**Test suite della classe MapAdapter.java**

**1° ClearTest():**

* Test del metodo public void clear()
* **Summary / Description**: verifica inizialmente il corretto funzionamento su una mappa vuota, vengono poi inseriti due entry nella mappa e invocato nuovamente il metodo. Si verifica che le due entry siano state effettivamente eliminate.
* **Design**: verifica che il metodo svuoti correttamente la mappa.
* **Pre-Condition**: un oggetto MapAdapter vuoto e correttamente inizializzato.
* **Post-Condition**: la mappa torna ad essere vuota, nessuna eccezione è stata generata.
* **Expected results**: il metodo elimina correttamente tutte le coppie contenute nella mappa.

**2° ContainsKeyTest():**

* Test del metodo public boolean containsKey(Object key)
* **Summary / Description**: viene prima testato il metodo su una mappa vuota. Questa viene poi riempita con delle coppie di riferimento e si verifica il corretto funzionamento del metodo su queste entry. Alcune coppie vengono rimosse e si ritesta il metodo su questi valori aspettandosi che restituisca false. La mappa viene infine svuotata del tutto ed il metodo testato ancora.
* **Design**: verifica il corretto comportamento del metodo applicandolo in relazione ad una lista di valori di riferimento.
* **Pre-Condition**: un oggetto MapAdapter vuoto e correttamente inizializzato, un parametro “key” diverso da “null”.
* **Post-Condition**: la mappa torna ad essere vuota, il metodo restituirà sempre false.
* **Expected results**: il metodo ritorna true se e solo se la chiave cercata è presente nella mappa.

**3° ContainsKeyExceptionsTest():**

* Test del metodo public boolean containsKey(Object key)
* **Summary / Description**: il lancio dell’eccezione viene prima verificato su una mappa vuota, viene poi inserita una coppia e si verifica nuovamente il corretto funzionamento del metodo. Viene poi verificato che il valore inserito in precedenza non abbia subito modifiche.
* **Design**: verifica il corretto lancio dell’eccezione NullPointerException sia quando la mappa è vuota sia quando contiene degli elementi.
* **Pre-Condition**: un oggetto MapAdapter vuoto e correttamente inizializzato.
* **Post-Condition**: la mappa contiene un elemento che non è stato modificato dal lancio delle eccezioni. Nessun altra eccezione è stata generata.
* **Expected results**: il corretto lancio di due NullPointerException quando il parametro passato è “null”.

**4° ContainsValueTest():**

* Test del metodo public boolean containsKey(Object key)
* **Summary / Description**: viene prima testato il metodo su una mappa vuota. Questa viene poi riempita con delle coppie di riferimento e si verifica il corretto funzionamento del metodo su queste entry. Alcune coppie vengono rimosse e si ritesta il metodo su questi valori aspettandosi che restituisca false. La mappa viene infine svuotata del tutto ed il metodo testato ancora.
* **Design**: verifica il corretto comportamento del metodo applicandolo in relazione ad una lista di coppie di riferimento.
* **Pre-Condition**: un oggetto MapAdapter vuoto e correttamente inizializzato, un parametro “value” diverso da “null”.
* **Post-Condition**: la mappa torna ad essere vuota, il metodo restituirà sempre false.
* **Expected results**: il metodo ritorna true se e solo se la chiave cercata è presente nella mappa.

**5° ContainsValueExceptionsTest():**

* Test del metodo public boolean containsValue(Object value)
* **Summary / Description**: il lancio dell’eccezione viene prima verificato su una mappa vuota, viene poi inserita una coppia e si verifica nuovamente il corretto funzionamento del metodo. Viene poi verificato che il valore inserito in precedenza non abbia subito modifiche.
* **Design**: verifica il corretto lancio dell’eccezione NullPointerException sia quando la mappa è vuota sia quando contiene degli elementi.
* **Pre-Condition**: un oggetto MapAdapter vuoto e correttamente inizializzato.
* **Post-Condition**: la mappa contiene un elemento che non è stato modificato dal lancio delle eccezioni. Nessun altra eccezione è stata generata.
* **Expected results**: il corretto lancio di due NullPointerException quando il parametro passato è “null”.

**6° EntrySetTest():**

* Test del metodo public HSet entrySet()
* **Summary / Description**: verifica inizialmente il funzionamento del metodo su una mappa vuota. La mappa viene poi riempita e il metodo invocato nuovamente. Si verifica che il set abbia la stessa dimensione della mappa e che i valori al suo interno siano effettivamente delle istanze di HMap.Entry come richiesto dall’interfaccia.
* **Design**: verifica che il metodo funzioni correttamente a prescindere dalle dimensioni della mappa su cui viene invocato.
* **Pre-Condition**: un oggetto MapAdapter vuoto e correttamente inizializzato.
* **Post-Condition**: la mappa ed il set vengono inizializzati con successo.
* **Expected results**: il metodo restituisce sempre un set contenente le entry presenti nella mappa.

**7° EqualsTest():**

* Test del metodo public boolean equals(Object o)
* **Summary / Description**: viene verificato che, date due mappe:
  + Il metodo restituisce false quando vengono confrontati oggetti di classi diverse.
  + Il metodo restituisce false quando il parametro “o” è uguale a “null”.
  + Dopo aver riempito due mappe con gli stessi elementi il metodo restituisce true.
  + Dopo aver estratto un elemento dalla mappa il metodo restituisce false.
  + Dopo aver inserito nuovamente l’elemento precedentemente eliminato dalla mappa il metodo torna a restituire true.
  + Due mappe vuote sono considerate uguali.
* **Design**: verifica che il metodo si comporti adeguatamente in tutte le condizioni esaminando due mappa contenenti degli elementi di riferimento.
* **Pre-Condition**: un oggetto MapAdapter vuoto e correttamente inizializzato.
* **Post-Condition**: le due mappa tornano ad essere vuote, un confronto tra esse ritornerà true.
* **Expected results**: il metodo soddisfa la condizione m1.equals(m2) se e solo se m1.entrySet().equals(m2.entrySet()).

**8° GetTest():**

* Test del metodo public Object get(Object key)
* **Summary / Description**: verifica inizialmente il corretto funzionamento su una lista vuota. Viene poi riempita la mappa ed il metodo viene testato su dei valori di riferimento controllando che il valore ritornata sia effettivamente presente nella mappa. La mappa viene poi nuovamente svuotata e viene verificato che il metodo torni a restituite sempre il valore null.
* **Design**: verifica il corretto fu
* **Pre-Condition**: un oggetto MapAdapter vuoto e correttamente inizializzato. Un parametro “key” con valore diverso da “null”.
* **Post-Condition**: la mappa torna ad essere vuota, nuove chiamate al metodo restituiranno sempre null.
* **Expected results**: il metodo restituisce un valore diverso da null se e solo se la chiave cercata è effettivamente presente nella mappa.

**9° GetExceptionsTest():**

* Test del metodo public Object get(Object key)
* **Summary / Description**: il lancio dell’eccezione viene prima verificato su una mappa vuota, viene poi inserita una coppia e si verifica nuovamente il corretto funzionamento del metodo. Viene poi verificato che il valore inserito in precedenza non abbia subito modifiche.
* **Design**: verifica il corretto lancio dell’eccezione NullPointerException sia quando la mappa è vuota sia quando contiene degli elementi.
* **Pre-Condition**: un oggetto MapAdapter vuoto e correttamente inizializzato.
* **Post-Condition**: la mappa contiene un elemento che non è stato modificato dal lancio delle eccezioni. Nessun altra eccezione è stata generata.
* **Expected results**: il corretto lancio di due NullPointerException quando il parametro passato è “null”.

**10° HashCodeTest():**

* Test del metodo public int hashCode()
* **Summary / Description**: viene verificato che, date due mappe:
  + Il metodo sia sempre applicabile ad una mappa correttamente inzializzata.
  + Dopo aver riempito due mappe con gli stessi elementi queste hanno lo stesso hashCode.
  + Dopo aver estratto un elemento dalla mappa i due hashCode sono distinti.
  + Dopo aver inserito nuovamente l’elemento precedentemente eliminato le due mappe tornano ad avere lo stesso hashCode.
  + Mappe vuote hanno un hashCode pari a zero.
* **Design**: verifica che il metodo si comporti adeguatamente in tutte le condizioni esaminando due mappa contenenti degli elementi di riferimento.
* **Pre-Condition**: un oggetto MapAdapter vuoto e correttamente inizializzato.
* **Post-Condition**: le due mappa tornano ad essere vuote, il loro hashCode è uguale a zero
* **Expected results**: il metodo soddisfa la condizione: se m1.equals(m2) allora m1.hashCode() == m2.hashCode().

