

**INGEGNERIA DEL SOFTWARE 2016-2017**

**15/01/2017**

**Documento di Testing di Unità 1.0**

**Fiorazon**

**15/12/2016**

**Sommario**

[**1.** **Introduction** 2](#_Toc472203771)

[**2.** **Dettagli per il seguente livello di test plan** 3](#_Toc472203772)

[**3.** **Glossario** 7](#_Toc472203773)

[**4.** **Team** 8](#_Toc472203774)

[**5.** **Storia delle versioni** 8](#_Toc472203775)

### **Introduction**

Lo scopo di questo documento è quello di riportare i risultati dell’esecuzione dei test case di unità dei bean di Fiorazon. 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 Fiorazon. 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 Fiorazon.

### **Dettagli per il seguente livello di test plan**

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

Classi testate:

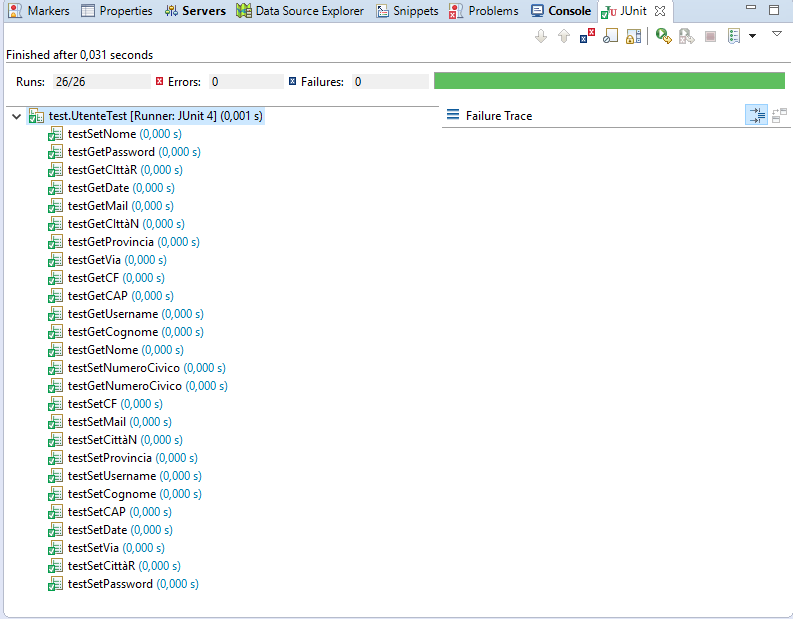
- utentiPackage: Utente.java;

- ordinePackage: Ordine.java;

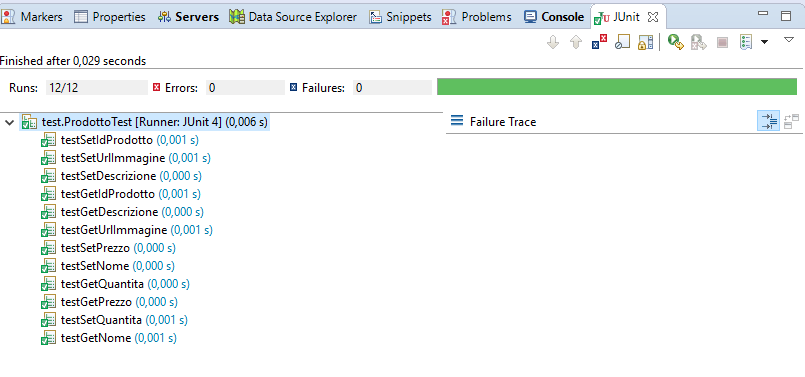
- prodottiPackage: Prodotto.java;

- carrelloPackage: Carrello.java.

