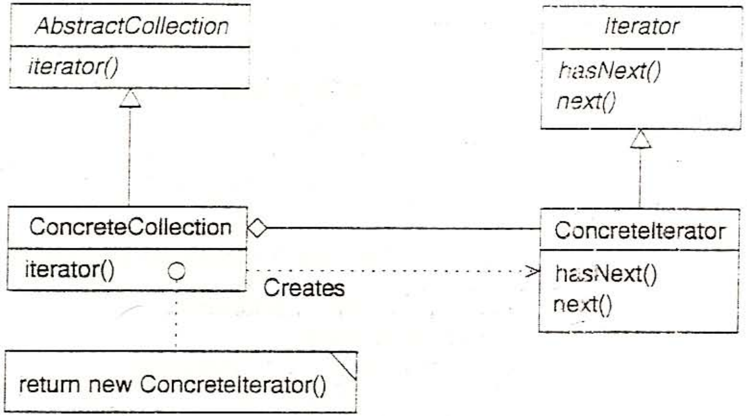
The structure of the Iterator pattern is illustrated in the following diagram:



The participants of the Iterator pattern are as follows:

- *Iterator* (e.g., `Iterator`), which defines an interface for accessing and traversing the elements.
- *ConcreteIterator* (e.g., `LinkedListIterator`), which implements the iterator interface and keeps track of the current position in the traversal of the collection.
- *AbstractCollection* (e.g., `List`), which defines an interface for creating a concrete iterator (e.g., the `iterator()` method).
- *ConcreteCollection* (e.g., `LinkedList`), which implements the `iterator()` method to return an instance of a proper concrete iterator.

The Iterator design pattern is used in the Java Collections Framework to provide