|  |  |  |  |
| --- | --- | --- | --- |
|  | Наследование | Агрегация | Композиция |
| Определение | это когда класс-наследник имеет все поля и методы родительского класса, и, как правило, добавляет какой-то новый функционал или/и поля. | Это когда один класс может существовать отдельно от другого класса, но при один класс включает объект другого класса | Это когда один класс не может существовать без другого. |
| Пример | Легковой автомобиль ***является*** автомобилем. Вполне естественно, если он будет его наследником | Автомобиль ***имеет*** двигатель.  Когда экземпляр двигателя создается где-то в другом месте кода, и передается в конструктор автомобиля в качестве параметра. | Автомобиль ***имеет*** двигатель.  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. |
| Применение | Возьму часть или все + добавлю свое | Воспользуюсь объектом двигателя в классе автомобиль, так как планирую еще использовать этот объект в классе грузовик | Воспользуюсь объектом двигателя в классе автомобиль. Не планирую пользоваться двигателем в других классах. |
| Описание классов | Класс-наследник и класс-родитель описываются друг от друга отдельно | Агрегационный класс(объект, которого включается в состав другого класса) описывается отдельно | Композиционные класс(который неотделим) может быть описан отдельно, а также внутри другого класса |
| код | class Vehicle  {  bool hasWheels;  }  class Car : Vehicle  {  string model = "Porshe";  int numberOfWheels = 4  }``` | class Engine  {  int power;  public Engine(int p)  {  power = p;  }  }  class Car  {  string model = "Porshe";  Engine engine;  public Car(Engine someEngine)  {  this.engine = someEngine;  }  }  Engine goodEngine = new Engine(360);  Car porshe = new Car(goodEngine); | class Engine  {  int power;  public Engine(int p)  {  power = p;  }  }  class Car  {  string model = "Porshe";  Engine engine;  public Car()  {  this.engine = new Engine(360);  }  } |
| class Car  {  string model = "Porshe";  class Engine  {  int power;  public Engine(int p)  {  power = p;  }  }  public Car()  {  this.engine = new Engine(360);  }  } |