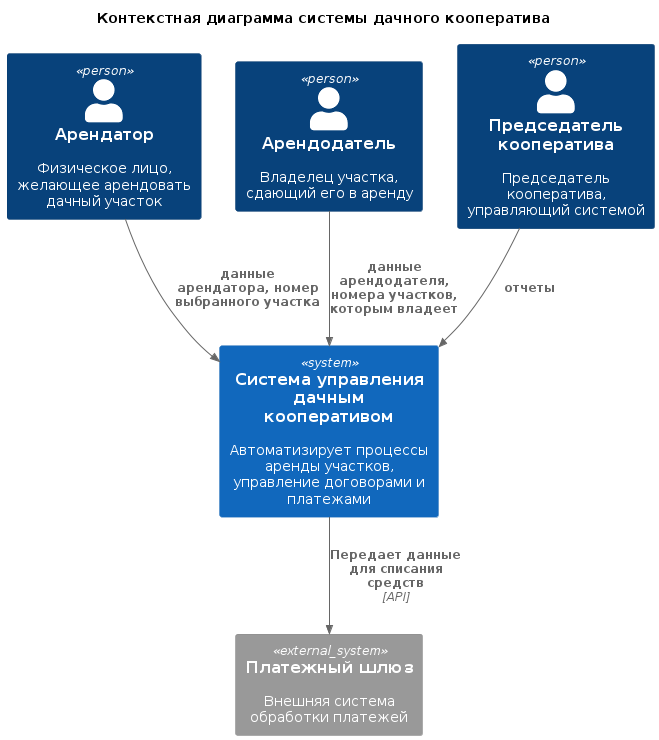


Строка прейс -> композиция

Диаграмма вариантов использования



Контекстная диаграмма (C4)



Сценарии вариантов использования

Основной поток

|  |
| --- |
| Вариант использования : заключение договора аренды |
| ID: UC-014 |
| Краткое описание: процесс оформления договора аренды участка между арендатором и арендодателем |
| Главное действующее лицо: арендатор |
| Второстепенные действующие лица: нет |
| Предусловия:   1. Арендатор авторизован в системе 2. Участок доступен для аренды на выбранные даты 3. Заявка на аренду подтверждена арендодателем 4. Прейскурант актуален и содержит стоимость аренды |
| Основной поток:   1. Арендодатель просматривает заявку 2. Арендодатель принимает заявку на аренду, генерируется шаблон договора 3. Арендатор смотрит договор и подписывает его 4. Система регистрирует договор, присваивает уникальный номер 5. Система сохраняет подписанный договор в базе данных 6. Система отправляет копии договора обеим сторонам 7. Статус договора изменяется на "Активен" |
| Постусловия:   1. Договор аренды зарегистрирован в системе 2. Участок помечен как "Занят" на период аренды 3. Обе стороны имеют доступ к электронной копии договора |
| Альтернативные потоки: нет |

