**Ассоциация** означает, что объекты двух классов могут ссылаться один на другой, иметь некоторую связь между друг другом.

Композиция – это когда двигатель не существует отдельно от автомобиля. Он создается при создании автомобиля и полностью управляется автомобилем. В типичном примере, экземпляр двигателя будет создаваться в конструкторе автомобиля.

Композиция – это когда объект не только является частью другого объекта, но и вообще не может принадлежат еще кому-то.

Агрегация – это когда экземпляр двигателя создается где-то в другом месте кода, и передается в конструктор автомобиля в качестве параметра.

**Агрегация** — отношение когда один объект является частью другого.

## **Назовите основные принципы ООП.**

* Инкапсуляция - сокрытие реализации.
* Наследование - создание новой сущности на базе уже существующей.
* Полиморфизм - возможность иметь разные формы для одной и той же сущности.
* Абстракция - набор общих характеристик.

## **Что такое статическое и динамическое связывание?**

Присоединение вызова метода к телу метода называется связыванием. Если связывание проводится компилятором (компоновщиком) перед запуском программы, то оно называется статическим или ранним связыванием (early binding).

В свою очередь, позднее связывание (late binding) это связывание, проводимое непосредственно во время выполнения программы, в зависимости от типа объекта. Позднее связывание также называют динамическим (dynamic) или связыванием на стадии выполнения (runtime binding). В языках, реализующих позднее связывание, должен существовать механизм определения фактического типа объекта во время работы программы, для вызова подходящего метода. Иначе говоря, компилятор не знает тип объекта, но механизм вызова методов определяет его и вызывает соответствующее тело метода. Механизм позднего связывания зависит от конкретного языка, но нетрудно предположить, что для его реализации в объекты должна включаться какая-то дополнительная информация.

Для всех методов Java используется механизм позднего (динамического) связывания, если только метод не был объявлен как final (приватные методы являются final по умолчанию).