

# Università degli Studi di Salerno

Corso di Ingegneria del Software

## **Auto Shop Unit Test Report Versione 1.0**



Data: 01/02/2018

| Cognome Nome            | Matricola  |
|-------------------------|------------|
| Corrado Mancino Alfredo | 0512102506 |

|                |            |
|----------------|------------|
| Carbè Daniele  | 0512102326 |
| Caloia Gennaro | 0512102332 |

## **Indice**

### 1. Introduzione

#### 1.1 Scopo

#### 1.2 Riferimenti

### 2. Dettagli

#### 2.1 Overview test results

#### 2.2 Dettagli test results

##### 2.2.1 Classi Entity it.unisa.docs.data

##### 2.2.2 Classi DB it.unisa.docs.storage

#### 2.3 Motivazioni delle decisioni

#### 2.4 Conclusioni

# 1. Introduzione

## 1.1 Scopo

Lo scopo è quello di riportare l'esecuzione dei test case d'unità precedentemente pianificati e definiti nel documento di UTP.

## 1.2 Riferimenti

### Riferimenti interni:

- RAD;
- SDD;
- ODD;
- UTP.

## 2. Dettagli

### 2.1 Overview test results

Dopo le correzioni della versione 1.0 non sono state trovate fault nelle componenti del sistema.

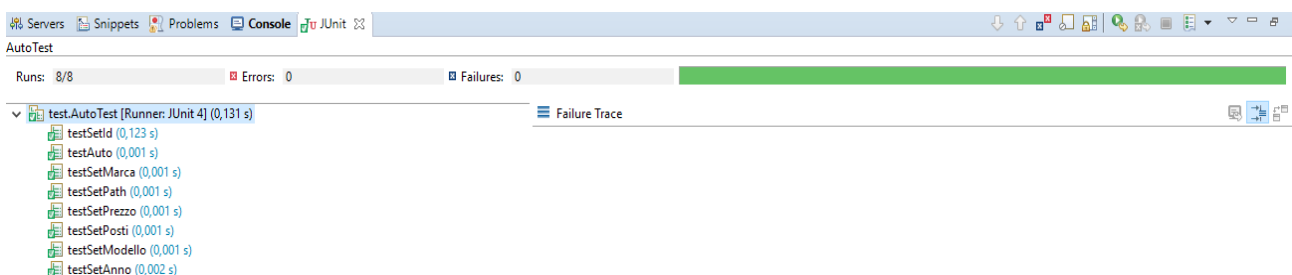
Per informazioni dettagliate sugli item testati e sui risultati si faccia riferimento al paragrafo successivo.

### 2.2 Dettagli test results

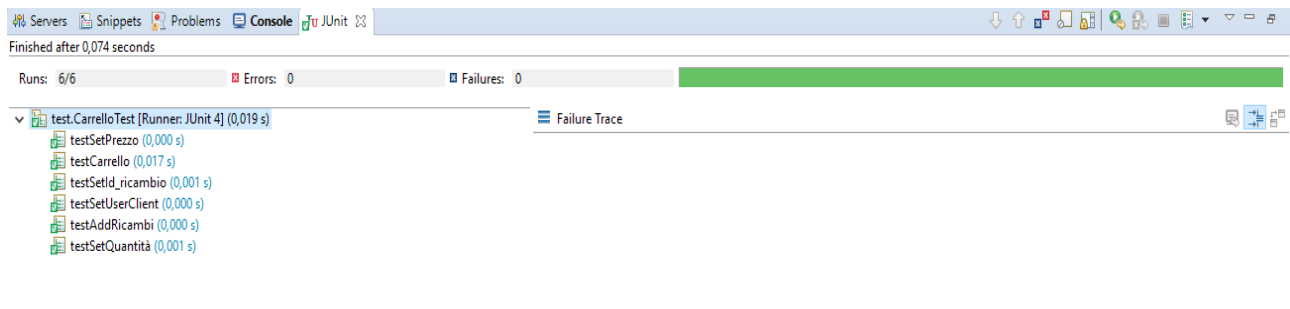
#### 2.2.1 Classi Entity model

Di seguito sono riportati tutti i test effettuati per le classi 'data' del sistema tramite testing JUNIT.