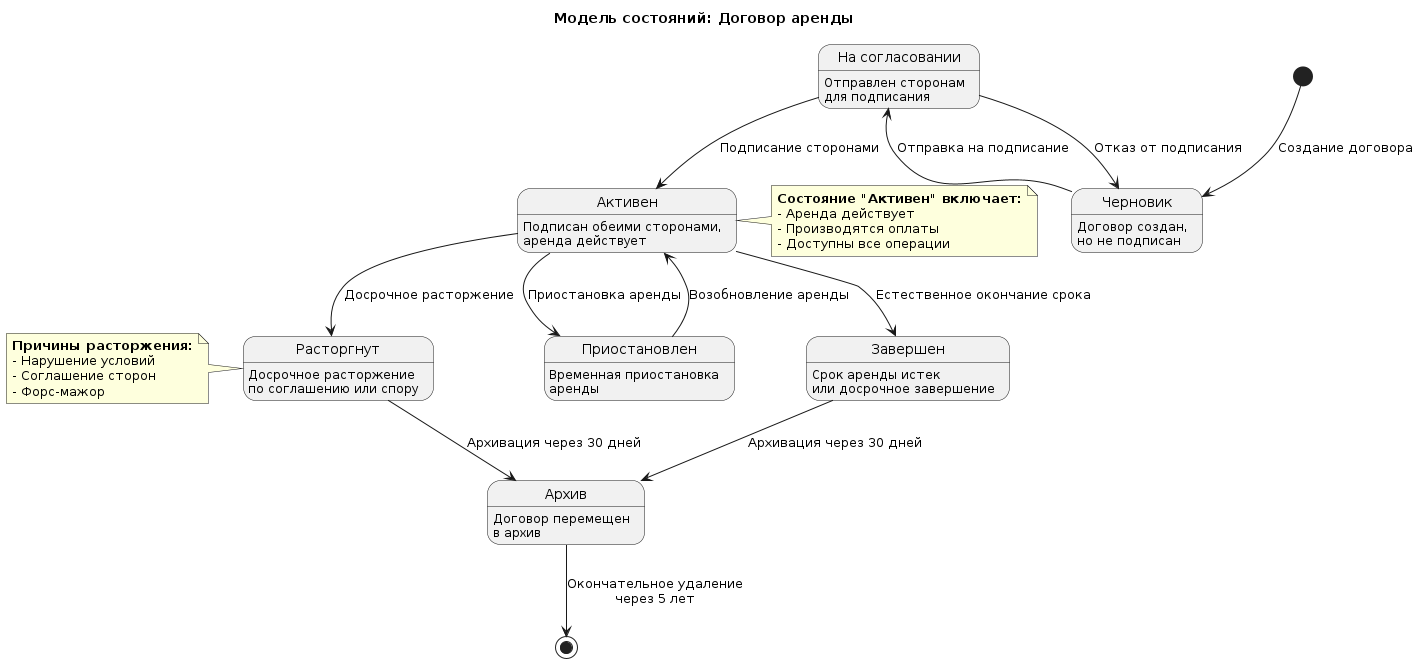
Альтернативные потоки

|  |
| --- |
| Вариант использования : заключение договора аренды |
| ID: UC-014 |
| Краткое описание: процесс оформления договора аренды участка между арендатором и арендодателем |
| Главное действующее лицо: арендатор |
| Второстепенные действующие лица: нет |
| Предусловия:   1. Арендатор авторизован в системе 2. Участок доступен для аренды на выбранные даты 3. Заявка на аренду подтверждена арендодателем 4. Прейскурант актуален и содержит стоимость аренды |
| Альтернативные потоки:  А1: отказ от арендодателя  Точка входа 1 шаг основного потока   1. Арендодатель проверяет условия договора 2. Арендодатель обнаруживает конфликт расписания или иные причины для отклонения 3. Арендодатель отклоняет заявку через интерфейс системы 4. Система фиксирует причину отклонения и время операции 5. Система автоматически освобождает забронированные даты участка 6. Система изменяет статус участка на "Доступен" 7. Система отправляет арендатору уведомление об отклонении заявки   А2: ошибка генерации договора  Точка входа 2 шаг основного потока   1. Система пытается сгенерировать договор аренды 2. Возникает ошибка генерации (отсутствует шаблон, некорректные данные) 3. Система сохраняет лог ошибки для технической поддержки 4. Система отображает арендатору сообщение о временной технической проблеме 5. Система предлагает повторить попытку через 15 минут   А3: отказ арендатора от аренды  Точка входа 3 шаг основного потока   1. Арендатор просматривает подготовленный договор 2. Арендатор обнаруживает несоответствие условий или принимает решение об отказе 3. Арендатор отказывается от подписания договора 4. Система запрашивает подтверждение отказа 5. Система уведомляет арендодателя об отказе арендатора 6. Система освобождает даты аренды |
| Постусловия: нет |

Поток с включением

|  |
| --- |
| Вариант использования : проведение инструктажа |
| ID: UC6 |
| Краткое описание:ознакомление арендатора с правилами и особенностями участка |
| Главное действующее лицо: арендодатель |
| Второстепенные действующие лица: арендатор |
| Предусловия:   1. Акт приема передачи подписан |
| Основной поток:   1. Арендодатель проводит инструктаж по правилам пользования участком 2. Арендодатель объясняет правила пользования общими зонами кооператива 3. Арендодатель знакомит с правилами противопожарной безопасности 4. Арендатор задает вопросы при необходимости 5. Арендатор подтверждает ознакомление с правилами в системе 6. Система фиксирует дату и время проведения инструктажа 7. Система сохраняет запись о проведении инструктажа |
| Постусловия:   1. Инструктаж проведен и зафиксирован в системе 2. Арендатор ознакомлен с правилами пользования участком |
| Альтернативные потоки:  А1: арендатор не понял инструктаж   1. Повторное проведение инструктажа |