**11° IsEmptyTest():**

* Test del metodo public boolean isEmpty()
* **Summary / Description**: viene inizialmente verificato che per una mappa appena inizializzata il metodo restituisca true, la mappa viene poi riempita e si verifica che il metodo restituisca false. La mappa viene nuovamente svuotata e si verifica che il metodo torni a restituire true.
* **Design**: verifica il corretto comportamento del metodo sia quando una mappa è vuota sia quando questa contiene degli elementi.
* **Pre-Condition**: un oggetto MapAdapter vuoto e correttamente inizializzato.
* **Post-Condition**: la mappa torna ad essere vuota, nuove invocazioni del metodo restituiranno sempre true.
* **Expected results**: il metodo restituisce true se e solo se la mappa è effettivamente vuota.

**12° KeySetTest():**

* Test del metodo public HSet keySet()
* **Summary / Description**: verifica inizialmente il funzionamento del metodo su una mappa vuota. La mappa viene poi riempita e il metodo invocato nuovamente. Si verifica che il set abbia la stessa dimensione della mappa e che i valori al suo interno siano effettivamente delle chiavi della mappa come richiesto dall’interfaccia.
* **Design**: verifica che il metodo funzioni correttamente a prescindere dalle dimensioni della mappa su cui viene invocato.
* **Pre-Condition**: un oggetto MapAdapter vuoto e correttamente inizializzato.
* **Post-Condition**: la mappa ed il set vengono inizializzati con successo.
* **Expected results**: il metodo restituisce sempre un set contenente le chiavi presenti nella mappa.

**13° PutTest():**

* Test del metodo public Object put(Object key, Object value)
* **Summary / Description**: partendo da una mappa inizialmente vuota viene inizialmente inserita una nuova coppia nella mappa e si verifica che: la dimensione di questa sia stata aggiornata, “key” ed “value” siano stati memorizzati correttamente, il metodo restituisca true. Vengono poi inseriti altri valori, alcuni dei quali aventi stessa chiave, e si verifica che il metodo operi correttamente.
* **Design**: si verifica la correttezza del metodo effettuando aggiunte ripetute alla lista.
* **Pre-Condition**: un oggetto MapAdapter vuoto e correttamente inizializzato. Parametri “key” e “value” con valori diversi da “null”.
* **Post-Condition**: alla mappa sono stati aggiunte tre coppie. La sua dimensione è stata modificata correttamente.
* **Expected results**: il metodo inserisce correttamente una nuova coppia all’interno della mappa, eventualmente sovrascrivendo un valore, e aggiorna la dimensione della mappa.

**14° PutExceptionsTest():**

* Test del metodo public Object put(Object key, Object value)
* **Summary / Description**: il lancio dell’eccezione viene prima verificato su una mappa vuota. Viene poi inserita una coppia e si verifica nuovamente il corretto funzionamento del metodo. Viene poi verificato che il valore inserito in precedenza non abbia subito modifiche.
* **Design**: verifica il corretto lancio dell’eccezione NullPointerException sia quando la mappa è vuota sia quando contiene degli elementi.
* **Pre-Condition**: un oggetto MapAdapter vuoto e correttamente inizializzato.
* **Post-Condition**: la mappa contiene un elemento che non è stato modificato dal lancio delle eccezioni. Nessun altra eccezione è stata generata.
* **Expected results**: il corretto lancio di due NullPointerException quando almeno uno dei parametri passati è “null”.

