

Instrukcja Iteratora jako wzorca projektowego

Krzysztof Brylski

23 kwietnia 2025



Politechnika
Wrocławska

1 Wstęp

Niniejsza instrukcja przeprowadzi cię przez zasady implementacji wzorca projektowego nazwanego "Iterator"

1.1 Co to jest iterator

Iterator to obiekt, który umożliwia sekwencyjne przechodzenie (iterowanie) przez elementy jakiejś kolekcji (np. listy, zbioru, drzewa), bez konieczności znajomości jej wewnętrznej struktury. Kluczową ideą iteratora jest dostarczanie jednolitego interfejsu do pobierania kolejnych elementów z kolekcji, zazwyczaj w sposób „leniwy” — tzn. tylko wtedy, gdy są potrzebne.

2 Opis problemu

W przypadku w którym mamy jakąś złożoną strukturę danych jak graf albo drzewo, iterator daje nam możliwość przejścia przez każdy jej element bez znajomości szczegółów implementacyjnych

3 Główne cechy iteratora:

Typowy iterator powinien zawierać kilka cech (metod) które umożliwią mu współpracę z wybraną strukturą danych

1. Stan wewnętrzny - iterator powinien przechowywać swoją pozycję w strukturze danych
2. Możliwość pobrania elementu na którym znajduje się iterator
3. Możliwość sprawdzenia czy istnieje następny element w strukturze danych
4. Metoda zwracająca następny element (oraz przesuwająca Stan wewnętrzny na następny element)
5. Metoda pozwalająca na zresetowanie Stanu wewnętrznego

4 Ogólne kroki implementacji

1. Zdefiniuj strukturę danych na której iterator ma pracować np. lista
2. Utwórz klasę iteratora
3. zaimplementuj sposób inicjalizacji iteratora dla wybranego startowego stanu wewnętrznego (domyślnie pierwszy element)
4. Zaimplementuj metodę dostępu do obecnego elementu
5. Zaimplementuj metody do sprawdzania czy następny element istnieje, oraz możliwość pobrania następnego elementu (pamiętaj żeby nie było możliwości pobrania następnego elementu kiedy nie istnieje)
6. zaimplementuj możliwość zresetowania iteratora oraz jeśli czas pozwoli opcje edycji i usuwania aktualnego obiektu na który wskazuje iterator

5 Wady i zalety

Główną zaletą iteratorów jest możliwość przejścia po wszystkich elementach w strukturze danych bez konieczności znajomości jej szczegółów implementacyjnych

Wadom natomiast jest konieczność utrzymywania w pamięci dodatkowego obiektu (iteratora) oraz przypadkowe nadpisanie jakiś danych za pomocą iteratora (jeśli daje takie możliwości)