Модель состояний предметной области



**Ответы на вопросы по UML**

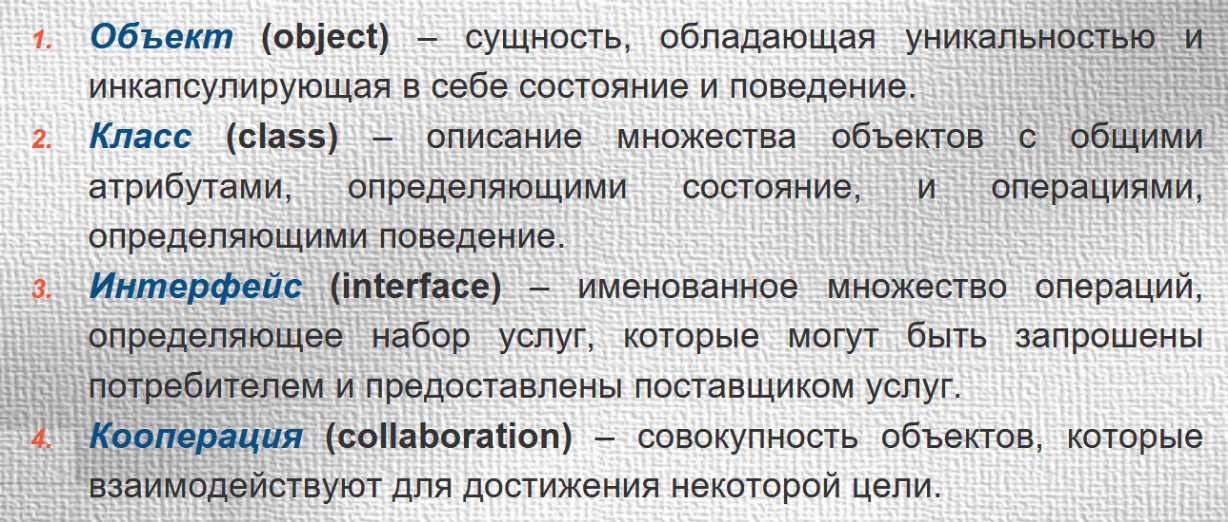
**1. Назовите типы сущностей в UML.**

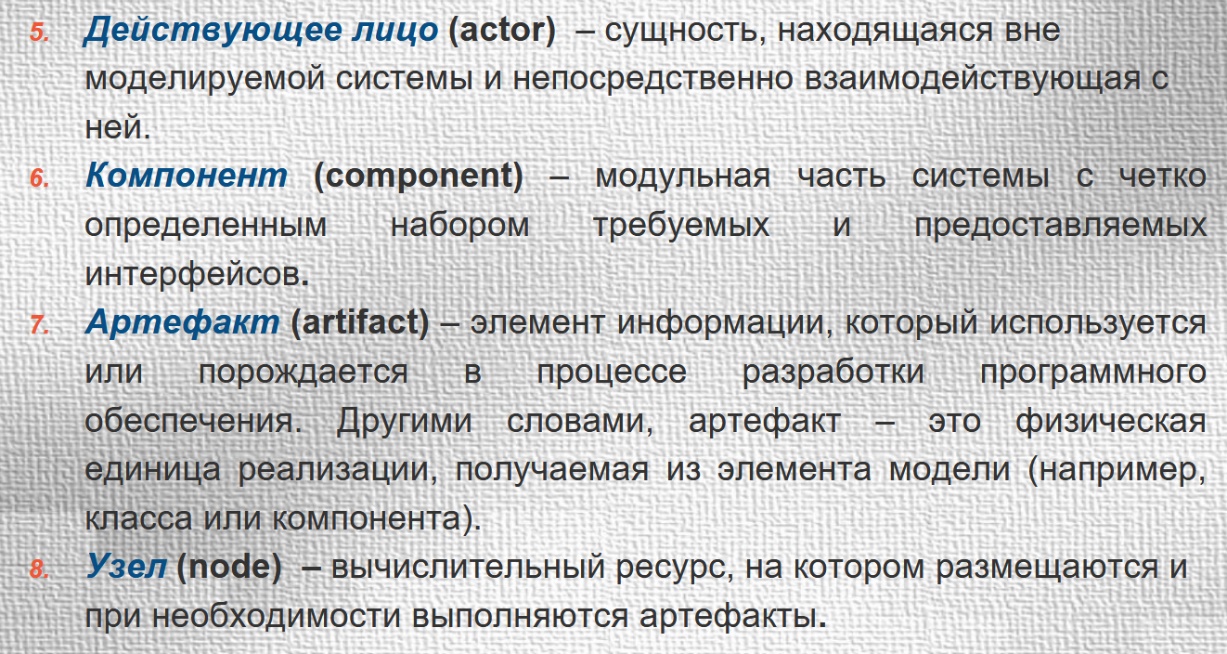
В UML есть четыре основных типа сущностей:

* Структурные — статические части модели, соответствующие концептуальным или физическим элементам системы (объекты, компоненты, интерфейсы).
* Поведенческие — динамические части модели, описывающие поведение системы во времени (взаимодействия, состояния, деятельность).
* Группирующие — организационные механизмы для объединения элементов модели (пакеты).
* Аннотационные — пояснительные комментарии к другим элементам модели для улучшения понимания (примечания).