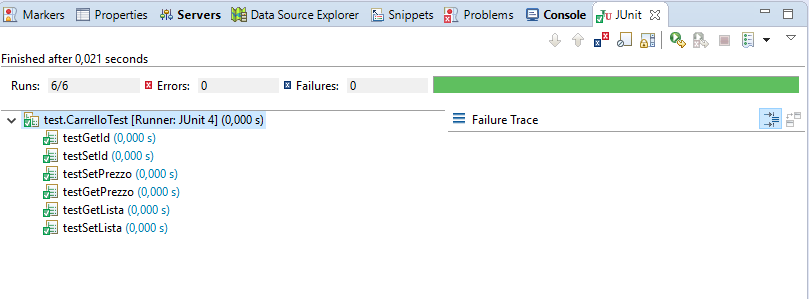
2.1 Utente.java



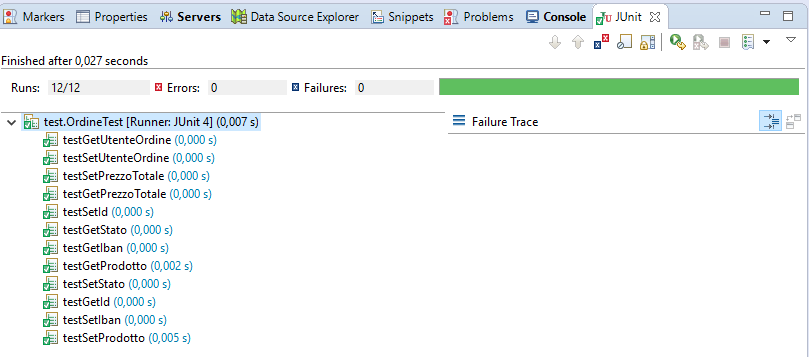
2.2 Prodotto.java



2.3 Carrello.java



2.4 Ordine.java



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

### **Glossario**

**Acronimi e abbreviazioni**

|  |  |
| --- | --- |
| **Acronimo** | **Definizione** |
| DB | Database (sistema di memorizzazione per l’archiviazione dei dati permanenti) |
| DBMS | Database Management System (gestore del sistema di memorizzazione) |
| HTML | HyperText Markup Language (linguaggio di markup utilizzato per la definizione della struttura della pagina web) |
| HTTP | HyperTextTransferProtocol (Protocollo per la gestione di richieste e risposte scambiate tra client e server) |
| JAVA | Linguaggio orientato agli oggetti |
| JAVASCRIPT | Linguaggio di scripting utilizzato lato client per rendere il portale web dinamico |
| Layout | Impaginazione struttura grafica del portale e dell'applicazione |
| MYSQL | DBMS utilizzato |
| SQL | Structured Query Language (linguaggio per l'invocazione delle richieste al database) |
| Query | Interrogazione al database |
| RAD | Requirementes Analysis Document |
| SDD | System Design Document |
| JUNIT | è un framework di unit testing per il linguaggio di programmazione Java. |

**Definizioni**

**Bean**: componente software riusabile, usata per rappresentare le informazioni di interesse.  
 **Sito di e-commerce**: sito dedicato all’attività di vendita e acquisto di prodotti tramite Internet.

**Database**: sistema che memorizza i dati.

**Utente**: è un utente generico che ha accesso al sito, e può essere “registrato” o “non registrato”.

**Utente registrato**: è un utente che può acquistare prodotti dal catalogo.

**Utente non registrato**: può solamente visualizzare informazioni, ma non può acquistare.

**Utente loggato**: utente che ha effettuato il login.

**Utente non loggato:** utente che non ha effettuato il login.

**Amministratore**: è il gestore del negozio. Può modificare il catalogo e gestire gli ordini.

**Amministratore loggato**: amministratore che ha effettuato il login.

**Form**: è una “scheda di compilazione” per l’inserimento dei dati, dove l’utente inserisce e invia i dati al server.

**Login**: è la procedura di accesso (procedura di autenticazione) ad un sistema informatico o ad un’applicazione informatica con le credenziali registrate al sito web.

**Logout**: è la procedura di uscita da un sistema informatico o un’applicazione informatica.

**Password**: è una parola chiave con cui l’utente accede al sito web.

**Registrazione**: è la procedura di salvataggio dei dati dell’utente nel database con i quali può procedere agli acquisti.

**Carrello**: oggetto contente i prodotti che l’utente vuole acquistare.

**Ordine**: ordine effettuato dall’utente dopo l’acquisto del carrello.

**Catalogo**: è l’insieme dei prodotti disponibili.  
**Package:** collezione di classi e interfacce correlate.

**Classe:** è identificabile come un tipo di dato astratto.

**Interfaccia:** è un tipo analogo alla classe, ma vincolato a non definire l'implementazione dei propri metodi.

**Metodo:** è un sottoprogramma associato in modo esclusivo ad una classe e che rappresenta.

**Variabile:** è un insieme di dati modificabili situati in una porzione di memoria destinata a contenere dei dati **.**

**Costante:** identifica una porzione di memoria il cui valore non varia nel corso dell'esecuzione di un programma.

### **Team**

|  |  |
| --- | --- |
| **Nome** | **Matricola** |
| Giuseppe Siani | 0512102958 |
| Carmine D’ Alessandro | 0512103402 |
| Aleandro Giuseppe Libano | 0512102976 |

### **Storia delle versioni**

|  |  |  |  |
| --- | --- | --- | --- |
| DATA | VERSIONE | DESCRIZIONE | AUTORE |
| 15/12/2016 | 1.0 | Completamento documento | Aleandro Giuseppe Libano |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |