Homework 1

- toArrayTest
- toArrayTest

KeySetTester
- constructorTester
- addTest
- addAllTest
- clearTest
- containsTest
- containsAllTest
- equalsTest
- hashTest
- isEmptyTest
- iteratorTest
- removeTest
- removeAllTest
- retainAllTest
- sizeTest
- toArrayTest
- toArrayTest

ValueCollectionTester
- constructorTester
- addTest
- addAllTest
- clearTest
- containsTest
- containsAllTest
- equalsTest
- hashTest
- isEmptyTest
- iteratorTest
- removeTest
- removeAllTest
- retainAllTest
- sizeTest
- toArrayTest
- toArrayTest

IteratorTester
- constructorTester
- hasNextTest
- nextTest
- removeTest

ListIteratorTester

- constructorTester
- addTest
- hasNextTest
- hasPrevTest
- nextTest
- nextIndexTest
- prevTest
- prevIndexTest
- removeTest
- setTest

ListTester
- constructorTester
- addTest
- addIndexTest
- addAllTest
- clearTest
- containsTest
- containsAllTest
- equalsTest
- getTest
- hashTest
- indexOfTest
- isEmptyTest
- iteratorTest
- lastIndexOfTest
- listIteratorTest
- removeIndexTest
- removeObjectTest
- removeAllTest
- retainAllTest
- setTest
- sizeTest
- subListTest
- toArrayTest
- toArrayObjectTest

LimitedListTester
- constructorTester
- addTest
- addIndexTest
- addAllTest
- clearTest
- containsTest

containsAllTest
equalsTest
getTest
hashTest
indexOfTest
isEmptyTest
lastIndexOfTest
listIteratorTest
removeIndexTest
removeObjectTest
removeAllTest
retainAllTest
setTest
sizeTest
subListTest
toArrayTest
toArrayObjectTest

MapTester

▼ Summary

Questo test mira a verificare il corretto funzionamento dei metodi della mappa, partendo dal test del costruttore si va verificare che ogni metodo ritorni i valori corretti o effettui le corrette modifiche all'interno della mappa. I metodi che ritornano Set e Collection vengono testati a parte per verificarne in maniera più metodica il corretto funzionamento

▼ Test Suite Design

Il test è disegnato in maniera tale da testare casi generali di creazione di mappe e "sottoprodotti" (Set e Collection), aggiunte, rimozioni, confronti, ottenimento di valori e sostituzioni dei valori interni alla mappa. Ogni metodo di test è autonomo e non necessita di variabili private interne alla classe di test.

▼ Pre-Condition

Deve esistere una classe MapAdapter implementata secondo l'interfaccia di HMap

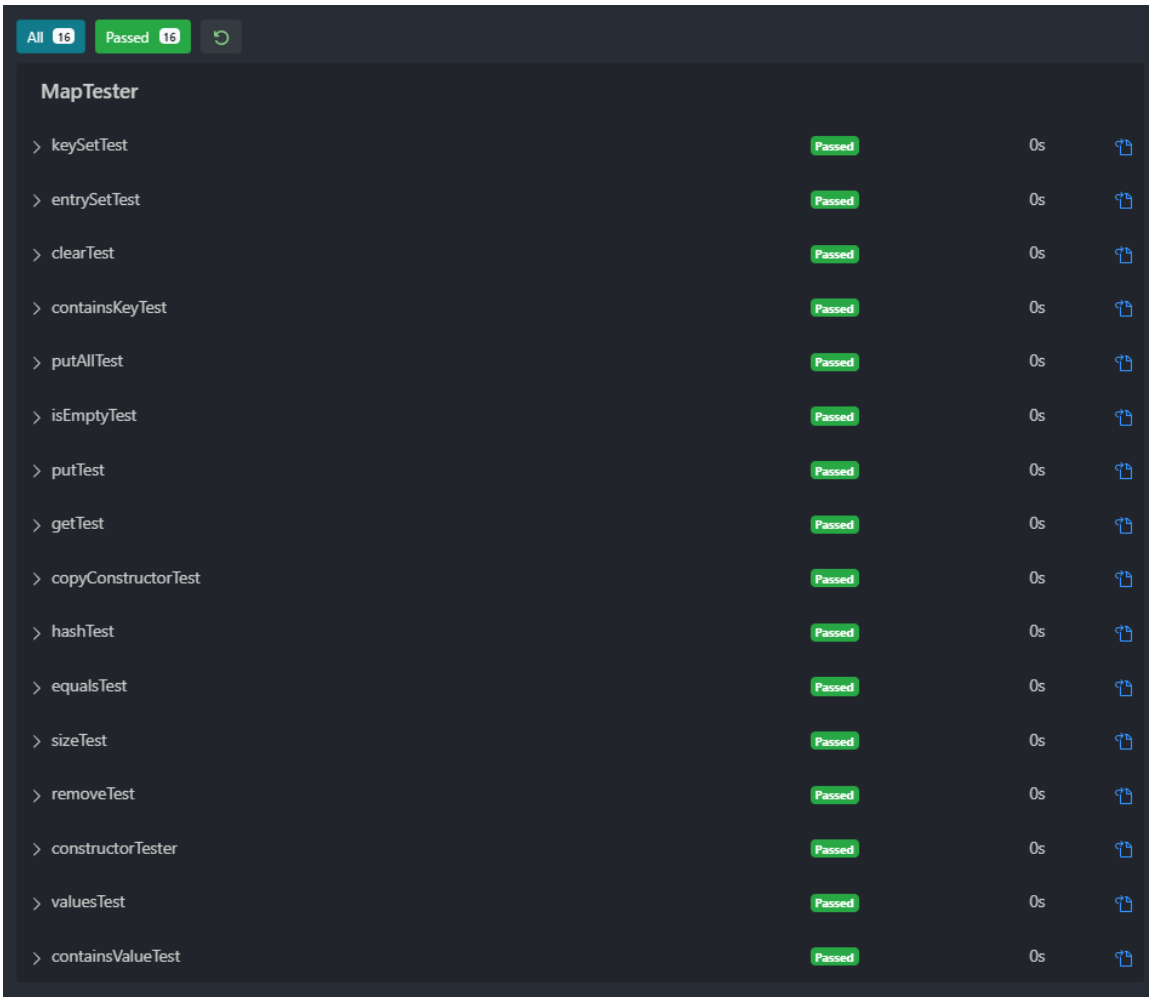
▼ Post-Condition

La classe MapAdapter funziona correttamente

▼ Test Cases

See below

▼ Test Suite Execution Records



▼ Execution Variables

None

constructorTester

▼ Summary

Test del costruttore della Map

▼ Test Case Design

Si testa che il costruttore di default e il costruttore con parametro di capacità iniziale non ritornino oggetti vuoti e che la mappa creata risulti

vuota. Si verifica inoltre che due mappe vuote appena create siano identiche

▼ Test Description

Creo una mappa con il costruttore di default e verifico che questa non corrisponda a null
Creo una mappa con il costruttore con un parametro e verifico che questa non corrisponda a null

Verifico che le due mappe create siano vuote e che siano uguali

▼ Pre-Condition

None

▼ Post-Condition

None

▼ Expected Results

Il test passa con successo senza lanciare eccezioni o contraddire un assert

copyConstructorTest

▼ Summary

Test del costruttore di copia della Map

▼ Test Case Design

Si testa che il costruttore di copia della mappa lanci correttamente eccezioni in caso venga passata una mappa null come valore e che non venga moficato l'oggetto a cui si effettua l'assegnamento. Si crea poi una mappa con dei valori e si verifica che una seconda mappa inizializzata con costruttore di copia sia identica alla prima

▼ Test Description

Creo una mappa e la riempio con dei valori generici
Testo che una mappa inizializzata con null lanci NullPointerException e poi testo che pur lanciando l'eccezione il valore precedentemente assegnato non sia variato
Creo una mappa con il costruttore di copia dalla prima mappa creata
Verifico che questa mappa e la prima siano identiche

▼ Pre-Condition

None

▼ Post-Condition

None

▼ Expected Results

Il test passa con successo senza lanciare eccezioni o contraddire un assert

clearTest

▼ Summary

Test del metodo clear della Map

▼ Test Case Design

Si riempie una mappa di valori e si testa che la dimensione sia diversa da 0 poi si utilizza il metodo clear e si verifica che la nuova dimensione sia 0

▼ Test Description

Creo una mappa e inserisco dei valori generici, verifico che la dimensione dopo gli inserimenti sia corretta
Pulisco la mappa e verifico che la nuova dimensione sia 0

▼ Pre-Condition

None

▼ Post-Condition

None

▼ Expected Results

Il test passa con successo senza lanciare eccezioni o contraddire un assert

containsKeyTest

▼ Summary

Test del metodo containsKey della Map

▼ Test Case Design

Si testa che chiavi inserite nella mappa siano effettivamente presenti e che chiavi non inserite non vi siano. Si verifica anche che dopo un clear non vengano più trovate le chiavi precedentemente inserite

▼ Test Description

Si crea una mappa e vi si inseriscono delle chiavi generiche
Utilizzando il metodo containsKey si verifica che le chiavi inserite siano trovate e poi si testa che non vi siano chiavi di altro tipo all'interno o che valori vengano scambiati per chiavi
Si effettua un clear sulla mappe e si verifica che non vi sia più contenta alcuna chiave
Si effettua un insert sulla mappa verificando che la chiave inserita sia di nuovo presente

▼ Pre-Condition

None

▼ Post-Condition

None

▼ Expected Results

Il test passa con successo senza lanciare eccezioni o contraddire un assert

containsValueTest

▼ Summary

Test del metodo containsValue della Map

▼ Test Case Design

Si testa che valori inseriti nella mappa siano effettivamente presenti e che valori non inseriti non vi siano. Si verifica anche che dopo un clear non vengano più trovati i valori precedentemente inseriti

▼ Test Description

Si crea una mappa e vi si inseriscono dei valori generici
Utilizzando il metodo containsValue si verifica che i valori inseriti siano trovati e poi si testa che non vi siano valori di altro tipo all'interno o che chiavi vengano scambiate per valori
Si effettua un clear sulla mappe e si verifica che non vi sia più contento alcun valore
Si effettua un insert sulla mappa verificando che il valore inserito sia di nuovo presente

▼ Pre-Condition

None

▼ Post-Condition

None

▼ Expected Results

Il test passa con successo senza lanciare eccezioni o contraddire un assert

entrySetTest

▼ Summary

Test del metodo entrySet della Map

▼ Test Case Design

Si testa che il metodo restituisca un oggetto non nullo. Ulteriori test verranno effettuati su una test suite a parte

▼ Test Description

Si crea una mappa e si verifica che il metodo entrySet non restituisca un oggetto null

▼ Pre-Condition

None

▼ Post-Condition

None

▼ Expected Results

Il test passa con successo senza lanciare eccezioni o contraddire un assert

equalsTest

▼ Summary

Test del metodo equals della Map

▼ Test Case Design

Si testa che due mappe vuote e due mappe contenenti le stesse entry siano ritenute uguali
Si verifica inoltre che mappe diverse o mappe e altri tipi di oggetto vengano ritenuti diversi

▼ Test Description

Si creano due mappe e si verifica che queste siano uguali
Si inseriscono gli stessi valori all'interno di entrambe le mappe e si verifica che queste siano rimaste uguali
Si inserisce un valore all'interno di una sola delle due mappe e si verifica che queste ora siano diverse
Si verifica che un oggetto generico sia diverso da una mappa contenente valori

▼ Pre-Condition

None

▼ Post-Condition

None

▼ Expected Results

Il test passa con successo senza lanciare eccezioni o contraddire un assert

getTest

▼ Summary

Test del metodo get della Map

▼ Test Case Design

Si testa che valori inseriti nella mappa siano effettivamente presenti e che valori non inseriti non vi siano. Si verifica anche che dopo un clear non vengano più trovati i valori precedentemente inseriti. Si verifica che passando chiave nulla il metodo lanci NullPointerException

▼ Test Description

Si crea una mappa e si verifica che lanci NullPointerException se chiamata passando chiave nulla
Si verifica che il metodo get ritorni un valore null se la chiave cercata non è presente
Si inseriscono delle entry e si verifica che i corrispettivi valore e chiave siano correttamente presenti
Si effettua clear sulla mappa e si verifica che non siano più presenti entry inserite in precedenza

▼ Pre-Condition

None

▼ Post-Condition

None

▼ Expected Results

Il test passa con successo senza lanciare eccezioni o contraddire un assert

hashTest

▼ Summary

Test del metodo hashCode della Map

▼ Test Case Design

Si testa che gli hash code ritornati dalla mappa con valori diversi siano quelli attesi svolgendo i calcoli a mano

▼ Test Description

Si crea una mappa e si verifica che l'hash code di una mappa vuota sia 0
Si inseriscono entry di diverso tipo e si verifica che l'hash code sia corrispondente a quello atteso

▼ Pre-Condition

None

▼ Post-Condition

None

▼ Expected Results

Il test passa con successo senza lanciare eccezioni o contraddire un assert

isEmptyTest

▼ Summary

Test del metodo isEmpty della Map

▼ Test Case Design

Si testa che su una mappa vuota il metodo torni true e che su una mappa con degli elementi torni false

▼ Test Description

Si crea una mappa e si verifica che isEmpty resituisca true
Si aggiunge una entry alla mappa e si verifica che isEmpty ritorni false

Si effettua un clear e si verifica che isEmpty torni di nuovo true

▼ Pre-Condition

None

▼ Post-Condition

None

▼ Expected Results

Il test passa con successo senza lanciare eccezioni o contraddire un assert

keySetTest

▼ Summary

Test del metodo keySet della Map

▼ Test Case Design

Si testa che il metodo restituisca un oggetto non nullo. Ulteriori test verranno effettuati su una test suite a parte

▼ Test Description

Si crea una mappa e si verifica che il metodo keySet non restituisca un oggetto null

▼ Pre-Condition

None

▼ Post-Condition

None

▼ Expected Results

Il test passa con successo senza lanciare eccezioni o contraddire un assert

putTest

▼ Summary

Test del metodo put della Map

▼ Test Case Design

Si testa che diversi tentativi di inserimento di tipo diverso riportino correttamente all'interno della mappa il valore e la chiave passati come

parametro. Si verifica inoltre che l'inserimento di chiavi o valori nulli lancino correttamente NullPointerException

▼ Test Description

Si crea una mappa e vi si inserisce una coppia chiave valore generica, si verifica poi che la dimensione sia quella attesa, che contenga la chiave e il valore inseriti
Si inserisce una seconda coppia e si ripetono i test
Si inserisce una terza coppia con argomenti di tipo diverso e si verifica che questi vengano correttamente memorizzati
Si chiama put con argomenti nulli e si verifica che non vi sia stato un inserimento di questi e che abbiano correttamente lanciato NullPointerException

▼ Pre-Condition

None

▼ Post-Condition

None

▼ Expected Results

Il test passa con successo senza lanciare eccezioni o contraddire un assert

putAllTest

▼ Summary

Test del metodo putAll della Map

▼ Test Case Design

Si testa che l'inserimento tramite putAll di elementi già presenti non modifichi la mappa, che vengano aggiunti correttamente gli elementi non già presenti e che i valori vengano aggiornati se necessario. Si verifica che venga lanciato NullPointerException se viene passata una mappa nulla

▼ Test Description

Si crea una mappa e si verifica che addAll di null lanci NullPointerException
Si crea una seconda mappa e assieme alla prima vi si inserisce una entry (uguale per entrambe)
Si usa putAll di una mappa sull'altra e si verifica che nulla sia cambiato
Si pulisce la prima mappa e si verifica che l'inserimento di una mappa vuota

non apporti modifiche
Si inseriscono elementi coppie chiave valore diverse nella prima mappa e una coppia con chiave già presente nella seconda ma valore diverso
Si usa putAll della prima mappa nella seconda e si verifica che siano avvenute le corrette modifiche