**2. Что относится к структурным сущностям?**

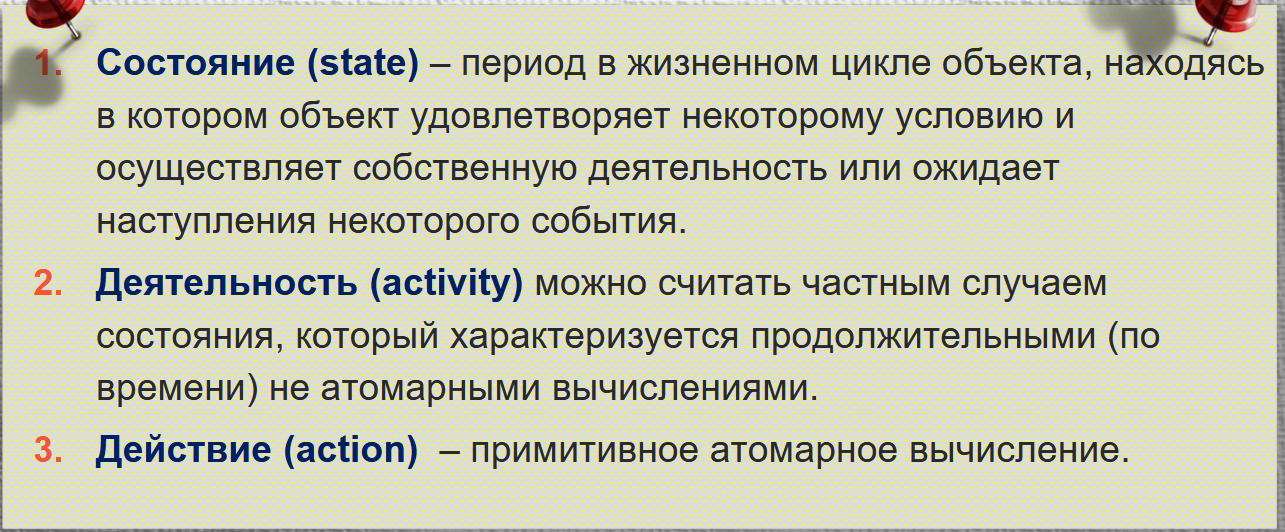
Это статические части модели, соответствующие концептуальным или физическим элементам системы.



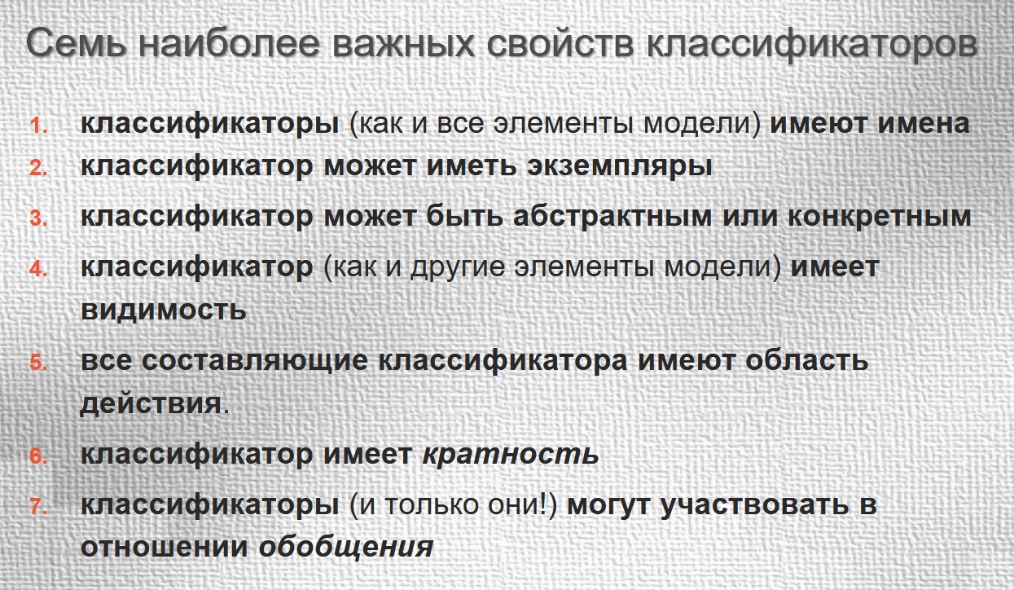


**3. Что относится к поведенческим сущностям?**

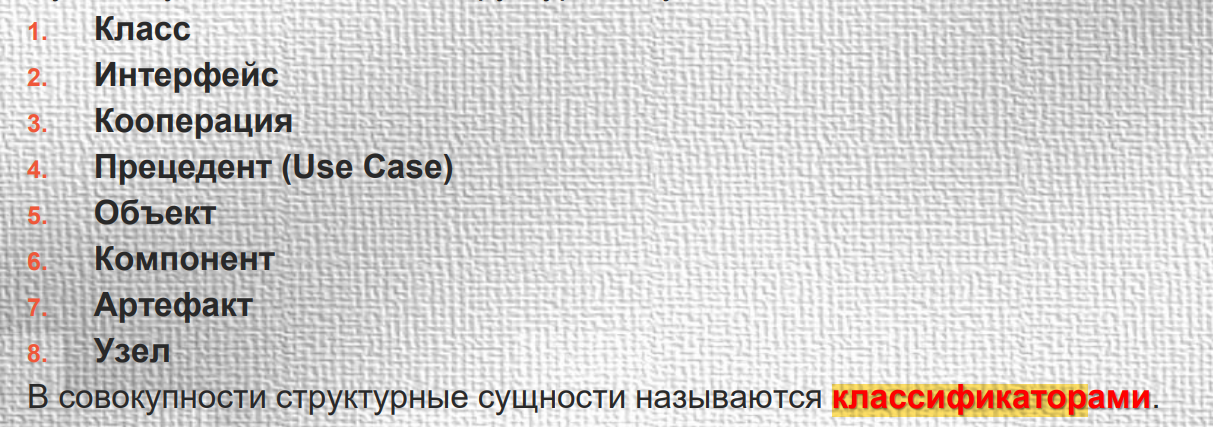
Это динамические составляющие модели UML, описывающими поведение модели во времени и пространстве.



**4. Назовите семь наиболее важных свойств классификаторов.**



**5. Что в UML относится к классификаторам?**



**6. Что такое абстрактный классификатор?**

* Абстрактный (abstract) классификатор не может иметь прямых экземпляров и в этом случае его имя выделяется курсивом.
* Конкретный (concrete) классификатор может иметь прямые экземпляры и в этом случае его имя записывается прямым шрифтом.

**7. Что такое классификатор?**

Классификатором называется механизм описания структурных и поведенческих свойств элемента системы.

**8. Что такое дескриптор?**

Дескриптор (descriptor) ‒ это описание общих свойств множества объектов, включая их структуру, отношения, поведение, ограничения, назначение и т. д.

Дескриптор имеет две стороны:

* это само описание множества (intent)
* и множество значений, описываемых дескриптором (extent)

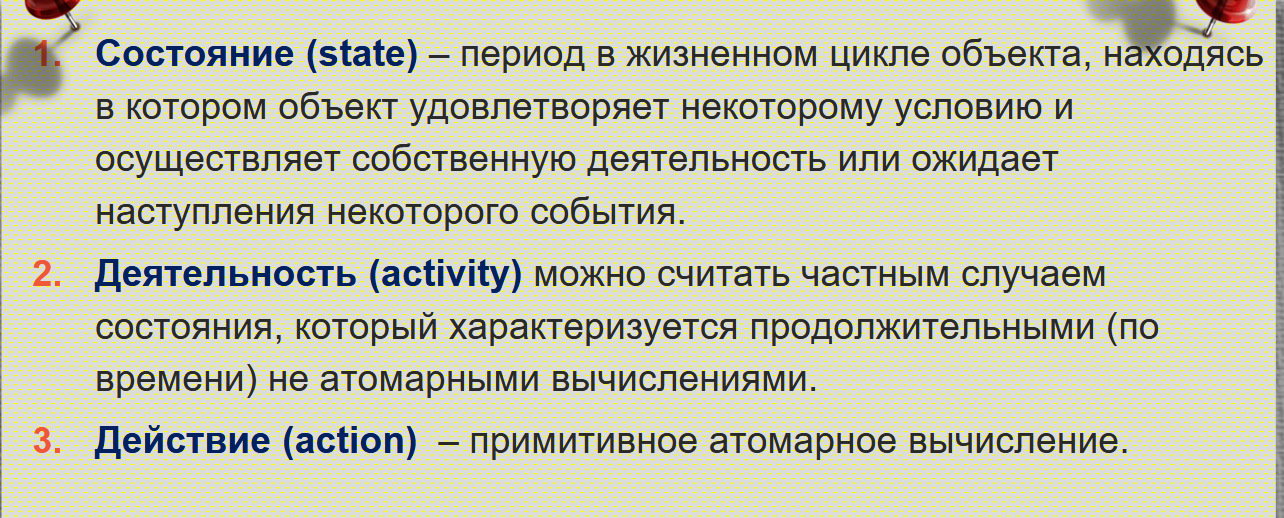
**9. Что такое литерал? Чем отличается от дескриптора?**

Литерал — это конкретный объект, созданный по чертежу.

Пример:

* Дескриптор (чертеж): Класс Student { String name; }.
* Литерал (объект): student1 = new Student(); student1.name = "Мария";.

**10. Дайте понятия состояния, деятельности, действия.**



**11. Является ли сущность Use Case классификатором? Обоснуйте ответ.**

Вариант использования (use case) представляет собой последовательность действий (транзакций), выполняемых системой в ответ на событие, инициируемое некоторым внешним объектом (действующим лицом).

