

Patterns You Already Use

Iterator

Jeremy Clark
www.jeremybytes.com
jeremy@jeremybytes.com



pluralsight 
hardcore developer training

Iterator Pattern



Provide a way to access the elements of an aggregate object sequentially without exposing its underlying representation.

Gamma, et al, *Design Patterns*. Addison-Wesley, 1994.

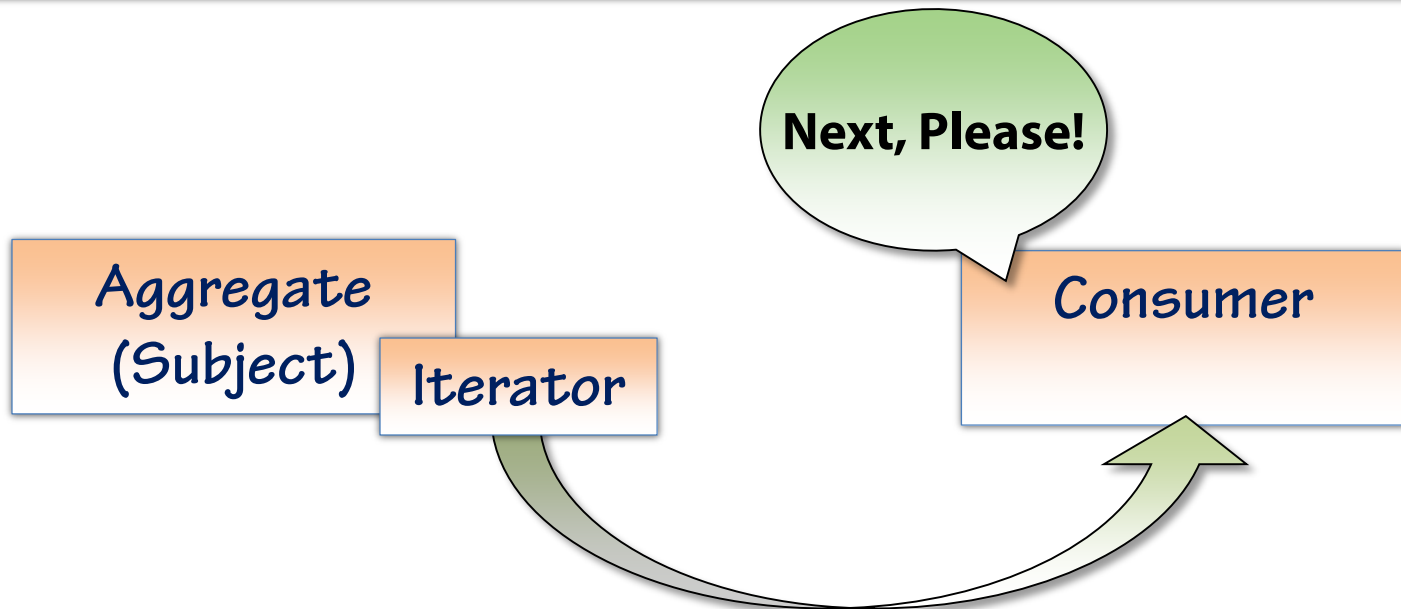
Real World Iterator

- **TV Remote**



Iterator Translation

Provide a way to access the elements of an aggregate object sequentially without exposing its underlying representation.



Iterator Consequences

- **Adding or removing items from an aggregate during iteration can invalidate the Iterator.**
- **Multiple Iterators can act on a single aggregate object.**

IEnumerable<T> Implementers

Collections

List<T>
Array
ArrayList
SortedList<TKey, TValue>
HashTable
Queue / Queue<T>
Stack / Stack<T>
Dictionary<TKey, TValue>
ObservableCollection<T>
+
Custom Types

IEnumerable<T> and IEnumerator<T>

IEnumerable<T>

IEnumerator<T> GetEnumerator()

IEnumerator<T>

T Current

bool MoveNext()

void Reset()

void Dispose()

Fibonacci Sequence

- Each number is the sum of the previous 2 numbers.

1, 1, 2, 3, 5, 8, 13

$$1 + 1 = 2$$

$$1 + 2 = 3$$

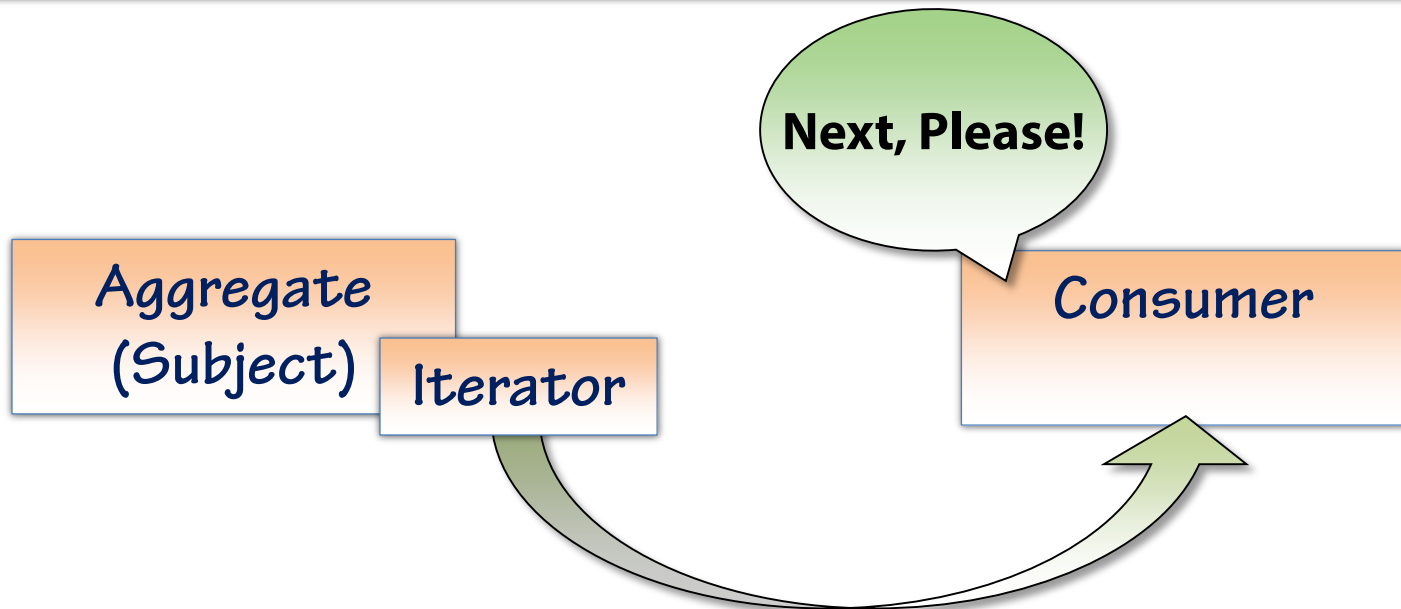
$$2 + 3 = 5$$

$$3 + 5 = 8$$

$$5 + 8 = 13$$

Iterator Review

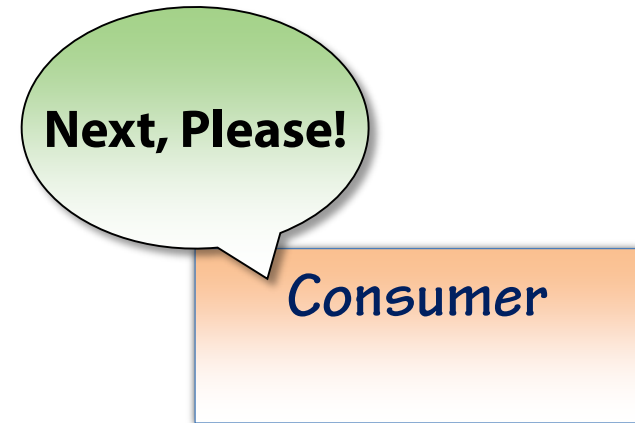
Provide a way to access the elements of an aggregate object sequentially without exposing its underlying representation.



Summary

■ Iterator

- Next, Please!
- Everyday usage: foreach
- IEnumerable<T>
- yield return
- MP3 Library
- LINQ



- ## ■ Next Up: Patterns You Already Use
- Chain of Responsibility and Facade

