Iterator

Nazwa: Facade

Drugie nazwy: Glue

Klasyfikacja: według celu – czynnościowy, według stosowania – obiektów

Jak często się używa: bardzo często

Przeznaczenie: zapewnić obiek-iterator dla dostępu do kolekcji

Opis – Iterator to wzorzec a lepiej powiedzieć obiekt który gwarantuje biezpieczność dostępu do elementu kolekcji. Dobrą metaforą będzie bank czyli kolekcja pieniendzy. W danym przypadku iteratorem będzie osoba obsługująca przy kasie. Taka osoba nie pozwoli nam wziąć nie swoje pieniądze czyli ograniczy dostęp do elementów kolekcji. Trzeba rozumieć że iterator nie tylko używamy do wrapowania łatwych tablic lub wektorów, a też przy pracy ze złożonymi obiektami. Nie wolno dawać możliwość developerowi pracować ze złożonymi konstrukcjami a tylko z pięknymi obiektami-wraperami. Developer prosi element a już iterator wie gdzie pójść i jak dostać potrzebny element. Jeżeli potrzebujemy stworzyć iterator który będzie działał na różnych kolekcjach, możemy użyć technikę tworzenia polimorfnego iteratora.

\*Strona architekta:

Zastosowanie iteratora pozwoli ukryć złożone obiekty-elementy kolekcji oraz otrzymać biezpieczny dostęp do elementów.

\*Strona programisty-realizatora:

Programista wykorzystuje przygotowany dla niego mechanizm przy pracy z kolekcją.

Kiedy użyć:

- kiedy potrzebujemy poręczny oraz biezpieczny sposób dostępu do elementów kolekcji.

- kiedy potrzebujemy mieć kylka sposobów iterowania po kolekcji

- dla polimorficznej iteracji

C# osobliwości:

Wzorzec Iterator jest ekskludowany z .NET o ile Microsoft daje alternatywny wzorzec, nazwimy go ‘Enumerator’. Do implementacje tego wzorcu w .NET wykorzystujemy implementację 2 najważniejszych interfejsów wbudowanych w .NET: Ienumerable oraz Ienumerator. Także mamy ICollection, IList oraz inne interfejsy. Ienumerable oraz Ienumerator – najbardziej prymitywne interfejsy do zadania interfejsu złożonego obiektu kolekcji.

-

Nazewnictwo:

Niema żadnych reguł dotyczących nazewnictwa

UML

C#