

Report dei casi di test - Carloan

Test 001

Descrizione

Si testa la connessione al *DBMS MySQL* utilizzato dal software **Carloan**. Il test è strutturato in maniera tale da catturare l'eccezione **ConnessioneAlDataBaseException** sollevata dalla classe `MySQLConnector`. Se l'eccezione catturata mentre si effettua la connessione con il database dovesse risultare quella citata poc'anzi allora il test risulta corretto.

Dipendenze

Nessuna.

Stato Sistema

Quando viene eseguito il test il *processo* che avvia il DBMS MySQL può essere sia **attivo che disattivo**. In entrambi i casi avremo l'eccezione lanciata dalla classe sotto test poichè le *credenziali di accesso* al DBMS saranno errate.

Input

Credenziali di accesso al DBMS MySQL errate. Le credenziali corrette necessarie per effettuare l'accesso al DBMS sono:

- **Server path**, nel nostro caso: `jdbc:mysql://localhost/carloan`
- **Username**, nel nostro caso: `root`
- **Password**, nel nostro caso vi è una password vuota.

Note

Poichè l'eccezione lanciata dalla classe `MySQLConnector` viene attivata quando le credenziali sono errate, possiamo catturare la medesima eccezione e verificare l'assenza di connessione verso il DBMS anche quando il processo del software MySQL è terminato.

Procedura Esecuzione

Avvio del caso di test denominato: `MySQLConnectionTester.java`.

Valore Atteso

Si prevede che per passare correttamente il test di connessione numero `001`, l'eccezione **ConnessioneAlDataBaseException** venga catturata.

Valore Ottenuto

L'eccezione **ConnessioneAlDataBaseException** viene correttamente catturata durante la fase di connessione al database.

Impatti

Se tale test dovesse passare ciò vuol dire che i successivi test eseguiti sulle classi

`DAOAgenzia`, `DAOCliente`, `DAOFascia` risulteranno **falliti** poichè necessitano della presenza di connessione verso il *DBMS MySQL*.

Test 002

Descrizione

Viene testata la classe `DAOAgenzia.java`, in modo tale da controllare che il metodo per la **scrittura** delle istanze nel database funzioni correttamente.

Dipendenze

E' necessario che ci sia connessione al **DBMS MySQL** e che quindi il test numero `001` passi correttamente.

Stato Sistema

Il database MySQL deve essere attivo. **Non devono** essere presenti *istanze* nella tabella **Agenzie**. Al termine del test le istanze di prova vengono cancellate dal database rendendo lo stato del database *invariato*.

Input

Per effettuare il test sul metodo `create` della classe `DAOAgenzia.java`, ci occuperemo di creare un *oggetto* di tipo *Agenzia* contenente i seguenti campi:

- Nome: `Carloan SRL`
- Numero di telefono: `0805412359`
- Indirizzo: `Via Amendola 184`

Passeremo successivamente l'oggetto creato al metodo da testare.

Note

Poichè utilizzeremo la create per testare anche altri casi di test come ad esempio l'eliminazione o l'aggiornamento dal database, inseriremo le istanze nel database nel metodo `setUp` messo a disposizione dalla libreria **Junit**. Successivamente controlliamo che il valore restituito dalla create sia effettivamente un numero che corrisponde all'**id** dell'istanza nella tabella MySQL.

Procedura Esecuzione

L'applicazione MySQL deve essere attiva e devono essere presente la tabella agenzia vuota all'interno del database. Può poi essere eseguito il caso di test.

Valore Atteso

Ci attendiamo dal test che l'istanza di prova venga correttamente scritta nel database con i campi inseriti come input.

Valore Ottenuto

Controlliamo che il metodo testato per la creazione delle istanze nel database restituisca una **stringa** contenente l'id dell'istanza all'interno del database.

Impatti

Il risultato positivo di tale test ci consente di testare altri metodi messi a disposizione della classe come: `read, readAll, update, delete`. Se tale test dovesse fallire ciò vuol dire che le istanze nel database non sono state create e non possiamo eseguire ulteriori operazioni.

Test 003

Descrizione

Viene testata la classe `DAOAgenzia.java`, in modo tale da controllare che il **metodo** per la **scrittura** delle istanze nel database funzioni correttamente.

Dipendenze

E' necessario che ci sia connessione al **DBMS MySQL** e che quindi il test numero `001` passi correttamente.

Stato Sistema

Il database MySQL deve essere attivo. **Devono** essere presenti *istanze* nella tabella **Agenzie**. Al termine del test le istanze di prova vengono cancellate dal database rendendo lo stato del database *invariato*.

Input

Per effettuare il test sul metodo `create` della classe `DAOAgenzia.java`, ci occuperemo di creare un oggetto di tipo *Agenzia* contenente i seguenti campi:

- Nome: `Carloan SRL`
- Numero di telefono: `0805412359`
- Indirizzo: `Via Amendola 184`

Passeremo successivamente l'oggetto creato al metodo da testare.

Note

Poichè utilizzeremo la `create` per testare anche altri casi di test come ad esempio l'eliminazione o l'aggiornamento dal database, inseriremo le istanze nel database nel metodo `setUp` messo a disposizione dalla libreria **Junit**. Successivamente controlliamo che il valore restituito dalla `create` sia effettivamente un numero che corrisponde all'**id** dell'istanza nella tabella MySQL.

Procedura Esecuzione

L'applicazione MySQL deve essere attiva e devono essere pressente la tabella *agenzia con delle istanze già presenti* all'interno del database. Può poi essere eseguito il caso di test.

Valore Atteso

Ci attendiamo dal test che l'istanza di prova venga correttamente scritta nel database con i campi inseriti come input.

Valore Ottenuto

Controlliamo che il metodo testato per la creazione delle istanze nel database restituisca una **stringa** contenente l'id dell'istanza all'interno del database.

Impatti

Il risultato positivo di tale test ci consente di testare altri metodi messi a disposizione della classe come: `read`, `readAll`, `update`, `delete`. Se tale test dovesse fallire ciò vuol dire che le istanze nel database non sono state create e non possiamo eseguire ulteriori operazioni.

Test 004

Descrizione

Viene testata la classe `DAOAgenzia.java`, in modo tale da controllare che il metodo per la **lettura** delle istanze nel database funzioni correttamente.

Dipendenze

E' necessario che ci sia connessione al **DBMS MySQL** e che quindi il test numero `001` passi correttamente. Questo test necessita che sia stata correttamente scritta l'**istanza di prova** all'interno del database e che quindi il test numero `002` passi correttamente.

Stato Sistema

Il database MySQL deve essere attivo. Non devono essere presenti ulteriori *istanze* nella tabella **Agenzie** se non quelle create dal metodo `create` nel test numero `002`. Al termine del test le istanze di prova vengono cancellate dal database rendendo lo stato del database *invariato*.

Input

Per effettuare il test sul metodo `read` della classe `DAOAgenzia.java`, ci occuperemo di specificare l'**id** dell'istanza di prova creata nel metodo `setUp` del test case.

Note

Nessuna nota.

Procedura Esecuzione

L'applicazione che fornisce accesso al database MySQL deve essere eseguita. Può quindi essere eseguito il caso di test.

Valore Atteso

Ci attendiamo dal test che l'istanza di prova scritta precedentemente nel database, venga correttamente letta utilizzando il metodo di lettura presente all'interno della classe da testare.

Valore Ottenuto

Controlliamo che il metodo testato per la lettura delle istanze nel database restituisca i tre campi dell'istanza di prova creata precedentemente a questo test. I tre campi che otteniamo sono:

- Nome dell'agenzia.
- Numero di telefono dell'agenzia.
- Indirizzo dell'agenzia.

Impatti

L'esito positivo di questo test non impatta su alcun altro caso di test.

Test 005

Descrizione

Viene testata la classe `DAOAgenzia.java`, in modo tale da controllare che il metodo per la **lettura** di tutte le istanze della tabella *Agenzie* all'interno del database funzioni correttamente.

Dipendenze

E' necessario che ci sia connessione al **DBMS MySQL** e che quindi il test numero `001` passi correttamente. Questo test necessita che sia stata correttamente scritta l'**istanza di prova** all'interno del database e che quindi il test numero `002` passi correttamente.

Stato Sistema

Il database MySQL deve essere attivo. Non devono essere presenti ulteriori *istanze* nella tabella **Agenzie** se non quelle create dal metodo create nel test numero `002`. Al termine del test le istanze di prova vengono cancellate dal database rendendo lo stato del database *invariato*.

Input

Per effettuare il test sul metodo `readAll` della classe `DAOAgenzia.java` non abbiamo bisogno di input esterni.

Note

Il test funziona in modo tale che una volta scritta l'istanza di prova nel metodo `setUp` ci occupiamo poi di prelevare tutti i valori all'interno della tabella *Agenzie*. Avendo tutti i valori all'interno di una lista di oggetti di tipo Agenzia, andiamo a leggere l'ultimo valore della lista che corrisponde esattamente all'ultima istanza scritta nella tabella. Tale istanza quindi risulterà quella inserita in fase di test.

Procedura Esecuzione

L'applicazione che fornisce accesso al database MySQL deve essere eseguita. Può quindi essere eseguito

il caso di test.

Valore Atteso

Ci attendiamo dal test che l'istanza di prova scritta precedentemente nel database, venga correttamente letta utilizzando il metodo di lettura di tutte le istanze nella tabella *Agenzie*, presente all'interno della classe da testare.

Valore Ottenuto

Otteniamo una lista di oggetti di tipo *Agenzia*. Andando successivamente a leggere l'ultimo valore della lista (che corrisponde all'istanza di prova creata nel caso di test), controlliamo che ci siano i tre campi dell'istanza di prova. I tre campi che otteniamo sono:

- Nome dell'agenzia.
- Numero di telefono dell'agenzia.
- Indirizzo dell'agenzia.

Impatti

L'esito positivo di questo test non impatta su alcun altro caso di test.

Test 006

Descrizione

Viene testata la classe `DAOAgenzia.java`, in modo tale da controllare che il metodo per l'**aggiornamento** dell'istanze di prova della tabella *Agenzie* all'interno del database funzioni correttamente.

Dipendenze

E' necessario che ci sia connessione al **DBMS MySQL** e che quindi il test numero `001` passi correttamente. Questo test necessita che sia stata correttamente scritta l'**istanza di prova** all'interno del database e che quindi il test numero `002` passi correttamente.

Stato Sistema

Il database MySQL deve essere attivo. Non devono essere presenti ulteriori *istanze* nella tabella **Agenzie** se non quelle create dal metodo create nel test numero `002`. Al termine del test le istanze di prova vengono cancellate dal database rendendo lo stato del database *invariato*.

Input

Per effettuare il test sul metodo `update` della classe `DAOAgenzia.java`, ci occuperemo di specificare l'**id** dell'istanza di prova creata nel metodo `setUp` del test case. Viene successivamente creato un oggetto temporaneo di tipo *Agenzia* che contiene i campi modificati dell'istanza da aggiornare. I campi possono essere:

- Nome
- Numero di telefono
- Indirizzo

Note

All'interno di questo test case c'è anche il metodo di lettura dalla classe dao per controllare se l'istanza modificata con il metodo di **aggiornamento** ha correttamente aggiornato i dati nel database.

Procedura Esecuzione

L'applicazione che fornisce accesso al database MySQL deve essere eseguita. Può quindi essere eseguito il caso di test.

Valore Atteso

Ci attendiamo dal test che l'istanza di prova scritta precedentemente nel database, venga correttamente aggiornata utilizzando il metodo di aggiornamento `update`, presente all'interno della classe da testare.

Valore Ottenuto

Otteniamo un oggetto temporaneo di tipo *Agenzia*. Controlliamo successivamente con il metodo di lettura che i campi aggiornati con `update` siano stati correttamente modificati. I tre campi che otteniamo sono:

- Nome dell'agenzia.
- Numero di telefono dell'agenzia.
- Indirizzo dell'agenzia.

Impatti

L'esito positivo di questo test non impatta su alcun altro caso di test.

Test 007

Descrizione

Viene testata la classe `DAOAgenzia.java`, in modo tale da controllare che il metodo per l'**eliminazione**

dell'istanze di prova della tabella *Agenzie* all'interno del database funzioni correttamente.

Dipendenze

E' necessario che ci sia connessione al **DBMS MySQL** e che quindi il test numero `001` passi correttamente. Questo test necessita che sia stata correttamente scritta l'**istanza di prova** all'interno del database e che quindi il test numero `002` passi correttamente.

Stato Sistema

Il database MySQL deve essere attivo. Non devono essere presenti ulteriori *istanze* nella tabella **Agenzie** se non quelle create dal metodo `create` nel test numero `002`. Al termine del test le istanze di prova vengono cancellate dal database rendendo lo stato del database *invariato*.

Input

Per effettuare il test sul metodo `delete` della classe `DAOAgenzia.java`, ci occuperemo di specificare l'**id** dell'istanza di prova creata nel metodo `setUp` del test case.

Note

Poichè alla fine di ogni test eseguendo il metodo `tearDown` messo a disposizione dal framework **Junit**, eliminiamo l'istanza di prova creata nel metodo `setUp`, ci occupiamo di controllare mediante un *flag* che il metodo di eliminazione sia già stato testato. In questa maniera quando il framework eseguirà il metodo `tearDown` non eliminiamo ulteriormente le istanze dal database.

Procedura Esecuzione

L'applicazione che fornisce accesso al database MySQL deve essere eseguita. Può quindi essere eseguito il caso di test.

Valore Atteso

Ci attendiamo dal test che l'istanza di prova scritta precedentemente nel database, venga correttamente eliminata utilizzando il metodo di aggiornamento `delete`, presente all'interno della classe da testare.

Valore Ottenuto

Controlliamo che dopo l'eliminazione dell'istanza di prova, effettuando una lettura completa delle istanze nella tabella **Agenzia**, non si apresente quella che abbiamo eliminato durante il test.

Impatti

L'esito positivo di questo test non impatta su alcun altro caso di test.

Test 008

Descrizione

Si testa il metodo `create` messo a disposizione della classe `DAOAgenzia.java`. Il caso di test è strutturato in maniera tale da creare una nuova istanza all'interno del database che contenga valori *uguali* ad una istanza già creata precedentemente. In questa maniera verrà sollevata una eccezione di tipo **CampoDuplicatoException**.

Dipendenze

E' necessario che ci sia connessione al **DBMS MySQL** e che quindi il test numero `001` passi correttamente. Questo test necessita che sia stata correttamente scritta l'**istanza di prova** all'interno del database e che quindi il test numero `002` passi correttamente.

Stato Sistema

Il database MySQL deve essere attivo. Non devono essere presenti ulteriori *istanze* nella tabella **Agenzie** se non quelle create dal metodo `create` nel test numero `002`. Al termine del test le istanze di prova vengono cancellate dal database rendendo lo stato del database *invariato*.

Input

Per effettuare il test sul metodo `create` della classe `DAOAgenzia.java`, ci occuperemo di creare un *oggetto* di tipo *Agenzia* contenente i seguenti campi:

- Nome: `Carloan SRL`
- Numero di telefono: `0805412359`
- Indirizzo: `Via Amendola 184`

Passeremo successivamente l'oggetto creato al metodo da testare. Poichè i valori corrispondono esattamente ai valori inseriti nella fase di `setUp` di **Junit** avremo un duplicato all'interno del database e questo ci verrà segnalato mediante una eccezione.

Note

Non sono presenti note.

Procedura Esecuzione

L'applicazione che fornisce accesso al database MySQL deve essere eseguita. Può quindi essere eseguito il caso di test.

Valore Atteso

Attendiamo che l'eccezione di tipo **CampoDuplicatoException** venga correttamente sollevata in fase di creazione dell'istanza.

Valore Ottenuto

Otteniamo una eccezione che verrà poi confrontata con l'eccezione **CampoDuplicatoException**. Se l'eccezione corrisponde a quella attesa allora il test passerà correttamente.

Impatti

L'esito positivo di questo test non impatta su alcun altro caso di test.

Test 009

Descrizione

Si testa il metodo `update` messo a disposizione della classe `DAOAgenzia.java`. Il caso di test è strutturato in maniera tale da creare una nuova istanza all'interno del database che contenga valori *uguali* ad una istanza già creata precedentemente. In questa maniera verrà sollevata una eccezione di tipo **CampoDuplicatoException**.

Dipendenze

E' necessario che ci sia connessione al **DBMS MySQL** e che quindi il test numero `001` passi correttamente. Questo test necessita che sia stata correttamente scritta l'**istanza di prova** all'interno del database e che quindi il test numero `002` passi correttamente.

Stato Sistema

Il database MySQL deve essere attivo. Non devono essere presenti ulteriori *istanze* nella tabella **Agenzie** se non quelle create dal metodo create nel test numero `002`. Al termine del test le istanze di prova vengono cancellate dal database rendendo lo stato del database *invariato*.

Input

Per effettuare il test sul metodo `update` della classe `DAOAgenzia.java`, ci occuperemo di specificare l'**id** dell'istanza da aggiornare e successivamente creare un *oggetto* di tipo *Agenzia* contenente i

seguenti campi:

- Nome: `Carloan 2 SRL`
- Numero di telefono: `0805412359`
- Indirizzo: `Via Amendola 184`

Passeremo successivamente l'oggetto creato al metodo da testare. Poichè i valori dell'istanza nella `setUp` corrispondono esattamente ad altri valori di prova inseriti in questo caso di test avremo un duplicato all'interno del database e questo ci verrà segnalato mediante una eccezione.

Note

Non sono presenti note.

Procedura Esecuzione

L'applicazione che fornisce accesso al database MySQL deve essere eseguita. Può quindi essere eseguito il caso di test.

Valore Atteso

Attendiamo che l'eccezione di tipo **CampoDuplicatoException** venga correttamente sollevata in fase di aggiornamento dell'istanza.

Valore Ottenuto

Otteniamo una eccezione che verrà poi confrontata con l'eccezione **CampoDuplicatoException**. Se l'eccezione corrisponde a quella attesa allora il test passerà correttamente.

Impatti

L'esito positivo di questo test non impatta su alcun altro caso di test.

Riguardo il report

In questo report sono stati riportati i casi di test per la classe dao: **DAOAgenzia**. Analogamente possiamo verificare gli stessi test anche per le altre classi dao non riportate in questo documento, quali:

- DAOCliente
- DAOFascia

Documento realizzato da *Simone Marzulli e Morena De Liddo*.