▼ Pre-Condition

None

▼ Post-Condition

None

▼ Expected Results

Il test passa con successo senza lanciare eccezioni o contraddire un assert

removeTest

▼ Summary

Test del metodo remove della Map

▼ Test Case Design

Si verifica che elementi inseriti vengano correttamente rimossi dalla mappa e che remove ritorni l'elemento eliminato corretto. Si verifica inoltre che viene lanciata NullPointerException alla richiesta di eliminare una chiave null

▼ Test Description

Si crea una mappa e si verifica che la rimozione di una chiave null lanci NullPointerException
Si inserisce un valore nella mappa e si verifica che remove ritorni il valore corretto dalla rimozione della chiave
Si verifica che il remove abbia correttamente modificato la dimensione della mappa
Si verifica che se viene richiesta la rimozione di chiavi non presenti remove ritorna null come valore

▼ Pre-Condition

None

▼ Post-Condition

None

▼ Expected Results

Il test passa con successo senza lanciare eccezioni o contraddire un assert

sizeTest

▼ Summary

Test del metodo size della Map

▼ Test Case Design

Si testa che il metodo restituisca la corretta dimensione della mappa dopo alcuni inserimenti e rimozioni

▼ Test Description

Si crea una mappa e si verifica che la dimensione sia zero
Si inseriscono degli elementi e si verifica che la dimensione aumenti correttamente ad ogni inserimento
Si rimuovono degli elementi e si verifica che la dimensione diminuisca correttamente ad ogni rimozione

▼ Pre-Condition

None

▼ Post-Condition

None

▼ Expected Results

Il test passa con successo senza lanciare eccezioni o contraddire un assert

valuesTest

▼ Summary

Test del metodo values della Map

▼ Test Case Design

Si testa che il metodo restituisca un oggetto non nullo. Ulteriori test verranno effettuati su una test suite a parte

▼ Test Description

Si crea una mappa e si verifica che il metodo values non restituisca un oggetto null

▼ Pre-Condition

None

▼ Post-Condition

None

▼ Expected Results

Il test passa con successo senza lanciare eccezioni o contraddire un assert

EntryTester

▼ Summary

Questo test mira a verificare il corretto funzionamento delle entry all'interno della mappa, partendo dal test del costruttore e del metodo equals si va verificare che ogni metodo getter ritorni i valori corretti inseriti all'interno della Entry e che il setter per i valori abbia un effetto effettivo sulla mappa sottostante e non apporti modifiche esclusivamente ai valori interni alla entry

▼ Test Suite Design

Il test è disegnato in maniera tale da testare casi generali di creazione, confronto, ottenimento e sostituzione dei valori interni alla Entry utilizzando una mappa (necessaria alla creazione delle entry) e gli assert per verificare la corretta (o non corretta) esecuzione dei metodi. Ogni metodo di test è autonomo e non necessita di variabili private interne alla classe di test.

▼ Pre-Condition

Deve esistere una classe EntryAdapter e una classe MapAdapter implementate secondo l'interfaccia di HMap

▼ Post-Condition

La classe EntryAdapter funziona correttamente

▼ Test Cases

See below

▼ Test Suite Execution Records

▼ Execution Variables

None

constructorTester

▼ Summary

Test del costruttore delle Map Entry

▼ Test Case Design

Si testa con chiavi nulle e valori nulli se il costruttore ritorna un oggetto costruito e non genera errori e poi si testa con oggetti generici per verificare la stessa cosa

▼ Test Description

Creo un oggetto Entry con chiave e valore nulli, verifico che l'oggetto creato non sia null
Creo un oggetto Entry con chiave e valori generici, verifico che l'oggetto creato non sia null

▼ Pre-Condition

None

▼ Post-Condition

None

▼ Expected Results

Il test passa con successo senza lanciare eccezioni o contraddire un assert

equalsTester

▼ Summary

Test del metodo equals delle Map Entry

▼ Test Case Design

Si testa che Entry contenenti chiavi e valori diversi non risultino uguali e che chiavi uguali invece siano correttamente riportate come uguali. Si verifica inoltre che una chiave è identica a sé stessa

▼ Test Description

Creo due entry con chiavi e valori diversi e teso che siano diverse fra loro e che siano identiche fra sé stesse
Modifico una entry per renderla uguale all'altra e testo che vengano ritenute uguali
Testo che un valore null non venga ritenuto uguale ad una entry

▼ Pre-Condition

None

▼ Post-Condition

None

▼ Expected Results

Il test passa con successo senza lanciare eccezioni o contraddire un assert

getKeyTester

▼ Summary

Test del metodo getKeys delle Map Entry

▼ Test Case Design

Si testano Entry contenenti chiavi e valori diversi per verificare che le chiavi ritornate corrispondano a quelle inserite

▼ Test Description

Creo due entry con chiavi e valori diversi sia null che generici e verifico una ad una che la chiave ritornata dal metodo corrisponda a quella inserita in fase di creazione

▼ Pre-Condition

None

▼ Post-Condition

None

▼ Expected Results

Il test passa con successo senza lanciare eccezioni o contraddire un assert

getValueTester

▼ Summary

Test del metodo getValues delle Map Entry

▼ Test Case Design

Si testano Entry contenenti chiavi e valori diversi per verificare che i valori ritornati corrispondano a quelli inseriti

▼ Test Description

Creo due entry con chiavi e valori diversi sia null che generici e verifico una ad una che i valori ritornati dal metodo corrispondano a quelli inseriti in fase di creazione

▼ Pre-Condition

None

▼ Post-Condition

None

▼ Expected Results

Il test passa con successo senza lanciare eccezioni o contraddire un assert

setValueTester

▼ Summary

Test del metodo setValue delle Map Entry

▼ Test Case Design

Si testa che Entry contenenti chiavi e valori diversi vengano effettivamente modificati apportando le corrette modifiche alla mappa d'origine e si verifica che vengano lanciate le eccezioni in caso di valori non accettabili

▼ Test Description

Creo una mappa e vi inserisco una coppia chiave valore, creo una nuova entry da quella mappa con gli stessi valori e effettuo il set di un valore

diverso nella entry
Verifico che la modifica sia stata apportata sia nella entry che nella mappa in presenza della chiave corrispondente

Testo che setValue lanci correttamente NullPointerException quando viene passato un valore null

▼ Pre-Condition

None

▼ Post-Condition

None

▼ Expected Results

Il test passa con successo senza lanciare eccezioni o contraddire un assert

EntrySetTester

▼ Summary

Questo test mira a verificare il corretto funzionamento della classe EntrySet, partendo dal test del costruttore si va verificare che ogni metodo ritorni i valori corretti inseriti all'interno del set e che i remove per le entry abbino un effetto effettivo sulla mappa sottostante e non apporti modifiche esclusivamente ai valori interni al set

▼ Test Suite Design

Il test è disegnato in maniera tale da testare casi generali di creazione di set da mappe, effettuare rimozioni, confronti e ottenimento dei valori interni al set e quindi alla mappa. Ogni metodo di test è autonomo e non necessita di variabili private interne alla classe di test.

▼ Pre-Condition

Deve esistere una classe EntryAdapter, una classe MapAdapter e una classe EntrySet implementate secondo l'interfaccia di HMap e HSet rispettivamente. Deve essere inoltre presente java.util.Hashtable

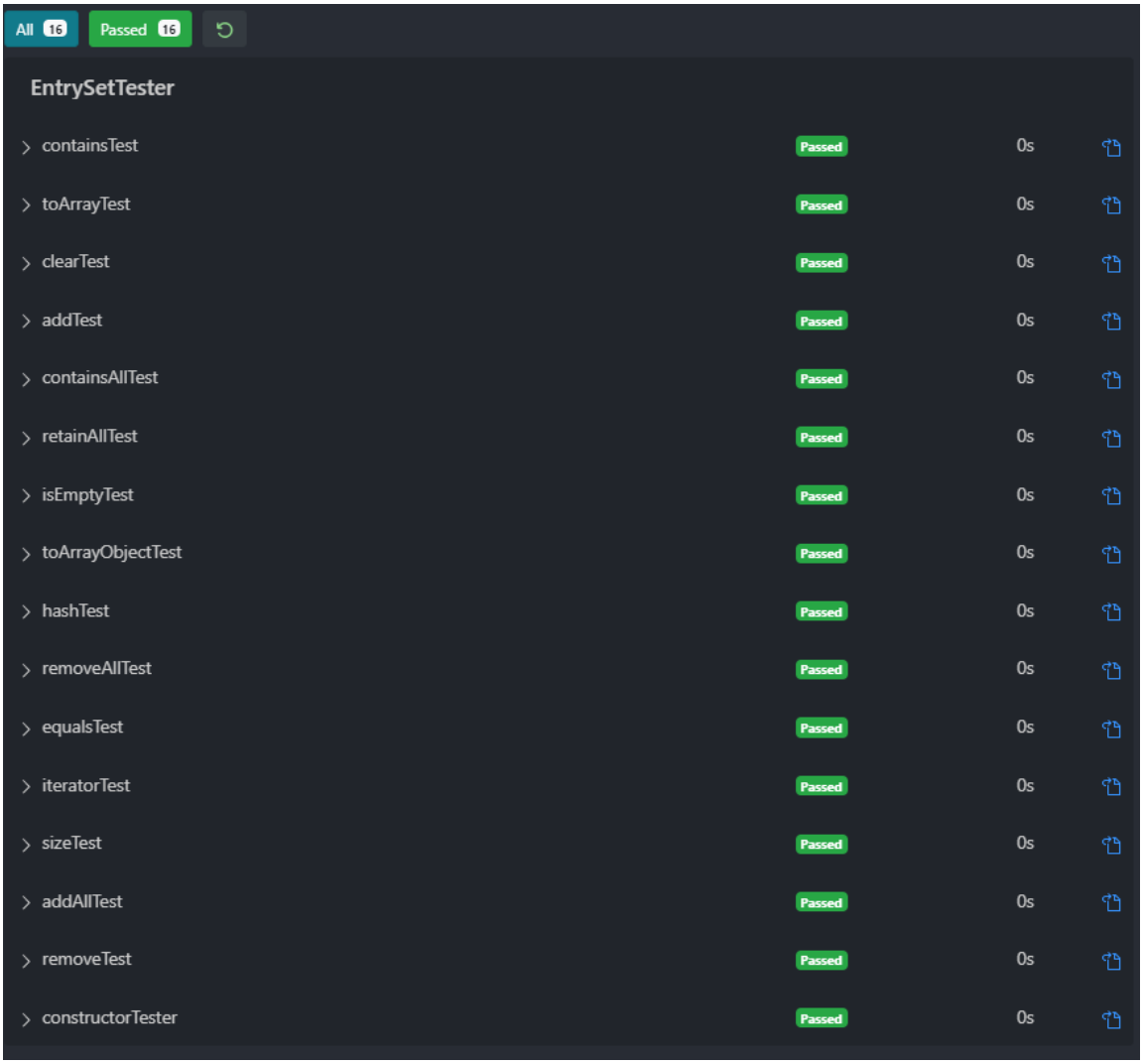
▼ Post-Condition

La classe EntryAdapter funziona correttamente

▼ Test Cases

See below

▼ Test Suite Execution Records



▼ Execution Variables

None

constructorTester

▼ Summary

Test del costruttore del EntrySet

▼ Test Case Design

Si testa che l'oggetto ritornato dal costruttore non sia nullo e che lanci NullPointerException se viene passato un valore null

▼ Test Description

Creo una Hashtable e verifico che l'entrySet derivato dalla mappa passando l'hashtable non sia null
Verifico che viene lanciata NullPointerException quando creao un EntrySet passando valore null

▼ Pre-Condition

None

▼ Post-Condition

None

▼ Expected Results

Il test passa con successo senza lanciare eccezioni o contraddire un assert

addTest

▼ Summary

Test del metodo add dell' EntrySet

▼ Test Case Design

Si testa che add lanci sempre l'eccezione UnsupportedOperationException

▼ Test Description

Creo una Hashtable e la uso per creare un entrySet
Verifico che venga lanciato UnsupportedOperationException quando si usa addAll

▼ Pre-Condition

None

▼ Post-Condition

None

▼ Expected Results

Il test passa con successo senza lanciare eccezioni o contraddire un assert

addAllTest

▼ Summary

Test del metodo addAll dell' EntrySet

▼ Test Case Design

Si testa che addAll lanci sempre l'eccezione UnsupportedOperationException

▼ Test Description

Creo una Hashtable e la uso per creare un entrySet
Verifico che venga lanciato UnsupportedOperationException quando si usa addAll

▼ Pre-Condition

None

▼ Post-Condition

None

▼ Expected Results

Il test passa con successo senza lanciare eccezioni o contraddire un assert

clearTest

▼ Summary

Test del metodo clear dell' EntrySet

▼ Test Case Design

Si riempie una hashtable di valori e si testa che il metodo sul set ottenuto da questa faccia in modo che la dimensione della hashtable torni a 0 così come quella del set

▼ Test Description

Creo una hashtable, inserisco dei valori generici e ne creo un EntrySet
Verifico che la dimensione dopo clear diventi 0 per l'entrySet e per la hashtable relativa

▼ Pre-Condition

None

▼ Post-Condition

None

▼ Expected Results

Il test passa con successo senza lanciare eccezioni o contraddire un assert

containsTest

▼ Summary

Test del metodo contains dell'EntrySet

▼ Test Case Design

Si riempie una hashtable di valori e si testa che il metodo sul set ottenuto da questa riferisca correttamente la presenza o meno degli elementi inseriti. Si testa inoltre che non riferisca il contenimento di elementi diversi da entry

▼ Test Description

Creo una hashtable, inserisco dei valori generici e ne creo un EntrySet
Verifico che contains torni vero per entry uguali a quelle inserite
Verifico che contains torni falso per elementi che non sono entry
Verifico che contains torni falso per entry diverse da quelle inserite

▼ Pre-Condition

None

▼ Post-Condition

None

▼ Expected Results

Il test passa con successo senza lanciare eccezioni o contraddire un assert

containsAllTest

▼ Summary

Test del metodo containsAll dell'EntrySet

▼ Test Case Design

Si riempie una hashtable di valori e si testa che il metodo sul set ottenuto da questa riferisca correttamente la presenza o meno di liste contenenti entry inserite. Si testa inoltre che non riferisca il contenimento di elementi diversi da Collection. Verifico inoltre che venga lanciato NullPointerException se viene passata una collection nulla

▼ Test Description

Creo una hashtable, inserisco dei valori generici e ne creo un EntrySet
Creo una lista e inserisco gli stessi elementi
Verifico che containsAll torni vero per la lista contenente le stesse entry
Modifico la lista con una entry diversa
Verifico che containAll torni falso per la lista contenente entry diverse
Verifico che containAll torni vero per una lista vuota
Verifico che containAll lanci NullPointerException passando null come parametro

▼ Pre-Condition

None

▼ Post-Condition

None

▼ Expected Results

Il test passa con successo senza lanciare eccezioni o contraddire un assert

equalsTest

▼ Summary

Test del metodo equals dell'EntrySet

▼ Test Case Design

Si riempiono due hashtable di valori e si testa che il metodo sui set ottenuti da questa riferisca correttamente l'uguaglianza o meno degli entrySet. Si testa inoltre che set diversi siano ritenuti diversi.

