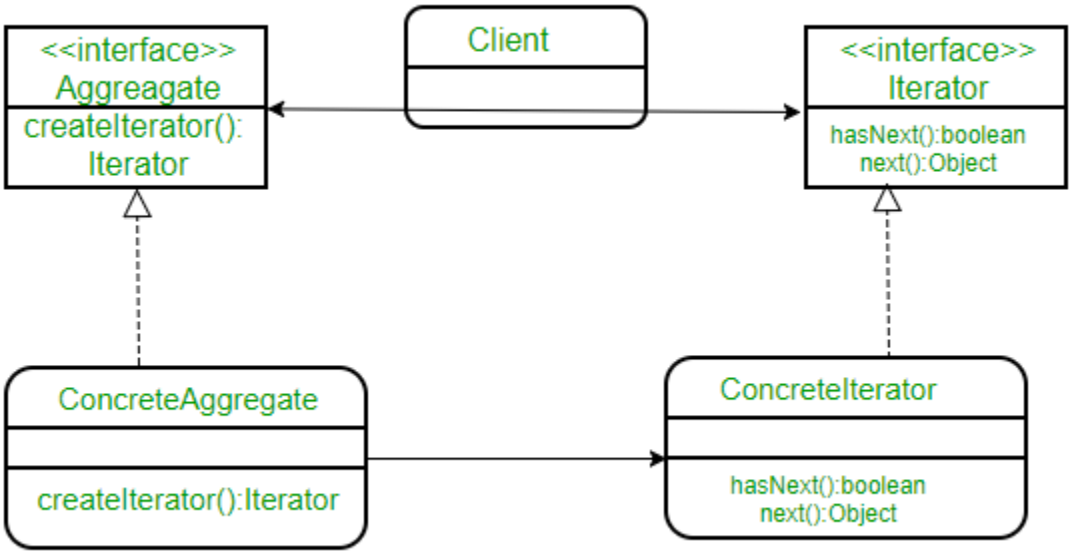
Iterator Pattern

Kurzbeschreibung & Zweck

Der Begriff Iterator stammt aus dem Bereich der Softwareentwicklung und bezeichnet einen Zeiger, mit dem die Elemente einer Menge durchlaufen werden können (z. B. eine Liste). Der Begriff leitet sich aus der mathematischen Methode der Iteration ab. Der Iterator wird insbesondere im Bereich der Datenbank meist Cursor genannt.

UML-Diagramm



Beschreibung

Ein Iterator ist ein spezieller Zeiger, der innerhalb eines Programms vom Software-Entwickler dazu verwendet werden kann, um auf Elemente einer Menge, vereinfacht eine Liste, zuzugreifen. Iteratoren arbeiten nach dem Grundprinzip „Wenn es ein weiteres Element in der Liste gibt, dann stelle es zur Verfügung.“

Dies ist vereinfacht damit vergleichbar, wie man einen Text, der eine Liste von Worten ist, liest: „Wenn es ein nächstes Wort gibt, dann lies es. Wenn kein weiteres Wort mehr folgt, ist der Text beendet.“ In jedem als Iteration bezeichneten Zugriffs-Schritt steht somit exakt ein Wort des Textes zur Bearbeitung zur Verfügung.

Viele der in der Programmierpraxis verwendeten Iteratoren stellen über die lesende Zugriffsmöglichkeit hinaus Mechanismen zur Verfügung, die ein aktuell gelesenes Element aus der Liste entfernen oder ein neues Element in die Liste aufnehmen, so, wie bei der Bearbeitung eines Textes Worte eingefügt oder gelöscht werden können.

Beispiel-Code in Java

Main:

*MainIt.java | Person.java | PersonIterator.java

```java
package business;

import business.entity.Person;
import persistence.PersonIterator;

public class MainIt {

    public static void main(String[] args) {

        PersonIterator persit = new PersonIterator();
        // Testdaten
        for (int i = 0; i <= 9; i++) {
            Person pers = new Person();
            pers.setVorname("vorname"+i);
            pers.setNachname("nachname"+i);
            pers.setEmail(pers.getVorname()+pers.getNachname()+"@email.ch");
            persit.add(pers);
        }

        // Ausgabe generieren
        while (persit.hasNext()) {
            Person pers = persit.next();
            System.out.println(pers.getVorname() + " " + pers.getNachname() + ": " + pers.getEmail());
        }
    }
}
```

Iterator:

*MainIt.java | Person.java | PersonIterator.java

```java
package persistence;

import java.util.ArrayList;
import java.util.Iterator;
import java.util.List;

import business.entity.Person;

public class PersonIterator implements Iterator<Person> {

    List<Person> data = new ArrayList<Person>();
    int index = 0;

    public void add(Person pers) {
        this.data.add(pers);
    }

    @Override
    public boolean hasNext() {
        if (this.index+1 <= this.data.size()) {
            return true;
        }
        return false;
    }

    @Override
    public Person next() {
        Person pers = data.get(this.index);
        this.index += 1;
        return pers;
    }

}
```