

Ingegneria del Software

Homework 2

Supponete di essere in ambiente Java Micro Edition, precisamente CLDC1.1 (<https://docs.oracle.com/javame/config/cldc/ref-impl/cldc1.1/jsr139/index.html>)

Supponete di voler utilizzare in questo ambiente una libreria di classi (myLib) nata in ambiente J2SE 1.4.2 (<https://www2.cs.duke.edu/csed/java/jdk1.4.2/docs/api/index.html>), in particolare, la libreria contiene classi che fanno uso delle interfacce List, Map, Set e Collection di del Java2 Collections Framework versione 1.4.2.

Sviluppate gli adapter per le interfacce sotto elencate, si suggerisce (ma non è obbligatorio) di utilizzare le classi adaptee sotto elencate:

Target	Adaptee
Map	Hashtable
Set	Hashtable
List	Vector
Collection	Vector

Il comportamento dei zz adapters e dei loro metodi deve essere quello descritto dalla documentazione di J2SE 1.4.2.

Potete lavorare effettivamente con J2ME CLDC 1.1, nel qual caso avrete la necessità di definire sia le interfacce target (ad esempio facendo copia e incolla dalla documentazione ai link di sopra) sia le interfacce ritornate da alcuni metodi (e.g., Iterator).

Potete lavorare in J2SE, nel qual caso sia le interfacce target che le interfacce ritornate da alcuni metodi sono già presenti, MA dovete comunque implementare gli adapters utilizzando classi (ad esempio Vectorized Hashtable) disponibili in J2ME CLDC 1.1. Per tali classi considerare le sole funzionalità disponibili in J2ME CLDC 1.1.

Inoltre, se utilizzerete l'ambiente standard di Java con una versione > 1.4.2, troverete metodi aggiuntivi nelle interfacce target, uso di generics etc. Se lavorerete in ambiente J2SE con versione superiore alla 1.4.2 suggerisco di definire comunque una vostra versione delle interfacce target (e.g., HMap per Map) e delle interfacce ritornate da alcuni metodi (e.g., HListIterator per ListIterator) in modo da poter tralasciare la presenza delle funzionalità introdotte dalle versioni di Java superiori alla 1.4.2.

È OBBLIGATORIO fornire le implementazioni degli iteratori (Iterator e ListIterator) ma NON È NECESSARIO preoccuparsi di replicare il comportamento di failure per modifica concorrente delle classi che implementano Iterator e ListIterator in J2SE.

Utilizzate la metodologia Test Driven Development, e, quindi, definite ed Implementate le test suite Junit per le classi sviluppate. Documentate la vostra test suite utilizzando il template “SAFE” descritto sul sito https://jazz.net/help-dev/clm/index.jsp?re=1&topic=/com.ibm.rational.test.qm.doc/topics/r_testsuite_template_ref.html&scope=null.

Potete utilizzare il framework Junit nella versione usata a lezione o più recente, dovete dichiarare nella documentazione la versione utilizzata, le componenti del framework utilizzate e le eventuali librerie di matcher utilizzate.

Scrivate la documentazione delle classi (utilizzate il tool javadoc) fornendo almeno la descrizione delle classi e la documentazione di base (paragrafi parameters, returns, throws) dei metodi. Non vi è impedito l'uso di annotazioni avanzate, ma non è obbligatorio.

Scrivate una breve relazione che identifichi tutti i design pattern utilizzati, ne descriva l'esatto uso nel vostro software e ne giustifichi la scelta con vantaggi e svantaggi.