▼ Test Description

Creo due hashtable, inserisco dei valori generici e ne creo i rispettivi EntrySet
Verifico che equals torni vero confrontando i due entry set
Modifico una hashtable con valori diversi
Verifico che equals torni falso confrontando i due entry set
Pulisco le due hashtable e verifico che risultino uguali
Inserisco più valori uguali sulle due hashtable e verifico che risultino uguali gli entrySet

▼ Pre-Condition

None

▼ Post-Condition

None

▼ Expected Results

Il test passa con successo senza lanciare eccezioni o contraddire un assert

hashTest

▼ Summary

Test del metodo hashCode dell'EntrySet

▼ Test Case Design

Si verifica che un entry set da una hashTable vuota e con valori ritorni il numero corretto dell'hashcode

▼ Test Description

Creo una hashtable ne creo un EntrySet
Utilizzo hashcode sul set e verifico che l'hash code torni 0
Inserisco degli elementi e verifico che i set tornino il valore corretto di hashcode

▼ Pre-Condition

None

▼ Post-Condition

None

▼ Expected Results

Il test passa con successo senza lanciare eccezioni o contraddire un assert

isEmptyTest

▼ Summary

Test del metodo isEmpty dell'EntrySet

▼ Test Case Design

Si verifica che un entry set da una hashTable vuota e con valori ritorni correttamente vero o falso

▼ Test Description

Creo una hashtable ne creo un EntrySet
Utilizzo isEmpty sul set e verifico che isEmpty sia vero
Inserisco degli elementi e verifico che isEmpty sia falso

▼ Pre-Condition

None

▼ Post-Condition

None

▼ Expected Results

Il test passa con successo senza lanciare eccezioni o contraddire un assert

iteratorTest

▼ Summary

Test del metodo iterator dell'EntrySet

▼ Test Case Design

Si verifica che la creazione di un iteratore a entry set da una hashTable non ritorni un valore null

▼ Test Description

Creo una hashtable ne creo un EntrySet
Utilizzo iterator sul set e verifico che il valore di ritorno non sia null

▼ Pre-Condition

None

▼ Post-Condition

None

▼ Expected Results

Il test passa con successo senza lanciare eccezioni o contraddire un assert

removeTest

▼ Summary

Test del metodo remove dell'EntrySet

▼ Test Case Design

Si verifica che un entry set da una hashTable vuota non possa rimuovere oggetti e che rimuova o meno correttamente gli elementi da un entry set e dalla rispettiva hashtable non vuota. Si verifica inoltre che la rimozione di chiavi nulle lanci NullPointerException

▼ Test Description

Creo una hashtable ne creo un EntrySet
Testo che la rimozione di null dal set lanci NullPointerException
Testo che non vengano effettuate rimozioni se il set è vuoto
Inserisco degli elementi e verifico che avvenga correttamente la rimozione della entry dal set e dalla hashtable relativa
Tento la rimozione della stessa entry per verificare che non si possa eseguire nuovamente
Reinserisco la entry eliminata e provo a rimuovere enrty con chiave corretta e valore diverso e con chiave diversa e valore corretto

▼ Pre-Condition

None

▼ Post-Condition

None

▼ Expected Results

Il test passa con successo senza lanciare eccezioni o contraddire un assert

removeAllTest

▼ Summary

Test del metodo removeAll dell'EntrySet

▼ Test Case Design

Si verifica che un entry set da una hashTable vuota non possa rimuovere collection di oggetti e che rimuova o meno correttamente gli elementi da un entry set e dalla rispettiva hashtable non vuota. Si verifica inoltre che il passaggio di una collection nulla lanci NullPointerException

▼ Test Description

Creo una hashtable ne creo un EntrySet
Testo che removeAll di null dal set lanci NullPointerException
Inserisco degli elementi nell'entry set

Testo che non vengano effettuate rimozioni passando una lista vuota
Pulisco la hashtable e inserisco una entry nella collection
Verifico che non avvengano modifiche all'entry set e alla hashtable vuote effettuando removeAll di una collection piena
Aggiungo la stessa entry della collection alla hashtable e verifico che removeAll della collection svuoti la hashtable e l'entrySet

▼ Pre-Condition

None

▼ Post-Condition

None

▼ Expected Results

Il test passa con successo senza lanciare eccezioni o contraddire un assert

retainAllTest

▼ Summary

Test del metodo retainAll dell'EntrySet

▼ Test Case Design

Si verifica che retainAll contenente le stesse entry del set relativo alla hashtable non apporti modifiche al set e alla hashtable, che una lista vuota o con meno elementi della hashtable effettui correttamente le eliminazioni e che una lista contenente più elementi del set apporti le corrette modifiche.
Si verifica inoltre che il passaggio di una collection nulla lanci NullPointerException

▼ Test Description

Creo una hashtable ne creo un EntrySet
Testo che retainAll di null dal set lanci NullPointerException
Inserisco degli elementi nell'entry set
Testo che venga svuotato il set e la hashtable passando una lista vuota
Pulisco la hashtable e inserisco una entry nella collection
Verifico che non avvengano modifiche all'entry set e alla hashtable vuote effettuando retainAll di una collection piena
Aggiungo più entry della collection alla hashtable e verifico che retainAll della collection mantenga solo le entry contenute in essa

▼ Pre-Condition

None

▼ Post-Condition

None

▼ Expected Results

Il test passa con successo senza lanciare eccezioni o contraddire un assert

sizeTest

▼ Summary

Test del metodo size dell'EntrySet

▼ Test Case Design

Si verifica che un entry set da una hashTable vuota abbia dimensione 0 e che aggiungendo una entry alla hashtable la dimensione del set aumenti

▼ Test Description

Creo una hashtable ne creo un EntrySet
Utilizzo size sul set e verifico che torni 0
Inserisco degli elementi e verifico che i set tornino il valore corretto di dimensione

▼ Pre-Condition

None

▼ Post-Condition

None

▼ Expected Results

Il test passa con successo senza lanciare eccezioni o contraddire un assert

toArrayTest

▼ Summary

Test del metodo toArray dell'EntrySet

▼ Test Case Design

Si verifica che un entry set da una hashTable vuota ritorni un array di dimensione 0 e che aggiungendo elementi venga tornato un array identico a quello atteso

▼ Test Description

Creo una hashtable ne creo un EntrySet
Utilizzo toArray sul set e verifico che torni un array di object di dimensione 0
Inserisco degli elementi nella hashtable e verifico che toArray torni un array contenente le stesse entry inserite e di dimensione minima

▼ Pre-Condition

None

▼ Post-Condition

None

▼ Expected Results

Il test passa con successo senza lanciare eccezioni o contraddire un assert

toArrayTest

▼ Summary

Test del metodo toArray dell'EntrySet

▼ Test Case Design

Si verifica che un entry set da una hashTable vuota ritorni un array di dimensione 0 passando un array vuoto e che aggiungendo elementi alla hashtable e passando un array di dimensione maggiore torni l'array parzialmente pieno mentre tornandone uno di dimensione inferiore o corretta

▼ Test Description

Creo una hashtable e ne creo un EntrySet
Utilizzo toArray con array vuoto sul set e verifico che torni un array di object di dimensione 0
Inserisco degli elementi nella hashtable e verifico che toArray passando un array vuoto torni un array contenente le stesse entry inserite e di dimensione minima
Verifico che toArray passando un array di dimensione maggiore di quella richiesta ritorni un array parzialmente riempito
Verifico che toArray passando un array di dimensione corretta ritorni un array riempito della stessa dimensione

▼ Pre-Condition

None

▼ Post-Condition

None

▼ Expected Results

Il test passa con successo senza lanciare eccezioni o contraddire un assert

KeySetTester

▼ Summary

Questo test mira a verificare il corretto funzionamento della classe KeySet, partendo dal test del costruttore si va verificare che ogni metodo ritorni i valori corretti inseriti all'interno del set e che i remove per le chiavi abbino un effetto effettivo sulla mappa sottostante e non apporti modifiche esclusivamente ai valori interni al set

▼ Test Suite Design

Il test è disegnato in maniera tale da testare casi generali di creazione di set da mappe, effettuare rimozioni, confronti e ottenimento dei valori interni al set e quindi alla mappa. Ogni metodo di test è autonomo e non necessita di variabili private interne alla classe di test.

▼ Pre-Condition

Deve esistere una classe MapAdapter e una classe KeySet implementate secondo l'interfaccia di HMap e HSet rispettivamente. Deve essere inoltre presente java.util.Hashtable

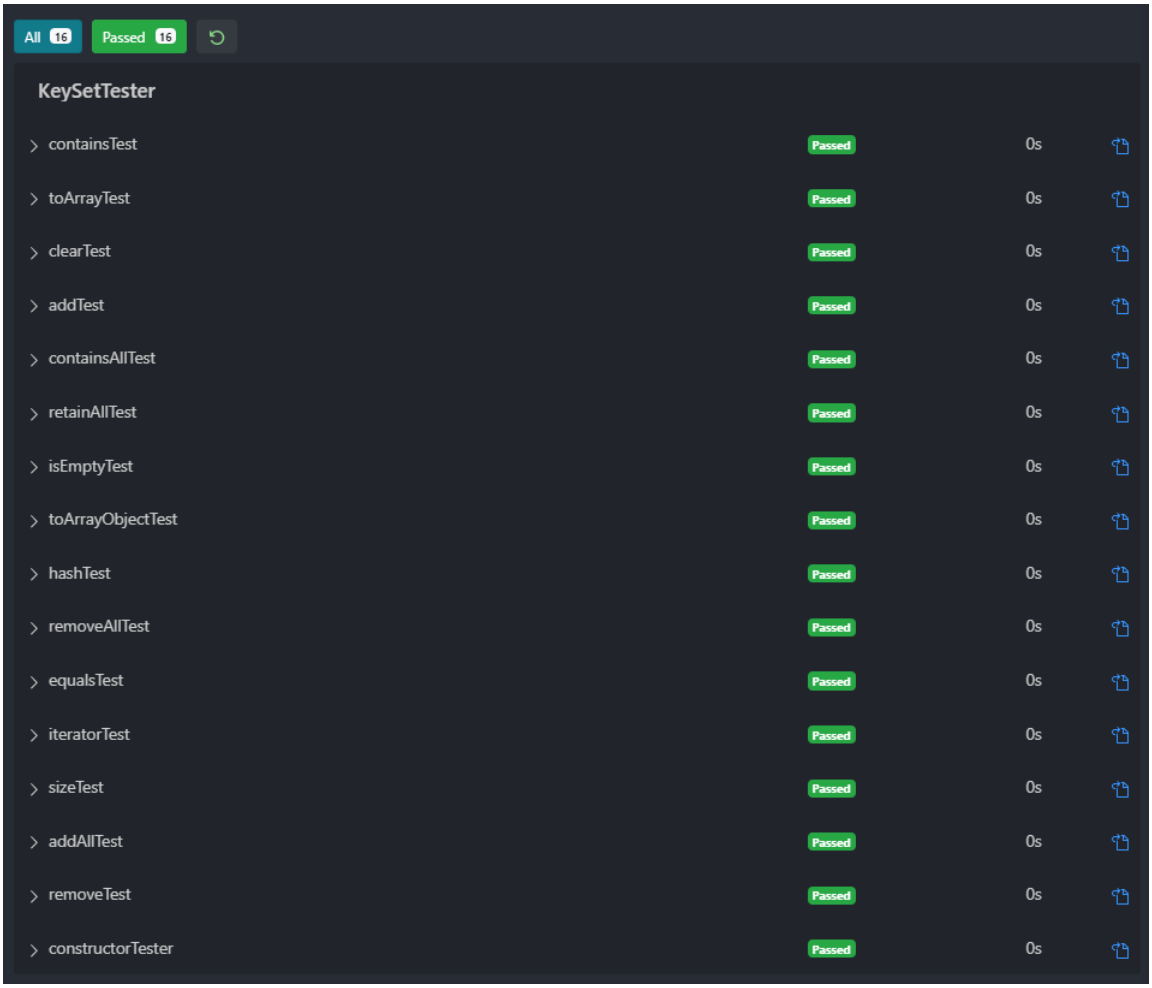
▼ Post-Condition

La classe KeySet funziona correttamente

▼ Test Cases

See below

▼ Test Suite Execution Records



▼ Execution Variables

None

constructorTester

▼ Summary

Test del costruttore del KeySet

▼ Test Case Design

Si testa che l'oggetto ritornato dal costruttore non sia nullo e che lanci NullPointerException se viene passato un valore null

▼ Test Description

Creo una Hashtable e verifico che il KeySet derivato dalla mappa passando l'hashtable non sia null
Verifico che viene lanciata NullPointerException quando creao un KeySet passando valore null

▼ Pre-Condition

None

▼ Post-Condition

None

▼ Expected Results

Il test passa con successo senza lanciare eccezioni o contraddire un assert

addTest

▼ Summary

Test del metodo add dell' KeySet

▼ Test Case Design

Si testa che add lanci sempre l'eccezione UnsupportedOperationException

▼ Test Description

Creo una Hashtable e la uso per creare un KeySet
Verifico che venga lanciato UnsupportedOperationException quando si usa add

▼ Pre-Condition

None

▼ Post-Condition

None

▼ Expected Results

Il test passa con successo senza lanciare eccezioni o contraddire un assert

addAllTest

▼ Summary

Test del metodo addAll dell' KeySet

▼ Test Case Design

Si testa che addAll lanci sempre l'eccezione UnsupportedOperationException

▼ Test Description

Creo una Hashtable e la uso per creare un KeySet
Verifico che venga lanciato UnsupportedOperationException quando si usa addAll

▼ Pre-Condition

None

▼ Post-Condition

None

▼ Expected Results

Il test passa con successo senza lanciare eccezioni o contraddire un assert

clearTest

▼ Summary

Test del metodo clear dell' KeySet

▼ Test Case Design

Si riempie una hashtable di valori e si testa che il metodo sul set ottenuto da questa faccia in modo che la dimensione della hashtable torni a 0 così come quella del set

▼ Test Description

Creo una hashtable, inserisco dei valori generici e ne creo un KeySet
Verifico che la dimensione dopo clear diventi 0 per l'KeySet e per la hashtable relativa

▼ Pre-Condition

None

▼ Post-Condition

None

▼ Expected Results

Il test passa con successo senza lanciare eccezioni o contraddire un assert

containsTest

▼ Summary

Test del metodo contains del KeySet

▼ Test Case Design

Si riempie una hashtable di valori e si testa che il metodo sul set ottenuto da questa riferisca correttamente la presenza o meno degli elementi inseriti. Si testa inoltre che non riferisca il contenimento di elementi diversi da key

▼ Test Description

Creo una hashtable, inserisco dei valori generici e ne creo un KeySet
Verifico che contains torni vero per chiavi uguali a quelle inserite
Verifico che contains torni falso per elementi che non sono key
Verifico che contains torni falso per chiavi diverse da quelle inserite

▼ Pre-Condition

None

▼ Post-Condition

None

▼ Expected Results

Il test passa con successo senza lanciare eccezioni o contraddire un assert

containsAllTest

▼ Summary

Test del metodo containsAll del KeySet

▼ Test Case Design

Si riempie una hashtable di valori e si testa che il metodo sul set ottenuto da questa riferisca correttamente la presenza o meno di liste contenenti chiavi inserite. Si testa inoltre che non riferisca il contenimento di elementi diversi da Collection. Verifico inoltre che venga lanciato NullPointerException se viene passata una collection nulla

▼ Test Description

Creo una hashtable, inserisco dei valori generici e ne creo un KeySet
Creo una lista e inserisco gli stessi elementi
Verifico che containsAll torni vero per la lista contenente le stesse chiavi

Modifico la lista con una chiave diversa
Verifico che containAll torni falso per la lista contenente chiavi diverse
Verifico che containAll torni vero per una lista vuota
Verifico che containAll lanci NullPointerException passando null come parametro

▼ Pre-Condition

None

▼ Post-Condition

None

▼ Expected Results

Il test passa con successo senza lanciare eccezioni o contraddire un assert

equalsTest

▼ Summary

Test del metodo equals del KeySet

▼ Test Case Design

Si riempiono due hashtable di valori e si testa che il metodo sui set ottenuti da questa riferisca correttamente l'uguaglianza o meno dei KeySet. Si testa inoltre che set diversi siano ritenuti diversi.

