User - Task → Aggregation
Planner - Task → Composition

Aggregation bu bir “Parça Bütün” ilişkisidir ama parçalar bir üst katmansız yaşayabilir yani onlardan bağımsızdır.

UML’de gösterimi Boş Elmas ile gösterilir.

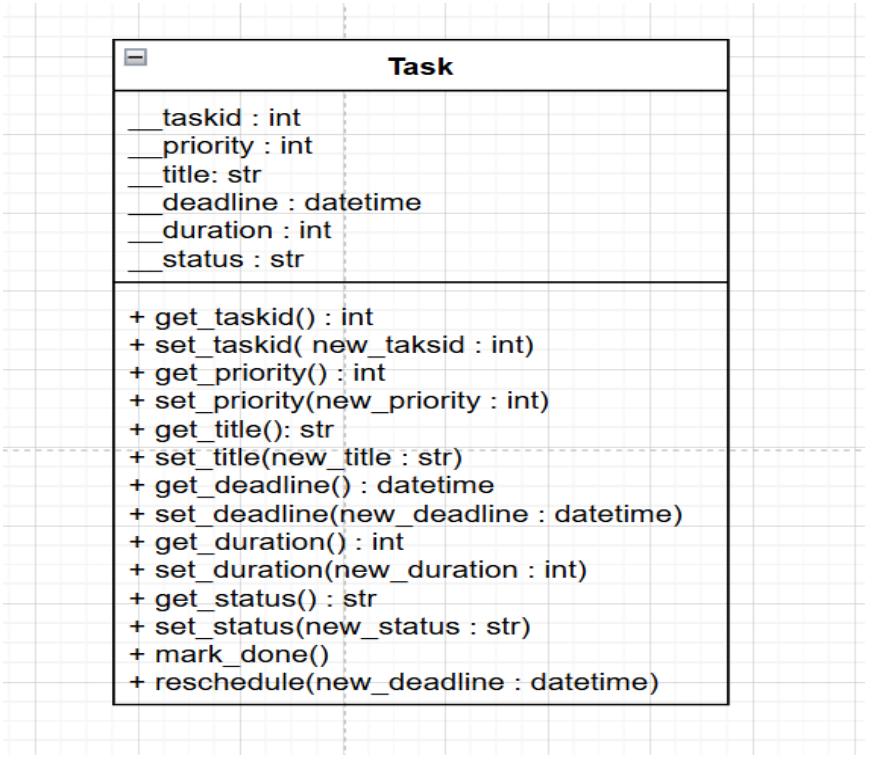
mesela
User → Task
burda Task class’ı User Classı var olmadan da var olabilir demek anlamına geliyor.

Composition dediğimiz şey ise de bir “Parça Bütün” ilişkisidir ama parçalar bir üst katmandan bağımsız DEĞİLDİR. Yani bütün silinirse parçalar da silinir anlamına geliyor.

UML’de gösterini Dolu Elmas ile gösterilir.

Planner ◆——> Task

TASK SINIFI



__ kullanarak Encapsulation uyguladık.
getter ve setter methodları ile kontrollü bir erişim sağladık.

filter_tasks → Görevleri belirli bir statüye göre getirir.

sort_tasks → Görevleri deadline veya başka kritere göre sıralar.

`sum(1 for t in self.__tasks if t.get_status() == "done")` kısmı, Python'daki "**generator expression**" + `sum()` kullanımına çok güzel bir örnek.
Bunu adım adım açıklayayım:

🧠 Genel Mantık

Bu ifade şu işe yarıyor:

> "Her görev (`t`) için, eğer status = 'done' ise 1 say, sonra bunların toplamını döndür."