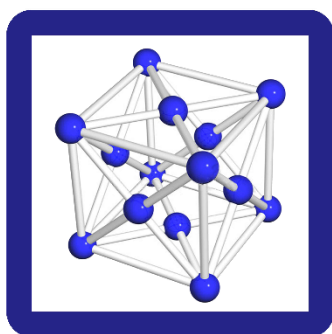




UNIVERSITÀ DEGLI STUDI DI SALERNO



UNIVERSITÀ DEGLI STUDI DI SALERNO
DIPARTIMENTO DI INFORMATICA



earth minerals
EMC
EUROPEAN METALS CORPORATION

CORSO DI INGEGNERIA DEL SOFTWARE PROF. A. DE LUCIA

PROGETTO EMC

UNIT TEST

2020/2021

PARTECIPANTI	MATRICOLA
ALESSANDRA POTESA'	06188
ROSARIO ANNUNZIATA	05810
GIOVANNI TAVOLO	05912

Introduzione

Per testing di unità si intende il testing delle singole unità software del sistema. Il test verrà effettuato con il framework JUnit insieme alla libreria Mockito.

Relazione con altri documenti

Per Individuare i test da effettuare si utilizzerà la tecnica Black-Box, quindi ci baseremo sui documenti prodotti precedentemente:

- TestPlanEMC
- TestCaseEMC

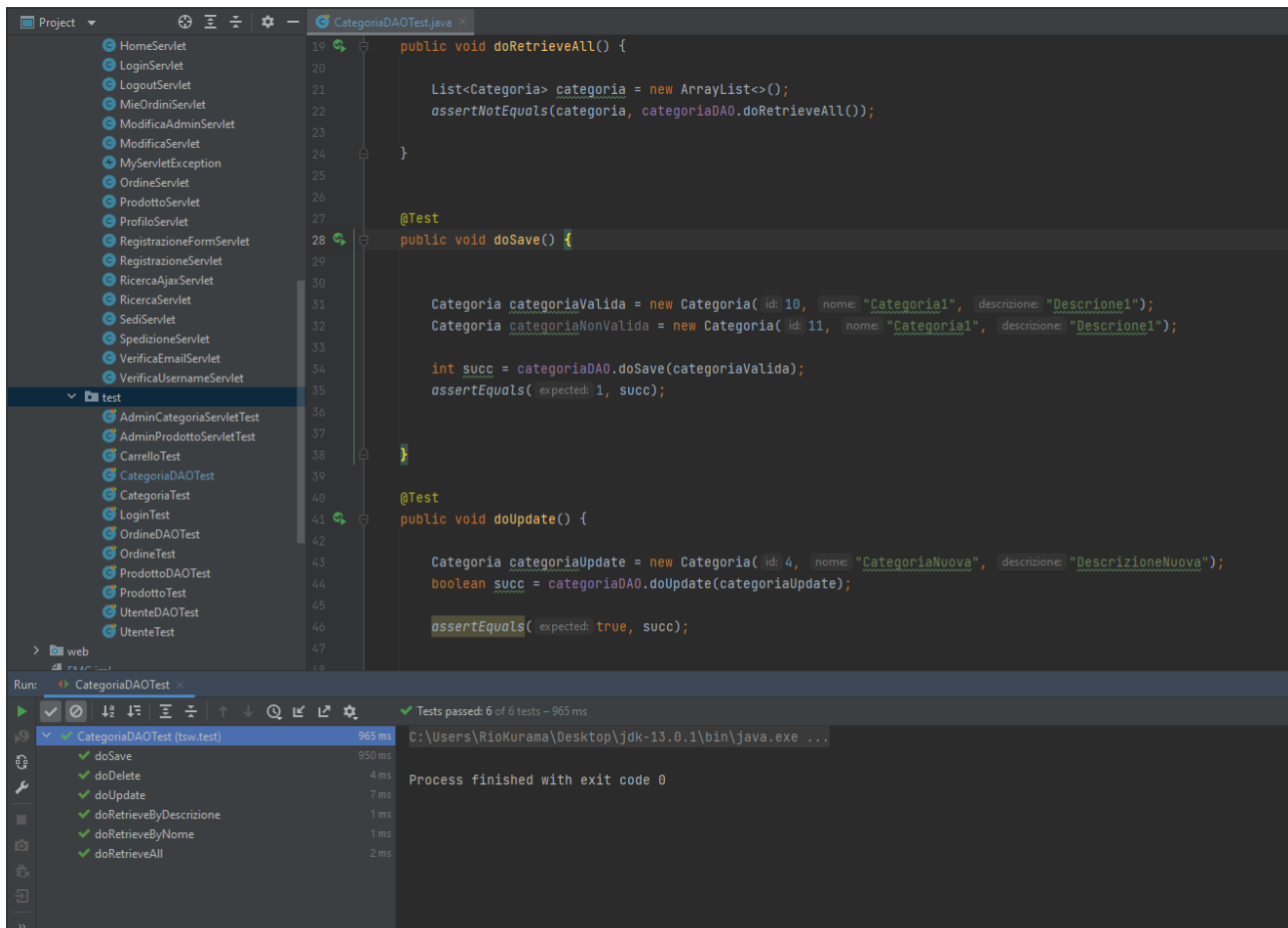
Approccio

Il testing di unità verrà diviso in testingDAO, testingBean e testingServlet, dove verranno testati singolarmente i dao, i bean e le servlet del sistema.

1 - Testing di unità

1.0 - TestDAO

1.0.1 – Test Categoria



The screenshot displays an IDE with the following components:

- Project Explorer:** Shows a list of Java files. The 'test' folder is expanded, showing various test classes including 'CategoriaDAOTest'.
- Editor:** Displays the code for 'CategoriaDAOTest.java'. The code includes:
 - A `doRetrieveAll()` method that creates an empty `ArrayList` and asserts it is not equal to the result of `categoriaDAO.doRetrieveAll()`.
 - A `@Test` annotated `doSave()` method that creates two `Categoria` objects, calls `categoriaDAO.doSave()`, and asserts the success is 1.
 - A `@Test` annotated `doUpdate()` method that creates a `Categoria` object, calls `categoriaDAO.doUpdate()`, and asserts the success is true.