▼ Test Description

Creo due hashtable, inserisco dei valori generici e ne creo i rispettivi KeySet
Verifico che equals torni vero confrontando i due KeySet
Modifico una hashtable con valori diversi
Verifico che equals torni falso confrontando i due KeySet
Pulisco le due hashtable e verifico che risultino uguali
Inserisco più valori uguali sulle due hashtable e verifico che risultino uguali i KeySet

▼ Pre-Condition

None

▼ Post-Condition

None

▼ Expected Results

Il test passa con successo senza lanciare eccezioni o contraddire un assert

hashTest

▼ Summary

Test del metodo hashCode del KeySet

▼ Test Case Design

Si verifica che un key set da una hashTable vuota e con valori ritorni il numero corretto dell'hashcode

▼ Test Description

Creo una hashtable ne creo un KeySet
Utilizzo hashcode sul set e verifico che l'hash code torni 0
Inserisco degli elementi e verifico che i set tornino il valore corretto di hashcode

▼ Pre-Condition

None

▼ Post-Condition

None

▼ Expected Results

Il test passa con successo senza lanciare eccezioni o contraddire un assert

isEmptyTest

▼ Summary

Test del metodo isEmpty del KeySet

▼ Test Case Design

Si verifica che un key set da una hashTable vuota e con valori ritorni correttamente vero o falso

▼ Test Description

Creo una hashtable ne creo un KeySet
Utilizzo isEmpty sul set e verifico che isEmpty sia vero
Inserisco degli elementi e verifico che isEmpty sia falso

▼ Pre-Condition

None

▼ Post-Condition

None

▼ Expected Results

Il test passa con successo senza lanciare eccezioni o contraddire un assert

iteratorTest

▼ Summary

Test del metodo iterator del KeySet

▼ Test Case Design

Si verifica che la creazione di un iteratore a key set da una hashTable non ritorni un valore null

▼ Test Description

Creo una hashtable ne creo un KeySet
Utilizzo iterator sul set e verifico che il valore di ritorno non sia null

▼ Pre-Condition

None

▼ Post-Condition

None

▼ Expected Results

Il test passa con successo senza lanciare eccezioni o contraddire un assert

removeTest

▼ Summary

Test del metodo remove dell'KeySet

▼ Test Case Design

Si verifica che un key set da una hashTable vuota non possa rimuovere oggetti e che rimuova o meno correttamente gli elementi da un key set e dalla rispettiva hashtable non vuota. Si verifica inoltre che la rimozione di chiavi nulle lanci NullPointerException

▼ Test Description

Creo una hashtable ne creo un KeySet
Testo che la rimozione di null dal set lanci NullPointerException
Testo che non vengano effettuate rimozioni se il set è vuoto
Inserisco degli elementi e verifico che avvenga correttamente la rimozione della chiave dal set e dalla hashtable relativa
Tento la rimozione della stessa chiave per verificare che non si possa eseguire nuovamente

▼ Pre-Condition

None

▼ Post-Condition

None

▼ Expected Results

Il test passa con successo senza lanciare eccezioni o contraddire un assert

removeAllTest

▼ Summary

Test del metodo removeAll del KeySet

▼ Test Case Design

Si verifica che un key set da una hashTable vuota non possa rimuovere collection di oggetti e che rimuova o meno correttamente gli elementi da un key set e dalla rispettiva hashtable non vuota. Si verifica inoltre che il passaggio di una collection nulla lanci NullPointerException

▼ Test Description

Creo una hashtable ne creo un KeySet
Testo che removeAll di null dal set lanci NullPointerException
Inserisco degli elementi nel KeySett
Testo che non vengano effettuate rimozioni passando una lista vuota
Pulisco la hashtable e inserisco una chiave nella collection
Verifico che non avvengano modifiche al key set e alla hashtable vuote effettuando removeAll di una collection piena
Aggiungo la stessa chiave della collection alla hashtable e verifico che removeAll della collection svuoti la hashtable e il KeySet

▼ Pre-Condition

None

▼ Post-Condition

None

▼ Expected Results

Il test passa con successo senza lanciare eccezioni o contraddire un assert

retainAllTest

▼ Summary

Test del metodo retainAll del KeySet

▼ Test Case Design

Si verifica che retainAll contenente le stesse chiavi del set relativo alla hashtable non apporti modifiche al set e alla hashtable, che una lista vuota o con meno elementi della hashtable effettui correttamente le eliminazioni e che una lista contenente più elementi del set apporti le corrette modifiche.
Si verifica inoltre che il passaggio di una collection nulla lanci NullPointerException

▼ Test Description

Creo una hashtable ne creo un KeySet
Testo che retainAll di null dal set lanci NullPointerException
Inserisco degli elementi nel KeySet
Testo che venga svuotato il set e la hashtable passando una lista vuota
Pulisco la hashtable e inserisco una key nella collection
Verifico che non avvengano modifiche al KeySet e alla hashtable vuote effettuando retainAll di una collection piena
Aggiungo più key della collection alla hashtable e verifico che retainAll della collection mantenga solo le key contenute in essa

▼ Pre-Condition

None

▼ Post-Condition

None

▼ Expected Results

Il test passa con successo senza lanciare eccezioni o contraddire un assert

sizeTest

▼ Summary

Test del metodo size del KeySet

▼ Test Case Design

Si verifica che un key set da una hashTable vuota abbia dimensione 0 e che aggiungendo una chiave alla hashtable la dimensione del set aumenti

▼ Test Description

Creo una hashtable ne creo un KeySet
Utilizzo size sul set e verifico che torni 0
Inserisco degli elementi e verifico che i set tornino il valore corretto di dimensione

▼ Pre-Condition

None

▼ Post-Condition

None

▼ Expected Results

Il test passa con successo senza lanciare eccezioni o contraddire un assert

toArrayTest

▼ Summary

Test del metodo toArray del KeySet

▼ Test Case Design

Si verifica che un key set da una hashTable vuota ritorni un array di dimensione 0 e che aggiungendo elementi venga tornato un array identico a quello atteso

▼ Test Description

Creo una hashtable ne creo un KeySet
Utilizzo toArray sul set e verifico che torni un array di object di dimensione 0

Inserisco degli elementi nella hashtable e verifico che toArray torni un array contenente le stesse chiavi inserite e di dimensione minima

▼ Pre-Condition

None

▼ Post-Condition

None

▼ Expected Results

Il test passa con successo senza lanciare eccezioni o contraddire un assert

toArrayTest

▼ Summary

Test del metodo toArray del KeySet

▼ Test Case Design

Si verifica che un key set da una hashTable vuota ritorni un array di dimensione 0 passando un array vuoto e che aggiungendo elementi alla hashtable e passando un array di dimensione maggiore torni l'array parzialmente pieno mentre tornandone uno di dimensione inferiore o corretta

▼ Test Description

Creo una hashtable e ne creo un KeySet
Utilizzo toArray con array vuoto sul set e verifico che torni un array di object di dimensione 0
Inserisco degli elementi nella hashtable e verifico che toArray passando un array vuoto torni un array contenente le stesse chiavi inserite e di dimensione minima
Verifico che toArray passando un array di dimensione maggiore di quella richiesta ritorni un array parzialmente riempito
Verifico che toArray passando un array di dimensione corretta ritorni un array riempito della stessa dimensione

▼ Pre-Condition

None

▼ Post-Condition

None

▼ Expected Results

Il test passa con successo senza lanciare eccezioni o contraddire un assert

ValueCollectionTester

▼ Summary

Questo test mira a verificare il corretto funzionamento della classe ValueCollection, partendo dal test del costruttore si va verificare che ogni metodo ritorni i valori corretti inseriti all'interno della collection e che i remove per le chiavi abbiano un effetto effettivo sulla mappa sottostante e non apporti modifiche esclusivamente ai valori interni alla collection

▼ Test Suite Design

Il test è disegnato in maniera tale da testare casi generali di creazione di collection da mappe, effettuare rimozioni, confronti e ottenimento dei valori interni alla collection e quindi alla mappa. Ogni metodo di test è autonomo e non necessita di variabili private interne alla classe di test.

▼ Pre-Condition

Deve esistere una classe MapAdapter e una classe ValueCollection implementate secondo l'interfaccia di HMap e HCollection rispettivamente. Deve essere inoltre presente java.util.Hashtable

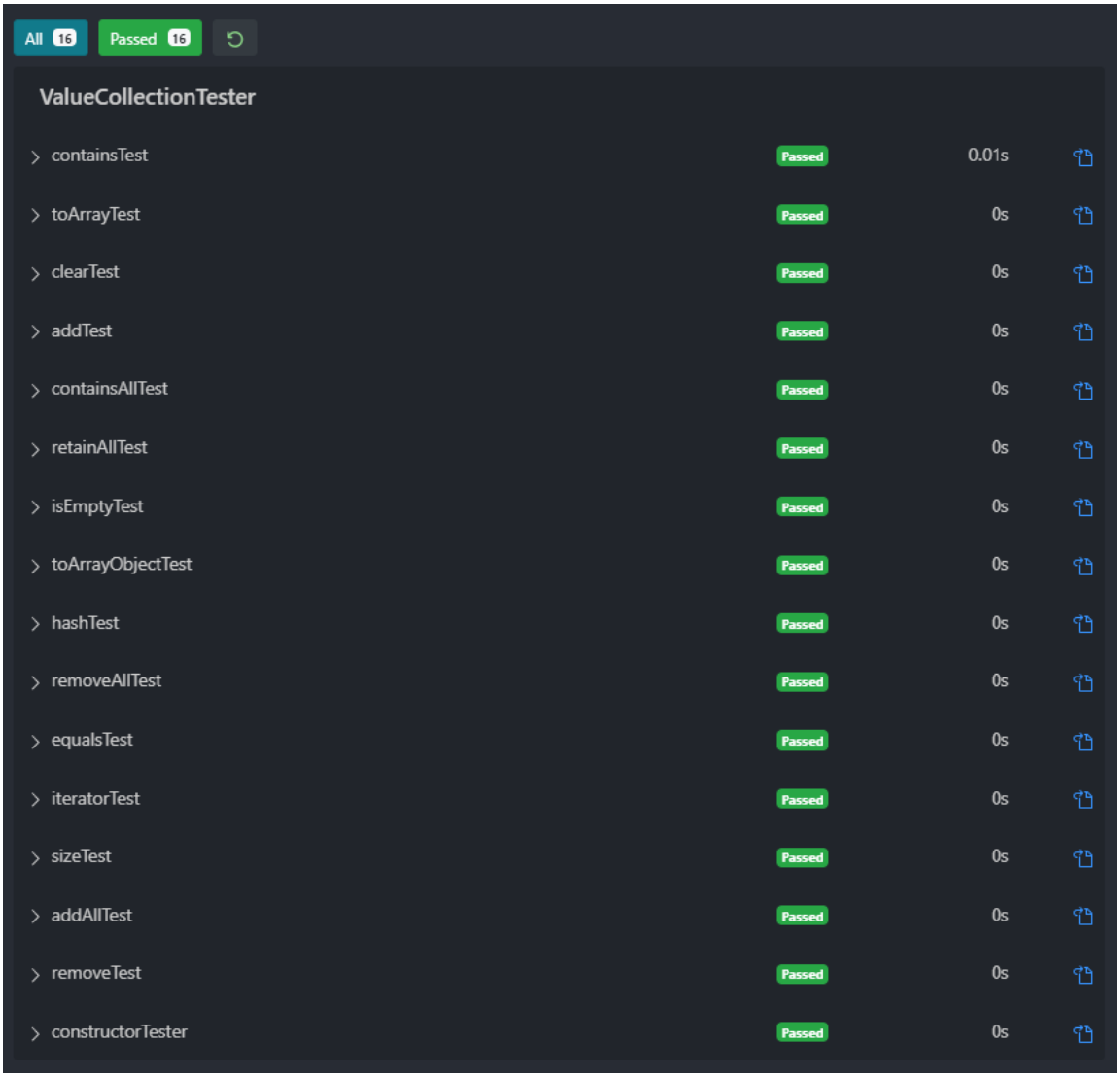
▼ Post-Condition

La classe ValueCollection funziona correttamente

▼ Test Cases

See below

▼ Test Suite Execution Records



▼ Execution Variables

None

constructorTester

▼ Summary

Test del costruttore di ValueCollection

▼ Test Case Design

Si testa che l'oggetto ritornato dal costruttore non sia nullo e che lanci NullPointerException se viene passato un valore null

▼ Test Description

Creo una Hashtable e verifico che la ValueCollection derivata dalla mappa passando l'hashtable non sia null

Verifico che viene lanciata NullPointerException quando creo una ValueCollection passando valore null

▼ Pre-Condition

None

▼ Post-Condition

None

▼ Expected Results

Il test passa con successo senza lanciare eccezioni o contraddire un assert

addTest

▼ Summary

Test del metodo add di ValueCollection

▼ Test Case Design

Si testa che add lanci sempre l'eccezione UnsupportedOperationException

▼ Test Description

Creo una Hashtable e la uso per creare una ValueCollection
Verifico che venga lanciato UnsupportedOperationException quando si usa add

▼ Pre-Condition

None

▼ Post-Condition

None

▼ Expected Results

Il test passa con successo senza lanciare eccezioni o contraddire un assert

addAllTest

▼ Summary

Test del metodo addAll di ValueCollection

▼ Test Case Design

Si testa che addAll lanci sempre l'eccezione UnsupportedOperationException

▼ Test Description

Creo una Hashtable e la uso per creare una ValueCollection
Verifico che venga lanciato UnsupportedOperationException quando si usa addAll

▼ Pre-Condition

None

▼ Post-Condition

None

▼ Expected Results

Il test passa con successo senza lanciare eccezioni o contraddire un assert

clearTest

▼ Summary

Test del metodo clear di ValueCollection

▼ Test Case Design

Si riempie una hashtable di valori e si testa che il metodo sulla collection ottenuta da questa faccia in modo che la dimensione della hashtable torni a 0 così come quella della collection

▼ Test Description

Creo una hashtable, inserisco dei valori generici e ne creo una ValueCollection
Verifico che la dimensione dopo clear diventi 0 per la ValueCollection e per la hashtable relativa

▼ Pre-Condition

None

▼ Post-Condition

None

▼ Expected Results

Il test passa con successo senza lanciare eccezioni o contraddire un assert

containsTest

▼ Summary

Test del metodo contains di ValueCollection

▼ Test Case Design

Si riempie una hashtable di valori e si testa che il metodo sulla collection ottenuta da questa riferisca correttamente la presenza o meno degli elementi inseriti. Si testa inoltre che non riferisca il contenimento di elementi diversi da value

▼ Test Description

Creo una hashtable, inserisco dei valori generici e ne creo una ValueCollection
Verifico che contains torni vero per valori uguali a quelli inseriti
Verifico che contains torni falso per elementi che non sono valori
Verifico che contains torni falso per valori diversi da quelli inseriti

▼ Pre-Condition

None

▼ Post-Condition

None

▼ Expected Results

Il test passa con successo senza lanciare eccezioni o contraddire un assert

containsAllTest

▼ Summary

Test del metodo containsAll di ValueCollection

▼ Test Case Design

Si riempie una hashtable di valori e si testa che il metodo sulla collection ottenuta da questa riferisca correttamente la presenza o meno di liste contenenti valori inseriti. Si testa inoltre che non riferisca il contenimento di elementi diversi da Collection. Verifico inoltre che venga lanciato NullPointerException se viene passata una collection nulla

▼ Test Description

Creo una hashtable, inserisco dei valori generici e ne creo una ValueCollection
Creo una lista e inserisco gli stessi elementi
Verifico che containsAll torni vero per la lista contenente gli stessi valori
Modifico la lista con valori diversi
Verifico che containAll torni falso per la lista contenente valori diversi
Verifico che containAll torni vero per una lista vuota
Verifico che containAll lanci NullPointerException passando null come parametro

▼ Pre-Condition

None

▼ Post-Condition

None

▼ Expected Results

Il test passa con successo senza lanciare eccezioni o contraddire un assert

equalsTest

▼ Summary

Test del metodo equals di ValueCollection

▼ Test Case Design

Si riempiono due hashtable di valori e si testa che il metodo sulle collection ottenute da questa riferisca correttamente l'uguaglianza o meno delle ValueCollection. Si testa inoltre che collection diverse siano ritenute diverse.