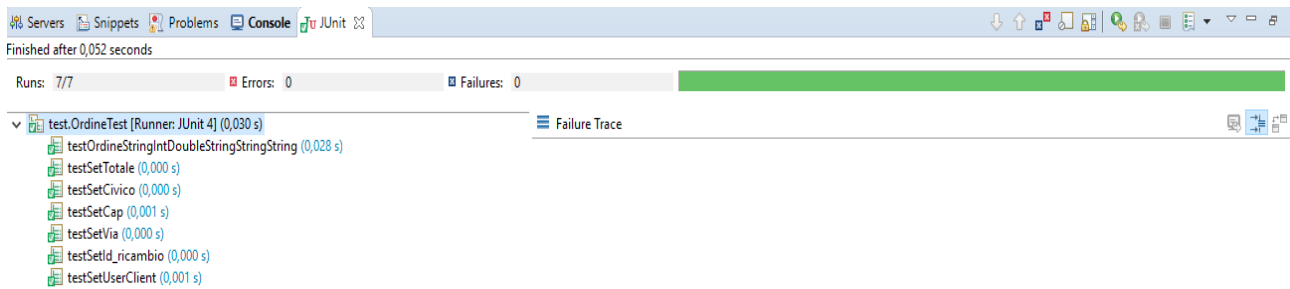
#### 1. Classe Auto



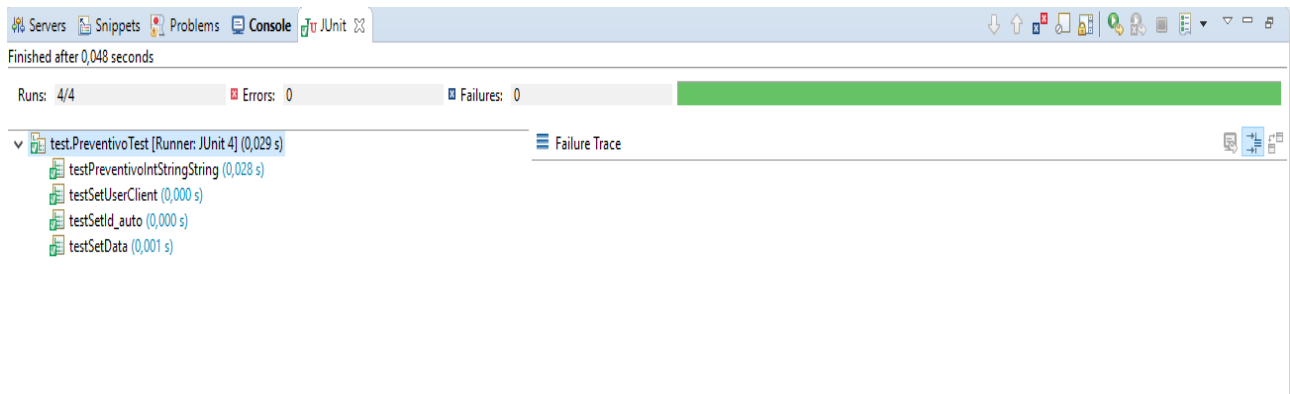
#### 2. Classe Carrello



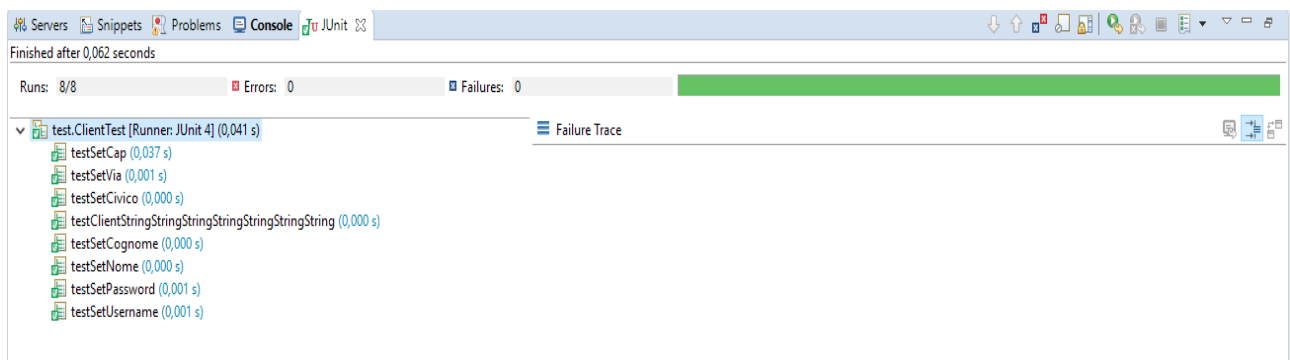
### 3. Classe Ordine



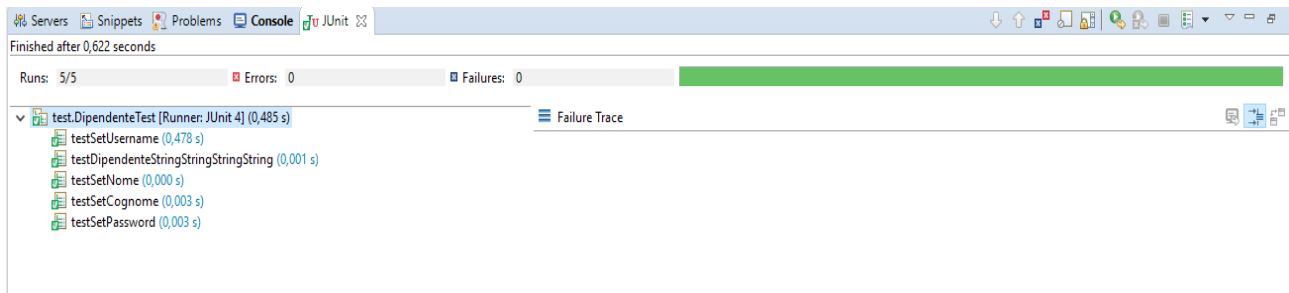
### 4. Classe Preventivo



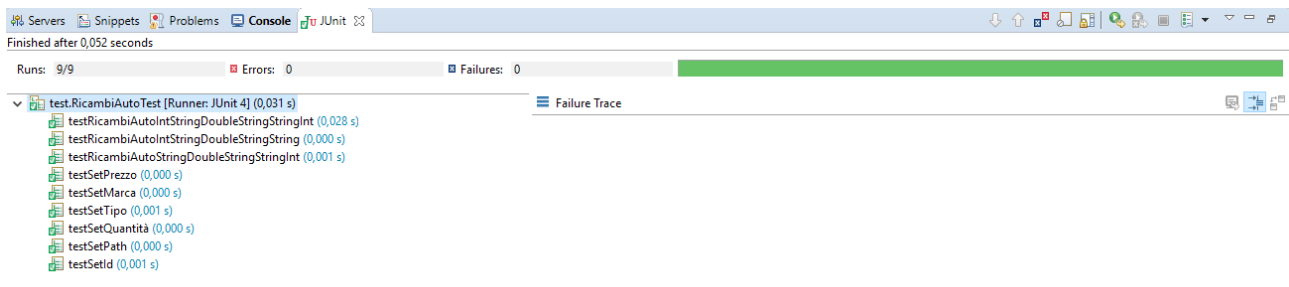
## 5. Classe Client



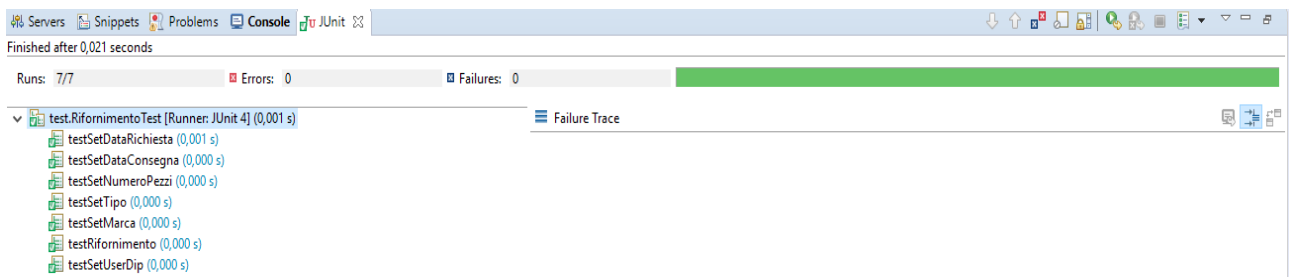
## 6. Classe Dipendente



## 7. Classe RicambiAuto



## 8. Classe Rifornamento



### **2.3 Motivazioni delle decisioni**

Onde evitare problemi futuri nel funzionamento del sistema, o problemi di incoerenza nei dati, si vuole correggere i fault tramite la correzione del codice stesso, in quanto ciò semplificherà le attività di testing successive all'UT. L'Integration Testing e il System Testing, infatti, sono test che mirano a correggere fault a livello di componenti e di sistema. Nel caso in cui, quindi, fossero presenti dei fault nel codice (e quindi uno o più metodi della classe fossero stati realizzati in maniera scorretta), questi sarebbero difficilmente individuabili durante l'IT e l'ST, poiché non vengono testati i singoli metodi.

### **2.4 Conclusioni**

I risultati ottenuti mostrano che i metodi testati singolarmente sono funzionanti in isolamento.

Si può passare all'attività di testing successiva.