- Run Console:** Shows the execution results for 'CategoriaDAOTest (tw.test)'. It lists six tests, all of which passed:
 - `doSave`: 950 ms
 - `doDelete`: 4 ms
 - `doUpdate`: 7 ms
 - `doRetrieveByDescrizione`: 1 ms
 - `doRetrieveByNome`: 1 ms
 - `doRetrieveAll`: 2 msThe total time for all tests is 965 ms. The console also shows 'Tests passed: 6 of 6 tests - 965 ms' and 'Process finished with exit code 0'.

1.0.2 – Test Ordine

The screenshot displays an IDE with the following components:

- Project Explorer (Left):** A tree view of the project structure. The 'test' package is expanded, showing various test classes. 'OrdineDAOTest' is selected and highlighted.
- Editor (Center):** Displays the source code for 'OrdineDAOTest.java'. The code includes:

```
1 package tsw.test;
2
3 import ...
4
13
14 public class OrdineDAOTest {
15     OrdineDAO ordineDAO = new OrdineDAO();
16     @Test
17     public void doSave() {
18
19         int pippo = ordineDAO.doSave( indirizzo: "indirizzo", idu: 2, prezzo: "prezzo");
20         assertEquals( expected: 1, pippo);
21     }
22
23
24     @Test
25     public void doRetrieveByUtente() {
26
27         List<Ordine> ordineList = new ArrayList<>();
28         assertEquals(ordineList, ordineDAO.doRetrieveByUtente(2));
29     }
30
31 }
```
- Run Console (Bottom):** Shows the execution results of the tests. The status bar indicates 'Tests passed: 2 of 2 tests - 951 ms'. The console output shows:

```
Run: OrdineDAOTest
✓ OrdineDAOTest (tsw.test) 951 ms
  ✓ doSave 946 ms
  ✓ doRetrieveByUtente 5 ms
Process finished with exit code 0
```

1.0.3 – Test Prodotto

The screenshot displays an IDE with a project named "test" containing various servlets and tests. The "ProdottaDAOTest.java" file is open, showing three test methods: `doRetrieveByCategoria()`, `doSave()`, and `doUpdate()`. The `doSave()` method creates a `Prodotto` object and asserts that `prodottaDAO.doSave(prodotta)` returns 1. The `doUpdate()` method creates a `Prodotto` object and asserts that `prodottaDAO.doUpdate(prodotta)` returns true.

```
30
31
32 @Test
33 public void doRetrieveByCategoria() {
34     List<Prodotto> prodottolist = new ArrayList<>();
35     assertNotNull(prodottolist, prodottaDAO.doRetrieveByCategoria(1));
36 }
37
38
39 @Test
40 public void doSave() {
41
42     Prodotto prodotta = new Prodotto( id: 11, nome: "nome", descrizione: "descrizione", prezzoBase: 3, iva: 2, idcategoria: 2);
43     int pippo = prodottaDAO.doSave(prodotta);
44     assertEquals( expected: 1, pippo);
45 }
46
47
48 @Test
49 public void doUpdate() {
50
51     Prodotto prodotta = new Prodotto( id: 3, nome: "nome", descrizione: "descrizione", prezzoBase: 3, iva: 2, idcategoria: 2);
52     boolean succ = prodottaDAO.doUpdate(prodotta);
53     assertEquals( expected: true, succ);
54 }
55
56
```

The Run window shows the test results for "ProdottaDAOTest (tsw.test)". All 12 tests passed, with a total time of 977 ms. The process finished with exit code 0.

Test Method	Time (ms)
doRetrieveByCategoria	937
doSave	10
doRetrieveById	4
doDelete	6
doUpdate	4
doRetrieveByNomeOrDescrizione	3
doRetrieveByDescrizione	2
doRetrieveByNome	1
doRetrieveByPrezzo	2
doRetrieveByCategoria	2
doRetrieveAll	4
doRetrieveByNomeSingolo	2

1.0.4 – Test Utente

The screenshot shows an IDE with the following components:

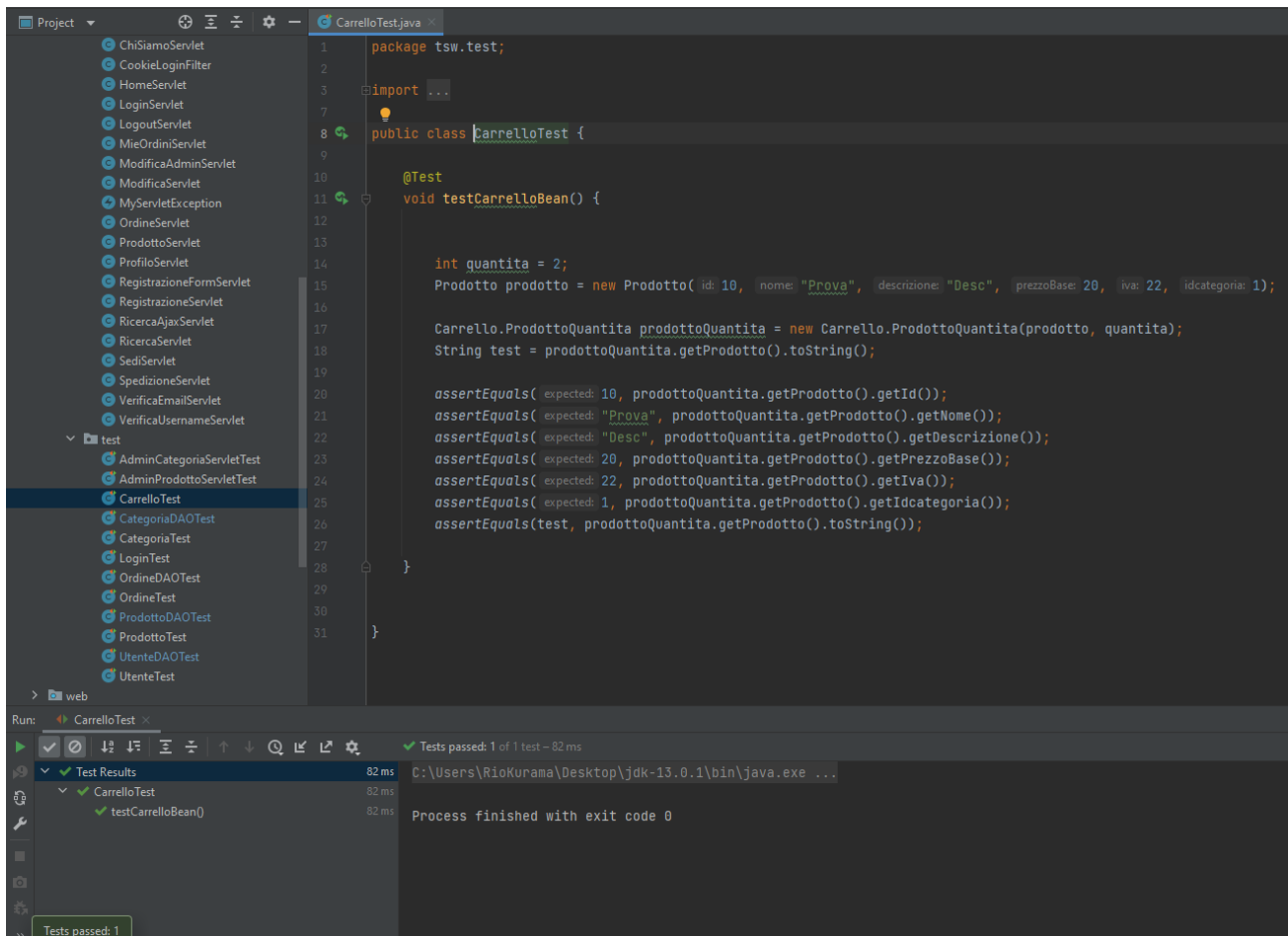
- Project Explorer:** Lists various servlets and tests. The **test** folder is expanded, showing **UtenteDAOTest** selected.
- Code Editor:** Displays the **UtenteDAOTest.java** file with the following code:

```
64 public void doSave() {  
65  
66     Utente utente = new Utente( id: 5, username: "Prova", passwordhash: "Prova", nome: "Email", email: "Prova", admin: false);  
67  
68     boolean succ = utenteDAO.doSave(utente);  
69     assertEquals( expected: true, succ);  
70  
71 }  
72  
73  
74 @Test  
75 public void doUpdate() {  
76  
77     int succ = utenteDAO.doUpdate( username: "Prova1", nome: "Email1", email: "Prova1", id: 5);  
78     assertEquals( expected: 1, succ);  
79  
80  
81 }  
82  
83  
84 @Test  
85 public void doDelete() {  
86  
87     int succ = utenteDAO.doDelete( id: 1);  
88     assertEquals( expected: 1, succ);  
89  
90 }  
91 }
```
- Run Console:** Shows the execution of **UtenteDAOTest** with the following output:

```
Run: UtenteDAOTest  
Tests passed: 8 of 8 tests - 1 sec 147 ms  
C:\Users\RioKurama\Desktop\jdk-13.0.1\bin\java.exe ...  
Process finished with exit code 0
```

1.1 - TestBean

1.1.0 – Test Carrello



1.1.1 – Test Categoria

The screenshot displays an IDE with the following components:

- Project Explorer:** A list of project files on the left. The `test` folder is expanded, showing various test classes. `CategoriaTest` is selected and highlighted.
- Editor:** The main window shows the code for `CategoriaTest.java`. The code defines a class `CategoriaTest` with a single test method `testCategoriaBean()` annotated with `@Test`. The method creates a `Categoria` object, clones it, and uses `assertEquals` to verify the object's properties and its string representation.
- Run Console:** At the bottom, the execution results are shown. It indicates that 1 of 1 tests passed in 36 ms. The command executed was `C:\Users\RioKurama\Desktop\jdk-13.0.1\bin\java.exe ...`, and the process finished with exit code 0.

```
2
3 import ...
7
8 class CategoriaTest {
9
10     @Test
11     void testCategoriaBean() {
12
13         int id = 1;
14         String nome = "Gina";
15         String descrizione = "La gallina";
16
17         Categoria categoria = new Categoria(id, nome, descrizione);
18         Categoria categoria1 = new Categoria(id, nome, descrizione);
19         String test = categoria.toString();
20         String test1 = categoria1.toString();
21
22         assertEquals(id, categoria.getId());
23         assertEquals(nome, categoria.getNome());
24         assertEquals(descrizione, categoria.getDescrizione());
25         assertEquals(test, categoria.toString());
26
27         categoria1.setId(id);
28         categoria1.setNome(nome);
29         categoria1.setDescrizione(descrizione);
30         assertEquals(nome, categoria1.getNome());
31         assertEquals(descrizione, categoria1.getDescrizione());
32         assertEquals(test1, categoria1.toString());
33
34
35 }
```

Run: **CategoriaTest** ×

✓ Tests passed: 1 of 1 test – 36 ms

Test Results	Duration
✓ CategoriaTest	36 ms
✓ testCategoriaBean()	36 ms

Process finished with exit code 0

Tests passed: 1

1.1.2 – Test Login

The screenshot displays an IDE interface with the `LoginTest.java` file open. The left sidebar shows a project tree with a 'test' folder containing various test classes, including `LoginTest`. The main editor shows the following Java code:

```
7  
8 public class LoginTest {  
9  
10  
11     @Test  
12     void testLogin() {  
13  
14         String id = "pippo";  
15         int idutente = 1;  
16         String token = "token";  
17  
18         Login login = new Login(id, idutente, token);  
19         Login login1 = new Login(id, idutente, token);  
20         assertNotNull(login);  
21         String test = login.toString();  
22         String test1 = login1.toString();  
23  
24         assertEquals(id, login.getId());  
25         assertEquals(idutente, login.getIdutente());  
26         assertEquals(token, login.getToken());  
27         assertEquals(test, login.toString());  
28  
29         login1.setId(id);  
30         login1.setIdutente(idutente);  
31         login1.setToken(token);  
32         assertEquals(id, login1.getId());  
33         assertEquals(idutente, login1.getIdutente());  
34         assertEquals(token, login1.getToken());  
35         assertEquals(test1, login.toString());  
36  
37     }  
}
```

Below the editor, the 'Run' tab shows the execution results for `LoginTest`. The output indicates that the test passed successfully.

Run: LoginTest

Tests passed: 1 of 1 test – 46 ms

Test Results 46 ms

- LoginTest 46 ms
 - testLogin() 46 ms

C:\Users\RioKurama\Desktop\jdk-13.0.1\bin\java.exe ...

Process finished with exit code 0

1.1.3 – Test Ordine

The screenshot displays an IDE with a project explorer on the left, a code editor in the center, and a run console at the bottom.

Project Explorer: The 'test' package is expanded, showing various test classes. 'OrdineTest' is selected.

Code Editor: The file 'OrdineTest.java' is open. It contains the following code:

```
1 package tsw.test;
2
3 import ...
4
10
11 public class OrdineTest {
12
13     @Test
14     void testOrdine() {
15
16         int id = 1;
17         String indirizzo = "Indirizzo";
18         String prezzo = "token";
19         Utente utente = new Utente(id, "Pippo", "pass", "Nome", "Email", true);
20         Ordine ordine = new Ordine(id, indirizzo, prezzo, utente);
21
22         assertNotNull(ordine);
23         String test = ordine.toString();
24
25         assertEquals(id, ordine.getId());
26         assertEquals(indirizzo, ordine.getIndirizzo());
27         assertEquals(prezzo, ordine.getPrezzotot());
28         assertEquals(utente, ordine.getUtente());
29         assertEquals(test, ordine.toString());
30
31     }
32
33 }
34
35
36 }
```

Run Console: The console shows the execution of the test. The status bar indicates 'Tests passed: 1 of 1 test - 63 ms'. The output shows:

```
Test Results 63 ms
  OrdineTest 63 ms
    testOrdine() 63 ms
    Process finished with exit code 0
```

The bottom status bar confirms 'Tests passed: 1'.

1.1.4 – Test Prodotto

The screenshot shows an IDE with a project named 'web' on the left. The 'test' folder is expanded, showing various test classes. The 'ProdottoTest' class is selected and its code is displayed in the main editor. The code defines private fields for 'prezzoBase', 'iva', 'categorie', and 'Rame', and a test method 'testProdotto()' that creates two 'Prodotto' objects and asserts their properties.

```
18 private long prezzoBase;
19 private int iva;
20 private List<Categoria> categorie;
21 private Categoria Rame = new Categoria();
22
23
24 @Test
25 void testProdotto() {
26
27     int id = 1;
28     String nome = "Indirizzo";
29     String descrizione = "token";
30     long prezzoBase = 2000;
31     int iva = 22;
32     int idcategoria = 1;
33
34     Prodotto prodotto = new Prodotto(id, nome, descrizione, prezzoBase, iva, idcategoria);
35     Prodotto prodotto1 = new Prodotto(id, nome, descrizione, prezzoBase, iva, idcategoria);
36     String test = prodotto.toString();
37     String test1 = prodotto1.toString();
38
39
40     assertEquals(id, prodotto.getId());
41     assertEquals(nome, prodotto.getNome());
42     assertEquals(descrizione, prodotto.getDescrizione());
43     assertEquals(prezzoBase, prodotto.getPrezzoBase());
44     assertEquals(iva, prodotto.getIva());
45     assertEquals(idcategoria, prodotto.getIdcategoria());
46     assertEquals(test, prodotto.toString());
47
48 }
```

The bottom panel shows the test results for 'ProdottoTest' and 'testProdotto()', both of which passed successfully. The output indicates that the process finished with exit code 0.

Run: **ProdottoTest** Tests passed: 1 of 1 test - 49 ms

Test Results 49 ms

- ProdottoTest 49 ms
 - testProdotto() 49 ms

Process finished with exit code 0

Tests passed: 1

1.1.5 – Test Utente

The screenshot displays an IDE interface with a project explorer on the left, a code editor in the center, and a run console at the bottom.

Project Explorer: Lists various servlets and test classes. The 'test' folder is expanded, showing classes like `AdminCategoriaServletTest`, `AdminProdottoServletTest`, `CarrelloTest`, `CategoriaDAOTest`, `CategoriaTest`, `LoginTest`, `OrdineDAOTest`, `OrdineTest`, `ProdottoDAOTest`, `ProdottoTest`, `UtenteDAOTest`, and `UtenteTest`.

Code Editor: Shows the `UtenteTest.java` file. The code defines a `public class UtenteTest` with a `@Test` annotated `void testUtente()` method. The method creates a `Utente` object, sets its attributes, and performs assertions.

```
public class UtenteTest {  
  
    @Test  
    void testUtente() {  
  
        int id = 1;  
        String username = "Indirizzo";  
        String passwordhash = "token";  
        String nome = "Nome";  
        String email = "Email";  
        boolean admin = false;  
  
        Utente utente = new Utente(id, username, passwordhash, nome, email, admin);  
        Utente utente1 = new Utente(id, username, passwordhash, nome, email, admin);  
        String test = utente.toString();  
        String test1 = utente1.toString();  
  
        assertEquals(id, utente.getId());  
        assertEquals(username, utente.getUsername());  
        assertEquals(passwordhash, utente.getPasswordhash());  
        assertEquals(nome, utente.getNome());  
        assertEquals(email, utente.getEmail());  
        assertEquals(admin, utente.isAdmin());  
        assertEquals(test, utente.toString());  
  
        utente1.setId(id);  
        utente1.setUsername(username);  
        utente1.setPasswordhash(passwordhash);  
        utente1.setNome(nome);  
        utente1.setEmail(email);  
    }  
}
```

Run Console: Shows the execution of the `UtenteTest` class. The output indicates that all tests passed.

```
Run: UtenteTest  
Tests passed: 1 of 1 test - 63 ms  
C:\Users\RioKurama\Desktop\jdk-13.0.1\bin\java.exe ...  
Process finished with exit code 0
```

Test Results	Time
UtenteTest	63 ms
testUtente()	63 ms