▼ Test Description

Creo due hashtable, inserisco dei valori generici e ne creo le rispettive ValueCollection
Verifico che equals torni vero confrontando le due ValueCollection
Modifico una hashtable con valori diversi
Verifico che equals torni falso confrontando le due ValueCollection
Pulisco le due hashtable e verifico che risultino uguali
Inserisco più valori uguali sulle due hashtable e verifico che le ValueCollection risultino uguali

▼ Pre-Condition

None

▼ Post-Condition

None

▼ Expected Results

Il test passa con successo senza lanciare eccezioni o contraddire un assert

hashTest

▼ Summary

Test del metodo hashCode di ValueCollection

▼ Test Case Design

Si verifica che una ValueCollection da una hashTable vuota e con valori ritorni il numero corretto dell'hashcode

▼ Test Description

Creo una hashtable ne creo una ValueCollection
Utilizzo hashcode sulla collection e verifico che l'hash code torni 0
Inserisco degli elementi e verifico che le collection tornino il valore corretto di hashcode

▼ Pre-Condition

None

▼ Post-Condition

None

▼ Expected Results

Il test passa con successo senza lanciare eccezioni o contraddire un assert

isEmptyTest

▼ Summary

Test del metodo isEmpty di ValueCollection

▼ Test Case Design

Si verifica che una ValueCollection da una hashTable vuota e con valori ritorni correttamente vero o falso

▼ Test Description

Creo una hashtable ne creo una ValueCollection
Utilizzo isEmpty sulla collection e verifico che isEmpty sia vero
Inserisco degli elementi e verifico che isEmpty sia falso

▼ Pre-Condition

None

▼ Post-Condition

None

▼ Expected Results

Il test passa con successo senza lanciare eccezioni o contraddire un assert

iteratorTest

▼ Summary

Test del metodo iterator di ValueCollection

▼ Test Case Design

Si verifica che la creazione di un iteratore a ValueCollection da una hashTable non ritorni un valore null

▼ Test Description

Creo una hashtable ne creo una ValueCollection
Utilizzo iterator sulla collection e verifico che il valore di ritorno non sia null

▼ Pre-Condition

None

▼ Post-Condition

None

▼ Expected Results

Il test passa con successo senza lanciare eccezioni o contraddire un assert

removeTest

▼ Summary

Test del metodo remove di ValueCollection

▼ Test Case Design

Si verifica che una ValueCollection da una hashTable vuota non possa rimuovere oggetti e che rimuova o meno correttamente gli elementi dalla ValueCollection e dalla rispettiva hashtable non vuota. Si verifica inoltre che la rimozione di valori nulle lanci NullPointerException

▼ Test Description

Creo una hashtable ne creo una ValueCollection
Testo che la rimozione di null dalla collection lanci NullPointerException
Testo che non vengano effettuate rimozioni se la collection è vuota
Inserisco degli elementi e verifico che avvenga correttamente la rimozione del valore dalla ValueCollection e dalla hashtable relativa
Tento la rimozione dello stesso valore per verificare che si possa eseguire nuovamente

▼ Pre-Condition

None

▼ Post-Condition

None

▼ Expected Results

Il test passa con successo senza lanciare eccezioni o contraddire un assert

removeAllTest

▼ Summary

Test del metodo removeAll di ValueCollection

▼ Test Case Design

Si verifica che una ValueCollection da una hashTable vuota non possa rimuovere collection di oggetti e che rimuova o meno correttamente gli elementi da una ValueCollection e dalla rispettiva hashtable non vuota. Si verifica inoltre che il passaggio di una collection nulla lanci NullPointerException

▼ Test Description

Creo una hashtable ne creo una ValueCollection
Testo che removeAll di null dalla collection lanci NullPointerException
Inserisco degli elementi nella ValueCollection
Testo che non vengano effettuate rimozioni passando una lista vuota

Pulisco la hashtable e inserisco un valore nella lista
Verifico che non avvengano modifiche alla ValueCollection e alla hashtable vuote effettuando removeAll di una collection piena
Aggiungo lo stesso valore della lista alla hashtable e verifico che removeAll della lista svuoti la hashtable e la ValueCollection

▼ Pre-Condition

None

▼ Post-Condition

None

▼ Expected Results

Il test passa con successo senza lanciare eccezioni o contraddire un assert

retainAllTest

▼ Summary

Test del metodo retainAll di ValueCollection

▼ Test Case Design

Si verifica che retainAll contenente gli stessi valori della ValueCollection relativa alla hashtable non apporti modifiche al set e alla hashtable, che una lista vuota o con meno elementi della hashtable effettui correttamente le eliminazioni e che una lista contenente più elementi della ValueCollection apporti le corrette modifiche. Si verifica inoltre che il passaggio di una collection nulla lanci NullPointerException

▼ Test Description

Creo una hashtable ne creo una ValueCollection
Testo che retainAll di null dalla collection lanci NullPointerException
Inserisco degli elementi nella ValueCollection
Testo che venga svuotata la collection e la hashtable passando una lista vuota
Pulisco la hashtable e inserisco un valore nella collection
Verifico che non avvengano modifiche ai valori della collection e alla hashtable vuote effettuando retainAll di una collection piena
Aggiungo più valori della lista alla hashtable e verifico che retainAll della lista mantenga solo i valori contenuti in essa

▼ Pre-Condition

None

▼ Post-Condition

None

▼ Expected Results

Il test passa con successo senza lanciare eccezioni o contraddire un assert

sizeTest

▼ Summary

Test del metodo size di ValueCollection

▼ Test Case Design

Si verifica che una ValueCollection da una hashTable vuota abbia dimensione 0 e che aggiungendo un valore alla hashtable la dimensione del set aumenti

▼ Test Description

Creo una hashtable ne creo una ValueCollection
Utilizzo size sulla collection e verifico che torni 0
Inserisco degli elementi e verifico che le collection tornino il valore corretto di dimensione

▼ Pre-Condition

None

▼ Post-Condition

None

▼ Expected Results

Il test passa con successo senza lanciare eccezioni o contraddire un assert

toArrayTest

▼ Summary

Test del metodo toArray di ValueCollection

▼ Test Case Design

Si verifica che una ValueCollection da una hashTable vuota ritorni un array di dimensione 0 e che aggiungendo elementi venga tornato un array

identico a quello atteso

▼ Test Description

Creo una hashtable ne creo una ValueCollection
Utilizzo toArray sulla collection e verifico che torni un array di object di dimensione 0
Inserisco degli elementi nella hashtable e verifico che toArray torni un array contenente gli stessi valori inseriti e di dimensione minima

▼ Pre-Condition

None

▼ Post-Condition

None

▼ Expected Results

Il test passa con successo senza lanciare eccezioni o contraddire un assert

toArrayTest

▼ Summary

Test del metodo toArray di ValueCollection

▼ Test Case Design

Si verifica che una ValueCollection da una hashTable vuota ritorni un array di dimensione 0 passando un array vuoto e che aggiungendo elementi alla hashtable e passando un array di dimensione maggiore torni l'array parzialmente pieno mentre tornandone uno di dimensione inferiore o corretta

▼ Test Description

Creo una hashtable e ne creo una ValueCollection
Utilizzo toArray con array vuoto sulla collection e verifico che torni un array di object di dimensione 0
Inserisco degli elementi nella hashtable e verifico che toArray passando un array vuoto torni un array contenente gli stessi valori inseriti e di dimensione minima
Verifico che toArray passando un array di dimensione maggiore di quella richiesta ritorni un array parzialmente riempito

Verifico che toArray passando un array di dimensione corretta ritorni un array riempito della stessa dimensione

▼ Pre-Condition

None

▼ Post-Condition

None

▼ Expected Results

Il test passa con successo senza lanciare eccezioni o contraddire un assert

IteratorTester

▼ Summary

Questo test mira a verificare il corretto funzionamento della classe Iterator, partendo dal test del costruttore si va verificare che ogni metodo ritorni i valori corretti inseriti all'interno della lista, che i remove e i set per chiavi e valori abbiano un effetto effettivo sulle liste sottostanti

▼ Test Suite Design

Il test è disegnato in maniera tale da testare casi generali di creazione di iterator di set e collection, effettuare rimozioni e iterazioni sui valori interni al set o collection. Ogni metodo di test è autonomo e non necessita di variabili private interne alla classe di test.

▼ Pre-Condition

Deve esistere una classe EntryAdapter, una classe MapAdapter, una classe EntrySet, una classe KeySet e una classe ValuesCollection implementate secondo l'interfaccia di HMap, HSet e HCollection rispettivamente.

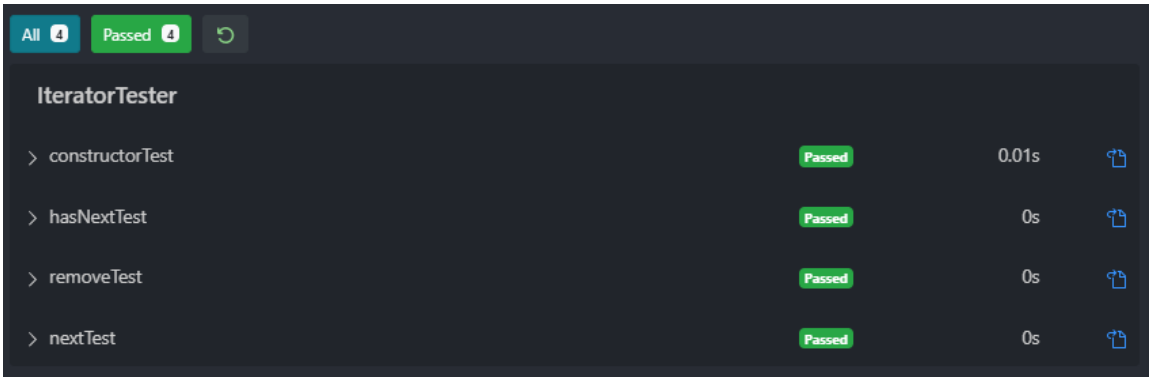
▼ Post-Condition

La classe Iterator funziona correttamente

▼ Test Cases

See below

▼ Test Suite Execution Records



▼ Execution Variables

None

constructorTester

▼ Summary

Test del costruttore dell'Iteratore

▼ Test Case Design

Si testa che l'oggetto ritornato dal costruttore non sia nullo e che lanci NullPointerException se viene passato un valore null per set e collection o se viene inizializzato con indici non validi

▼ Test Description

Creo una mappa e ottengo i relativi Set e Collection
Verifico che viene lanciata NullPointerException quando creo un Iteratore passando valore null
Verifico che viene lanciata NullPointerException quando creo un Iteratore passando valore null e indice valido
Verifico che viene lanciata NullPointerException quando creo un Iteratore passando Collection accettabile e indice non valido
Creo iteratori validi per set e collection e verifico che non siano creati iteratori nulli

▼ Pre-Condition

None

▼ Post-Condition

None

▼ Expected Results

Il test passa con successo senza lanciare eccezioni o contraddire un assert

hasNextTest

▼ Summary

Test del metodo hasNext dell'Iteratore

▼ Test Case Design

Si testa che hasNext ritorni correttamente vero o falso se è o non possibile iterare sulla collection perchè in ultima posizione o all'interno di collection vuote

▼ Test Description

Creo una mappa e ottengo i relativi Set e Collection iterator
Verifico che torni false su collection vuote
Verifico che torni true in delle posizioni su collection piene
Verifico che torni false se si trova nell'ultima posizione di collection piene

▼ Pre-Condition

None

▼ Post-Condition

None

▼ Expected Results

Il test passa con successo senza lanciare eccezioni o contraddire un assert

nextTest

▼ Summary

Test del metodo next dell'Iteratore

▼ Test Case Design

Si testa che next ritorni correttamente l'elemento su cui ha iterato e che lanci correttamente NoSuchElementException se si trova in ultima posizione o collection vuote

▼ Test Description

Creo una mappa e ottengo i relativi Set e Collection iterator
Verifico che lanci NoSuchElementException su collection vuote

Verifico che ritorni il corretto valore iterando sulle collection
Verifico che lanci NoSuchElementException se viene chiamato in ultima posizione

▼ Pre-Condition

None

▼ Post-Condition

None

▼ Expected Results

Il test passa con successo senza lanciare eccezioni o contraddire un assert

removeTest

▼ Summary

Test del metodo remove dell'Iteratore

▼ Test Case Design

Si testa che return rimuova correttamente gli elementi ritornati da next in precedenza e che lanci IllegalStateException se viene chiamato prima di next

▼ Test Description

Creo una mappa e ottengo i relativi Set e Collection iterator
Verifico che lanci IllegalStateException prima di aver fatto next
Verifico che elimini il valore corretto dalla Collection e mappa rispettiva dopo una chiamata a next
Verifico che lanci IllegalStateException se viene chiamato due volte di seguito

▼ Pre-Condition

None

▼ Post-Condition

None

▼ Expected Results

Il test passa con successo senza lanciare eccezioni o contraddire un assert

ListIteratorTester

▼ Summary

Questo test mira a verificare il corretto funzionamento della classe ListIterator, partendo dal test del costruttore si va verificare che ogni metodo ritorni i valori corretti inseriti all'interno del set o della collection e che i remove per le chiavi abbiano un effetto effettivo su set e collection sottostanti

▼ Test Suite Design

Il test è disegnato in maniera tale da testare casi generali di creazione di ListIterator di liste, effettuare rimozioni, alterazioni e iterazioni sui valori interni alla collection. Ogni metodo di test è autonomo e non necessita di variabili private interne alla classe di test.

▼ Pre-Condition

Deve esistere una classe ListIteratorAdapter e ListAdapter implementate secondo l'interfaccia di HListIterator e HList rispettivamente.

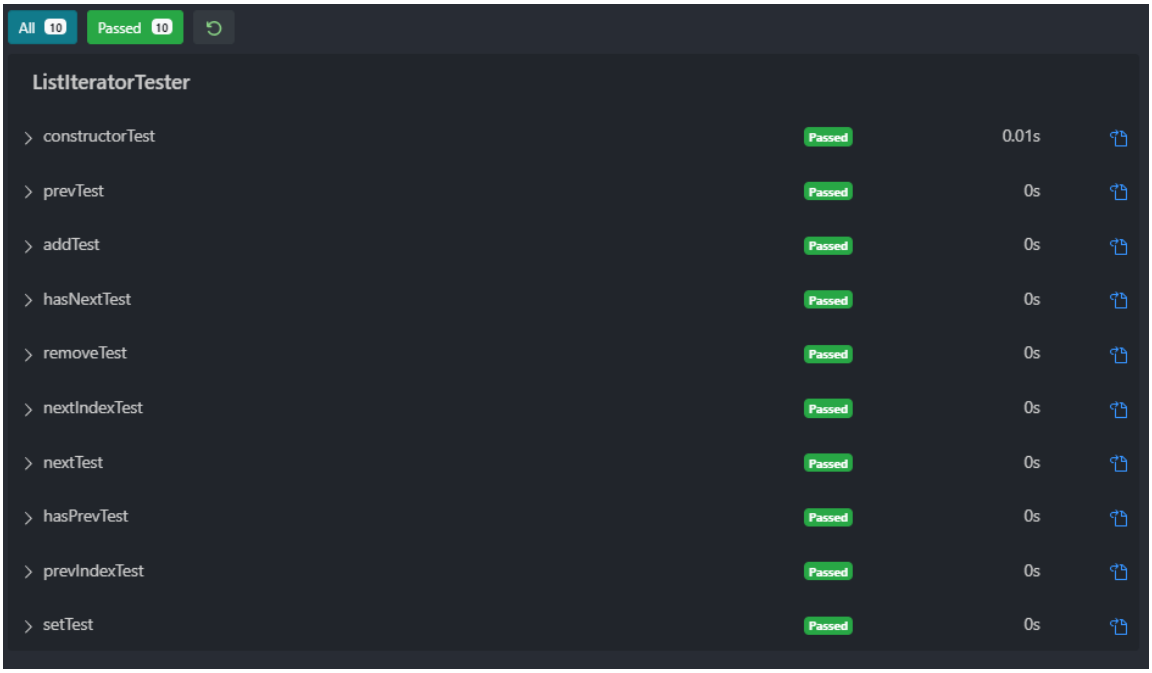
▼ Post-Condition

La classe ListIteratorAdapter funziona correttamente

▼ Test Cases

See below

▼ Test Suite Execution Records



▼ Execution Variables

None

constructorTester

▼ Summary

Test del costruttore del ListIterator

▼ Test Case Design

Si testa che l'oggetto ritornato dal costruttore non sia nullo se correttamente istanziato e che lanci IllegalArgumentException se viene passato un valore null o che lanci IndexOutOfBoundsException viene inizializzato con indici non validi

