

Aula Prática - 16 Padrões de Comportamento - Iterator

Intenção

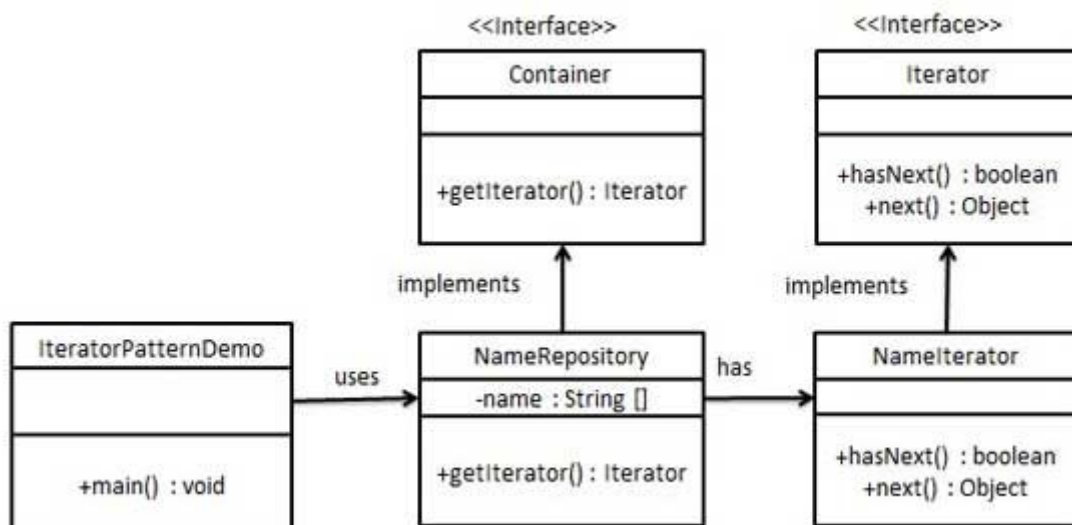
Prover uma forma de acessar os elementos de um conjunto em sequência sem expor a representação interna deste conjunto. Também conhecido como: Cursor.

Usar este padrão quando...

- Quiser acessar objetos agregados (coleções) sem expor a estrutura interna;
- Quiser prover diferentes meios de acessar tais objetos;
- Quiser especificar uma interface única e uniforme para este acesso.

Vantagens e desvantagens

- Múltiplas formas de acesso:
 - Basta implementar um novo iterador com uma nova lógica de acesso.
- Interface simplificada:
 - Acesso é simples e uniforme para todos os tipos de coleções.
- Mais de um iterador:
 - É possível ter mais de um acesso à coleção em pontos diferentes.



Passo 1

Crie interfaces.

Iterator.java

```
public interface Iterator {  
    public boolean hasNext();  
    public Object next();  
}
```

Container.java

```
public interface Container {  
    public Iterator getIterator();  
}
```

Passo 2

Crie uma classe concreta implementando a interface Container . Esta classe tem a classe interna NameIterator implementando a interface Iterator.

NameRepository.java

```
public class NameRepository implements Container {  
    public String names[] = {"Robert" , "John" ,"Julie" , "Lora"};  
  
    @Override  
    public Iterator getIterator() {  
        return new NameIterator();  
    }  
  
    private class NameIterator implements Iterator {  
  
        int index;  
  
        @Override  
        public boolean hasNext() {  
  
            if(index < names.length){  
                return true;  
            }  
            return false;  
        }  
  
        @Override  
        public Object next() {  
  
            if(this.hasNext()){  
                return names[index++];  
            }  
            return null;  
        }  
    }  
}
```

Passo 3

Use o NameRepository para obter o iterador e os nomes de impressão.

IteratorPatternDemo.java

```
public class IteratorPatternDemo {  
  
    public static void main(String[] args) {  
        NameRepository namesRepository = new NameRepository();  
  
        for(Iterator iter = namesRepository.getIterator(); iter.hasNext();){  
            String name = (String)iter.next();  
            System.out.println("Name : " + name);  
        }  
    }  
}
```

Passo 4

Teste sua implementação!