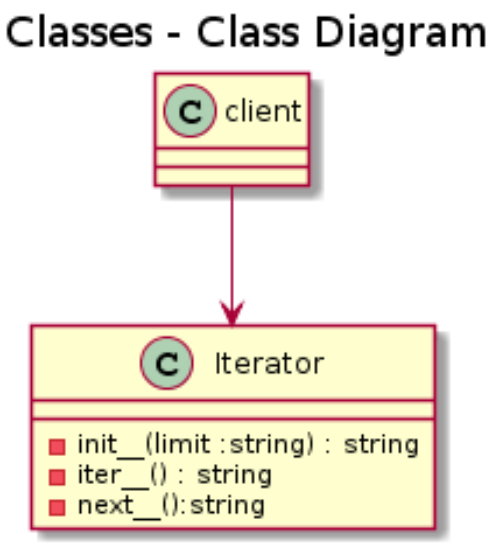
ITERATOR DESİGN PATTERN

Iterator (tekrarlayıcı) tasarım deseni, nesne koleksyonlarının (list,array,queue) elemanlarını belirlenen kurallara göre elde edilmesini düzenleyen tasarım desenidir. Nesne tabanlı dillerde uygulama geliştirilirken en sık kullanılan yapılardan biri de koleksiyonlardır. Iterator tasarım deseni ile koleksiyon yapısı bilinmesine ihtiyaç olmadan koleksiyon elemanları üzerinde işlem yapılabilmesini sağlar. Yani iterator tasarım deseni kullanılarak koleksiyonun array, queue, list olması önemli olmadan, aynı şekilde elemanlarının elde edilmesi sağlanır. Koleksiyon içindeki nesnelerin nasıl elde edileceği tercihe göre belirlenebilir. Yani sonraki, ilk, son, 3. Eleman gibi istenilen şekilde elemanlara ulaşılabilir. Iterator tasarım deseninde 5 temel yapı bulunur.

- Iterator: Koleksiyon elemanları elde edilebilmesi için gerekli işlemleri tanımlar.
- Aggregate: Koleksiyon barındıran nesnelerin Iterator tipinden nesne oluşturacağını belirten arayüzdür.
- Concrete Aggregate: Koleksiyon barındıran nesnedir. Aggregate arayüzünü uygular ve ilgili ConcreteIterator nesnesini oluşturur.
- ConcreteIterator: Aggregate yapısında ki koleksiyon elemanlarının elde edilmesini sağlayan metotları barındıran yani Iterator arayüzünü uygulayan gerçek iterator nesnesidir.
- Client: Bu yapıyı kullanarak koleksiyon içindeki elemanlara erişen yapıdır. Iterator tasarım deseni için uml diyagramı aşağıdadır.




```
@startuml

title Classes - Class Diagram


class Iterator {
  -init__(limit :string) : string
  -iter__() : string
  -next__():string



  }
  class client
  client--> Iterator

@enduml
```