▼ Test Description

Creo una lista
Verifico che viene lanciata IllegalArgumentException quando creo un Iteratore passando valore null
Verifico che viene lanciata IllegalArgumentException quando creo un Iteratore passando valore null e indice valido
Verifico che viene lanciata IndexOutOfBoundsException quando creo un Iteratore passando liste accettabili e indici non validi
Istanzio correttamente iteratori con ogni costruttore e verifico che i valori ritornati non siano nulli

▼ Pre-Condition

None

▼ Post-Condition

None

▼ Expected Results

Il test passa con successo senza lanciare eccezioni o contraddire un assert

addTest

▼ Summary

Test del metodo add del ListIterator

▼ Test Case Design

Si testa che add inserisca correttamente l'elemento all'interno della lista partendo da inizio, metà o fine

▼ Test Description

Creo una lista e ottengo il relativo iteratore
Eseguo add e verifico che l'elemento sia stato inserito nella posizione corretta più volte
Creo un nuovo iteratore a partire da una posizione diversa
Eseguo add e verifico che l'elemento sia stato inserito nella posizione corretta
Creo un nuovo iteratore limitato
Eseguo add e verifico che l'elemento sia stato inserito nella posizione corretta

▼ Pre-Condition

None

▼ Post-Condition

None

▼ Expected Results

Il test passa con successo senza lanciare eccezioni o contraddire un assert

hasNextTest

▼ Summary

Test del metodo hasNext del ListIterator

▼ Test Case Design

Si testa che hasNext ritorni correttamente vero o falso se è o non possibile iterare sulla lista perchè in ultima posizione o all'interno di liste vuote

▼ Test Description

Creo una lista e ottengo i relativi ListIterator
Verifico che torni false su liste vuote
Aggiungo elementi alla lista
Verifico che torni true in delle posizioni su liste con elementi
Creo un nuovo iteratore a partire da una posizione diversa
Verifico che torni true in delle posizioni centrali su liste con elementi
Creo un nuovo iteratore limitato
Verifico che torni false se si trova nell'ultima posizione di liste piene o con iteratore limitato

▼ Pre-Condition

None

▼ Post-Condition

None

▼ Expected Results

Il test passa con successo senza lanciare eccezioni o contraddire un assert

hasPrevTest

▼ Summary

Test del metodo hasPrev del ListIterator

▼ Test Case Design

Si testa che hasPrev ritorni correttamente vero o falso se è o non possibile iterare sulla lista perchè in ultima prima o all'interno di liste vuote

▼ Test Description

Creo una lista e ottengo i relativi ListIterator
Verifico che torni false su liste vuote

Aggiungo elementi alla lista
Verifico che torni true in delle posizioni su liste con elementi
Creo un nuovo iteratore a partire da una posizione diversa
Verifico che torni true in delle posizioni centrali su liste con elementi
Creo un nuovo iteratore limitato
Verifico che torni false se si trova in prima posizione di liste piene o con iteratore limitato

▼ Pre-Condition

None

▼ Post-Condition

None

▼ Expected Results

Il test passa con successo senza lanciare eccezioni o contraddire un assert

nextTest

▼ Summary

Test del metodo next del ListIterator

▼ Test Case Design

Si testa che next ritorni correttamente l'elemento su cui ha iterato e che lanci correttamente NoSuchElementException se si trova in ultima posizione o collection vuote

▼ Test Description

Creo una lista e ottengo i relativi ListIterator
Verifico che lanci NoSuchElementException su liste vuote
Inserisco elementi nella lista
Verifico che ritorni il corretto valore iterando sulla lista
Verifico che lanci NoSuchElementException se viene chiamato in ultima posizione o se usato da iteratori limitati

▼ Pre-Condition

None

▼ Post-Condition

None

▼ Expected Results

Il test passa con successo senza lanciare eccezioni o contraddire un assert

nextIndexTest

▼ Summary

Test del metodo nextIndex del ListIterator

▼ Test Case Design

Si testa che nextIndex ritorni l'indice corretto che si trovi all'inizio, metà o fine della lista

▼ Test Description

Creo una lista e ottengo i relativi ListIterator
Verifico che torni 0 all'inizio della lista
Aggiungo elementi alla lista
Verifico che chiamate successive a next e previous tornino l'indice corretto
Creo un nuovo iteratore a partire da una posizione diversa
Verifico che torni l'indice corretto in delle posizioni centrali su liste con elementi
Creo un nuovo iteratore limitato
Verifico che torni la posizione corretta se usato da iteratori limitati

▼ Pre-Condition

None

▼ Post-Condition

None

▼ Expected Results

Il test passa con successo senza lanciare eccezioni o contraddire un assert

prevTest

▼ Summary

Test del metodo previous del ListIterator

▼ Test Case Design

Si testa che previous ritorni correttamente l'elemento su cui ha iterato e che lanci correttamente NoSuchElementException se si trova in prima posizione o collection vuote

▼ Test Description

Creo una lista e ottengo i relativi ListIterator
Verifico che lanci NoSuchElementException su liste vuote
Inserisco elementi nella lista
Verifico che ritorni il corretto valore iterando sulla lista
Verifico che lanci NoSuchElementException se viene chiamato in prima posizione o se usato da iteratori limitati

▼ Pre-Condition

None

▼ Post-Condition

None

▼ Expected Results

Il test passa con successo senza lanciare eccezioni o contraddire un assert

prevIndexTest

▼ Summary

Test del metodo previousIndex del ListIterator

▼ Test Case Design

Si testa che previousIndex ritorni l'indice corretto che si trovi all'inizio, metà o fine della lista

▼ Test Description

Creo una lista e ottengo i relativi ListIterator
Verifico che torni -1 all'inizio della lista
Aggiungo elementi alla lista
Verifico che chiamate successive a next e previous tornino l'indice corretto
Creo un nuovo iteratore a partire da una posizione diversa
Verifico che torni l'indice corretto in delle posizioni centrali su liste con elementi

Creo un nuovo iteratore limitato
Verifico che torni la posizione corretta se usato da iteratori limitati

▼ Pre-Condition

None

▼ Post-Condition

None

▼ Expected Results

Il test passa con successo senza lanciare eccezioni o contraddire un assert

removeTest

▼ Summary

Test del metodo remove del ListIterator

▼ Test Case Design

Si testa che return rimuova correttamente gli elementi ritornati da next e previous in precedenza e che lanci IllegalStateException se viene chiamato prima di next o previous

▼ Test Description

Creo una lista e ottengo i relativi ListIterator
Verifico che lanci IllegalStateException se viene eseguito prima di next o previous
Aggiungo elementi alla lista
Effettuo delle rimozioni tramite iteratore alternate da next e previous
Creo un nuovo iteratore a partire da una posizione diversa
Verifico che elimini l'elemento corretto e che posizioni l'indice al posto corretto
Creo un nuovo iteratore limitato
Verifico che elimini correttamente se usato da iteratori limitati

▼ Pre-Condition

None

▼ Post-Condition

None

▼ Expected Results

Il test passa con successo senza lanciare eccezioni o contraddire un assert

setTest

▼ Summary

Test del metodo set del ListIterator

▼ Test Case Design

Si testa che set sostituisca correttamente l'ultimo elemento ritornato da next e previous e che lanci IllegalStateException se viene chiamato prima di next o previous

▼ Test Description

Creo una lista e ottengo i relativi ListIterator
Verifico che lanci IllegalStateException se viene eseguito prima di next o previous
Aggiungo elementi alla lista
Effettuo delle sostituzioni tramite iteratore alternate da next e previous
Creo un nuovo iteratore a partire da una posizione diversa
Verifico che sostituisca l'elemento corretto
Creo un nuovo iteratore limitato
Verifico che sostituisca correttamente se usato da iteratori limitati

▼ Pre-Condition

None

▼ Post-Condition

None

▼ Expected Results

Il test passa con successo senza lanciare eccezioni o contraddire un assert

ListTester

▼ Summary

Questo test mira a verificare il corretto funzionamento della classe ListAdapter, partendo dal test del costruttore si va verificare che ogni metodo ritorni i valori corretti inseriti all'interno della lista, che i remove e gli add abbiano un effetto effettivo sulla lista e che i metodi per il controllo ritornino correttamente i valori attesi

▼ Test Suite Design

Il test è disegnato in maniera tale da testare casi generali di creazione di liste, effettuare rimozioni, aggiunte, confronti e ottenimento dei valori interni alla lista. Ogni metodo di test è autonomo e non necessita di variabili private interne alla classe di test.

▼ Pre-Condition

Deve esistere una classe Listdapter implementata secondo l'interfaccia di HList. List filler (metodo privato del test) funziona correttamente

▼ Post-Condition

La classe ListAdapter funziona correttamente

▼ Test Cases

See below

▼ Test Suite Execution Records

▼ Execution Variables

None

constructorTester

▼ Summary

Test del costruttore della lista

▼ Test Case Design

Si testa che l'oggetto ritornato dal costruttore non sia nullo e che la lista correttamente istanziata abbia valori consoni di dimensione

▼ Test Description

Creo una lista e verifico non sia null
Creo una lista con capacità iniziale e verifico che non sia null
Verifico che le liste abbiano dimensione 0 e che siano tra loro uguali

▼ Pre-Condition

None

▼ Post-Condition

None

▼ Expected Results

Il test passa con successo senza lanciare eccezioni o contraddire un assert

addTest

▼ Summary

Test del metodo add della lista

▼ Test Case Design

Si testa che add inserisca correttamente l'oggetto passato nella posizione corretta e che non vada a modificare altri valori già presenti

▼ Test Description

Creo una lista e inserisco un valore
Verifico che questo sia correttamente inserito all'interno della lista e che non siano presenti altri elementi
Inserisco elementi di altro tipo e verifico nuovamente la loro presenza
Verifico che tutti gli elementi al termine degli inserimenti siano al posto corretto

▼ Pre-Condition

None

▼ Post-Condition

None

▼ Expected Results

Il test passa con successo senza lanciare eccezioni o contraddire un assert

addIndexTest

▼ Summary

Test del metodo add della lista

▼ Test Case Design

Si testa che add inserisca correttamente l'oggetto passato nella posizione corretta e che non vada a modificare altri valori già presenti. Si verifica inoltre che lanci IndexOutOfBoundsException se si tenta l'inserimento al di fuori della lista

▼ Test Description

Creo una lista e verifico che add lanci IndexOutOfBoundsException inserendo elementi in posizioni negative e diverse da 0
Inserisco degli elementi in posizioni diverse e verifico che questi siano correttamente inseriti all'interno della lista e che non siano presenti altri elementi
Verifico che tutti gli elementi al termine degli inserimenti siano al posto corretto

▼ Pre-Condition

None

▼ Post-Condition

None

▼ Expected Results

Il test passa con successo senza lanciare eccezioni o contraddire un assert

addAllTest

▼ Summary

Test del metodo addAll della lista

▼ Test Case Design

Si testa che addAll inserisca correttamente collection all'interno di liste e che in caso si tenti l'inserimento in indici non validi o di collection null lanci IndexOutOfBoundsException e NullPointerException rispettivamente

▼ Test Description

Creo una lista e verifico che venga lanciato NullPointerException tentando l'inserimento di collection nulla con e senza indice
Verifico che venga lanciata IndexOutOfBoundsException quando si tenta l'inserimento di collection valide a indici non validi
Inserisco una collection vuota e verifico che non cambi la lista originale
Inserisco una collection con elementi e verifico il corretto inserimento
Inserisco una collection ad un indice valido e verifico il suo corretto inserimento

▼ Pre-Condition

None

▼ Post-Condition

None

▼ Expected Results

Il test passa con successo senza lanciare eccezioni o contraddire un assert

clearTest

▼ Summary

Test del metodo clear della lista

▼ Test Case Design

Si riempie una lista di valori e si testa che il metodo faccia in modo che la dimensione della lista torni a 0 e non siano più presenti gli elementi inseriti

▼ Test Description

Creo una lista e inserisco dei valori generici
Verifico che la dimensione dopo clear diventi 0 per la lista
Verifico che non ci siano più gli elementi precedentemente inseriti

▼ Pre-Condition

None

▼ Post-Condition

None

▼ Expected Results

Il test passa con successo senza lanciare eccezioni o contraddire un assert

containsTest

▼ Summary

Test del metodo contains della lista

▼ Test Case Design

Si riempie una lista di valori e si testa che il metodo riferisca correttamente la presenza o meno degli elementi inseriti.

▼ Test Description

Creo una lista e inserisco dei valori generici
Verifico che contains torni vero per valori uguali a quelli inseriti
Verifico che contains torni falso per valori diversi da quelli inseriti

▼ Pre-Condition

None

▼ Post-Condition

None

▼ Expected Results

Il test passa con successo senza lanciare eccezioni o contraddire un assert

containsAllTest

▼ Summary

Test del metodo containsAll della lista

▼ Test Case Design

Si riempie una lista di valori e si testa che il metodo riferisca correttamente la presenza o meno di collection contenenti elementi. Verifico inoltre che venga lanciato NullPointerException se viene passata una collection nulla

▼ Test Description

Creo una lista e inserisco dei valori generici
Verifico che containAll lanci NullPointerException passando null come parametro
Verifico che la lista contenga collection vuote
Verifico che la lista contenga sottoinsiemi di questa
Verifico che containsAll torni falso per collection contenenti elementi parzialmente uguali a quelli della lista
Verifico che containsAll torni vero per la lista contenente gli stessi elementi

▼ Pre-Condition

None

▼ Post-Condition

None

▼ Expected Results

Il test passa con successo senza lanciare eccezioni o contraddire un assert

equalsTest

▼ Summary

Test del metodo equals della lista

▼ Test Case Design

Si riempiono due liste di valori e si testa che il metodo riferisca correttamente l'uguaglianza o meno delle liste. Si testa inoltre che oggetti diversi da liste siano ritenuti diversi.