Да, является. Потому что у Use Case есть экземпляры. Например – каждый раз, когда пользователь запускает сценарий "Купить билет" — это создаёт конкретный экземпляр этого Use Case.

**12. Что относится к группирующим сущностям?**

Группирующие сущности являются организующими частями модели UML. Это блоки, на которые можно разложить модель. Есть только одна первичная группирующая сущность, а именно пакет.

Пакеты (Packages) представляют собой универсальный механизм организации элементов в группы. В пакет можно поместить структурные, поведенческие и даже другие группирующие сущности. В отличие от компонентов, существующих во время работы программы, пакеты носят чисто концептуальный характер, то есть существуют только во время разработки.

**13. Что такое программный интерфейс? Что такое сигнатура?**

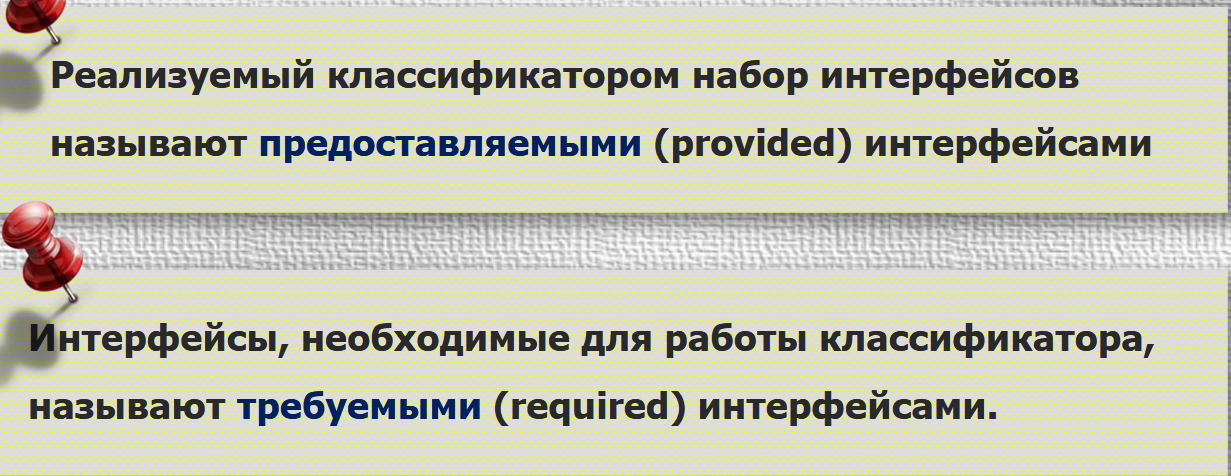
Интерфейс ‒ это именованный набор составляющих, описывающий контракт между поставщиками и потребителями услуг.

Имя операции вместе со списком типов формальных аргументов называется сигнатурой (signature) операции. Сигнатура в пределах класса должна быть уникальной. Это

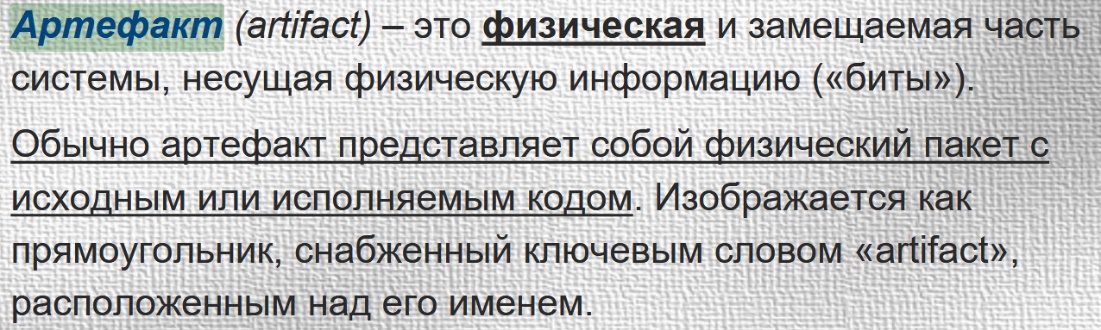
значит, что класс может обладать множеством операций с одним и тем же именем, при условии, что списки типов параметров этих операций отличаются.

**14. Дайте определение компонента UML.**

Компонент (component) ‒ это модульный фрагмент логического представления системы, взаимодействие с которым описывается набором обеспеченных и требуемых интерфейсов.



