UNIVERSITÀ DEGLI STUDI DI PADOVA

HOMEWORK 2

DOCUMENTAZIONE

10/06/2020

Corso di **Ingegneria del Software**

Descrizione

Svolgimento progetto

- Ambiente Java: Java Micro Edition CLDC1.1.
- Comportamento degli Adapter secondo la documentazione di J2SE 1.4.2.
- Utilizzo dei tipi Generics.

Design Patterns

In questo progetto sono stati utilizzati 2 Design Patterns:

- **Object pattern** (tipo di Adapter Pattern):
  Prevede un rapporto di associazione tra Adapter e Adaptee, in cui un Adapter istanzia uno o più Adaptee, avendo così un'associazione tra essi. Utilizzato quando si intende utilizzare un componente software ma occorre adattare la sua interfaccia per motivi di integrazione con l'applicazione esistente.
  Questo pattern è stato utilizzato per tutte e 4 le classi Adapter di questo progetto:
  - CollectionAdapter: contiene ed utilizza un'istanza di Vector
  - ListAdapter: contiene ed utilizza un'istanza di Vector
  - SetAdapter: contiene ed utilizza un'istanza di MapAdapter che a sua volta contiene ed utilizza un'istanza di Hashtable
  - MapAdapter: contiene ed utilizza un'istanza di Hashtable

  Questo tipo di pattern è stato utilizzato anche per le classi secondarie implementate: SubList (come List), EntrySet (come Set), KeySet (come Set), Values (come Collection).

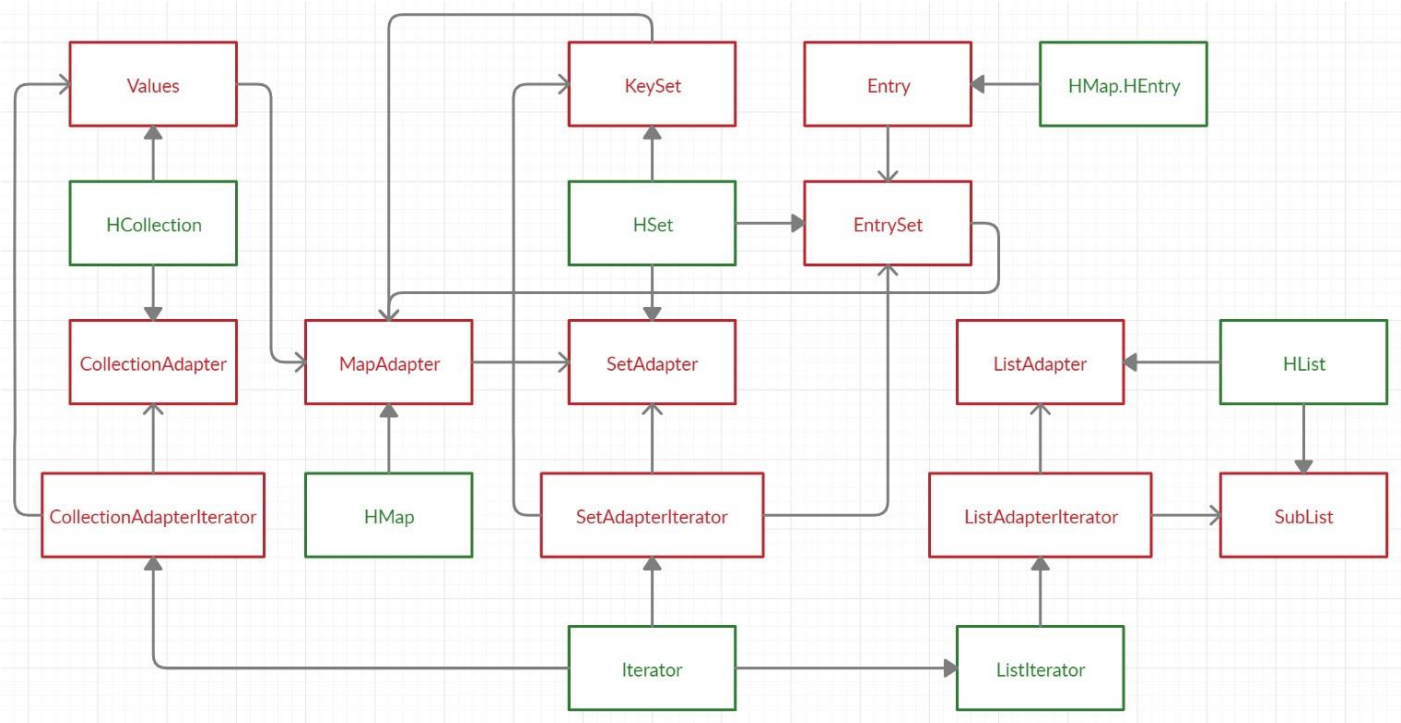
- **Iterator Pattern**:
  Nasconde le strutture dati esponendo sempre gli stessi metodi di integrazione, con l'obbiettivo di disaccoppiare l'utilizzatore con colui che implementa l'aggregazione di dati. Utilizzato quando, dato un aggregato di oggetti, si vuole accedere ai suoi elementi senza doverne esporre la struttura.
  Per ogni classe Adapter (tranne per MapAdapter) sono stati usati degli iteratori per poter scorrere e gestire i dati:
  - CollectionAdapter: CollectionAdapterIterator che implementa Iterator
  - ListAdapter: ListAdapterIterator che implementa Iterator
  - SetAdapter: SetAdapterIterator che implementa ListIterator

  Questo tipo di pattern è stato utilizzato anche per le classi secondarie implementate: SubList (come List), EntrySet (come Set), KeySet (come Set), Values (come Collection).

Test

Sono state create 4 classi test (CollectionAdapterTest, ListAdapterTest, SetAdapterTest, MapAdapterTest), una per ogni Adapter, al fine di controllare che tutte le funzioni di tutti gli Adapter venissero eseguite nel modo corretto per ogni tipologia di input, controllando anche che il lancio delle eccezioni previste avvenisse correttamente. Per eseguire i test è stata utilizzata una metodologia Test Driven Development, definendo ed implementando le Test Suite Junit per le classi sviluppate. Questo è stato fatto utilizzando il framework Junit 5 così da eseguire le 4 classi test tramite una apposita classe TestRunner. Per fare i controlli in ogni classe test sono state usati 3 tipi di Assert di Junit: assertEquals, assertArrayEquals, assertThrows. La Test Suite è stata documentata utilizzando il template "SAFe" fornito dalla consegna.

Diagramma progetto



Legenda: interfacce in verde, classi in rosso

Documentazione progetto

- Java Project: corrispondente alla cartella "Homework2".
- JavaDoc classi: corrispondente alla cartella "JavaDoc".
- SAFe TestDoc: corrispondente alla cartella "SAFe TestDoc".