Appello 4 A.A. 24-25

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 dell'interfaccia Map, e quindi delle interfacce ad essa connesse (Collection, Set, Iterator) del Java2 Collections Framework versione 1.4.2.

Sviluppate l'***adapter*** per Map utilizzando come ***adaptee*** per Map la classe Hashtable di CLDC 1.1. Siete liberi di decidere come realizzare Collection, Set e Iterator, ma ricordate che la vostra implementazione deve essere consistente con quanto documentato in J2SE 1.4.2.

Lavorate in ambiente Java versione corrente, ma ricordate che e' FONDAMENTALE che il vostro codice utilizzi solo le funzionalita' presenti in CLDC 1.1 per realizzare l'adapter.

Per evitare collisioni con le interfacce Map, Collection, Set e Iterator della versione corrente di Java ***dovete*** definire localmente al package del vostro adapter (package che ***dovete*** chiamare *myAdapter* senza ulteriori livelli di nidificazione) le interfacce HMap, HCollection, HSet e HIterator con tutti i metodi delle interfacce Map, Collection, Set ed Iterator della versione 1.4.2 di Java completamente funzionanti. La vostra classe adapter ***deve*** chiamarsi MapAdapter e ***deve*** appartenere al package myAdapter.

Il comportamento dei vostri adapters e dei loro metodi ***deve*** essere esattamente quello descritto dalla documentazione di J2SE 1.4.2 (attenzione al backing delle viste) e ***devono*** essere implementate tutte le optional operations.

Devono essere compliant con la documentazione della versione 1.4.2 anche gli iteratori e ***devono*** essere implementate anche tutte le optional operations degli iteratori stessi. La o le classi che implementano gli iteratori ***devono*** far parte del package myAdapter e ***deve/devono*** implementare l'interfaccia HIterator.

Dovete utilizzare la metodologia **Test Driven Development**, e, quindi, definire ed Implementare i test case Junit per le classi sviluppate. Le classi di test ***devono*** essere contenute in un package myTest (senza ulteriori livelli di nidificazione).

Il package deve contenere una classe TestRunner che possa essere invocata da linea di comando, eseguire tutti i test da voi definiti, fornire il risultato dei test ed il numero complessivo di test eseguiti, il numero di test falliti ed il tempo richiesto dall'esecuzione dei test.

Dovete documentare il/i vostri test case utilizzando il template nella tabella 1 di questo documento.

Dovete documentare ogni test method secondo il template descritto nella tabella 2 di questo documento.

E' possibile fornire la documentazione in formato pdf o in formato javadoc, in entrambi i casi le diverse voci per i diversi test{case, method} devono essere facilmente leggibili e distinguibili dalle altre.
La documentazione ***deve*** essere fornita nella vostra consegna.

Si suggerisce di utilizzare il framework JUnit nella versione usata a lezione, dovete comunque dichiarare nella documentazione la versione utilizzata, e fornire tutte le componenti del framework utilizzate in formato di uno o piu' jar contenuti in una cartella JUnit posizionata allo stesso livello della radice del Classpath.

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

N.B. La consegna **deve** consistere in un file zip che contenga la struttura delle cartelle richiesta e tutta la documentazione (sia adapter che test). Tutto **deve** poter essere compilato utilizzando il compilatore da riga di comando in versione >= 1.4.2. TestRunner **deve** poter essere eseguito da riga di comando. Non devono essere presenti dipendenze dal vostro specifico ambiente di sviluppo.

Tabella 1

Section	Section description
Summary	A detailed description of the test suite. In this section, you can also set up test suite categories to help you organize your test suite into logical groups. This section uses a full-text editor.
Test Case Design	A description of the motivation and rationale of the design of the test case.

Tabella 2

Section	Section description
Summary	A description of the test case. In this section, you can also set up categories to organize your test cases into logical groups. This section uses a full-text editor.
Test Case Design	The overall design of the test case, including any background setup information or topologies. A description of the motivation and rationale of the design of the test method. This section uses a full-text editor.
Test Description	The detailed description of the test method, including the actions performed step by step. The description might include a project-unique identifier of the test case, prerequisites for running the test, a description of the test data, an explanation of the criteria to evaluate the results with, and any assumptions or constraints that are associated with the test case. This section uses a full-text editor.
Pre-Condition	The information that must be true before you run the test case. This section uses a full-text editor.
Post-Condition	The information that must be true after you run the test case. This section uses a full-text editor.
Expected Results	The conditions that must be met before a test case is considered to be successful. This section uses a full-text editor.