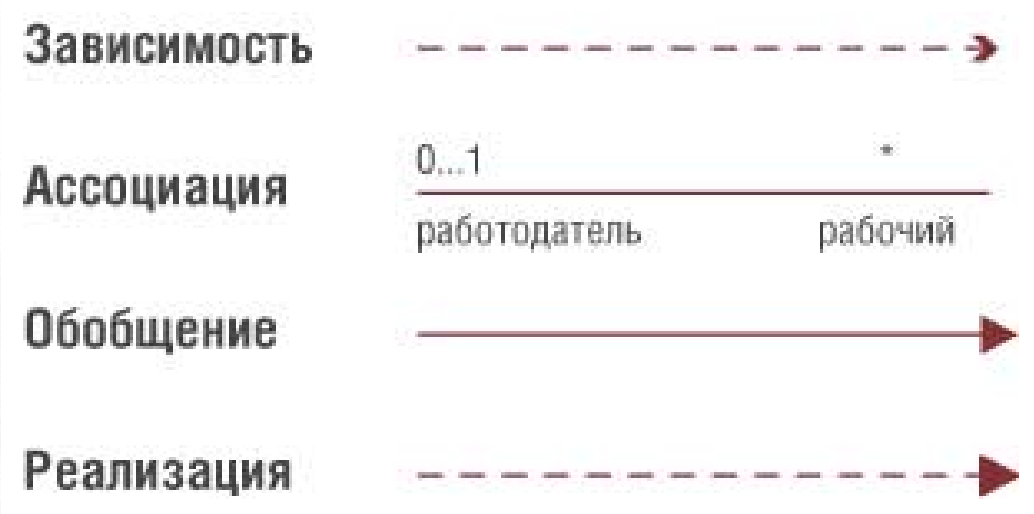
**15. Что такое артефакт? Приведите примеры.**



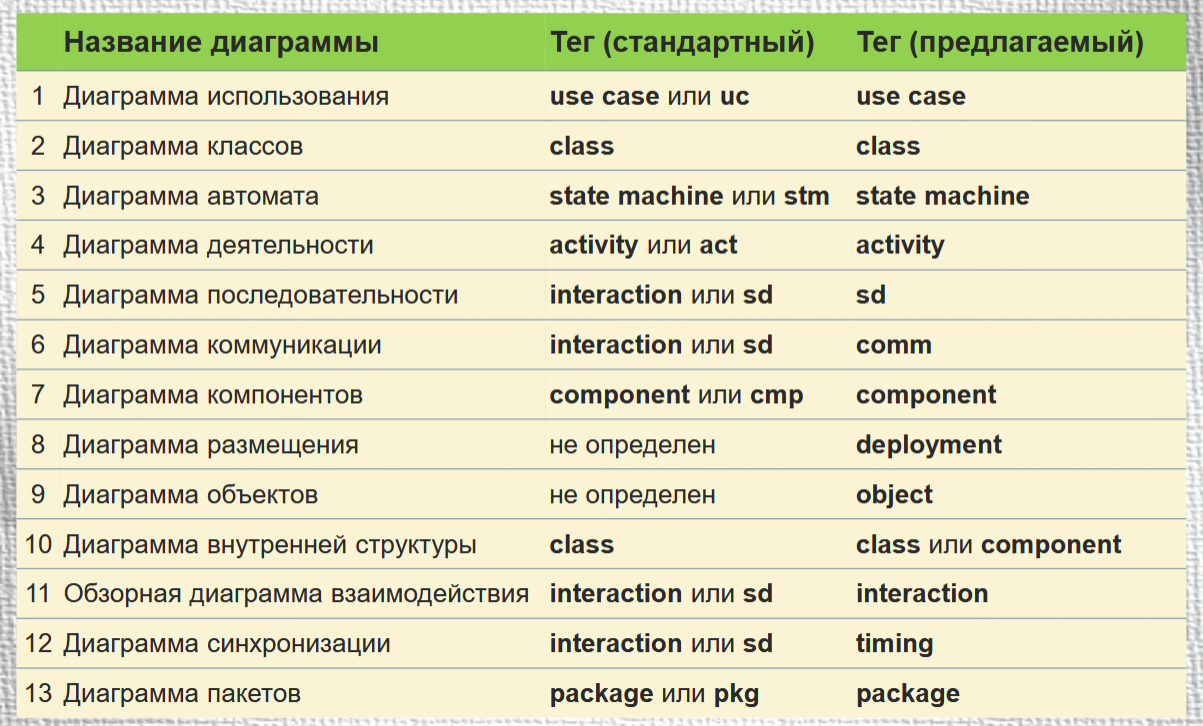
**16-17. Какие типы отношений определены в UML?**

Основные типы связей:

1. Ассоциация: просто факт связи. «Преподаватель — \*ведет\* — Курс».
   1. Агрегация: связь «часть-целое», где часть может жить без целого. «Компьютер — \*содержит\* — Монитор» (монитор можно отключить и подключить к другому компьютеру).
   2. Композиция: более жесткая связь. Часть не может жить без целого. «Дом — \*имеет\* — Комнату» ( если разрушить дом, комнаты не станет).
2. Зависимость: один класс \*использует\* другой временно. «Класс Заказ использует класс Logger для записи в журнал».
3. Обобщение: это наследование. «Класс Студент наследует классу Человек».
4. Реализация: класс реализует интерфейс. «Класс MySQLDatabase реализует интерфейс Database».



**18. Назовите основные типы диаграмм.**



**19. Какие представление включает в себя модель «4+1»?**



Логическое представление описывает статическую структуру системы (классы, объекты, связи) и динамические взаимодействия между ними. Отвечает на вопрос «Что система делает?»

Представление реализации описывает организацию программных модулей и компонентов в среде разработки (пакеты, библиотеки, зависимости). Отвечает на вопрос «Как это собрать?»

Процессное представление описывает динамические аспекты системы во время выполнения: потоки (нити), процессы, их взаимодействие и синхронизацию. Отвечает на вопрос «Как это работает в runtime?»

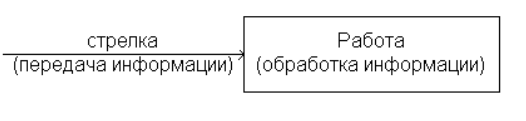
Физическое представление описывает размещение компонентов системы на физических аппаратных узлах (серверы, компьютеры, сети). Отвечает на вопрос «На чем это работает?»

+1. Представление сценариев описывает ключевые сценарии использования (Use Cases) системы, которые связывают все остальные представления воедино и проверяют их полноту. Отвечает на вопрос «Как система удовлетворяет потребности пользователей?»

**Контрольные вопросы**

1. Что такое модель домена и с какой целью она создаётся?

Часто используемые комбинации свойств можно именовать (дать имя). Такая комбинация свойств называется доменом, может наследоваться. Например, можно определить домен DATA для отображения всех колонок с датами для отображения в одном стиле. Домен DATA родителя ребенка наследует все атрибуты домена и может вносить свои. Связывание данных - входные/выходные данные - обработка данных



В домене показываем, какие сущности используем, грубо говоря это модель предметной области.

1. Поясните сценарий использования. Где и кем используется?

Сценарий представляет собой последовательность шагов, описывающих взаимодействие между пользователем и системой.

ВИ представляет собой множество сценариев, объединенных вместе некоторой общей целью пользователя.

Цель варианта использования заключается в том, чтобы определить законченный аспект или фрагмент поведения некоторой сущности без раскрытия внутренней структуры этой сущности.

1. Какие виды отношений используются в Use Case?

|  |  |
| --- | --- |
| Ассоциация | Указывает, что субъект принимает участие в варианте использования. |
| Включение | Включающий вариант использования вызывает включенный. Включение используется, чтобы показать, как разбить вариант использования на несколько более мелких шагов. Включенный вариант использования находится на окончании с наконечником стрелки. |
| Расширение | Расширяющий вариант использования добавляет цели и шаги в расширяемый вариант использования. Расширения работают только при определенных условиях. Расширенный вариант использования находится на окончании с наконечником стрелки. |
| Наследование | Устанавливает отношение между специализированным и обобщенным элементом. Обобщенный элемент находится на окончании с наконечником стрелки. Специализированный вариант использования наследует цели и субъекты своего обобщения и может добавлять более конкретные цели и шаги для их достижения. Специализированный субъект наследует варианты использования, атрибуты и ассоциации своего обобщения и может добавлять дополнительные объекты. |
| Зависимость | Указывает, что конструкция источника зависит от конструкции целевого объекта. |

1. Что такое точка расширения (extension point), условие активации (activationcondition)иветка расширения (extension branch)?

Точка расширения (Extension Point)

Что это: Конкретное место в основном Use Case, где может быть подключено дополнительное поведение из расширяющего Use Case.

Аналогия: Это как "розетка" в основном сценарии, в которую можно "включить" дополнительную функциональность.

Условие активации (Activation Condition)

Что это: Булево выражение, которое определяет, когда именно должно сработать расширение.

Аналогия: Это "выключатель", который решает, будет ли использована "розетка".

Формат: Логическое условие, которое должно быть истинно для активации расширения.

Ветка расширения (Extension Branch)

Что это: Последовательность шагов из расширяющего Use Case, которая выполняется когда условие активации истинно.

Аналогия: Это "удлинитель с прибором", который включается в розетку когда нужно.

Содержит: Конкретные действия, которые добавляются к основному потоку.

1. Какие другие элементы UML допустимо использовать на диаграмме UC, кроме основных?

Пакеты, артефакты, комментарии, зависимости, интерфейсы

1. Назовите основные типы актёров.

Пользователи, программные системы, аппаратные средства, время.

1. Что такое абстрактное действующее лицо?