**15° PutAllTest():**

* Test del metodo public void putAll(HMap t)
* **Summary / Description**: alla mappa, inizialmente vuota, si verifica che il metodo funzioni correttamente anche usando una mappa vuota come parametro. Vengono poi aggiunti alcuni valori alla mappa di base. Viene poi invocato il metodo passando come parametro una mappa appositamente create contenente dei valori di riferimento. Si verifica che le coppie della mappa “t” siano state aggiunte alla mappa di base e si verifica che i valori aventi chiave comune siano stati sovrascritti.
* **Design**: verifica che gli elementi della mappa passata come parametro vengano correttamente inseriti all’interno della mappa eventualmente sovrascrivendo il valore degli elementi che hanno la stessa chiave.
* **Pre-Condition**: un oggetto MapAdapter vuoto e correttamente inizializzato. Un parametro “t” diverso da “null”.
* **Post-Condition**: la mappa è formata da 9 coppie. Nessuna eccezione è stata generata.
* **Expected results**: la mappa contiene tutte le coppie della mappa t.

**16° PutAllExceptionsTest():**

* Test del metodo public void putAll(HMap t)
* **Summary / Description**: il lancio dell’eccezione viene prima verificato su una mappa vuota, viene poi inserita una coppia e si verifica nuovamente il corretto funzionamento del metodo. Viene poi verificato che il valore inserito in precedenza non abbia subito modifiche.
* **Design**: verifica il corretto lancio dell’eccezione NullPointerException sia quando la mappa è vuota sia quando contiene degli elementi.
* **Pre-Condition**: un oggetto MapAdapter vuoto e correttamente inizializzato.
* **Post-Condition**: la mappa contiene un elemento che non è stato modificato dal lancio delle eccezioni. Nessun altra eccezione è stata generata.
* **Expected results**: il corretto lancio di due NullPointerException quando il parametro passato è “null”.

**17° RemoveTest():**

* Test del metodo public Object remove(Object key)
* **Summary / Description**: la mappa è inizialmente vuota e si verifica che il metodo applicato ad un mappa vuota restituisca null. La mappa viene poi riempita con dei valori di riferimento e si verifica il corretto comportamento del metodo invocandolo sia con chiavi presenti nella mappa sia con chiavi non presenti. Il metodo viene poi invocato su tutte le chiavi rimanenti della mappa fino a svuotarla completamente verificando che i valori restituiti non siano effettivamente più presenti. Si verifica che la dimensione della mappa venga aggiornata correttamente.
* **Design**: applicare più volte il metodo su una mappa riempita con valori di riferimento per verificarne la correttezza.
* **Pre-Condition**: un oggetto MapAdapter vuoto e correttamente inizializzato. Un parametro “key” diverso da “null”.
* **Post-Condition**: la mappa torna ad essere vuota. Ulteriori invocazioni del metodo restituiranno il valore null.
* **Expected results**: il metodo ritorna un valore diverso da “null” se e solo se è stato un rimosso una coppia dalla mappa. La dimensione della mappa viene aggiornata correttamente.

**18° RemoveExceptionsTest():**

* Test del metodo public Object remove(Object key)
* **Summary / Description**: il lancio dell’eccezione viene prima verificato su una mappa vuota, viene poi inserita una coppia e si verifica nuovamente il corretto funzionamento del metodo. Viene poi verificato che il valore inserito in precedenza non abbia subito modifiche.
* **Design**: verifica il corretto lancio dell’eccezione NullPointerException sia quando la mappa è vuota sia quando contiene degli elementi.
* **Pre-Condition**: un oggetto MapAdapter vuoto e correttamente inizializzato.
* **Post-Condition**: la mappa contiene un elemento che non è stato modificato dal lancio delle eccezioni. Nessun altra eccezione è stata generata.
* **Expected results**: il corretto lancio di due NullPointerException quando il parametro passato è “null”.

**19° SizeTest():**

* Test del metodo public int size()
* **Summary / Description**: verifica che il metodo ritorni zero per una mappa vuota. Alla mappa vengono poi aggiunti e tolti degli elementi e si verifica il corretto comportamento del metodo. La mappa viene infine svuotata e si verifica che il metodo restituisca zero.
* **Design**: aggiunta e rimozione di elementi dalla mappa per verificare che il metodo descrive fedelmente il numero di elementi contenuti all’interno della mappa.
* **Pre-Condition**: un oggetto MapAdapter vuoto e correttamente inizializzato.
* **Post-Condition**: la mappa torna ad essere vuota. Ulteriori chiamate al metodo restituiscono sempre il valore zero.
* **Expected results**: il metodo restituisce un valore che rispecchia fedelmente il numero di elementi della mappa.

