1. **Что такое UML?**

**UML** – унифицированный язык моделирования (Unified Modeling Language) – это система обозначений, которую можно применять для объектно-ориентированного анализа и проектирования. Его можно использовать для визуализации, спецификации, конструирования и документирования программных систем.

1. **Перечислите типы диаграмм UML.**

UML 1.5 определял двенадцать типов диаграмм, разделенных на три группы:

* четыре типа диаграмм представляют статическую структуру приложения;
* пять представляют поведенческие аспекты системы;
* три представляют физические аспекты функционирования системы (диаграммы реализации).

·         диаграмма прецедентов;

·         диаграмма классов;

·         диаграмма объектов;

·         диаграмма последовательностей;

·         диаграмма взаимодействия;

·         диаграмма состояний;

·         диаграмма активности;

·         диаграмма развертывания.

**Диаграммы для изображения структуры системы**:

* Диаграмма компонентов (component diagram, тег *component*);
* Диаграмма размещения (deployment diagram, тег *deployment*);
* Диаграмма классов (class diagram, тег *class*);
* Диаграмма объектов (object diagram, тег *object*);
* Диаграмма структуры внутренней (composite structure diagram, тег *class*);

**Диаграммы для изображения поведения системы**:

* Диаграмма синхронизации (interaction diagram, тег *timing*);
* Диаграмма деятельности (activity diagram, тег *activity*);
* Диаграмма последовательности (sequence diagram, тег *sd*);
* Диаграмма коммуникации (communication diagram, тег *comm*);
* Диаграмма автомата (state machine diagram, тег *state machine*);
* Обзорная диаграмма взаимодействия (interaction overview diagram, тег *interaction*);

Особняком стоят диаграммы:

* Диаграмма использования (use case diagram, тег use case);
* Диаграмма пакетов (package diagram, тег *package*);

1. **Какие отношения между классами могут быть на диаграмме классов?**
2. **Обобщение**

Отношение обобщения — это наследование.

1. **Ассоциация**

Ассоциация показывает отношения между объектами-экземплярами класса.

1. **Бинарная**
2. **N-арная ассоциация**
3. **Агрегация**

Простая *агрегация* предполагает, что части, отделенные от целого, могут продолжать свое существование независимо от него.

1. **Композиция**

Под композитным же агрегированием понимается ситуация, когда целое владеет своими частями и их время жизни соответствует времени жизни целого, т. е. независимо от целой части существовать не могут.

1. **Зависимость**
2. **Реализация**

Реализация предполагает определение интерфейса и его реализация в классах.

1. **Как обозначаются абстрактные классы на диаграмме классов?**

Как и обычный класс.

1. **Как обозначаются интерфейсы на диаграмме классов?**



1. **Как отображается доступность членов класса на диаграмме классов?**

Для атрибутов и операций может быть указан один из трех типов видимости:

— — private (частный)

# — protected (защищенный)

+ — public (общий)

1. **Что такое агрегация? Как обозначается?**

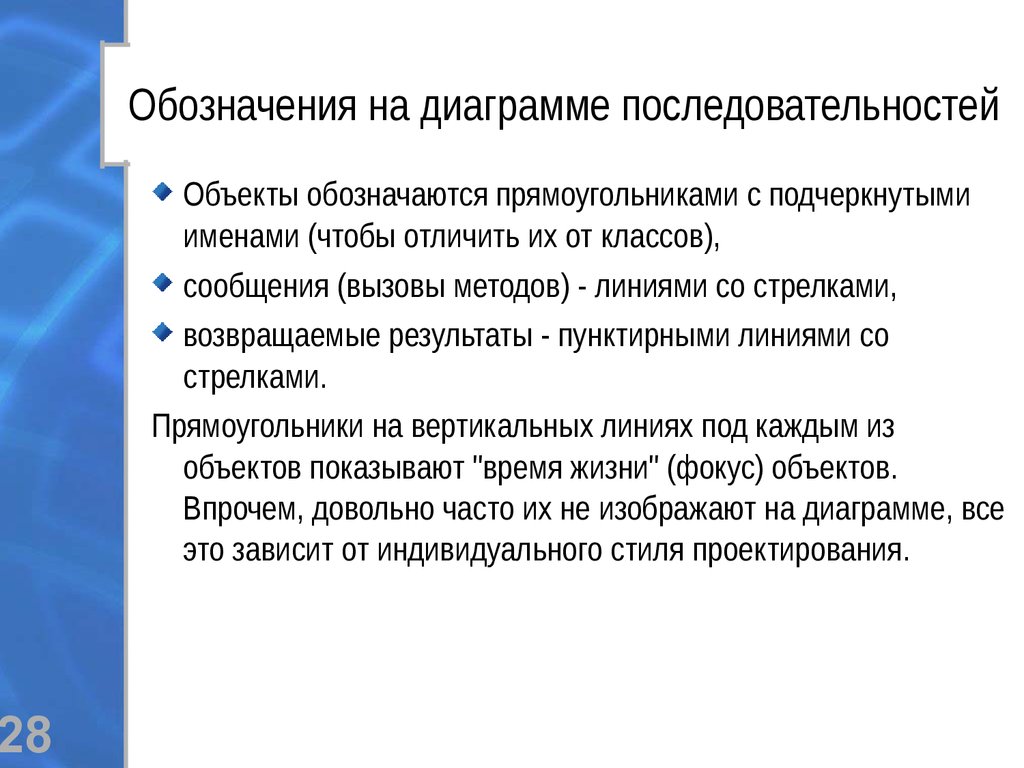
От композиции следует отличать агрегацию. Она также предполагает отношение **HAS A**, но реализуется она иначе:

При агрегации реализуется слабая связь, то есть в данном случае объекты Car и Engine будут равноправны. В конструктор Car передается ссылка на уже имеющийся объект Engine. И, как правило, определяется ссылка не на конкретный класс, а на абстрактный класс или интерфейс, что увеличивает гибкость программы.

1. **Что такое ассоциация?**

Ассоциация - это отношение, при котором объекты одного типа неким образом связаны с объектами другого типа. Например, объект одного типа содержит или использует объект другого типа.

1. **Какие обозначения используют на диаграмме последовательности?**



1. **Для чего нужна диаграмма последовательности?**

Диаграммы последовательности (sequence diagram) являются видом диаграмм взаимодействия языка UML, которые описывают отношения объектов в различных условиях. Условия взаимодействия задаются сценарием, полученным на этапе разработки [диаграмм вариантов использования](https://pro-prof.com/archives/2594)

1. **Каково назначение диаграммы использования, пакетов и активности?**

Вариант использования описывает, с точки зрения действующего лица, группу действий в системе, которые приводят к конкретному результату.  
Варианты использования являются описаниями типичных взаимодействий между пользователями системы и самой системой. Они отображают внешний интерфейс системы и указывают форму того, что система должна сделать (именно что, а не как).

Диаграмма пакетов служит, в первую очередь, для организации элементов в группы по какому-либо признаку с целью упрощения структуры и организации работы с моделью системы.

**Диаграмма активностей (видов деятельности)** - как и [диаграмма состояний](https://flexberry.github.io/ru/fd_statechart-diagram.html), **отражает динамические аспекты поведения системы**. По существу, эта диаграмма представляет собой блок-схему, которая наглядно показывает, как поток управления переходит от одной деятельности к другой.