▼ Test Description

Creo due liste e inserisco dei valori generici
Verifico che equals torni falso confrontando le due liste
Modifico una lista rendendola uguale all'altra
Verifico che equals torni vero confrontando le due liste
Verifico che anche gli hashcode risultino uguali
Confronto una lista con un oggetto generico per verificare che non risultino uguali

▼ Pre-Condition

None

▼ Post-Condition

None

▼ Expected Results

Il test passa con successo senza lanciare eccezioni o contraddire un assert

getTest

▼ Summary

Test del metodo get della lista

▼ Test Case Design

Si crea una lista sulla quale si testa che venga lanciato IndexOutOfBoundsException se si tenta di prendere elementi in posizioni non presenti, vi si inseriscono degli elementi e si verifica che get ritorni l'elemento corretto atteso passando una data posizione valida

▼ Test Description

Creo una lista e verifico che venga lanciato IndexOutOfBoundsException passando un qualsiasi indice
Aggiungo degli elementi e verifico che il get torni correttamente l'elemento relativo alla posizione passata

▼ Pre-Condition

None

▼ Post-Condition

None

▼ Expected Results

Il test passa con successo senza lanciare eccezioni o contraddire un assert

hashTest

▼ Summary

Test del metodo hashCode della lista

▼ Test Case Design

Si verifica che una lista vuota e con valori ritorni il numero corretto dell'hashcode

▼ Test Description

Creo una lista e utilizzo hashcode per verificare che l'hash code sia 0
Inserisco degli elementi e verifico che le liste tornino il valore corretto di hashcode

▼ Pre-Condition

None

▼ Post-Condition

None

▼ Expected Results

Il test passa con successo senza lanciare eccezioni o contraddire un assert

indexOfTest

▼ Summary

Test del metodo indexOf della lista

▼ Test Case Design

Si verifica che una lista vuota e con valori ritorni correttamente la prima posizione dell'elemento passato se questo è presente, -1 altrimenti

▼ Test Description

Creo una lista e verifico che indexOf torni -1 per degli elementi
Inserisco degli elementi e verifico che venga tornata la posizione corretta
Inserisco più elementi uguali e verifico che venga tornata la prima posizione in cui è presente l'elemento passato a indexOf

▼ Pre-Condition

None

▼ Post-Condition

None

▼ Expected Results

Il test passa con successo senza lanciare eccezioni o contraddire un assert

isEmptyTest

▼ Summary

Test del metodo isEmpty della lista

▼ Test Case Design

Si verifica che una lista vuota e con valori ritorni correttamente vero o falso

▼ Test Description

Creo una lista e verifico che isEmpty sia vero
Inserisco degli elementi e verifico che isEmpty sia falso
Pulisco la lista e verifico che isEmpty ritorni di nuovo vero

▼ Pre-Condition

None

▼ Post-Condition

None

▼ Expected Results

Il test passa con successo senza lanciare eccezioni o contraddire un assert

iteratorTest

▼ Summary

Test del metodo iterator della lista

▼ Test Case Design

Si verifica che la creazione di un iteratore a lista non ritorni un valore null

▼ Test Description

Creo una lista e il rispettivo iteratore
Verifico che il valore dell'iteratore creato non sia null

▼ Pre-Condition

None

▼ Post-Condition

None

▼ Expected Results

Il test passa con successo senza lanciare eccezioni o contraddire un assert

lastIndexOfTest

▼ Summary

Test del metodo lastIndexOf della lista

▼ Test Case Design

Si verifica che una lista vuota e con valori ritorni correttamente l'ultima posizione dell'elemento passato se questo è presente, -1 altrimenti

▼ Test Description

Creo una lista e verifico che lastIndexOf torni -1 per degli elementi
Inserisco degli elementi e verifico che venga tornata la posizione corretta
Inserisco più elementi uguali e verifico che venga tornata l'ultima posizione in cui è presente l'elemento passato a indexOf

▼ Pre-Condition

None

▼ Post-Condition

None

▼ Expected Results

Il test passa con successo senza lanciare eccezioni o contraddire un assert

listIteratorTest

▼ Summary

Test del metodo listIterator della lista

▼ Test Case Design

Si verifica che la creazione di un ListIterator a lista non ritorni un valore null e che venga lanciato IndexOutOfBoundsException se si passano indici non validi

▼ Test Description

Creo una lista e verifico che venga lanciato IndexOutOfBoundsException se questa è vuota o se viene passato un indice non valido
Aggiungo un elemento alla lista e verifico che venga lanciato IndexOutOfBoundsException per indici non validi e che venga invece creato un iteratore non null per indici validi

▼ Pre-Condition

None

▼ Post-Condition

None

▼ Expected Results

Il test passa con successo senza lanciare eccezioni o contraddire un assert

removeIndexTest

▼ Summary

Test del metodo remove della lista

▼ Test Case Design

Si verifica che una lista rimuova correttamente e ritorni l'elemento all'indice indicato a remove. Si verifica inoltre che la rimozione su indici non validi lanci IndexOutOfBoundsException

▼ Test Description

Creo una lista testo che la rimozione di di indici non validi lanci IndexOutOfBoundsException
Inserisco degli elementi e verifico che l'elemento ritornato da remove sia quello atteso e che la rimozione sia avvenuta correttamente e con successo mantenendo l'ordine corretto degli elementi

▼ Pre-Condition

None

▼ Post-Condition

None

▼ Expected Results

Il test passa con successo senza lanciare eccezioni o contraddire un assert

removeObjectTest

▼ Summary

Test del metodo remove della lista

▼ Test Case Design

Si verifica che una lista rimuova correttamente e ritorni true dalla rimozione di un oggetto se questo è presente. Si verifica inoltre che la rimozione di oggetti non presenti non modifichi la lista e torni false

▼ Test Description

Creo una lista e verifico che ritorni false dalla richiesta di rimozione di oggetti non presenti
Inserisco degli elementi e verifico che la rimozione di elementi presenti torni true e che sia avvenuta correttamente e con successo mantenendo l'ordine corretto degli elementi
Verifico che non si possano più rimuovere elementi già eliminati

▼ Pre-Condition

None

▼ Post-Condition

None

▼ Expected Results

Il test passa con successo senza lanciare eccezioni o contraddire un assert

removeAllTest

▼ Summary

Test del metodo removeAll della lista

▼ Test Case Design

Si verifica che una lista vuota non possa rimuovere collection di oggetti e che rimuova o meno correttamente gli elementi da una lista non vuota. Si verifica inoltre che il passaggio di una collection nulla lanci NullPointerException

▼ Test Description

Creo una lista
Testo che removeAll di null dalla lista lanci NullPointerException
Inserisco degli elementi nella lista
Testo che venga svuotata la lista passando una collection uguale
Testo che non vengano effettuate rimozioni passando una collection con elementi diversi
Testo che vengano effettuate le corrette rimozioni passando una collection con alcuni elementi diversi
Testo che non vengano effettuate rimozioni passando una lista vuota

▼ Pre-Condition

None

▼ Post-Condition

None

▼ Expected Results

Il test passa con successo senza lanciare eccezioni o contraddire un assert

retainAllTest

▼ Summary

Test del metodo retainAll della lista

▼ Test Case Design

Si verifica che retainAll di una collection contenente le stesse chiavi della lista non apporti modifiche a questa, che una collection vuota o con meno elementi della lista effettui correttamente le eliminazioni e che una collection contenente più elementi della lista apporti le corrette modifiche. Si verifica inoltre che il passaggio di una collection nulla lanci NullPointerException

▼ Test Description

Creo una lista
Testo che retainAll di null dalla lista lanci NullPointerException
Inserisco degli elementi nella lista
Verifico che non avvengano modifiche alla lista effettuando retainAll di una collection uguale alla lista
Testo che venga svuotata la lista passando una collection vuota
Inserisco elementi di tipo diverso nella lista e collection, alcuni in comune
Verifico che avvengano le corrette modifiche utilizzando retainAll della collection sulla lista

▼ Pre-Condition

None

▼ Post-Condition

None

▼ Expected Results

Il test passa con successo senza lanciare eccezioni o contraddire un assert

setTest

▼ Summary

Test del metodo set della lista

▼ Test Case Design

Si verifica il set di un elemento in una posizione di una lista esegua correttamente la sostituzione all'indice indicato o che lanci IndexOutOfBoundsException se viene passato un indice invalido

▼ Test Description

Creo una lista e verifico che venga lanciato IndexOutOfBoundsException per qualsiasi indice
Inserisco degli elementi nella lista e verifico che il set in posizioni corrette sostituisca correttamente gli elementi
Verifico che venga lanciato IndexOutOfBoundsException in posizioni non valide con la lista piena

▼ Pre-Condition

None

▼ Post-Condition

None

▼ Expected Results

Il test passa con successo senza lanciare eccezioni o contraddire un assert

sizeTest

▼ Summary

Test del metodo size della lista

▼ Test Case Design

Si verifica che una lista vuota abbia dimensione 0 e che aggiungendo e rimuovendo degli elementi la dimensione della lista aumenti coerentemente

▼ Test Description

Creo una lista e verifico che size torni 0
Inserisco degli elementi e verifico che size torni il valore corretto di dimensione

Rimuovo degli elementi e verifico che size torni il valore corretto di dimensione

▼ Pre-Condition

None

▼ Post-Condition

None

▼ Expected Results

Il test passa con successo senza lanciare eccezioni o contraddire un assert

subListTest

▼ Summary

Test del metodo subList della lista

▼ Test Case Design

Si verifica che il metodo subList torni un valore non nullo e che lanci correttamente IndexOutOfBoundsException passando indici non validi

▼ Test Description

Creo una lista e verifico che subList lanci IndexOutOfBoundsException su lista vuota
Inserisco degli elementi e verifico che subList non ritorni null con indici validi e che lanci IndexOutOfBoundsException se gli indici non sono validi

▼ Pre-Condition

None

▼ Post-Condition

None

▼ Expected Results

Il test passa con successo senza lanciare eccezioni o contraddire un assert

toArrayTest

▼ Summary

Test del metodo toArray della lista

▼ Test Case Design

Si verifica che una lista vuota ritorni un array di dimensione 0 e che aggiungendo elementi venga tornato un array identico a quello atteso

▼ Test Description

Creo una lista
Utilizzo toArray sul set e verifico che torni un array di object di dimensione 0
Inserisco degli elementi nella lista e verifico che toArray torni un array contenente gli stessi valori inseriti e di dimensione minima

▼ Pre-Condition

None

▼ Post-Condition

None

▼ Expected Results

Il test passa con successo senza lanciare eccezioni o contraddire un assert

toArrayObjectTest

▼ Summary

Test del metodo toArray della lista

▼ Test Case Design

Si verifica che una lista vuota ritorni un array di dimensione 0 passando un array vuoto e che aggiungendo elementi alla lista e passando un array di dimensione maggiore torni l'array parzialmente pieno mentre passandone uno di dimensione inferiore o corretta torni un array di dimensione minima

▼ Test Description

Creo una lista
Utilizzo toArray con array vuoto sulla lista e verifico che torni un array di object di dimensione 0
Inserisco degli elementi nella lista e verifico che toArray passando un array vuoto torni un array contenente le stesse chiavi inserite e di dimensione minima
Verifico che toArray passando un array di dimensione maggiore di quella richiesta ritorni un array parzialmente riempito

▼ Pre-Condition

None

▼ Post-Condition

None

▼ Expected Results

Il test passa con successo senza lanciare eccezioni o contraddire un assert

LimitedListTester

▼ Summary

Questo test mira a verificare il corretto funzionamento della classe LimitedListAdapter, partendo dal test del costruttore si va verificare che ogni metodo ritorni i valori corretti inseriti all'interno della lista limitata, che i remove e gli add abbiano un effetto effettivo sulla lista pur rimanendo nel range attribuito e che i metodi per il controllo ritornino correttamente i valori attesi

▼ Test Suite Design

Il test è disegnato in maniera tale da testare casi generali di creazione di sottoliste di liste, effettuare rimozioni, aggiunte, confronti e ottenimento dei valori interni alla sotto lista. Ogni metodo di test è autonomo e non necessita di variabili private interne alla classe di test.

▼ Pre-Condition

Deve esistere una classe Listdapter e una classe LimitedListAdapter implementate secondo l'interfaccia di HList.

▼ Post-Condition

La classe LimitedListAdapter funziona correttamente

▼ Test Cases

See below

▼ Test Suite Execution Records

▼ Execution Variables

None

constructorTester

▼ Summary

Test del costruttore della sotto lista

▼ Test Case Design

Si testa che l'oggetto ritornato dal costruttore non sia nullo e che la sottolista correttamente istanziata abbia valori consoni di dimensione. Si verifica inoltre che venga ritornato NullPointerException e IndexOutOfBoundsException se vengono passati rispettivamente ListAdapter null o indici di inizializzazione invalidi

▼ Test Description

Verifico che venga lanciato NullPointerException tentando di creare una sottolista con lista nulla
Creo una lista e verifico che lanci IndexOutOfBoundsException passando indici non validi
Creo una lista e verifico non sia null passando indici validi

▼ Pre-Condition

None

▼ Post-Condition

None

▼ Expected Results

Il test passa con successo senza lanciare eccezioni o contraddire un assert

addTest

▼ Summary

Test del metodo add della sotto lista

▼ Test Case Design

Si testa che add inserisca correttamente l'oggetto passato nella posizione corretta relativamente agli indici limitati e che non vada a modificare altri valori già presenti. Verifico inoltre che lanci IllegalStateException se viene chiamata più volte sulla stessa sotto lista

▼ Test Description

Creo una lista e la sua relativa sottolista e vi inserisco un valore
Verifico che questo sia correttamente inserito all'interno della lista, ritorni true e che se si tenta un altro add venga lanciata IllegalStateException
Verifico che pur lanciando IllegalStateException non sia stata modificata la lista sottostante

Creo una nuova sottolista e verifico che add funzioni allo stesso modo partendo da indici diversi

▼ Pre-Condition

None

▼ Post-Condition

None

▼ Expected Results

Il test passa con successo senza lanciare eccezioni o contraddire un assert

addIndexTest

▼ Summary

Test del metodo add della sotto lista

▼ Test Case Design

Si testa che add inserisca correttamente l'oggetto passato nella posizione corretta e che non vada a modificare altri valori già presenti. Si verifica inoltre che lanci IndexOutOfBoundsException se si tenta l'inserimento al di fuori della sotto lista e che lanci IllegalStateException se viene chiamata più volte sulla stessa sotto lista

▼ Test Description

Creo una lista e la relativa sottolista, verifico che add lanci IndexOutOfBoundsException inserendo elementi in posizioni negative e diverse da 0
Inserisco degli elementi in posizioni diverse e verifico che questi siano correttamente inseriti all'interno della lista e che non siano presenti altri elementi
Verifico che venga lanciata IllegalStateException se viene ripetuto add
Verifico che tutti gli elementi al termine degli inserimenti siano al posto corretto
Creo altre sottoliste e verifico che gli elementi vengano inseriti nella lista con il corretto offset

▼ Pre-Condition

None

▼ Post-Condition

None

▼ Expected Results

Il test passa con successo senza lanciare eccezioni o contraddire un assert

addAllTest

▼ Summary

Test del metodo addAll della sotto lista

▼ Test Case Design

Si testa che addAll inserisca correttamente collection all'interno di sottoliste e che in caso si tenti l'inserimento in indici non validi o di collection null lanci IndexOutOfBoundsException e NullPointerException rispettivamente. Verifico inoltre che lanci IllegalStateException se viene chiamata più volte sulla stessa sotto lista

▼ Test Description

Creo una lista e relativa sottolista e verifico che venga lanciato NullPointerException tentando l'inserimento di collection nulla con e senza indice
Verifico che venga lanciata IndexOutOfBoundsException quando si tenta l'inserimento di collection valide a indici non validi
Inserisco una collection e verifico che cambi coerentemente la lista originale
Verifico che non il tentativo di usare addAll più volte lanci IllegalStateException
Creo una nuova collection e la inserisco ad un indice valido su una nuova sottolista e verifico il suo corretto inserimento
Creo una nuova sottolista e tento l'inserimento di una collection vuota per verificare che non cambi il contenuto della lista

▼ Pre-Condition

None

▼ Post-Condition

None

▼ Expected Results

Il test passa con successo senza lanciare eccezioni o contraddire un assert

clearTest

▼ Summary

Test del metodo clear della sotto lista

▼ Test Case Design

Si riempie una lista di valori e si testa su una sottolista che il metodo faccia in modo che vengano eliminati solo i valori contenuti nella sottolista.
Verifico inoltre che lanci IllegalStateException se viene chiamata più volte sulla stessa sotto lista

▼ Test Description

Creo una lista e inserisco dei valori generici
Creo una sottolista comprendente tutta la lista e verifico che clear elimini tutti gli elementi della lista
Verifico che lanci IllegalStateException se viene chiamato clear più volte
Riempio la lista e ne creo una sottolista ridotta
Chiamo clear e verifico che vengano eliminati solo gli elementi contenuti nella sottolista

▼ Pre-Condition

None

▼ Post-Condition

None

▼ Expected Results

Il test passa con successo senza lanciare eccezioni o contraddire un assert

containsTest

▼ Summary

Test del metodo contains della sotto lista

▼ Test Case Design

Si riempie una lista di valori e si testa che il metodo sulla sottolista riferisca correttamente la presenza o meno degli elementi contenuti nella sottolista.
Verifico inoltre che lanci IllegalStateException se viene chiamata dopo una funzione che modifica la struttura della lista

▼ Test Description

Creo una lista e inserisco dei valori generici
Verifico che contains torni vero per i valori uguali a quelli contenuti nella sottolista
Verifico che venga lanciato IllegalStateException se eseguita dopo funzioni che apportano modifiche strutturali alla lista
Verifico che contains torni falso per valori contenuti nella lista ma non compresi nella sottolista

▼ Pre-Condition

None

▼ Post-Condition

None

▼ Expected Results

Il test passa con successo senza lanciare eccezioni o contraddire un assert

containsAllTest

▼ Summary

Test del metodo containsAll della sotto lista

▼ Test Case Design

Si riempie una lista di valori e si testa che il metodo applicato alla sottolista riferisca correttamente la presenza o meno di collection contenenti elementi. Verifico inoltre che venga lanciato NullPointerException se viene passata una collection nulla e che lanci IllegalStateException se viene chiamata più volte sulla stessa sotto lista

▼ Test Description

Creo una lista e inserisco dei valori generici
Verifico che containAll sulla sottolista lanci NullPointerException passando null come parametro
Verifico che la sotto lista contenga se stessa
Verifico che venga lanciato IllegalStateException se eseguita dopo funzioni che apportano modifiche strutturali alla lista
Verifico che containsAll torni vero per la lista contenente gli stessi elementi
Verifico che containsAll torni falso per collection contenenti elementi parzialmente uguali a quelli della lista

▼ Pre-Condition

None

▼ Post-Condition

None

▼ Expected Results

Il test passa con successo senza lanciare eccezioni o contraddire un assert

equalsTest

▼ Summary

Test del metodo equals della sotto lista

▼ Test Case Design

Si riempiono delle liste di valori e si testa sulle relative sottolista che il metodo riferisca correttamente l'uguaglianza o meno di queste. Verifico inoltre che lanci IllegalStateException se viene chiamata dopo una funzione che modifica la struttura della lista

▼ Test Description

Creo una lista e inserisco dei valori generici
Verifico che equals torni falso confrontando la lista con una sottolista identica
Modifico la lista e verifico che venga lanciato IllegalStateException da equals della sottolista
Verifico che equals torni falso confrontando due sottoliste diverse

▼ Pre-Condition

None

▼ Post-Condition

None

▼ Expected Results

Il test passa con successo senza lanciare eccezioni o contraddire un assert

getTest

▼ Summary

Test del metodo get della sotto lista

▼ Test Case Design

Si crea una lista sulla quale sottolista si testa che venga lanciato IndexOutOfBoundsException se si tenta di prendere elementi in posizioni non presenti, vi si inseriscono degli elementi e si verifica che get ritorni l'elemento corretto atteso passando una data posizione valida. Verifico inoltre che lanci IllegalStateException se viene chiamata dopo una funzione che modifica la struttura della lista

▼ Test Description

Creo una lista, inserisco degli elementi e verifico che dalla sottolista venga lanciato IndexOutOfBoundsException passando un indice invalido
Aggiungo degli elementi e verifico che il get sulla sottolista lanci IllegalStateException
Aggiungo degli elementi e verifico che il get sulla sottolista torni correttamente l'elemento della lista relativo alla posizione passata

▼ Pre-Condition

None

▼ Post-Condition

None

▼ Expected Results

Il test passa con successo senza lanciare eccezioni o contraddire un assert

hashTest

▼ Summary

Test del metodo hashCode della sotto lista

▼ Test Case Design

Si verifica che una lista vuota e con valori ritorni il numero corretto dell'hashcode. Verifico inoltre che lanci IllegalStateException se viene chiamata dopo una funzione che modifica la struttura della lista

▼ Test Description

Creo una lista e utilizzo hashcode per verificare che l'hash code della sottolista sia 0
Inserisco degli elementi e verifico che la sottolista lanci

IllegalStateException
Verifico poi che una nuova sottolista torni il valore corretto di hashcode

▼ Pre-Condition

None

▼ Post-Condition

None

▼ Expected Results

Il test passa con successo senza lanciare eccezioni o contraddire un assert

indexOfTest

▼ Summary

Test del metodo indexOf della sotto lista

▼ Test Case Design

Si verifica che una sottolista di una lista ritorni correttamente la prima posizione dell'elemento passato se questo è presente, -1 altrimenti. Verifico inoltre che lanci IllegalStateException se viene chiamata dopo una funzione che modifica la struttura della lista

▼ Test Description

Creo una lista e inserisco degli elementi
Verifico che indexOf torni -1 per degli elementi non presenti
Inserisco altri elementi nella lista e verifico che venga lanciata IllegalStateException dalla sottolista
Creo una nuova sottolista e verifico che indexOf ritorni la posizione relativa corretta per elementi presenti

▼ Pre-Condition

None

▼ Post-Condition

None

▼ Expected Results

Il test passa con successo senza lanciare eccezioni o contraddire un assert

isEmptyTest

▼ Summary

Test del metodo isEmpty della sotto lista

▼ Test Case Design

Si verifica che una sottolista sia sempre piena. Verifico inoltre che lanci IllegalStateException se viene chiamata dopo una funzione che modifica la struttura della lista

▼ Test Description

Creo una lista, inserisco degli elementi e verifico che isEmpty sulla sottolista sia falso
Pulisco la lista e verifico che venga lanciato IllegalStateException ad una nuova chiamata di isEmpty

▼ Pre-Condition

None

▼ Post-Condition

None

▼ Expected Results

Il test passa con successo senza lanciare eccezioni o contraddire un assert

lastIndexOfTest

▼ Summary

Test del metodo lastIndexOf della sotto lista

▼ Test Case Design

Si verifica che una lista ritorni correttamente l'ultima posizione dell'elemento passato se questo è presente, -1 altrimenti. Verifico inoltre che lanci IllegalStateException se viene chiamata dopo una funzione che modifica la struttura della lista

▼ Test Description

Creo una lista, la riempio e verifico che lastIndexOf torni l'indice corretto per elementi inseriti e entro i limiti della sottolista
Verifico che venga tornato -1 se l'elemento non è presente nella lista
Inserisco più elementi e verifico che venga lanciato IllegalStateException

Creo una nuova sottolista e verifico che elementi presenti nella lista ma non nella sottolista tornino -1 o altrimenti l'indice corretto

▼ Pre-Condition

None

▼ Post-Condition

None

▼ Expected Results

Il test passa con successo senza lanciare eccezioni o contraddire un assert

listIteratorTest

▼ Summary

Test del metodo listIterator della lista

▼ Test Case Design

Si verifica che la creazione di un ListIterator a sotto lista non ritorni un valore null e che venga lanciato IndexOutOfBoundsException se si passano indici non validi. Verifico inoltre che lanci IllegalStateException se viene chiamata dopo una funzione che modifica la struttura della lista

▼ Test Description

Creo una lista e verifico che venga lanciato IndexOutOfBoundsException se viene passato un indice non valido
Aggiungo elementi alla lista e verifico che venga lanciato IllegalStateException chiamando listIterator dalla sottolista
Creo una nuova sottolista e verifico che venga lanciato IndexOutOfBoundsException per indici non validi e che venga invece creato un iteratore non null per indici validi

▼ Pre-Condition

None

▼ Post-Condition

None

▼ Expected Results

Il test passa con successo senza lanciare eccezioni o contraddire un assert

removeIndexTest

▼ Summary

Test del metodo remove della sotto lista

▼ Test Case Design

Si verifica che una sotto lista rimuova correttamente e ritorni l'elemento all'indice indicato a remove. Si verifica inoltre che la rimozione su indici non validi lanci IndexOutOfBoundsException e che lanci IllegalStateException se viene chiamata dopo una funzione che modifica la struttura della lista

▼ Test Description

Creo una lista, inserisco degli elementi e creo una sottolista
Testo che la rimozione di di indici non validi lanci IndexOutOfBoundsException
Testo che avvenga correttamente la rimozione dalla sottolista e lista contemporaneamente
Inserisco degli elementi e verifico venga lanciata IllegalStateException
Creo una nuova sotto lista e verifico che l'elemento ritornato da remove sulla sottolista sia quello atteso e che la rimozione sia avvenuta correttamente e con successo mantenendo l'ordine corretto degli elementi

▼ Pre-Condition

None

▼ Post-Condition

None

▼ Expected Results

Il test passa con successo senza lanciare eccezioni o contraddire un assert

removeObjectTest

▼ Summary

Test del metodo remove della sotto lista

▼ Test Case Design

Si verifica che una sotto lista rimuova correttamente e ritorni true dalla rimozione di un oggetto se questo è presente. Si verifica inoltre che la rimozione di oggetti non presenti o fuori dai limiti della sottolista non

modifichi la lista e torni false. Verifico inoltre che lanci IllegalStateException se viene chiamata dopo una funzione che modifica la struttura della lista

▼ Test Description

Creo una lista, inserisco degli elementi e creo una sottolista relativa
Verifico che ritorni false dalla richiesta di rimozione di oggetti non presenti
Verifico che ritorni true e effettui correttamente la rimozione dalla lista di elementi presenti
Inserisco degli elementi e verifico che venga lanciata IllegalStateException
Creo una nuova sottolista e verifico che sia avvenuta la corretta rimozione di elementi a indici relativi alla sottolista
Verifico che non si possano più rimuovere elementi già eliminati

▼ Pre-Condition

None

▼ Post-Condition

None

▼ Expected Results

Il test passa con successo senza lanciare eccezioni o contraddire un assert

removeAllTest

▼ Summary

Test del metodo removeAll della sotto lista

▼ Test Case Design

Si verifica che una lista vuota non possa rimuovere collection di oggetti e che rimuova o meno correttamente gli elementi da una lista non vuota. Si verifica inoltre che il passaggio di una collection nulla lanci NullPointerException e che lanci IllegalStateException se viene chiamata dopo una funzione che modifica la struttura della lista

▼ Test Description

Creo una lista e vi inserisco degli elementi
Testo che removeAll di null dalla sotto lista lanci NullPointerException
Verifico che removeall della lista dalla sottolista abbia apportato le corrette modifiche
Inserisco degli elementi nella lista e verifico che venga lanciata

IllegalStateException se si tenta la rimozione data la modifica alla lista originaria
Testo che vengano rimossi correttamente gli elementi dalla lista passando una collection parzialmente uguale
Testo che non vengano effettuate rimozioni passando una collection con elementi diversi

▼ Pre-Condition

None

▼ Post-Condition

None

▼ Expected Results

Il test passa con successo senza lanciare eccezioni o contraddire un assert

retainAllTest

▼ Summary

Test del metodo retainAll della sotto lista

▼ Test Case Design

Si verifica che retainAll di una collection contenente le stesse chiavi della sotto lista non apporti modifiche a questa, che una collection vuota o con meno elementi della lista effettui correttamente le eliminazioni e che una collection contenente più elementi della lista apporti le corrette modifiche.
Si verifica inoltre che il passaggio di una collection nulla lanci NullPointerException e che lanci IllegalStateException se viene chiamata dopo una funzione che modifica la struttura della lista

▼ Test Description

Creo una lista, inserisco degli elementi e creo la relativa sottolista
Testo che retainAll di null dalla lista lanci NullPointerException
Rimuovo la sottolista dalla sottolista per verificare che avvenga correttamente la rimozione
Inserisco degli elementi nella lista
Inserisco degli elementi nella lista e verifico che venga lanciata IllegalStateException da removeAll
Verifico che avvengano le corrette modifiche utilizzando retainAll di una collection parzialmente uguale alla sottolista

Verifico che non avvengano modifiche alla lista effettuando retainAll di una collection uguale alla lista

▼ Pre-Condition

None

▼ Post-Condition

None

▼ Expected Results

Il test passa con successo senza lanciare eccezioni o contraddire un assert

setTest

▼ Summary

Test del metodo set della sotto lista

▼ Test Case Design

Si verifica il set di un elemento in una posizione di una sotto lista esegua correttamente la sostituzione all'indice relativo indicato, che lanci IndexOutOfBoundsException se viene passato un indice invalido, e che lanci IllegalStateException se viene chiamata dopo una funzione che modifica la struttura della lista

▼ Test Description

Creo una lista, inserisco un elemento e creo la relativa sottolista
Verifico che venga lanciato IndexOutOfBoundsException per qualsiasi indice non valido della sottolista
Verifico che il set in posizioni corrette sostituisca correttamente gli elementi e ritorni il valore corretto sostituito
Inserisco degli elementi e verifico che venga lanciata IllegalStateException
Verifico che vengano effettuate le corrette sostituzioni agli indici relativi della sottolista creata

▼ Pre-Condition

None

▼ Post-Condition

None

▼ Expected Results

Il test passa con successo senza lanciare eccezioni o contraddire un assert

sizeTest

▼ Summary

Test del metodo size della sotto lista

▼ Test Case Design

Si verifica che una sottolista abbia la corretta dimensione data dagli indici che limitano la sua azione. Verifico inoltre che lanci IllegalStateException se viene chiamata dopo una funzione che modifica la struttura della lista

▼ Test Description

Creo una lista, inserisco degli elementi e creo una sottolista
Verifico che la dimensione della sottolista sia coerente con i limiti imposti
Inserisco degli elementi e verifico che venga lanciata IllegalStateException
Creo una nuova sottolista e verifico che size ritorni valori coerenti

▼ Pre-Condition

None

▼ Post-Condition

None

▼ Expected Results

Il test passa con successo senza lanciare eccezioni o contraddire un assert

subListTest

▼ Summary

Test del metodo subList della lista

▼ Test Case Design

Si verifica che il metodo subList torni un valore non nullo e che lanci correttamente IndexOutOfBoundsException passando indici non validi e che lanci IllegalStateException se viene chiamata dopo una funzione che modifica la struttura della lista

▼ Test Description

Creo una lista inserisco degli elementi e creo una sottolista
Verifico che subList lanci IndexOutOfBoundsException per indici non validi

Verifico che subList non ritorni null con indici validi
Aggiungo degli elementi alla lista e verifico che venga lanciata IllegalStateException
Verifico che una sottolista di una sottolista contenga gli elementi corretti relativamente alla lista

▼ Pre-Condition

None

▼ Post-Condition

None

▼ Expected Results

Il test passa con successo senza lanciare eccezioni o contraddire un assert

toArrayTest

▼ Summary

Test del metodo toArray della sotto lista

▼ Test Case Design

Si verifica che una sottolista ritorni un array di dimensione minima contenente tutti gli elementi da questa contenuti. Verifico inoltre che lanci IllegalStateException se viene chiamata dopo una funzione che modifica la struttura della lista

▼ Test Description

Creo una lista, inserisco degli elementi e creo una sottolista
Utilizzo toArray sulla sotto lista e verifico che torni un array con gli stessi valori inseriti
Inserisco degli elementi nella lista e verifico che venga lanciata IllegalStateException
Creo una nuova sottolista e verifico che toArray torni un array contenente gli stessi valori inseriti e di dimensione minima

▼ Pre-Condition

None

▼ Post-Condition

None

▼ Expected Results

Il test passa con successo senza lanciare eccezioni o contraddire un assert

toArrayObjectTest

▼ Summary

Test del metodo toArray della sotto lista

▼ Test Case Design

Si verifica che una sotto ritorni un array di dimensione corretta passando un array vuoto e che aggiungendo elementi alla lista e passando un array di dimensione maggiore torni l'array parzialmente pieno mentre passandone uno di dimensione inferiore o corretta torni un array di dimensione minima. Verifico inoltre che lanci IllegalStateException se viene chiamata dopo una funzione che modifica la struttura della lista

▼ Test Description

Creo una lista, inserisco un elemento e creo una sottolista
Utilizzo toArray con array null sulla sotto lista e verifico che torni un array di object di dimensione 1 contenente l'elemento inserito
Verifico che toArray passando un array pieno torni un array contenente gli elementi inseriti nella sottolista ma che non abbia sovrascritto quelli di dimensione superiore
Inserisco degli elementi nella lista e verifico che venga lanciata IllegalStateException
Verifico che toArray passando un array di dimensione maggiore di quella richiesta ritorni un array parzialmente riempito

▼ Pre-Condition

None

▼ Post-Condition

None

▼ Expected Results

Il test passa con successo senza lanciare eccezioni o contraddire un assert