

BENEFITS

The following lists the benefits of using the Interpreter pattern:

- Easy to change and extend the grammar
- Implementing the grammar is easy

WHEN TO USE

You should use the Interpreter pattern when:

- The grammar of the language is simple.
- Efficiency is not a critical concern.

Iterator Pattern

The Iterator pattern provides a consistent way to sequentially access items in a collection that is independent of and separate from the underlying collection. Figure 4–16 illustrates the Iterator pattern.

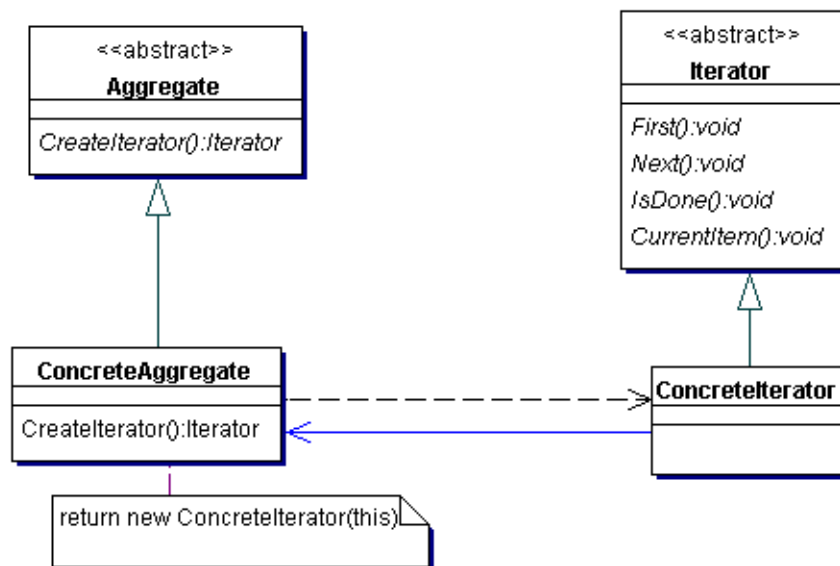


Figure 4–16 *The Iterator Pattern*

BENEFITS

The following lists the benefits of using the Iterator pattern:

- Supports variations in the traversal of a collection
- Simplifies the interface of the collection

78 Chapter 4 | Design Patterns

When to Use

You should use the Iterator pattern when you want to:

- Access collection object's contents without exposing its internal representation.
- Support multiple traversals of objects in a collection.
- Provide a uniform interface for traversing different structures in a collection.

Mediator Pattern

The Mediator pattern simplifies communication among objects in a system by introducing a single object that manages message distribution among other objects. The Mediator pattern promotes loose coupling by keeping objects from referring to each other explicitly, and it lets you vary their interaction independently. Figure 4–17 illustrates the Mediator pattern.

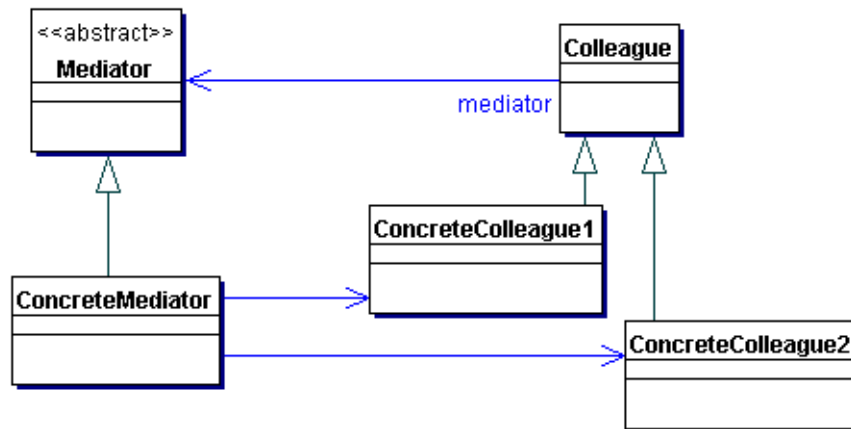


Figure 4–17 *The Mediator Pattern*

BENEFITS

The following lists the benefits of using the Mediator pattern:

- Decouples colleagues
- Simplifies object protocols
- Centralizes control