

# Elaborato di Ingegneria del Software

Claudia Manfredi  
Mattia Pavlovic  
Elena Tonini

Università degli Studi di Brescia

A.A. 2021/2022

# Sommario I

1 Sommario

2 Testing della  
Versione 5

3 Pattern  
MVC

4 Pattern  
GRASP

5 Principi  
SOLID

6 Pattern della  
Gang of Four

7 Refactoring

8 Testing  
finale

9 Bibliografia

1. Sommario

2. Testing della Versione 5

3. Pattern MVC

4. Pattern GRASP

5. Principi SOLID

6. Pattern della Gang of Four

7. Refactoring

8. Testing finale

9. Bibliografia

2022-06-21

Ingegneria Del Software

Sommario

Sommario

Sommario I

1. Sommario

2. Testing della Versione 5

3. Pattern MVC

4. Pattern GRASP

5. Principi SOLID

6. Pattern della Gang of Four

7. Refactoring

8. Testing finale

9. Bibliografia

# Test Classe Applicazione

- 1 Sommario
- 2 Testing della Versione 5
- 3 Pattern MVC
- 4 Pattern GRASP
- 5 Principi SOLID
- 6 Pattern della Gang of Four
- 7 Refactoring
- 8 Testing finale
- 9 Bibliografia

```
1 package tests;
2
3 import ...
4
14
15 class ApplicazioneTest {
16     Applicazione app = new Applicazione();
17
18     @Test
19     @DisplayName("Root name should be recognised")
20     void isHierarchyNameTaken() {
21         app.addGerarchia( rootName: "radice", new Gerarchia(new Foglia( _nome: "radice", _descrizione: "categoria radice")));
22         assertTrue(app.isHierarchyNameTaken("radice"));
23     }
24
25     @Test
26     void canAddHierarchy(){
27         app.addGerarchia( rootName: "radice", new Gerarchia(new Foglia( _nome: "radice", _descrizione: "categoria radice")));
28         assert app.getHierarchies().keySet().size() == 1;
29     }
30
31     @Test
32     void canRetrieveHierarchy(){
33         app.addGerarchia( rootName: "radice", new Gerarchia(new Foglia( _nome: "radice", _descrizione: "categoria radice")));
34         assertNotNull(app.getHierarchies().get("radice"));
35     }
36
37 }
```

2022-06-21

Ingegneria Del Software  
└─ Testing della Versione 5  
└─ Test Classe Applicazione

Test Classe Applicazione

```
1 package tests;
2
3 import ...
4
5 class ApplicazioneTest {
6     Applicazione app = new Applicazione();
7
8     @Test
9     @DisplayName("Root name should be recognised")
10    void isHierarchyNameTaken() {
11        app.addGerarchia( rootName: "radice", new Gerarchia(new Foglia( _nome: "radice", _descrizione: "categoria radice")));
12        assertTrue(app.isHierarchyNameTaken("radice"));
13    }
14
15    @Test
16    void canAddHierarchy(){
17        app.addGerarchia( rootName: "radice", new Gerarchia(new Foglia( _nome: "radice", _descrizione: "categoria radice")));
18        assert app.getHierarchies().keySet().size() == 1;
19    }
20
21    @Test
22    void canRetrieveHierarchy(){
23        app.addGerarchia( rootName: "radice", new Gerarchia(new Foglia( _nome: "radice", _descrizione: "categoria radice")));
24        assertNotNull(app.getHierarchies().get("radice"));
25    }
26
27 }
```

I test eseguiti sulla classe Applicazione, così come sulle altre, sono stati effettuati selezionando i metodi principali dalla cui correttezza dipendeva in modo ingente la fattibilità del resto del codice.

## isHierarchyNameTaken()

Il test si assicura che, dopo aver inserito una gerarchia nell'applicazione, la verifica della presenza del nome della gerarchia inserita abbia esito positivo.

## canAddHierarchy()

Il test si assicura che, dopo aver inserito una gerarchia nell'applicazione, la verifica della presenza di esattamente una gerarchia nell'applicazione abbia esito positivo.

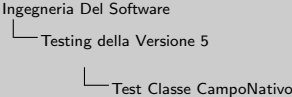
## canRetrieveHierarchy()

Il test si assicura che, dopo aver inserito una gerarchia non null nell'applicazione, essa possa essere recuperata e che non venga restituito un valore null, come in effetti ci si aspetta.

# Test Classe CampoNativo

- 1 Sommario
- 2 Testing della Versione 5
- 3 Pattern MVC
- 4 Pattern GRASP
- 5 Principi SOLID
- 6 Pattern della Gang of Four
- 7 Refactoring
- 8 Testing finale
- 9 Bibliografia

2022-06-21

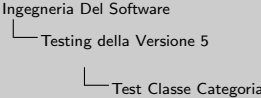


Test Classe CampoNativo

# Test Classe Categoria

- 1 Sommario
- 2 Testing della Versione 5
- 3 Pattern MVC
- 4 Pattern GRASP
- 5 Principi SOLID
- 6 Pattern della Gang of Four
- 7 Refactoring
- 8 Testing finale
- 9 Bibliografia

2022-06-21

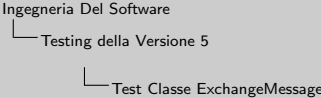


Test Classe Categoria

# Test Classe ExchangeMessage

- 1 Sommario
- 2 Testing della Versione 5
- 3 Pattern MVC
- 4 Pattern GRASP
- 5 Principi SOLID
- 6 Pattern della Gang of Four
- 7 Refactoring
- 8 Testing finale
- 9 Bibliografia

2022-06-21

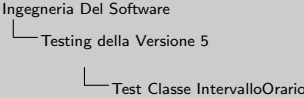


Test Classe ExchangeMessage

# Test Classe IntervalloOrario

- 1 Sommario
- 2 Testing della Versione 5
- 3 Pattern MVC
- 4 Pattern GRASP
- 5 Principi SOLID
- 6 Pattern della Gang of Four
- 7 Refactoring
- 8 Testing finale
- 9 Bibliografia

2022-06-21

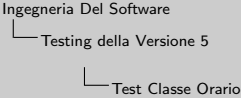


Test Classe IntervalloOrario

# Test Classe Orario

- 1 Sommario
- 2 Testing della Versione 5
- 3 Pattern MVC
- 4 Pattern GRASP
- 5 Principi SOLID
- 6 Pattern della Gang of Four
- 7 Refactoring
- 8 Testing finale
- 9 Bibliografia

2022-06-21



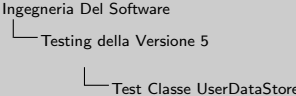
Test Classe Orario



# Test Classe UserDataStore

- 1 Sommario
- 2 Testing della Versione 5
- 3 Pattern MVC
- 4 Pattern GRASP
- 5 Principi SOLID
- 6 Pattern della Gang of Four
- 7 Refactoring
- 8 Testing finale
- 9 Bibliografia

2022-06-21

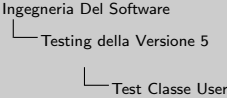


Test Classe UserDataStore

# Test Classe User

- 1 Sommario
- 2 Testing della Versione 5
- 3 Pattern MVC
- 4 Pattern GRASP
- 5 Principi SOLID
- 6 Pattern della Gang of Four
- 7 Refactoring
- 8 Testing finale
- 9 Bibliografia

2022-06-21

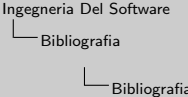


Test Classe User

# Bibliografia I

- 1 Sommario
- 2 Testing della Versione 5
- 3 Pattern MVC
- 4 Pattern GRASP
- 5 Principi SOLID
- 6 Pattern della Gang of Four
- 7 Refactoring
- 8 Testing finale
- 9 Bibliografia**

2022-06-21



Bibliografia I