**20° ValuesTest():**

* Test del metodo public HCollection values()
* **Summary / Description**: verifica inizialmente il funzionamento del metodo su una mappa vuota. La mappa viene poi riempita e il metodo invocato nuovamente. Si verifica che la HCollection restituita abbia la stessa dimensione della mappa e che gli elementi al suo interno siano effettivamente dei valori della mappa come richiesto dall’interfaccia.
* **Design**: verifica che il metodo funzioni correttamente a prescindere dalle dimensioni della mappa su cui viene invocato.
* **Pre-Condition**: un oggetto MapAdapter vuoto e correttamente inizializzato.
* **Post-Condition**: la mappa e la collection vengono inizializzati con successo.
* **Expected results**: il metodo restituisce sempre una collection contenente i valori presenti nella mappa.

**1° AddIndexTest():**

* Test del metodo public void add(Object o)
* **Summary / Description**:
* **Design**:
* **Pre-Condition**: un oggetto MapAdapter vuoto e correttamente inizializzato.
* **Post-Condition**:
* **Expected results**:

**T**

**1° AddIndexTest():**

* Test del metodo public void add(Object o)
* **Summary / Description**:
* **Design**:
* **Pre-Condition**: un oggetto MapAdapter vuoto e correttamente inizializzato.
* **Post-Condition**:
* **Expected results**:

**T**

**1° AddIndexTest():**

* Test del metodo public void add(Object o)
* **Summary / Description**:
* **Design**:
* **Pre-Condition**: un oggetto MapAdapter vuoto e correttamente inizializzato.
* **Post-Condition**:
* **Expected results**:

**T**

**1° AddIndexTest():**

* Test del metodo public void add(Object o)
* **Summary / Description**:
* **Design**:
* **Pre-Condition**: un oggetto MapAdapter vuoto e correttamente inizializzato.
* **Post-Condition**:
* **Expected results**:

**T**

**1° AddIndexTest():**

* Test del metodo public void add(Object o)
* **Summary / Description**:
* **Design**:
* **Pre-Condition**: un oggetto MapAdapter vuoto e correttamente inizializzato.
* **Post-Condition**:
* **Expected results**:

**T**

**1° AddIndexTest():**

* Test del metodo public void add(Object o)
* **Summary / Description**:
* **Design**:
* **Pre-Condition**: un oggetto MapAdapter vuoto e correttamente inizializzato.
* **Post-Condition**:
* **Expected results**:

**T**

**1° AddIndexTest():**

* Test del metodo public void add(Object o)
* **Summary / Description**:
* **Design**:
* **Pre-Condition**: un oggetto MapAdapter vuoto e correttamente inizializzato.
* **Post-Condition**:
* **Expected results**:

**T**

**1° AddIndexTest():**

* Test del metodo public void add(Object o)
* **Summary / Description**:
* **Design**:
* **Pre-Condition**: un oggetto MapAdapter vuoto e correttamente inizializzato.
* **Post-Condition**:
* **Expected results**:

**T**

**1° AddIndexTest():**

* Test del metodo public void add(Object o)
* **Summary / Description**:
* **Design**:
* **Pre-Condition**: un oggetto MapAdapter vuoto e correttamente inizializzato.
* **Post-Condition**:
* **Expected results**:

**T**

**1° AddIndexTest():**

* Test del metodo public void add(Object o)
* **Summary / Description**:
* **Design**:
* **Pre-Condition**: un oggetto MapAdapter vuoto e correttamente inizializzato.
* **Post-Condition**:
* **Expected results**:

**T**

**1° AddIndexTest():**

* Test del metodo public void add(Object o)
* **Summary / Description**:
* **Design**:
* **Pre-Condition**: un oggetto MapAdapter vuoto e correttamente inizializzato.
* **Post-Condition**:
* **Expected results**:

**T**

**1° AddIndexTest():**

* Test del metodo public void add(Object o)
* **Summary / Description**:
* **Design**:
* **Pre-Condition**: un oggetto MapAdapter vuoto e correttamente inizializzato.
* **Post-Condition**:
* **Expected results**:

**T**

**1° AddIndexTest():**

* Test del metodo public void add(Object o)
* **Summary / Description**:
* **Design**:
* **Pre-Condition**: un oggetto MapAdapter vuoto e correttamente inizializzato.
* **Post-Condition**:
* **Expected results**:

**T**

**1° AddIndexTest():**

* Test del metodo public void add(Object o)
* **Summary / Description**:
* **Design**:
* **Pre-Condition**: un oggetto MapAdapter vuoto e correttamente inizializzato.
* **Post-Condition**:
* **Expected results**:

**T**

**1° AddIndexTest():**

* Test del metodo public void add(Object o)
* **Summary / Description**:
* **Design**:
* **Pre-Condition**: un oggetto MapAdapter vuoto e correttamente inizializzato.
* **Post-Condition**:
* **Expected results**:

**T**

**1° AddIndexTest():**

* Test del metodo public void add(Object o)
* **Summary / Description**:
* **Design**:
* **Pre-Condition**: un oggetto MapAdapter vuoto e correttamente inizializzato.
* **Post-Condition**:
* **Expected results**:

**T**

**1° AddIndexTest():**

* Test del metodo public void add(Object o)
* **Summary / Description**:
* **Design**:
* **Pre-Condition**: un oggetto MapAdapter vuoto e correttamente inizializzato.
* **Post-Condition**:
* **Expected results**:

**T**

**1° AddIndexTest():**

* Test del metodo public void add(Object o)
* **Summary / Description**:
* **Design**:
* **Pre-Condition**: un oggetto MapAdapter vuoto e correttamente inizializzato.
* **Post-Condition**:
* **Expected results**:

**T**

**1° AddIndexTest():**

* Test del metodo public void add(Object o)
* **Summary / Description**:
* **Design**:
* **Pre-Condition**: un oggetto MapAdapter vuoto e correttamente inizializzato.
* **Post-Condition**:
* **Expected results**:

**T**

**1° AddIndexTest():**

* Test del metodo public void add(Object o)
* **Summary / Description**:
* **Design**:
* **Pre-Condition**: un oggetto MapAdapter vuoto e correttamente inizializzato.
* **Post-Condition**:
* **Expected results**:

**T**

**1° AddIndexTest():**

* Test del metodo public void add(Object o)
* **Summary / Description**:
* **Design**:
* **Pre-Condition**: un oggetto MapAdapter vuoto e correttamente inizializzato.
* **Post-Condition**:
* **Expected results**:

**T**

**1° AddIndexTest():**

* Test del metodo public void add(Object o)
* **Summary / Description**:
* **Design**:
* **Pre-Condition**: un oggetto MapAdapter vuoto e correttamente inizializzato.
* **Post-Condition**:
* **Expected results**:

**T**

**1° AddIndexTest():**

* Test del metodo public void add(Object o)
* **Summary / Description**:
* **Design**:
* **Pre-Condition**: un oggetto MapAdapter vuoto e correttamente inizializzato.
* **Post-Condition**:
* **Expected results**:

**T**

**1° AddIndexTest():**

* Test del metodo public void add(Object o)
* **Summary / Description**:
* **Design**:
* **Pre-Condition**: un oggetto MapAdapter vuoto e correttamente inizializzato.
* **Post-Condition**:
* **Expected results**:

**T**

**1° AddIndexTest():**

* Test del metodo public void add(Object o)
* **Summary / Description**:
* **Design**:
* **Pre-Condition**: un oggetto MapAdapter vuoto e correttamente inizializzato.
* **Post-Condition**:
* **Expected results**:

**T**

**1° AddIndexTest():**

* Test del metodo public void add(Object o)
* **Summary / Description**:
* **Design**:
* **Pre-Condition**: un oggetto MapAdapter vuoto e correttamente inizializzato.
* **Post-Condition**:
* **Expected results**:

**T**

**1° AddIndexTest():**

* Test del metodo public void add(Object o)
* **Summary / Description**:
* **Design**:
* **Pre-Condition**: un oggetto MapAdapter vuoto e correttamente inizializzato.
* **Post-Condition**:
* **Expected results**:

**T**

**1° AddIndexTest():**

* Test del metodo public void add(Object o)
* **Summary / Description**:
* **Design**:
* **Pre-Condition**: un oggetto MapAdapter vuoto e correttamente inizializzato.
* **Post-Condition**:
* **Expected results**:

**T**

**1° AddIndexTest():**

* Test del metodo public void add(Object o)
* **Summary / Description**:
* **Design**:
* **Pre-Condition**: un oggetto MapAdapter vuoto e correttamente inizializzato.
* **Post-Condition**:
* **Expected results**:

**T**

**1° AddIndexTest():**

* Test del metodo public void add(Object o)
* **Summary / Description**:
* **Design**:
* **Pre-Condition**: un oggetto MapAdapter vuoto e correttamente inizializzato.
* **Post-Condition**:
* **Expected results**:

**T**

**1° AddIndexTest():**

* Test del metodo public void add(Object o)
* **Summary / Description**:
* **Design**:
* **Pre-Condition**: un oggetto MapAdapter vuoto e correttamente inizializzato.
* **Post-Condition**:
* **Expected results**:

**T**

**1° AddIndexTest():**

* Test del metodo public void add(Object o)
* **Summary / Description**:
* **Design**:
* **Pre-Condition**: un oggetto MapAdapter vuoto e correttamente inizializzato.
* **Post-Condition**:
* **Expected results**:

**T**

**1° AddIndexTest():**

* Test del metodo public void add(Object o)
* **Summary / Description**:
* **Design**:
* **Pre-Condition**: un oggetto MapAdapter vuoto e correttamente inizializzato.
* **Post-Condition**:
* **Expected results**:

**T**

**1° AddIndexTest():**

* Test del metodo public void add(Object o)
* **Summary / Description**:
* **Design**:
* **Pre-Condition**: un oggetto MapAdapter vuoto e correttamente inizializzato.
* **Post-Condition**:
* **Expected results**:

**T**

**1° AddIndexTest():**

* Test del metodo public void add(Object o)
* **Summary / Description**:
* **Design**:
* **Pre-Condition**: un oggetto MapAdapter vuoto e correttamente inizializzato.
* **Post-Condition**:
* **Expected results**:

**T**

**1° AddIndexTest():**

* Test del metodo public void add(Object o)
* **Summary / Description**:
* **Design**:
* **Pre-Condition**: un oggetto MapAdapter vuoto e correttamente inizializzato.
* **Post-Condition**:
* **Expected results**:

**T**

**1° AddIndexTest():**

* Test del metodo public void add(Object o)
* **Summary / Description**:
* **Design**:
* **Pre-Condition**: un oggetto MapAdapter vuoto e correttamente inizializzato.
* **Post-Condition**:
* **Expected results**:

**T**

**1° AddIndexTest():**

* Test del metodo public void add(Object o)
* **Summary / Description**:
* **Design**:
* **Pre-Condition**: un oggetto MapAdapter vuoto e correttamente inizializzato.
* **Post-Condition**:
* **Expected results**:

**T**

**1° AddIndexTest():**

* Test del metodo public void add(Object o)
* **Summary / Description**:
* **Design**:
* **Pre-Condition**: un oggetto MapAdapter vuoto e correttamente inizializzato.
* **Post-Condition**:
* **Expected results**:

**T**

**1° AddIndexTest():**

* Test del metodo public void add(Object o)
* **Summary / Description**:
* **Design**:
* **Pre-Condition**: un oggetto MapAdapter vuoto e correttamente inizializzato.
* **Post-Condition**:
* **Expected results**:

**T**

**1° AddIndexTest():**

* Test del metodo public void add(Object o)
* **Summary / Description**:
* **Design**:
* **Pre-Condition**: un oggetto MapAdapter vuoto e correttamente inizializzato.
* **Post-Condition**:
* **Expected results**:

**T**

**1° AddIndexTest():**

* Test del metodo public void add(Object o)
* **Summary / Description**:
* **Design**:
* **Pre-Condition**: un oggetto MapAdapter vuoto e correttamente inizializzato.
* **Post-Condition**:
* **Expected results**:

**T**

**1° AddIndexTest():**

* Test del metodo public void add(Object o)
* **Summary / Description**:
* **Design**:
* **Pre-Condition**: un oggetto MapAdapter vuoto e correttamente inizializzato.
* **Post-Condition**:
* **Expected results**: