



UNIVERSITÀ DEGLI STUDI DI SALERNO
DIPARTIMENTO DI INFORMATICA
INGEGNERIA DEL SOFTWARE 2017-2018

Documento di Testing di Unità 1.0

Easybook

Data: 10/01/2018

Sommario

1.PARTECIPANTI:	2
2.REVISION HISTORY:	2
3.INTRODUZIONE:	2
4.DETTAGLI PER IL SEGUENTE LIVELLO DI TEST PLAN:.....	2

1.PARTECIPANTI:

Nome	Matricola
Andrea Di Lucia	0512103222

2.REVISION HISTORY:

Data	Versione	Descrizione	Autore
10/01/2018	1.0	Completamento documento	Andrea Di Lucia

3.INTRODUZIONE:

Lo scopo di questo documento è quello di riportare i risultati dell'esecuzione dei test case di unità dei bean di EasyBook. Tutti i test sono stati realizzati utilizzando JUnit e il debugger fornito dalla console di Eclipse per verificare eventuali failure.

1.1 SCOPO

Lo scopo del documento è quello di fornire una presentazione dettagliata della pianificazione del test di unità per il software di EasyBook. Nel test si verificano le singole unità di una classe e si controlla che i metodi e le classi abbiano il comportamento atteso..

1.2 RIFERIMENTI

Per realizzare il progetto, sono stati utilizzati:

- slide del docente;
- B. BRUEGGE, A.H. DUTOIT, OBJECT ORIENTED SOFTWARE ENGINEERING – USING UML, PATTERNS AND JAVA, PRENTICE HALL, 3D EDITION, 2009;
- documentazione di EasyBook.

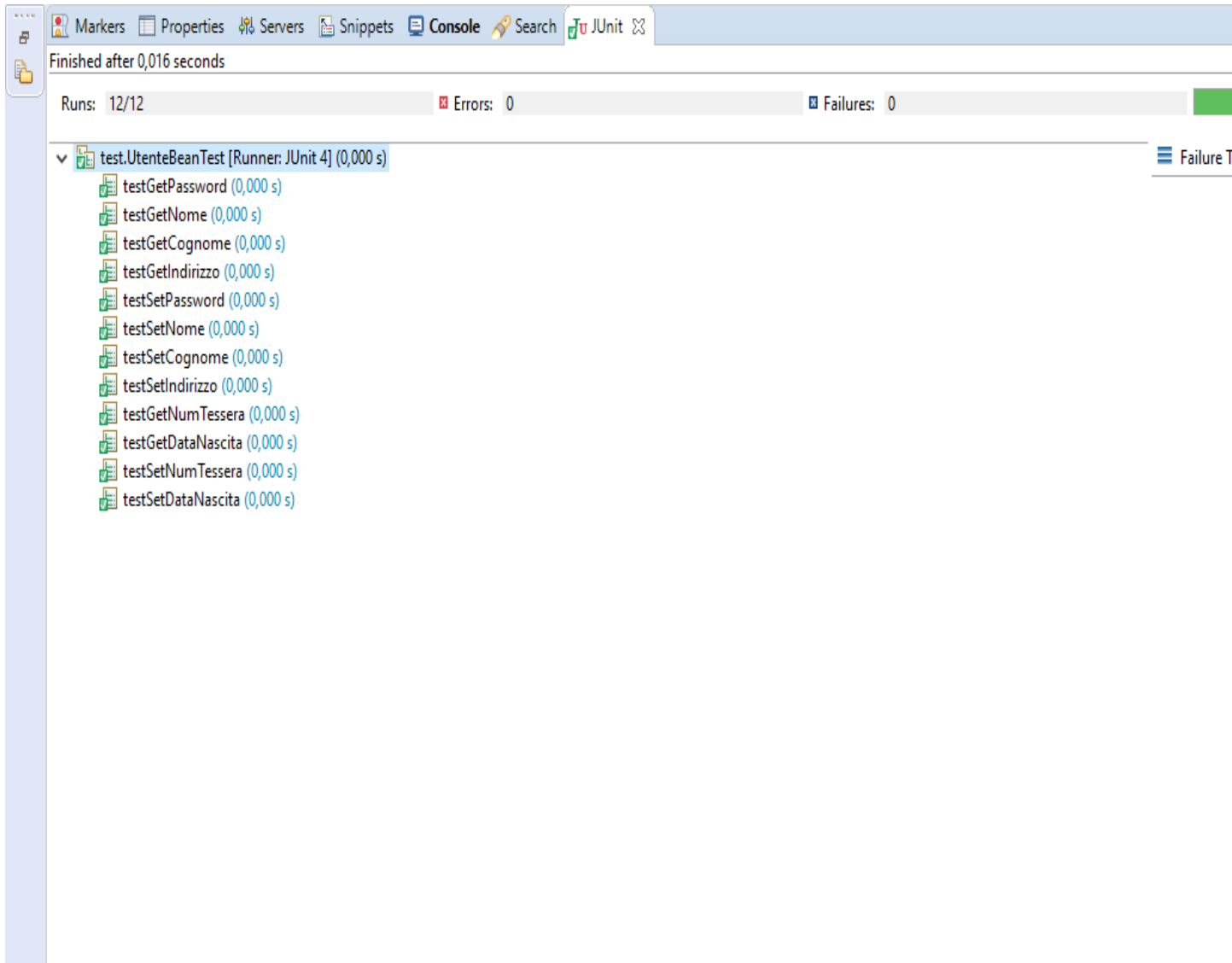
4.DETTAGLI PER IL SEGUENTE LIVELLO DI TEST PLAN:

In questa sezione del documento verranno forniti i dettagli relativi alle funzionalità ed agli strumenti utilizzati per eseguire il test di unità.

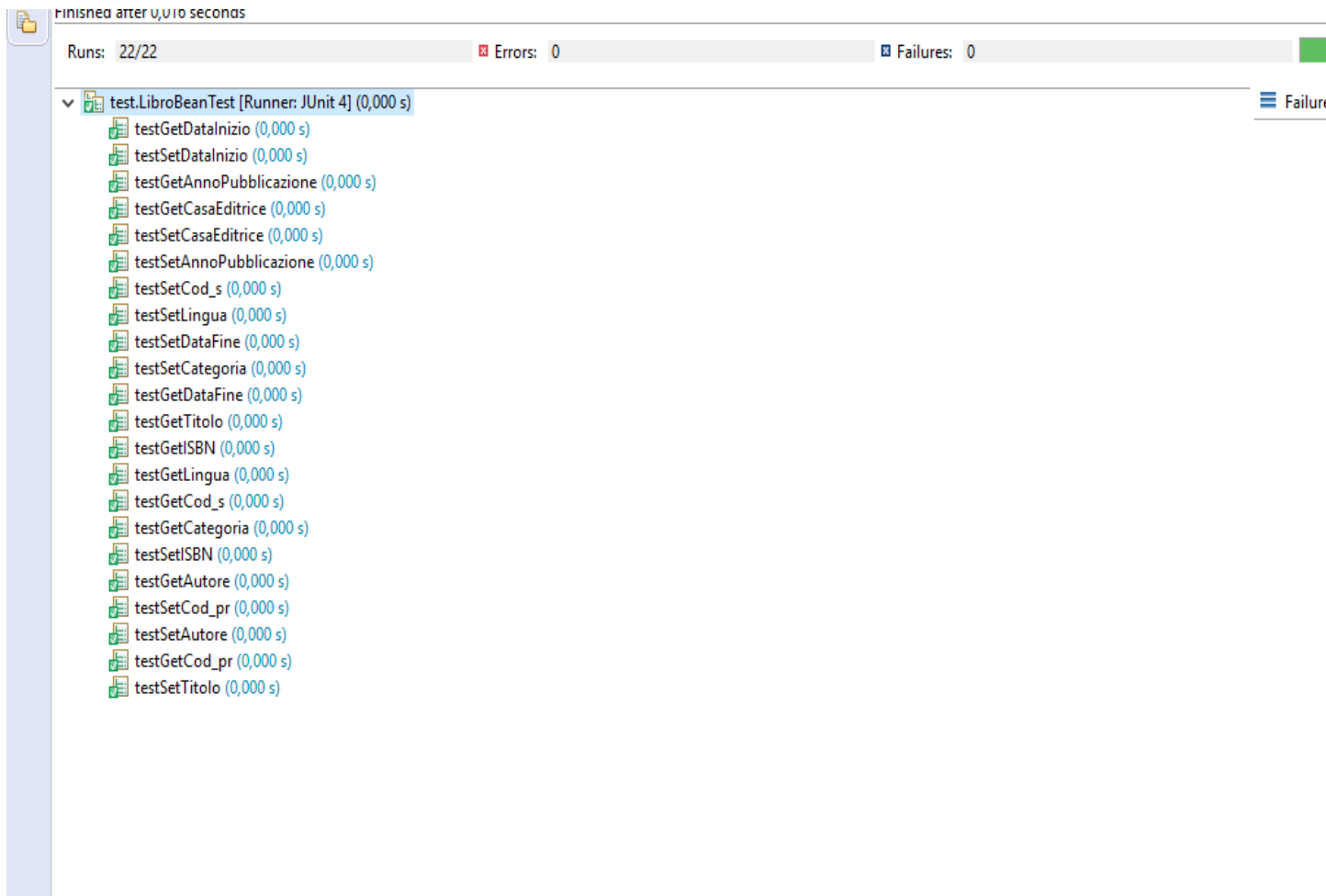
Classi testate:

- utentiPackage: UtenteBean.java;
- libriPackage: LibroBean.java;
- carrelloPackage: CarrelloBean.java.

4.1 UtenteBean.java



4.2 LibroBean.java

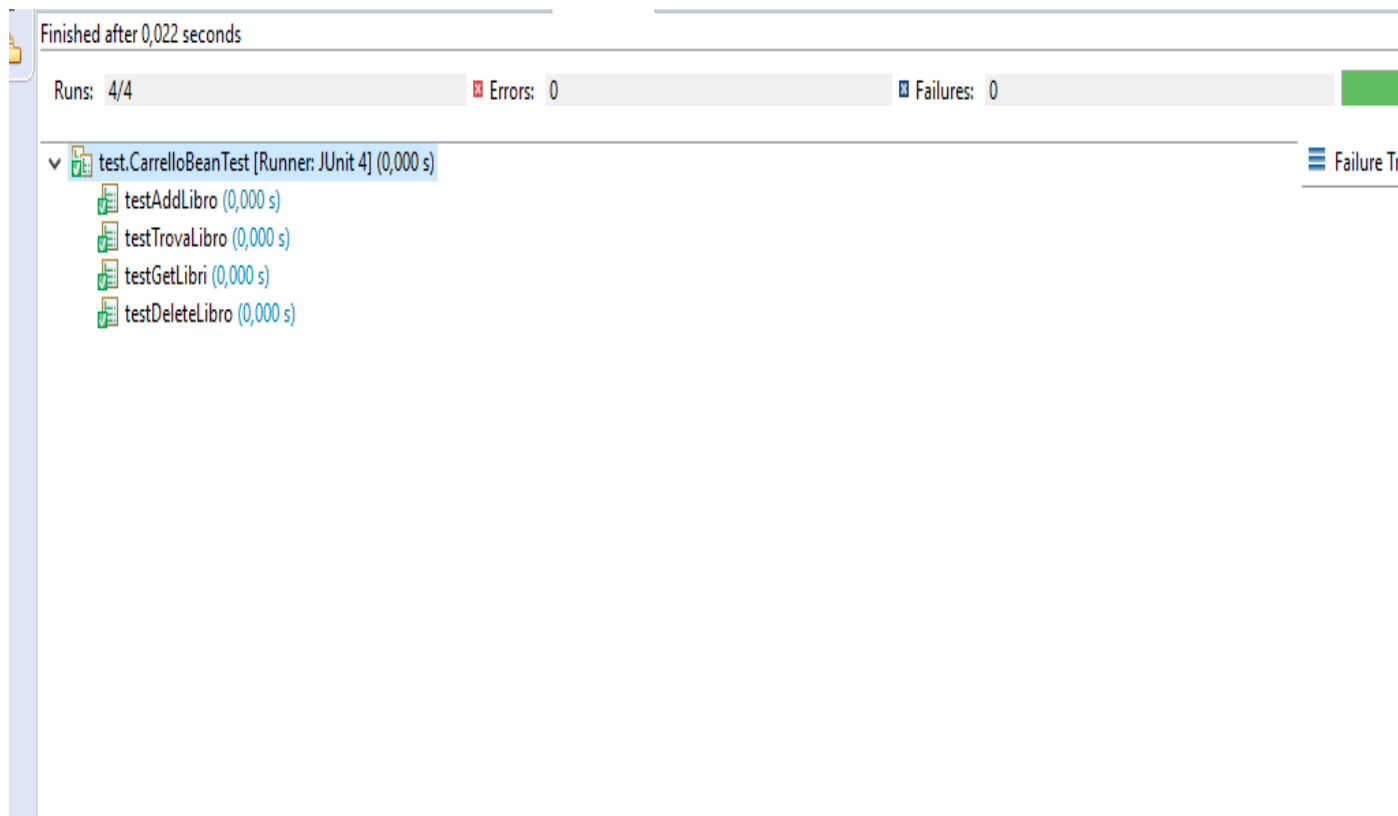


The image shows a screenshot of a JUnit test runner interface. At the top, a status bar indicates "Finished after 0,010 seconds". Below this, a summary bar shows "Runs: 22/22", "Errors: 0", and "Failures: 0". The main area displays a list of test methods under the package "test.LibroBeanTest [Runner: JUnit 4] (0,000 s)". Each method is preceded by a green checkmark icon, indicating successful execution. The methods listed are:

- testGetDataInizio (0,000 s)
- testSetDataInizio (0,000 s)
- testGetAnnoPubblicazione (0,000 s)
- testGetCasaEditrice (0,000 s)
- testSetCasaEditrice (0,000 s)
- testSetAnnoPubblicazione (0,000 s)
- testSetCod_s (0,000 s)
- testSetLingua (0,000 s)
- testSetDataFine (0,000 s)
- testSetCategoria (0,000 s)
- testGetDataFine (0,000 s)
- testGetTitolo (0,000 s)
- testGetISBN (0,000 s)
- testGetLingua (0,000 s)
- testGetCod_s (0,000 s)
- testGetCategoria (0,000 s)
- testSetISBN (0,000 s)
- testGetAutore (0,000 s)
- testSetCod_pr (0,000 s)
- testSetAutore (0,000 s)
- testGetCod_pr (0,000 s)
- testSetTitolo (0,000 s)

On the right side of the interface, there is a partially visible label "Failure" next to a list icon.

2.3 CarrelloBean.java



In conclusione, possiamo dire che le nostre classi testate non hanno evidenziato problemi.