

Università degli Studi di Salerno

Corso di laurea in Informatica

**Ingegneria, Gestione ed Evoluzione
del Software**

“DisneyStore”

Docente:

Andrea De Lucia

Studenti:

Buono Claudia

05225 01097

Fulgione Anna

05225 01103

Anno accademico 2020/2021

SOMMARIO

Introduzione	3
JUnit	3
Test di unità	3
Model	3
Servlet.....	7
Test di integrazione	15
TC_AddressOperationIntegration.....	15
Katalon	24
Gestione utente.....	24
Gestione prodotto	26

INTRODUZIONE

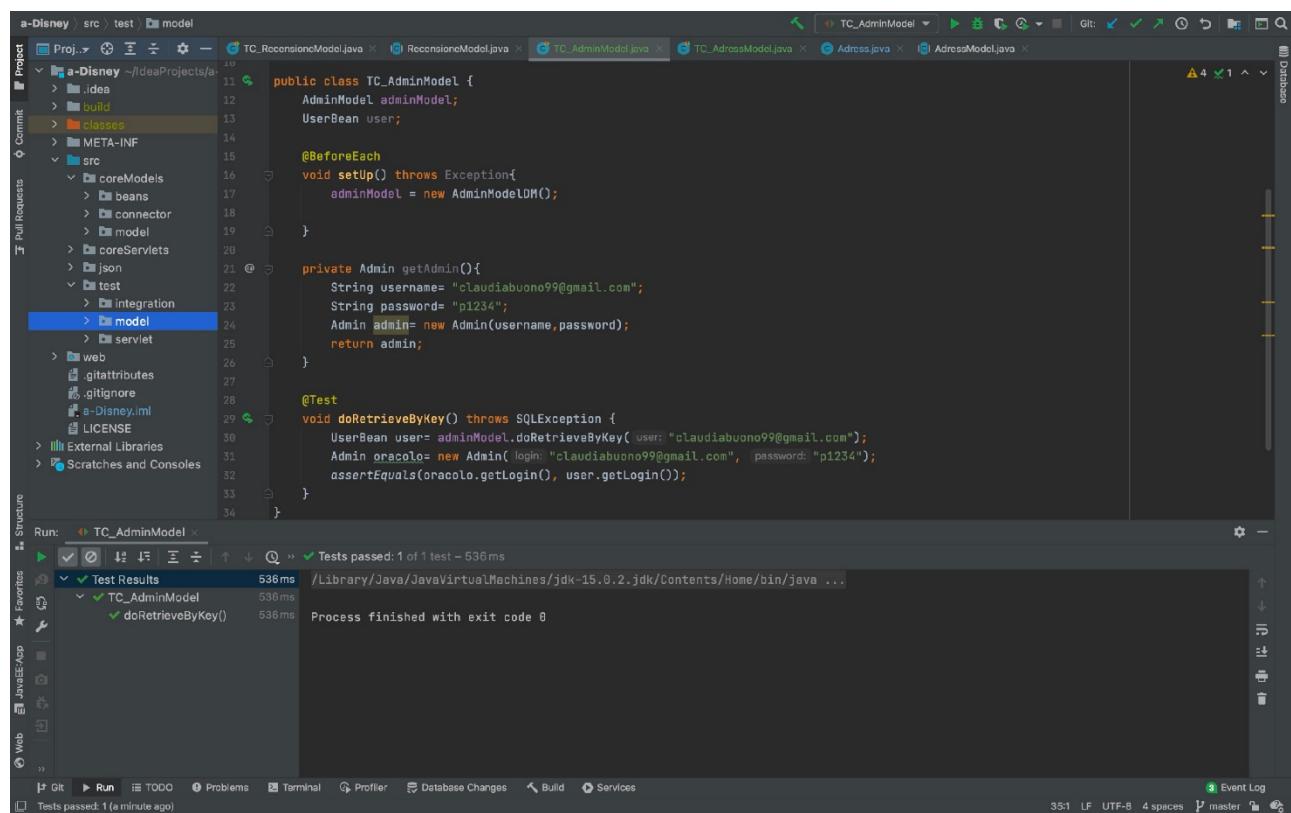
Tale documento si compone di due parti, nella prima parte effettueremo test di unità sulle classi Model e sulle Servlet, successivamente procederemo con il test di integrazione. Tali test verranno effettuati con JUnit. La seconda parte invece rappresenta l'attività di esecuzione dei test case descritti all'interno del Test Case Specification. I test verranno eseguiti con Katalon.

JUNIT

TEST DI UNITÀ

MODEL

ADMINMODEL



The screenshot shows the IntelliJ IDEA interface with the following details:

- Project View:** Shows the project structure under "a-Disney". The "model" package is selected, containing files like TC_AdminModel.java, RecensioneModel.java, AdressModel.java, and AdminModel.java.
- Code Editor:** Displays the content of TC_AdminModel.java. It includes an @BeforeEach annotation for setup, a private getAdmin() method, and a @Test doRetrieveByKey() method that asserts equality between an Admin object created via the AdminModel and one retrieved by key.
- Run Tab:** Shows the results of a run named "TC_AdminModel". It indicates 1 test passed in 536ms, with the process finishing with exit code 0.
- Bottom Status Bar:** Shows "Tests passed: 1 (a minute ago)" and other system information like "35:1 LF UTF-8 4 spaces master".

ADDRESSMODEL

The screenshot shows the IntelliJ IDEA interface with the project 'a-Disney' open. The code editor displays the file 'TC_AdressModel.java'. The 'Run' tool window at the bottom shows the test results for 'TC_AdressModel'. The results indicate 5 tests passed in 565 ms. The tests listed are: doRetrieve(), doSave(), doDelete(), doModify(), and doRetrieveAll(). All tests passed with execution times ranging from 5 ms to 565 ms.

```
package test.model;

import coreModels.beans.Admin;
import coreModels.beans.Adress;
import coreModels.beans.Registered;
import coreModels.model.AdressModel;
import coreModels.model.DM.AddressModelDM;
import coreModels.model.DM.AdminModelDM;
import org.junit.jupiter.api.BeforeEach;
import org.junit.jupiter.api.Test;
import java.sql.SQLException;
import java.util.ArrayList;
import java.util.Map;

import static org.junit.jupiter.api.Assertions.*;

public class TC_AdressModel {
    AdressModel address;
    Registered utente;

    @BeforeEach
    void setup() throws Exception {
        // Setup logic here
    }

    @Test
    void doRetrieve() {
        assertEquals("matteo@alice.it", address.getAdress());
    }

    @Test
    void doSave() {
        assertEquals("matteo@alice.it", address.getAdress());
    }

    @Test
    void doDelete() {
        assertEquals("matteo@alice.it", address.getAdress());
    }

    @Test
    void doModify() {
        assertEquals("matteo@alice.it", address.getAdress());
    }

    @Test
    void doRetrieveAll() {
        assertEquals("matteo@alice.it", address.getAdress());
    }
}
```

Test Results:

- TC_AdressModel
 - doRetrieve(): 565 ms
 - doSave(): 536 ms
 - doDelete(): 10 ms
 - doModify(): 5 ms
 - doRetrieveAll(): 6 ms

Process finished with exit code 0

FATTURAMODEL

The screenshot shows the IntelliJ IDEA interface with the project 'a-Disney' open. The code editor displays the file 'TC_FatturaModel.java'. The 'Run' tool window at the bottom shows the test results for 'TC_FatturaModel'. The results indicate 7 tests passed in 4 s 67 ms. The tests listed are: retrieveInvoiceOrders(), retrieveInvoices(), doSave(), updateQtyProducts(), retrieveInvoice(), retrieveInvoicesTest(), and hasPurchased(). All tests passed with execution times ranging from 15 ms to 4 s 67 ms.

```
package test.model;

import coreModels.model.FatturaModel;
import coreModels.model.DM.FatturaModelDM;
import org.junit.jupiter.api.Test;
import java.sql.SQLException;
import java.util.List;

import static org.junit.jupiter.api.Assertions.*;

public class TC_FatturaModel {
    FatturaModel fatturaModel;

    @BeforeEach
    void setup() {
        fatturaModel = new FatturaModel();
    }

    @Test
    void retrieveInvoiceOrders() {
        List<Fattura> fatture = fatturaModel.retrieveInvoiceOrders();
        assertEquals(1, fatture.size());
        assertEquals("Ordine 1", fatture.get(0).getTitolo());
    }

    @Test
    void retrieveInvoices() {
        List<Fattura> fatture = fatturaModel.retrieveInvoices();
        assertEquals(1, fatture.size());
        assertEquals("Ordine 1", fatture.get(0).getTitolo());
    }

    @Test
    void doSave() {
        Fattura fattura = new Fattura();
        fattura.setTitolo("Ordine 1");
        fattura.setModel("Modello 1");
        fattura.setPrezzoTotale(100);
        fattura.setQuantitaProdotti(1);
        fattura.setUtente("utente");
        fattura.setFatturato(true);

        boolean result = fatturaModel.doSave(fattura);
        assertEquals(true, result);
    }

    @Test
    void updateQtyProducts() {
        Fattura fattura = new Fattura();
        fattura.setTitolo("Ordine 1");
        fattura.setModel("Modello 1");
        fattura.setPrezzoTotale(100);
        fattura.setQuantitaProdotti(1);
        fattura.setUtente("utente");
        fattura.setFatturato(true);

        boolean result = fatturaModel.updateQtyProducts(fattura);
        assertEquals(true, result);
    }

    @Test
    void retrieveInvoice() {
        Fattura fattura = new Fattura();
        fattura.setTitolo("Ordine 1");
        fattura.setModel("Modello 1");
        fattura.setPrezzoTotale(100);
        fattura.setQuantitaProdotti(1);
        fattura.setUtente("utente");
        fattura.setFatturato(true);

        Fattura retrievedFattura = fatturaModel.retrieveInvoice("Ordine 1");
        assertEquals("Ordine 1", retrievedFattura.getTitolo());
    }

    @Test
    void retrieveInvoicesTest() {
        Fattura fattura = new Fattura();
        fattura.setTitolo("Ordine 1");
        fattura.setModel("Modello 1");
        fattura.setPrezzoTotale(100);
        fattura.setQuantitaProdotti(1);
        fattura.setUtente("utente");
        fattura.setFatturato(true);

        List<Fattura> retrievedFatture = fatturaModel.retrieveInvoices();
        assertEquals(1, retrievedFatture.size());
        assertEquals("Ordine 1", retrievedFatture.get(0).getTitolo());
    }

    @Test
    void hasPurchased() {
        Fattura fattura = new Fattura();
        fattura.setTitolo("Ordine 1");
        fattura.setModel("Modello 1");
        fattura.setPrezzoTotale(100);
        fattura.setQuantitaProdotti(1);
        fattura.setUtente("utente");
        fattura.setFatturato(true);

        boolean result = fatturaModel.hasPurchased("utente");
        assertEquals(true, result);
    }
}
```

Test Results:

- TC_FatturaModel
 - retrieveInvoiceOrders(): 4 s 67 ms
 - retrieveInvoices(): 2 s 483 ms
 - doSave(): 281 ms
 - updateQtyProducts(): 757 ms
 - retrieveInvoice(): 187 ms
 - retrieveInvoicesTest(): 344 ms
 - hasPurchased(): 15 ms

Process finished with exit code 0

PRODUCTMODEL

```
a-Disney > src > test > model > TC_ProductModel.java
```

```
ProductModel.java TC_ProductModel.java prodotto ProductBean.java Admin.java
```

```
Project Commit Pull Requests Favorites Structure
```

```
Run: TC_ProductModel <
```

```
File Edit View Navigate Code Analyze Refactor Build Run Tools Git Window Help a-Disney - TC_ProductModel.java
```

```
beans
    Admin
    Adress
    Cart
    FatturaBean
    Order
    ProductBean
    RecensioneBean
    Registered
    UserBean
```

```
connector
```

```
79
80     @Test
81     void testDoRetrieveList () throws SQLException{
82         int[] code= new int[1];
83         code[0] = 1;
84         List<ProductBean> listaprodotti= product.doRetrieveList(code, acquista: true);
85         assertEquals( expected: 1, listaprodotti.get(0).getCode());
86     }
87
88     @Test
89     void testDoDelete() throws SQLException{
90         Boolean flag= product.doDelete(-code: 60);
```

```
Run: TC_ProductModel <
```

```
Test Results 1 s 510 ms C:\Users\utente\.jdks\openjdk-15.0.2\bin\java.exe ...
```

```
TC_ProductModel 1 s 46 ms
    ✓ testDoRetrieveByKey()
    ✓ testDoRetrieveList()
    ✓ testDoRetrieveAll()
    ✓ testDoRetrieveBySearch()
    ✓ testDoDeleted()
    ✓ testDoUpdate()
    ✓ testDoSave()
    ✓ testDoRetrieveByCategory()
    ✓ testDoRetrieveByDiscount()
```

```
Process finished with exit code 0
```

```
84:55 CRLF UTF-8 4 spaces master Event Log
```

RECENSIONEMODEL

```
a-Disney > src > test > model > TC_RecensioneModel.java
```

```
ProductModel.java TC_ProductModel.java TC_RecensioneModel.java prodotto ProductBean.java Admin.java
```

```
Project Commit Pull Requests Favorites Structure
```

```
Run: TC_RecensioneModel <
```

```
File Edit View Navigate Code Analyze Refactor Build Run Tools Git Window Help a-Disney - TC_RecensioneModel.java
```

```
RegisteredModel
    UserModel
    coreServices
    json
    test
        integration
        model
            prova
            TC_ProductModel
            TC_RecensioneModel
                servlet
                    prova
    web
        contentJSP
        css
        db
            database_store.sql
        fonts
        images
        js
        WFR-INF
```

```
37
38     @Test
39     void testMediumVote () throws SQLException {
40         double media= recensione.mediumVote(getprodotto());
41         assertEquals( expected: 4.0, media);
42     }
43     @Test
44     void testUserComment()throws SQLException {
45         RecensioneBean recensioneBean= recensione.userComment(getUtenze(),getprodotto());
46         RecensioneBean oracolo= new RecensioneBean( description: "Prodotto curato nei minimi dettagli", valutazione: 4, name: "ca");
47         assertEquals(oracolo, recensioneBean);
48
49     @Test
50     void testGetComments()throws SQLException{
51         List<RecensioneBean> listRecensione= recensione.getComments (getprodotto());
52         List<RecensioneBean> oracolo= new ArrayList<>();
53         RecensioneBean r1= new RecensioneBean( description: "Buone qualità e consegna in tempo", valutazione: 4, name: "russo mat");
54         RecensioneBean r2= new RecensioneBean( description: "Prodotto curato nei minimi dettagli", valutazione: 4, name: "capozzo");
55         oracolo.add(r1);
56         oracolo.add(r2);
```

```
Run: TC_RecensioneModel <
```

```
Test Results 1 s 144 ms C:\Users\utente\.jdks\openjdk-15.0.2\bin\java.exe ...
```

```
TC_RecensioneModel 1 s 144 ms
    ✓ testUserComment()
    ✓ testGetComments()
    ✓ testMediumVote()
```

```
Process finished with exit code 0
```

```
58:6 CRLF UTF-8 4 spaces master Event Log
```

REGISTEREDMODEL

The screenshot shows the IntelliJ IDEA interface with the project 'a-Disney' open. The left sidebar displays the project structure with packages like DM, DS, AdminModel, AdressModel, FatturaModel, Paginator, ProductModel, RecensioneModel, RegisteredModel, and UserModel. The 'test' package contains sub-folders integration and model, which in turn contain TC_RegisteredModel and TC_RecensioneModel. The main editor window shows the code for TC_RegisteredModel.java, containing several @Test annotations for methods like testDoRetrieveByKey, testRegistration, testDoModify, and testDoRetrieveBySearch. The Database tool window on the right shows a connection to MySQL at localhost with a schema named 'store'. The bottom status bar indicates the code has passed 4 tests in 649 ms.

```
a-Disney > src > test > model > TC_RegisteredModel
a-Disney - TC_RegisteredModel.java
27     @Test
28     void testDoRetrieveByKey() throws SQLException{
29         Registered registered = registrato.doRetrieveByKey(user: "rosalia@libero.it");
30         assertEquals(expected: "rosalia@libero.it", registered.getLogin());
31     }
32
33     @Test
34     void testRegistration() throws SQLException{
35         Boolean flag= registrato.registration(getUtente());
36         assertEquals(expected: true, flag);
37     }
38
39     @Test
40     void testDoModify() throws SQLException{
41         Boolean flag= registrato.doModify(getUtente(), name: "leo", surname: "fulgione", login: "bla");
42         assertEquals(expected: true, flag);
43     }
44
45     @Test
46     void testDoRetrieveBySearch() throws SQLException{
        ArrayList<Registered> listaRegistrati= registrato.doRetrieveBySearch("anna@virgilio.it")
    }
```

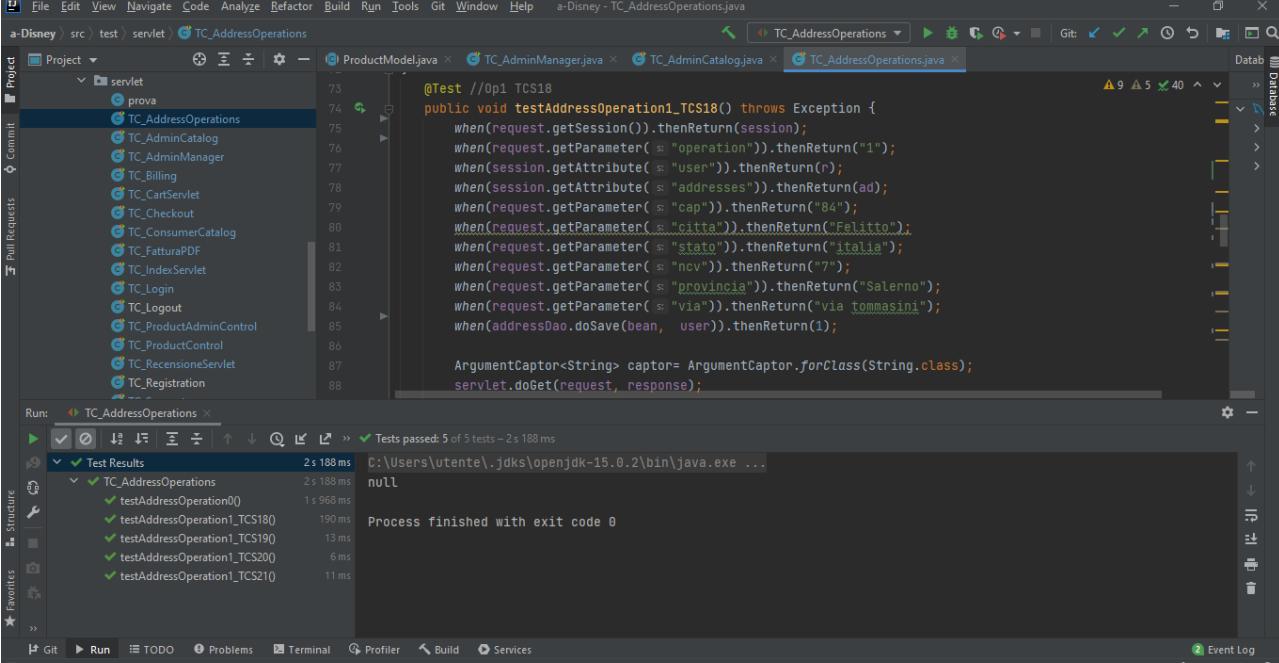
USERMODEL

The screenshot shows the IntelliJ IDEA interface with the project 'a-Disney' open. The left sidebar displays the project structure with packages like coreServices, connector, model, and test. The 'model' package contains sub-folders prova, TC_AdminModel, TC_AdressModel, TC_FatturaModel, TC_ProductModel, TC_RecensioneModel, TC_RegisteredModel, and TC_UserModel. The main editor window shows the code for TC_UserModel.java, containing @BeforeEach and @Test annotations for methods like testDoesExist and testLogin. The Database tool window on the right shows a connection to MySQL at localhost with a schema named 'store'. The bottom status bar indicates the code has passed 2 tests in 162 ms.

```
a-Disney > src > coreServices
a-Disney - TC_UserModel.java
19
20     @BeforeEach
21     void setUp() throws Exception {
22         userReg = new RegisteredModelDM();
23     }
24
25     private UserBean getUser(){
26         UserBean utente= new Registered( name: "rosalia", cognome: "capozzolo", login: "rosalia", password: "ro123456", email: "rosalia@libero.it", telefono: "010123456789", indirizzo: "via dei fiori 123", codicePostale: "00123", città: "roma", provincia: "roma", nazione: "italia", dataNascita: "1990-01-01", sesso: "f", ruolo: "utente");
27         return utente;
28     }
29
30     @Test
31     void testDoesExist() throws SQLException{
32         Boolean flag= userReg.exists(getUser());
33         assertEquals(expected: true, flag);
34     }
35
36     @Test
37     void testLogin () throws SQLException{
38         UserBean utente= userReg.login( username: "rosalia@libero.it", password: "ro123456");
39         assertEquals(expected: "rosalia@libero.it", utente.getLogin());
    }
```

SERVLET

TC_ADDRESSOPERATION



The screenshot shows the IntelliJ IDEA interface with the project 'servlet' open. The code editor displays a Java test class 'TC_AddressOperations.java'. The test method 'testAddressOperation1_TCS18()' is shown with its implementation using the Mockito framework. The test results pane at the bottom indicates 5 tests passed in 2 seconds.

```
    @Test //Op1 TCS18
    public void testAddressOperation1_TCS18() throws Exception {
        when(request.getSession()).thenReturn(session);
        when(request.getParameter("operation")).thenReturn("1");
        when(session.getAttribute("user")).thenReturn(r);
        when(session.getAttribute("addresses")).thenReturn(ad);
        when(request.getParameter("cap")).thenReturn("84");
        when(request.getParameter("citta")).thenReturn("Feltitto");
        when(request.getParameter("stato")).thenReturn("italia");
        when(request.getParameter("ncv")).thenReturn("");
        when(request.getParameter("provincia")).thenReturn("Salerno");
        when(request.getParameter("via")).thenReturn("via Tommasini");
        when(addressDao.doSave(bean, user)).thenReturn(1);

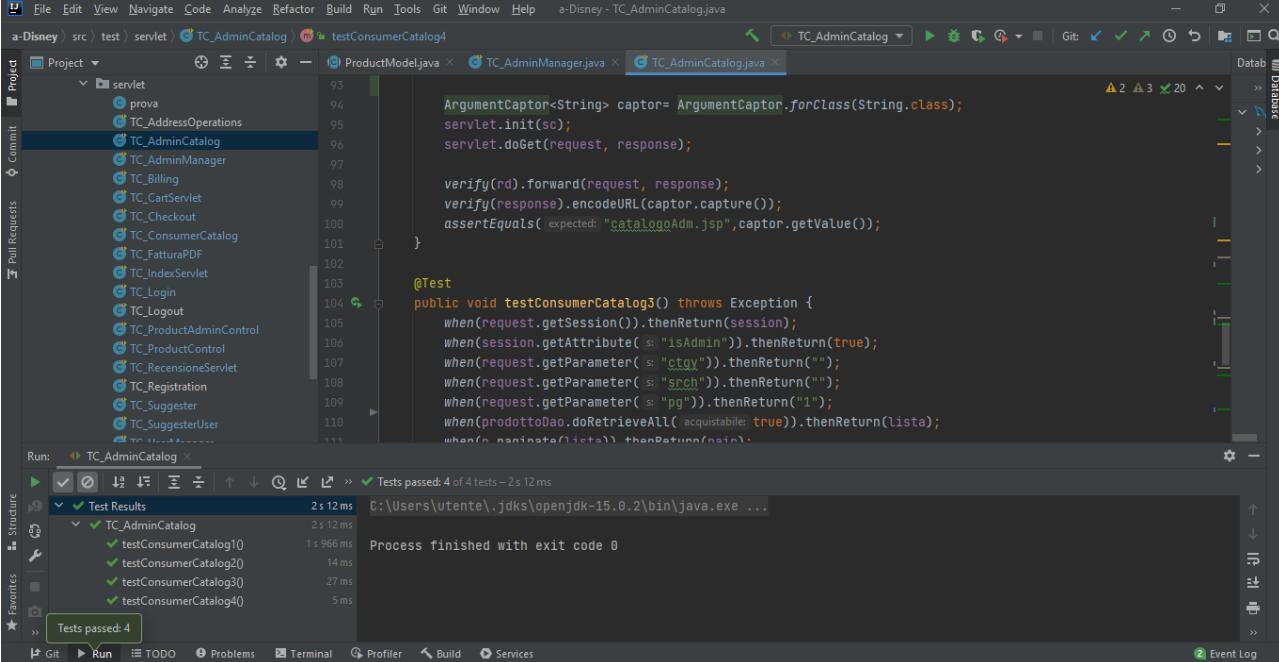
        ArgumentCaptor<String> captor= ArgumentCaptor.forClass(String.class);
        servlet doGet(request, response);
    }
```

Run: TC_AddressOperations x
Tests passed: 5 (moments ago)

Test Results
TC_AddressOperations
 ✓ testAddressOperation0() 2s 188 ms
 ✓ testAddressOperation1_TCS18() 1s 968 ms
 ✓ testAddressOperation1_TCS19() 190 ms
 ✓ testAddressOperation1_TCS20() 6 ms
 ✓ testAddressOperation1_TCS21() 11 ms

Process finished with exit code 0

TC_ADMINCATALOG



The screenshot shows the IntelliJ IDEA interface with the project 'servlet' open. The code editor displays a Java test class 'TC_AdminCatalog.java'. The test methods 'testConsumerCatalog1()' through 'testConsumerCatalog4()' are shown with their implementations. The test results pane at the bottom indicates 4 tests passed in 2 seconds.

```
    ArgumentCaptor<String> captor= ArgumentCaptor.forClass(String.class);
    servlet.init(sc);
    servlet doGet(request, response);

    verify(rd).forward(request, response);
    verify(response).encodeURL(captor.capture());
    assertEquals( expected: "catalogoAdm.jsp", captor.getValue());
}

@Test
public void testConsumerCatalog3() throws Exception {
    when(request.getSession()).thenReturn(session);
    when(session.getAttribute("isAdmin")).thenReturn(true);
    when(request.getParameter("ctgy")).thenReturn("");
    when(request.getParameter("srch")).thenReturn("");
    when(request.getParameter("pg")).thenReturn("1");
    when(prodottoDao.doRetrieveAll( acquistabile: true)).thenReturn(lista);
    when(categoriaDao.findAll()).thenReturn(categorie);
}
```

Run: TC_AdminCatalog x
Tests passed: 4 (moments ago)

Test Results
TC_AdminCatalog
 ✓ testConsumerCatalog1() 2s 12 ms
 ✓ testConsumerCatalog2() 1s 966 ms
 ✓ testConsumerCatalog3() 14 ms
 ✓ testConsumerCatalog4() 27 ms

Process finished with exit code 0

TC_ADMINMANAGER

```
a-Disney > src > test > servlet > TC_AdminManager > testAdminManager
```

```
ProductModel.java | CartServlet.java | TC_CartServlet.java | TC_Billing.java | TC_AdminManager.java
```

```
Project < Run < Git < Favorites
```

```
when(request.getSession()).thenReturn(session);
when(session.getAttribute("isAdmin")).thenReturn(true);
when(request.getParameter("search")).thenReturn("rosalia@libero.it");
when(request.getParameter("da")).thenReturn("2020-8-1");
when(request.getParameter("a")).thenReturn("2020-8-4");
when(request.getParameter("pg")).thenReturn("1");
when(fatturaDao.retrieveInvoices(da, a)).thenReturn(list);
when(fatturaDao.retrieveInvoices(r, da, a)).thenReturn(list);
when(paginator.paginate(list)).thenReturn(pair1);
when(response.encodeURL("/ordini.jsp")).thenReturn("/ordini.jsp");
when(request.getRequestDispatcher("/ordini.jsp")).thenReturn(rd);

ArgumentCaptor<String> captor= ArgumentCaptor.forClass(String.class);
servlet doGet(request, response);
verify(rd).forward(request, response);
verify(response).encodeURL(captor.capture());
assertEquals( expected: "/ordini.jsp", captor.getValue());
```

```
Run: > TC_AdminManager > Test Results > TC_AdminManager > testAdminManager()
Tests passed: 1 of 1 test - 2 s 9 ms
2 s 9 ms C:\Users\utente\.jdks\openjdk-15.0.2\bin\java.exe ...
Process finished with exit code 0
```

```
85:8 CRLF UTF-8 4 spaces P master
```

TC_BILLING

```
a-Disney > src > test > servlet > TC_Billing > testBilling2
```

```
ProductModel.java | CartServlet.java | TC_CartServlet.java | TC_Billing.java
```

```
Project < Run < Git < Favorites
```

```
ArgumentCaptor<String> captor= ArgumentCaptor.forClass(String.class);
servlet doGet(request, response);
verify(response).encodeURL(captor.capture());
assertEquals( expected: "Login.jsp", captor.getValue());
```

```
@Test
public void testBilling2() throws Exception {
    when(request.getSession()).thenReturn(session);
    when(request.getSession().getAttribute("cart")).thenReturn(c);
    when(request.getSession().getAttribute("user")).thenReturn(r);
    when(request.getRequestDispatcher("AddressOperations?operation=0")).thenReturn(rd1);
    when(response.encodeURL("Checkout.jsp")).thenReturn("Checkout.jsp");
    when(request.getRequestDispatcher("Checkout.jsp")).thenReturn(rd2);

    ArgumentCaptor<String> captor= ArgumentCaptor.forClass(String.class);
    servlet doGet(request, response);
```

```
Run: > TC_Billing > Test Results > TC_Billing > testBilling1()
Tests passed: 2 of 2 tests - 1 s 223 ms
1 s 223 ms C:\Users\utente\.jdks\openjdk-15.0.2\bin\java.exe ...
Process finished with exit code 0
```

```
71:9 CRLF UTF-8 4 spaces P master
```

TC_CARTSERVLET

The screenshot shows the IntelliJ IDEA interface with the following details:

- File menu:** File, Edit, View, Navigate, Code, Analyze, Refactor, Build, Run, Tools, Git, Window, Help.
- Project tree:** A package named `servlet` containing numerous test classes starting with `TC_`, such as `TC_Prova`, `TC_AddressOperations`, `TC_AdminCatalog`, etc.
- Code editor:** The `TC_CartServlet.java` file is open, showing a test method `testCartServlet()` using Mockito annotations to verify session attribute retrieval and dispatcher handling.
- Run tool window:** Shows a single test run for `TC_CartServlet` with a duration of 1 ms and a success status.
- Status bar:** Displays "Tests passed: 1" and other system information like file encoding and branch.

TC_CHECKOUT

File Edit View Navigate Code Analyze Refactor Build Run Tools Git Window Help a-Disney - TC_Checkout.java

a-Disney | src | test | servlet | TC_Checkout | testCheckout2

Project

- TC_ProductModel
- TC_RecensioneModel
- TC_RegisteredModel
- TC_UserModel
- servlet
 - prova
 - TC_AddressOperations
 - TC_AdminCatalog
 - TC_AdminManager
 - TC_Billing
 - TC_CartServlet
 - TC_Checkout
 - TC_ConsumerCatalog
 - TC_FatturaPDF
 - TC_IndexServlet
 - TC_Login
 - TC_Logout
 - TC_ProductAdminControl
 - TC_ProductControl
 - TC_RecensioneServlet
- TC_P...

TC_Checkout.java

```
83
84
85 @Test
86
87     public void testCheckout2() throws Exception {
88
89         when(request.getSession()).thenReturn(session);
90         when(session.getAttribute("isUser")).thenReturn(true);
91
92         when(session.getAttribute("addresses")).thenReturn(addresses);
93         when(session.getAttribute("cart")).thenReturn(c);
94         when(request.getParameter("address")).thenReturn("0");
95         when(addresses.get(0)).thenReturn(a);
96         when(session.getAttribute("user")).thenReturn(e);
97         when(response.encodeURL("Thanks.jsp")).thenReturn("Thanks.jsp");
98
99         ArgumentCaptor<String> captor = ArgumentCaptor.forClass(String.class);
100        servlet.doGet(request, response);
101        verify(response).sendRedirect(captor.capture());
102        assertEquals(expected: "Thanks.jsp", captor.getValue());
103    }
104}
```

Run: TC_Checkout

Test Results

- Test Results 2 s 296 ms C:\Users\utente\.jdks\openjdk-15.0.2\bin\java.exe ...
- TC_Checkout 2 s 296 ms
- testCheckout1 1 s 447 ms
- testCheckout2 849 ms

Process finished with exit code 0

Favorites

GitHub Run TODO Problems Terminal Profiler Build Services

Tests passed: 2 (moments ago)

Event Log master

TC_CONSUMERCATALOG

The screenshot shows the IntelliJ IDEA interface with the project 'a-Disney' open. The code editor displays `testConsumerCatalog.java` containing a test method `testConsumerCatalog1()`. The test uses Mockito annotations to verify various parameters and responses. The 'Run' tool window shows the test results: 3 tests passed in 2s 53ms. The terminal output shows the command `C:\Users\utente\.jdks\openjdk-15.0.2\bin\java.exe ...` and the message "Process finished with exit code 0".

```
public void testConsumerCatalog1() throws Exception {
    when(request.getParameter("city")).thenReturn("0");
    when(request.getParameter("search")).thenReturn("");
    when(request.getParameter("pg")).thenReturn("1");
    when(prodottoDao.doRetrieveByCategory(0)).thenReturn(lista);
    when(p.paginate(lista)).thenReturn(pair);
    when(response.encodeURL("Catalogo.jsp")).thenReturn("Catalogo.jsp");
    when(request.getRequestDispatcher("Catalogo.jsp")).thenReturn(rd);

    servlet.init(sc);
    servlet.doGet(request, response);
    verify(rd).forward(request, response);
    ArgumentCaptor<String> captor = ArgumentCaptor.forClass(String.class);
    verify(response).encodeURL(captor.capture());
    assertEquals(expected: "Catalogo.jsp", captor.getValue());
}
```

Test Results:

- TC_ConsumerCatalog
- testConsumerCatalog1()
- testConsumerCatalog2()
- testConsumerCatalog3()

Process finished with exit code 0

TC_INDEXSERVLET

The screenshot shows the IntelliJ IDEA interface with the project 'a-Disney' open. The code editor displays `TC_IndexServlet.java` containing a test method `testIndexServlet2()`. The test uses Mockito annotations to verify session retrieval, product discount retrieval, and request dispatcher. The 'Run' tool window shows the test results: 2 tests passed in 2s 43ms. The terminal output shows the command `C:\Users\utente\.jdks\openjdk-15.0.2\bin\java.exe ...` and the message "Process finished with exit code 0".

```
when(request.getSession()).thenReturn(session);
when(productDao.doRetrieveByDiscount(amount: 30, acquista: true)).thenReturn(list);
assertThrows(Exception.class, ()->servlet.doPost(request, response));
}

@Test //TCS
public void testIndexServlet2() throws Exception {
    when(request.getSession()).thenReturn(session);
    when(productDao.doRetrieveByDiscount(amount: 30, acquista: true)).thenReturn(list);
    when(request.getRequestDispatcher("/contentJSP/ProductCard.jsp")).thenReturn(rd);

    ArgumentCaptor<String> captor = ArgumentCaptor.forClass(String.class);
    servlet.doPost(request, response);
    verify(request).getDispatcher(captor.capture());
    assertEquals(expected: "/contentJSP/ProductCard.jsp", captor.getValue());
}
```

Test Results:

- TC_IndexServlet
- testIndexServlet()
- testIndexServlet2()

Process finished with exit code 0

TC_LOGIN

The screenshot shows the IntelliJ IDEA interface with the project 'a-Disney' open. The code editor displays the `TC_Login.java` file, which contains several test methods for the login functionality. The test results in the bottom panel show all 5 tests passed in 1s 869ms.

```
a-Disney > src > test > servlet > TC_Login > testLogin4
public void testLogin5() throws Exception {
    when(request.getParameter("username")).thenReturn("rosalia@libero.it");
    when(request.getParameter("password")).thenReturn("rosalia");
    when(request.getSession()).thenReturn(session);
    when(userDao.login(username: "rosalia@libero.it", password: "rosalia")).thenReturn(userBean);
    ArgumentCaptor<String> captor= ArgumentCaptor.forClass(String.class);
    servlet.doPost(request, response);
    verify(response).encodeURL(captor.capture());
    assertEquals(expected: "/Login.jsp", captor.getValue());
}
```

Test Results:

- TC_Login
- ✓ testLogin1()
- ✓ testLogin2()
- ✓ testLogin3()
- ✓ testLogin4()
- ✓ testLogin5()

Process finished with exit code 0

TC_LOGOUT

The screenshot shows the IntelliJ IDEA interface with the project 'a-Disney' open. The code editor displays the `TC_Logout.java` file, which contains a single test method for the logout functionality. The test results in the bottom panel show 1 test passed in 1s 138ms.

```
a-Disney > src > test > servlet > TC_Logout > testLogout
@MockitoAnnotations.initMocks(this);
when(request.getSession()).thenReturn(session);

ArgumentCaptor<String> captor= ArgumentCaptor.forClass(String.class);
servlet doGet(request, response);
verify(response).encodeURL(captor.capture());
assertEquals(expected: "index.jsp", captor.getValue());
```

Test Results:

- TC_Logout
- ✓ testLogout()

Process finished with exit code 0

TC_PRODUCTADMINCONTROL

The screenshot shows the IntelliJ IDEA interface with the project 'a-Disney' open. The code editor displays the file `TC_ProductAdminController.java`. The method `testPACDelete()` is highlighted. The code uses the Mockito framework to simulate request parameters and verify the response. The test passes with 5 tests in 2s 77ms.

```
a-Disney > src > test > servlet > TC_ProductAdminController > testPACDelete
```

```
    when(request.getParameter("act")).thenReturn("insert");
    when(request.getParameter("name")).thenReturn("");
    when(request.getParameter("description")).thenReturn("");
    when(request.getParameter("personaggio")).thenReturn("aurora");
    when(request.getParameter("url")).thenReturn("images/auroraCuscino.PNG");
    when(request.getParameter("titolo")).thenReturn("cuscino");
    when(request.getParameter("categoria")).thenReturn("0");
    when(request.getParameter("qty")).thenReturn("3");
    when(request.getParameter("price")).thenReturn("");
    when(request.getParameter("iva")).thenReturn("5.00");
    when(request.getParameter("sconto")).thenReturn("35.00");

    ArgumentCaptor<Integer> captor = ArgumentCaptor.forClass(Integer.class);
    servlet.doPost(request, response);
    verify(response).sendError(captor.capture());
    assertEquals(expected: 406, captor.getValue());
```

Run: TC_ProductAdminController > Test Results

Test	Time	Java Exec
TC_ProductAdminController	2s 77 ms	C:\Users\utente\.jdks\openjdk-15.0.2\bin\java.exe ...
testPACDelete()	1s 836 ms	
testPAC240	36 ms	
testPAC250	6 ms	
testPAC260	9 ms	
testPAC270	190 ms	

Process finished with exit code 0

TC_PRODUCTCONTROL

The screenshot shows the IntelliJ IDEA interface with the project 'a-Disney' open. The code editor displays the file `TC_ProductControl.java`. The method `testAddC()` is highlighted. The code uses the Mockito framework to simulate session attributes and product retrieval. The test passes with 3 tests in 2s 283ms.

```
a-Disney > src > test > servlet > TC_ProductControl > testView
```

```
    when(request.getSession()).thenReturn(session);
    Cart c = new Cart();
    when((Cart)request.getSession().getAttribute("cart")).thenReturn(c);
    when(request.getParameter("act")).thenReturn("addC");
    when(request.getParameter("id")).thenReturn("3");
    when(request.getParameter("qty")).thenReturn("2");

    StringWriter stringWriter = new StringWriter();
    PrintWriter writer = new PrintWriter(stringWriter);
    when(response.getWriter()).thenReturn(writer);
    ProductBean b = new ProductBean();
    when(productDao.doRetrieveByKey(code: 3, acquistabile: true)).thenReturn(b);

    servlet doGet(request, response);
```

Run: TC_ProductControl > Test Results

Test	Time	Java Exec
TC_ProductControl	2s 283 ms	C:\Users\utente\.jdks\openjdk-15.0.2\bin\java.exe ...
testAddC()	2s 257 ms	
testView()	12 ms	
testDelete()	14 ms	

Process finished with exit code 0

TC_RECENSIONESERVLET

The screenshot shows the IntelliJ IDEA interface with the project 'a-Disney' open. The code editor displays a Java test class 'TC_RecensioneServlet.java'. The test method 'testRecensioneView()' is annotated with '@Test'. The run tool window at the bottom shows the test results: 'Test Results' with 'TC_RecensioneServlet' expanded, showing 'testRecensioneInsert()' and 'testRecensioneView()' both passed. The status bar indicates 'Tests passed: 2 (moments ago)'. The code editor has syntax highlighting and code completion features.

```
a-Disney > src > test > servlet > TC_RecensioneServlet > testRecensioneInsert
```

```
Project: a-Disney
File Edit View Navigate Code Analyze Refactor Build Run Tools Git Window Help a-Disney - TC_RecensioneServlet.java
```

```
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
```

```
@Test //TCS
public void testRecensioneView() throws Exception {
    when(request.getSession()).thenReturn(session);
    when(request.getParameter("act")).thenReturn("view");
    when(request.getAttribute("product")).thenReturn(b);
    when(request.getSession().getAttribute("user")).thenReturn(r);
    when(recensioneDao.getComments(b)).thenReturn(list);
    when(recensioneDao.userComment(r, b)).thenReturn(recensioneBean);
    when(fatturaDao.hasPurchased(b, r)).thenReturn(true);
    when(recensioneDao.mediumVote(b)).thenReturn(4.5);

    servlet.doGet(request, response);
    verify(request).setAttribute("payed", false);
}
```

```
@Test
```

```
Run: TC_RecensioneServlet
Test Results: 2s 749 ms C:\Users\utente\.jdks\openjdk-15.0.2\bin\java.exe ...
  ✓ Test Results
    ✓ TC_RecensioneServlet
      ✓ testRecensioneInsert() 2s 723 ms
      ✓ testRecensioneView() 26 ms
Process finished with exit code 0
```

```
Git Run TODO Problems Terminal Profiler Build Services Event Log master
```

TC_REGISTRATION

The screenshot shows the IntelliJ IDEA interface with the project 'a-Disney' open. The code editor displays a Java test class 'TC_Registration.java'. The test method 'testRegistration6()' is annotated with '@Test'. The run tool window at the bottom shows the test results: 'Test Results' with 'TC_Registration' expanded, showing 'testRegistration6()' passed. The status bar indicates 'Tests passed: 1 (moments ago)'. The code editor has syntax highlighting and code completion features.

```
a-Disney > src > test > servlet > TC_Registration
```

```
File Edit View Navigate Code Analyze Refactor Build Run Tools Git Window Help a-Disney - TC_Registration.java
```

```
MockitoAnnotations.initMocks(testClass);
```

```
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
```

```
@Test
public void testRegistration6() throws Exception {
    when(request.getParameter("surname")).thenReturn("chicco");
    when(request.getParameter("email")).thenReturn("chicco@alice.it");
    when(request.getParameter("name")).thenReturn("chicco");
    when(request.getParameter("password")).thenReturn("chicco");
    when(request.getSession()).thenReturn(session);

    StringWriter stringWriter = new StringWriter();
    PrintWriter writer = new PrintWriter(stringWriter);
    when(response.getWriter()).thenReturn(writer);

    ArgumentCaptor<String> captor = ArgumentCaptor.forClass(String.class);
    servlet.doPost(request, response);
    verify(response).sendRedirect(captor.capture());
    assertEquals(expected: "index.jsp", captor.getValue());
}
```

```
Run: TC_Registration
Test Results: 3s 521 ms C:\Users\utente\.jdks\openjdk-15.0.2\bin\java.exe ...
  ✓ Test Results
    ✓ TC_Registration
      ✓ testRegistration6() 3s 521 ms
Process finished with exit code 0
```

```
Git Run TODO Problems Terminal Profiler Build Services Event Log master
```

TC_SUGGESTER

The screenshot shows the IntelliJ IDEA interface with the project 'a-Disney' open. The code editor displays `TC_Suggerster.java`. The code is a JUnit test for the `Suggerster` servlet. It uses Mockito annotations to mock `ProductBean` and `ProductModel`, and `MockitoAnnotations` to initialize mocks. The `@Test` method `testSuggerster()` checks if the search parameter is present and if the model retrieves the correct product based on the search term and acquistabile status.

```
a-Disney > src > test > servlet > TC_Suggerster > TC_Suggerster.java
...
    @Mock
    List<ProductBean> list;

    @Mock
    ProductModel model;

    @InjectMocks
    coreServlets.Suggerster servlet;

    @BeforeEach
    public void setUp() throws Exception {
        MockitoAnnotations.initMocks(this);
    }

    @Test //TCS
    public void testSuggerster() throws Exception {
        when(request.getParameter(s: "srch")).thenReturn("rosalia@libero.com");
        when(model.doRetrieveBySearch(search: "rosalia@libero.com", acquistabile: true)).thenReturn(list);

        StringWriter stringWriter = new StringWriter();
        PrintWriter writer = new PrintWriter(stringWriter);
        when(response.getWriter()).thenReturn(writer);
        servlet.init();
        servlet.service(request, response);
        String result = stringWriter.getBuffer().toString().trim();
        assertNotEqual(result);
    }
}
```

The run results show 1 test passed in 1s 246ms. The terminal output indicates the process finished with exit code 0.

TC_SUGGESTERUSER

The screenshot shows the IntelliJ IDEA interface with the project 'a-Disney' open. The code editor displays `TC_SuggersterUser.java`. The code is a JUnit test for the `SuggersterUser` servlet. It uses Mockito annotations to mock `Session` and `registeredDao`, and `MockitoAnnotations` to initialize mocks. The `@Test` method `testSuggersterUser()` checks if the search parameter is present, if the session is returned, and if the registered DAO retrieves the correct list of products.

```
a-Disney > src > test > servlet > TC_SuggersterUser > TC_SuggersterUser.java
...
    coreServlets.SuggersterUser servlet;

    @BeforeEach
    public void setUp() throws Exception {
        MockitoAnnotations.initMocks(this);
    }

    @Test //TCS
    public void testSuggersterUser() throws Exception {
        when(request.getParameter(s: "srch")).thenReturn("rosalia@libero.it");
        when(request.getSession()).thenReturn(session);
        when(registeredDao.doRetrieveBySearch("rosalia@libero.it")).thenReturn(list);

        StringWriter stringWriter = new StringWriter();
        PrintWriter writer = new PrintWriter(stringWriter);
        when(response.getWriter()).thenReturn(writer);
        servlet.init();
        servlet.service(request, response);
        String result = stringWriter.getBuffer().toString().trim();
        assertNotEqual(result);
    }
}
```

The run results show 1 test passed in 1s 359ms. The terminal output indicates the process finished with exit code 0.

TC_USERMANAGER

The screenshot shows the IntelliJ IDEA interface with the project 'a-Disney' open. The 'Test Results' tool window is active, displaying the results of the 'TC_UserManager' test class. The test class contains two methods: 'test0' and 'test2'. Both tests passed, with execution times of 2s 343 ms and 2s 431 ms respectively. The code editor shows the implementation of these tests, which interact with a session and a response object to verify URL encoding and session returns.

```
a-Disney src test servlet TC_UserManager > test2

Project > model > servlet > prova > TC_AddressOperations > TC_AdminCatalog > TC_AdminManager > TC_Billing > TC_CartServlet > TC_Checkout > TC_ConsumerCatalog > TC_FatturaPDF > TC_IndexServlet > TC_Login > TC_Logout > TC_ProductAdminControl > TC_ProductControl > TC_ReclensioneServlet > TC_Registration > TC_Suggerente > TC_SuggerenteUser > TC_UserManager

public void test0() throws Exception {
    when(request.getSession()).thenReturn(session);
    when(request.getParameter("op")).thenReturn("viewFatture");
    when(session.getAttribute("user")).thenReturn(userBean);
    when(request.getParameter("pg")).thenReturn("1");
    when(request.getHeader("x-requested-with")).thenReturn(null);
    when(fatturaDao.retrieveInvoices(userBean, date: null, date2: null)).thenReturn(list);
    when(response.encodeURL("/ordiniUtente.jsp")).thenReturn("/OrdiniUtente.jsp");
    when(request.getRequestDispatcher("/OrdiniUtente.jsp")).thenReturn(rd);
}

@Test
public void test2() throws Exception {
    when(request.getSession()).thenReturn(session);
    when(request.getParameter("op")).thenReturn("viewFatture");
    when(session.getAttribute("user")).thenReturn(userBean);
    when(request.getParameter("pg")).thenReturn("1");
    when(request.getHeader("x-requested-with")).thenReturn(null);
    when(fatturaDao.retrieveInvoices(userBean, date: null, date2: null)).thenReturn(list);
    when(response.encodeURL("/ordiniUtente.jsp")).thenReturn("/OrdiniUtente.jsp");
    when(request.getRequestDispatcher("/OrdiniUtente.jsp")).thenReturn(rd);

    ArgumentCaptor<String> captor = ArgumentCaptor.forClass(String.class);
    servlet doGet(request, response);
    verify(request).getRequestDispatcher(captor.capture());
    assertEquals(expected: "/OrdiniUtente.jsp", captor.getValue());
}
```

Test Results: 2 Tests passed: 2 moments ago

Process finished with exit code 0

TEST DI INTEGRAZIONE

TC_ADDRESSOPERATIONINTEGRATION

The screenshot shows the IntelliJ IDEA interface with the project 'a-Disney' open. The 'Test Results' tool window is active, displaying the results of the 'TC_AddressOperationsIntegration' test class. The test class contains three methods: 'testAddressOperation0', 'testAddressOperation1_TCS180', and 'testAddressOperation1_TCS210'. All three tests passed, with execution times of 4s 19 ms, 3s 861 ms, and 146 ms respectively. The code editor shows the implementation of these tests, which use ArgumentCaptor to verify parameter values and session returns.

```
a-Disney src test integration > TC_AddressOperationsIntegration

Project > integration > prova > TC_AddressOperationsIntegration > TC_AdminCatalogIntegration > TC_AdminManagerIntegration > TC_BillingIntegration > TC_CartServletIntegration > TC_CheckoutIntegration > TC_ConsumerCatalogIntegration > TC_FatturaPDFIntegration > TC_LoginIntegration > model > servlet > prova > TC_AddressOperations > TC_AdminCatalog > TC_AdminManager > TC_Billing > TC_CartServlet

when(request.getParameter("via")).thenReturn("via tommassini");

ArgumentCaptor<String> captor = ArgumentCaptor.forClass(String.class);
servlet doGet(request, response);
verify(response).encodeURL(captor.capture());
assertEquals(expected: "error.jsp", captor.getValue());

@Test //Op1 TCS21
public void testAddressOperation1_TCS21() throws Exception {
    when(request.getParameter("operation")).thenReturn("");
    when(request.getParameter("cap")).thenReturn("84061");
    when(request.getParameter("città")).thenReturn("Fellitto");
    when(request.getParameter("stato")).thenReturn("italia");
    when(request.getParameter("ncv")).thenReturn("7");
    when(request.getParameter("provincia")).thenReturn("Salerno");
    when(request.getParameter("via")).thenReturn("via tommassini");
    when(request.getSession()).thenReturn(session);
}
```

Test Results: 3 Tests passed: 3 moments ago

Process finished with exit code 0

TC_ADMINCATALOGINTEGRATION

The screenshot shows the IntelliJ IDEA interface with the project structure on the left and the code editor on the right. The code editor displays a Java test class `TC_AdminCatalogIntegration.java` containing several test methods for the `testConsumerCatalog` suite. The test results in the bottom panel show four tests passed, each taking between 15 ms and 221 ms.

```
when(request.getParameter("pg")).thenReturn("1");
when(response.encodeURL("catalogoAdm.jsp")).thenReturn("catalogoAdm.jsp");

when(request.getRequestDispatcher("catalogoAdm.jsp")).thenReturn(rd);
ArgumentCaptor<String> captor= ArgumentCaptor.forClass(String.class);
servlet.init(sc);
servlet doGet(request, response);

verify(rd).forward(request, response);
verify(response).encodeURL(captor.capture());
assertEquals(expected: "catalogoAdm.jsp",captor.getValue());

@Test
public void testConsumerCatalog4() throws Exception {
    when(request.getSession()).thenReturn(session);
    when(session.getAttribute("isAdmin")).thenReturn(false);

    ArgumentCaptor<String> captor= ArgumentCaptor.forClass(String.class);
```

Tests passed: 4

Test Results

- TC_AdminCatalogIntegration
 - testConsumerCatalog()
 - testConsumerCatalog2()
 - testConsumerCatalog3()
 - testConsumerCatalog4()

Process finished with exit code 0

TC_ADMINMANAGERINTEGRATION

The screenshot shows the IntelliJ IDEA interface with the project structure on the left and the code editor on the right. The code editor displays a Java test class `TC_AdminManagerIntegration.java` containing one test method `testAdminManager`. The test results in the bottom panel show one test passed, taking 227 ms.

```
public void testAdminManager() throws Exception {
    when(request.getSession()).thenReturn(session);
    when(session.getAttribute("isAdmin")).thenReturn(true);
    when(request.getParameter("search")).thenReturn("rosalia@libero.it");
    when(request.getParameter("da")).thenReturn("2020-8-1");
    when(request.getParameter("a")).thenReturn("2020-8-4");
    when(request.getParameter("pg")).thenReturn("1");
    when(response.encodeURL("/Ordini.jsp")).thenReturn("/Ordini.jsp");
    when(request.getRequestDispatcher("/Ordini.jsp")).thenReturn(rd);

    ArgumentCaptor<String> captor= ArgumentCaptor.forClass(String.class);
    servlet doGet(request, response);
    verify(rd).forward(request, response);
    verify(response).encodeURL(captor.capture());
    assertEquals(expected: "/Ordini.jsp",captor.getValue());
```

Tests passed: 1

Test Results

- TC_AdminManagerIntegration
 - testAdminManager()

Process finished with exit code 0

TC_BILLINGINTEGRATION

The screenshot shows the IntelliJ IDEA interface with the project 'a-Disney' open. The code editor displays `TC_BillingIntegration.java` containing a test method `testBilling`. The test uses the MockServer framework to verify a servlet response. The run results show 2 tests passed in 1s 401ms.

```
a-Disney > src > test > integration > TC_BillingIntegration > testBilling2
ProductBean prodotto= new ProductBean();
prodotto.setCode(5);
c.addOrder(new Order(prodotto, quantity: 4));
Registered r = new Registered();
when(request.getSession()).thenReturn(session);
when(request.getAttribute("cart")).thenReturn(c);
when(request.getSession().getAttribute("user")).thenReturn(r);
when(request.getRequestDispatcher("AddressOperations?operation=0")).thenReturn(rd1);
when(response.encodeURL("Checkout.jsp")).thenReturn("Checkout.jsp");
when(request.getRequestDispatcher("Checkout.jsp")).thenReturn(rd2);

ArgumentCaptor<String> captor= ArgumentCaptor.forClass(String.class);
servlet doGet(request, response);
verify(response).encodeURL(captor.capture());
assertEquals(expected: "Checkout.jsp", captor.getValue());
}

Run: TC_BillingIntegration <
Tests passed: 2 of 2 tests - 1s 401 ms
Test Results
TC_BillingIntegration
  ✓ testBilling1()
  ✓ testBilling2()
Process finished with exit code 0
Tests passed: 2 (moments ago)
```

TC_CARTSERVLETINTEGRATION

The screenshot shows the IntelliJ IDEA interface with the project 'a-Disney' open. The code editor displays `TC_CartServletIntegration.java` containing a test method `testCartServlet`. The test uses the MockServer framework to verify a servlet response. The run results show 1 test passed in 2s 306ms.

```
a-Disney > src > test > integration > TC_CartServletIntegration > testCartServlet
    }
    @Test
    public void testCartServlet() throws Exception {
        Cart c= new Cart();
        ProductBean prodotto= new ProductBean();
        prodotto.setCode(5);
        c.addOrder(new Order(prodotto, quantity: 4));

        when(request.getSession()).thenReturn(session);
        when(session.getAttribute("cart")).thenReturn(c);
        ArgumentCaptor<String> captor= ArgumentCaptor.forClass(String.class);
        when(request.getRequestDispatcher("/contentJSP/cartContent.jsp")).thenReturn(rd);
        servlet doGet(request, response);
        verify(request).getRequestDispatcher(captor.capture());
        assertEquals(expected: "/contentJSP/cartContent.jsp", captor.getValue());
    }

Run: TC_CartServletIntegration <
Tests passed: 1 of 1 test - 2s 306 ms
Test Results
TC_CartServletIntegration
  ✓ testCartServlet()
Process finished with exit code 0
Tests passed: 1 (moments ago)
```

TC_CHECKOUTINTEGRATION

The screenshot shows the IntelliJ IDEA interface with the project navigation bar at the top. Below it is the code editor with a Java file named `TC_CheckoutIntegration.java`. The code contains a series of `when` and `thenReturn` statements for a mockMvc setup. The `Run` tool window at the bottom shows the test results for `TC_CheckoutIntegration`, indicating 2 tests passed in 2 seconds.

```
a-Disney > src > test > integration > TC_CheckoutIntegration > TC_CheckoutIntegration.java
```

```
addresses.put(0,a);  
|  
when(request.getSession()).thenReturn(session);  
when(session.getAttribute("isUser")).thenReturn(true);  
when(session.getAttribute("addresses")).thenReturn(addresses);  
when(session.getAttribute("cart")).thenReturn(c);  
when(request.getParameter("address")).thenReturn("0");  
when(session.getAttribute("user")).thenReturn(e);  
when(response.encodeURL("Thanks.jsp")).thenReturn("Thanks.jsp");  
  
ArgumentCaptor<String> captor= ArgumentCaptor.forClass(String.class);  
servlet doGet(request, response);  
verify(response).sendRedirect(captor.capture());  
assertEquals(expected: "Thanks.jsp", captor.getValue());  
}
```

Run: TC_CheckoutIntegration >
Test Results: 2s 354 ms C:\Users\utente\.jdks\openjdk-15.0.2\bin\java.exe ...
TC_CheckoutIntegration 2s 354 ms
testCheckout1 1s 235 ms
testCheckout2 1s 119 ms
Process finished with exit code 0

Tests passed: 2 (moments ago)

TC_CONSUMERCATALOGINTEGRATION

The screenshot shows the IntelliJ IDEA interface with the project navigation bar at the top. Below it is the code editor with a Java file named `TC_ConsumerCatalogIntegration.java`. The code contains two test methods: `testConsumerCatalog1` and `testConsumerCatalog2`. The `Run` tool window at the bottom shows the test results for `TC_ConsumerCatalogIntegration`, indicating 3 tests passed in 2 seconds.

```
a-Disney > src > test > integration > TC_ConsumerCatalogIntegration > TC_ConsumerCatalogIntegration.java
```

```
public void testConsumerCatalog1() throws Exception {  
    when(request.getParameter("ctgy")).thenReturn("0");  
    when(request.getParameter("srch")).thenReturn("");  
    when(request.getParameter("pg")).thenReturn("1");  
    when(response.encodeURL("Catalogo.jsp")).thenReturn("Catalogo.jsp");  
    when(request.getRequestDispatcher("Catalogo.jsp")).thenReturn(rd);  
  
    servlet.init(sc);  
    servlet doGet(request, response);  
    verify(rd).forward(request, response);  
    ArgumentCaptor<String> captor= ArgumentCaptor.forClass(String.class);  
    verify(response).encodeURL(captor.capture());  
    assertEquals(expected: "Catalogo.jsp", captor.getValue());  
}  
  
@Test  
public void testConsumerCatalog2() throws Exception {  
}
```

Run: TC_ConsumerCatalogIntegration >
Test Results: 2s 151 ms C:\Users\utente\.jdks\openjdk-15.0.2\bin\java.exe ...
TC_ConsumerCatalogIntegration 2s 151 ms
testConsumerCatalog1 2s 111 ms
testConsumerCatalog2 13 ms
testConsumerCatalog3 27 ms
Process finished with exit code 0

Tests passed: 3

TC_INDEXSERVLETINTEGRATION

The screenshot shows the IntelliJ IDEA interface with the project navigation bar at the top. The left sidebar displays the project structure, showing a package named 'a-Disney' containing 'src' and 'test' directories. Under 'src', there are 'META-INF', 'coreModels', 'coreServices', 'json', and 'test' packages. The 'test' package contains 'integration' and 'prova' sub-packages, with 'TC_IndexServletIntegration' being the active file. The code editor shows Java test code using Mockito annotations. The 'Run' tab at the bottom shows a successful run of the tests, with 2 tests passed in 1s.

```
42     @InjectMocks
43     coreServices.IndexServlet servlet;
44
45     @BeforeEach
46     public void setUp() throws Exception {
47         MockitoAnnotations.initMocks(this);
48     }
49
50     @Test //TCS
51     public void testIndexServlet() throws Exception {
52         when(request.getSession()).thenReturn(session);
53         assertThrows(Exception.class, ()->servlet.doPost(request, response));
54     }
55
56     @Test //TCS
57     public void testIndexServlet2() throws Exception {
58         when(request.getSession()).thenReturn(session);
59         when(request.getRequestDispatcher( s: "/contentJSP/ProductCard.jsp")).thenReturn(rd);
60
61         ArgumentCaptor<String> captor= ArgumentCaptor.forClass(String.class);
62         servlet.doPost(request, response);
63         verify(request).getRequestDispatcher(captor.capture());
64     }

```

TC_LOGININTEGRATION

The screenshot shows the IntelliJ IDEA interface with the project navigation bar at the top. The left sidebar displays the project structure, showing a package named 'a-Disney' containing 'src' and 'test' directories. Under 'src', there are 'META-INF', 'coreModels', 'coreServices', 'json', and 'test' packages. The 'test' package contains 'integration' and 'prova' sub-packages, with 'TC_LoginIntegration' being the active file. The code editor shows Java test code using Mockito annotations. The 'Run' tab at the bottom shows a successful run of the tests, with 5 tests passed in 2s.

```
85     ArgumentCaptor<String> captor= ArgumentCaptor.forClass(String.class);
86     servlet.doPost(request, response);
87     verify(response).encodeURL(captor.capture());
88     assertEquals( expected: "/Login.jsp", captor.getValue());
89 }
90
91 @Test //TCS5
92 public void testLogin5() throws Exception {
93     when(request.getParameter( s: "username")).thenReturn("rosalia@libero.it");
94     when(request.getParameter( s: "password")).thenReturn("rosalia");
95     when(request.getSession()).thenReturn(session);
96
97     ArgumentCaptor<String> captor= ArgumentCaptor.forClass(String.class);
98     servlet.doPost(request, response);
99     verify(response).encodeURL(captor.capture());
100    assertEquals( expected: "index.jsp", captor.getValue());
101 }
102 }
```

TC_LOGOUTINTEGRATION

The screenshot shows the IntelliJ IDEA interface with the project navigation bar at the top. Below it is the code editor with the file `TC_LogoutIntegration.java`. The code is a JUnit test for a logout integration. It uses Mockito to mock `HttpServletResponse` and `HttpSession`, and `@InjectMocks` to inject a `Logout servlet`. The `@BeforeEach` method sets up the mocks. The `@Test` method `testLogout()` checks if the response URL is correctly encoded after a logout.

```
37     @Mock
38     HttpServletResponse response;
39
40     @Mock
41     HttpSession session;
42
43     @InjectMocks
44     coreServlets.Logout servlet;
45
46     @BeforeEach
47     public void setUp() throws Exception {
48         MockitoAnnotations.initMocks(this);
49     }
50
51     @Test //TCS
52     public void testLogout() throws Exception {
53         when(request.getSession()).thenReturn(session);
54
55         ArgumentCaptor<String> captor = ArgumentCaptor.forClass(String.class);
56         servlet.doGet(request, response);
57         verify(response).encodeURL(captor.capture());
58     }
59
60 }
```

The run results show one test passed in 664 ms.

TC_PRODUCTADMINCONTROLINTEGRATION

The screenshot shows the IntelliJ IDEA interface with the project navigation bar at the top. Below it is the code editor with the file `TC_ProductAdminControlIntegration.java`. The code is a JUnit test for a product admin control integration. It uses Mockito to mock `StringWriter` and `PrintWriter`, and `@Test` to perform a delete operation. The test passes in 2 s 362 ms.

```
127     when(response.getWriter()).thenReturn(writer);
128
129     servlet.doPost(request, response);
130     String result = stringWriter.getBuffer().toString().trim();
131     assertNotNull(result);
132
133
134     @Test //TCS Eliminazione
135     public void testPACDelete() throws Exception {
136
137         when(request.getSession()).thenReturn(session);
138         when(session.getAttribute("isAdmin")).thenReturn(true);
139         when(request.getParameter("act")).thenReturn("delete");
140         when(request.getParameter("id")).thenReturn("1");
141
142         StringWriter stringWriter = new StringWriter();
143         PrintWriter writer = new PrintWriter(stringWriter);
144
145     }
146
147 }
```

The run results show five tests passed in 2 s 362 ms.

TC_PRODUCTCONTROLINTEGRATION

```
a-Disney > src > test > integration > TC_ProductControlIntegration > testDelete
```

```
Project TC_ProductControlIntegration
```

```
TC_IndexServletIntegration
TC_LoginIntegration
TC_LogoutIntegration
TC_ProductAdminControlIntegration
TC_ProductControlIntegration
TC_UserManagerIntegration
```

```
model
servlet
 prova
TC_AddressOperations
TC_AdminCatalog
TC_AdminManager
TC_Billing
TC_CartServlet
TC_Checkout
TC_ConsumerCatalog
TC_FatturaPDF
TC_IndexServlet
TC_Login
TC_Logout
TC_ProductAdminController
TC_ProductControl
```

```
TC_ProductControlIntegration.java
```

```
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
```

```
@Test //TCS
public void testAddC() throws Exception {
    Cart c = new Cart();
    when(request.getSession()).thenReturn(session);
    when((Cart)request.getAttribute("cart")).thenReturn(c);
    when(request.getParameter("act")).thenReturn("addC");
    when(request.getParameter("id")).thenReturn("3");
    when(request.getParameter("qty")).thenReturn("2");

    StringWriter stringWriter = new StringWriter();
    PrintWriter writer = new PrintWriter(stringWriter);
    when(response.getWriter()).thenReturn(writer);

    servlet.doGet(request, response);

    String result = stringWriter.getBuffer().toString().trim();
    assertEquals(result, "Cart updated successfully!");
}
```

```
Run: TC_ProductControlIntegration
```

```
Test Results 1s 286 ms /Library/Java/JavaVirtualMachines/jdk-15.0.2.jdk/Contents/Home/bin/java ...
```

```
TC_ProductControlIntegration 1s 286 ms
  ✓ testAddC() 1s 257 ms
  ✓ testView() 13 ms
  ✓ testDelete() 16 ms
```

```
Process finished with exit code 0
```

TC_RECENSIONESERVLETINTEGRATION

```
a-Disney > src > test > integration > TC_RecensioneServletIntegration > testRecensioneInsert
```

```
Project TC_RecensioneServletIntegration
```

```
TC_LoginIntegration
TC_LogoutIntegration
TC_ProductAdminControlIntegration
TC_ProductControlIntegration
TC_RecensioneServletIntegration
```

```
model
servlet
 web
.classpath
.gitattributes
.gitignore
.project
a-Disney.iml
fattura0.pdf
```

```
TC_RecensioneServletIntegration.java
```

```
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
```

```
    servlet.doGet(request, response);
    verify(request).setAttribute("payed", false);
}

@Test
public void testRecensioneInsert() throws Exception {
    ProductBean b = new ProductBean();
    Registered r = new Registered();
    r.setLogin("andrea@libero.it");
    when(request.getSession()).thenReturn(session);
    when(request.getParameter("act")).thenReturn("insert");
    when(request.getAttribute("product")).thenReturn(b);
    when(session.getAttribute("user")).thenReturn(r);
    when(request.getParameter("vote")).thenReturn("2");
    when(request.getParameter("comment")).thenReturn("Prodotto curato nei minimi dettagli");
    when(request.getParameter("id")).thenReturn("3");
}
```

```
Run: TC_RecensioneServletIntegration
```

```
Test Results 2s 275 ms C:\Users\utente\.jdks\openjdk-15.0.2\bin\java.exe ...
```

```
TC_RecensioneServletIntegration 2s 275 ms
  ✓ testRecensioneInsert() 2s 257 ms
  ✓ testRecensioneView() 18 ms
```

```
Process finished with exit code 0
```

```
Tests passed: 2
```

```
Git Run TODO Profiler Build Services
```

```
Tests passed: 2 (moments ago)
```

TC_REGISTRATIONINTEGRATION

The screenshot shows the IntelliJ IDEA interface with the project navigation bar at the top. The main area displays the code for `TC_RegistrationIntegration.java`. The code is a JUnit test for the `TC_RegistrationIntegration` class, which contains a single test method `testRegistration6()`. The test uses Mockito annotations to set up a `Request` object with parameters for surname, email, name, and password, and then verifies that the response.sendRedirect() method is called with the expected URL ("index.jsp"). The code editor has syntax highlighting and code completion features visible.

```
43     @Test
44     public void testRegistration6() throws Exception {
45         when(request.getParameter("surname")).thenReturn("chicco");
46         when(request.getParameter("email")).thenReturn("chicco@alice.it");
47         when(request.getParameter("name")).thenReturn("chicco");
48         when(request.getParameter("password")).thenReturn("chicco");
49         when(request.getSession()).thenReturn(session);
50
51         StringWriter stringWriter = new StringWriter();
52         PrintWriter writer = new PrintWriter(stringWriter);
53         when(response.getWriter()).thenReturn(writer);
54
55         ArgumentCaptor<String> captor = ArgumentCaptor.forClass(String.class);
56         servlet.doPost(request, response);
57         verify(response).sendRedirect(captor.capture());
58         assertEquals(expected: "index.jsp", captor.getValue());
59     }
60 }
```

The bottom panel shows the run results for the `TC_RegistrationIntegration` test, indicating 1 test passed in 2s 274ms. The terminal output shows the command used to run the test and the exit code 0.

TC_SUGGESTERINTEGRATION

The screenshot shows the IntelliJ IDEA interface with the project navigation bar at the top. The main area displays the code for `TC_SuggesterIntegration.java`. The code is a JUnit test for the `TC_SuggesterIntegration` class, which contains a single test method `testSuggester()`. The test uses Mockito annotations to set up a `Request` object with a parameter for "srgh" and then verifies that the response's buffer content is "rosalia@libero.com". The code editor has syntax highlighting and code completion features visible.

```
48
49
50     @Test //TCS
51     public void testSuggester() throws Exception {
52         when(request.getParameter("srgh")).thenReturn("rosalia@libero.com");
53
54         StringWriter stringWriter = new StringWriter();
55         PrintWriter writer = new PrintWriter(stringWriter);
56         when(response.getWriter()).thenReturn(writer);
57
58         servlet.init();
59         servlet.service(request, response);
60         String result = stringWriter.getBuffer().toString().trim();
61         assertNotNull(result);
62     }
63
64 }
```

The bottom panel shows the run results for the `TC_SuggesterIntegration` test, indicating 1 test passed in 1s 184ms. The terminal output shows the command used to run the test and the exit code 0.

TC_SUGGESTERUSERINTEGRATION

The screenshot shows the IntelliJ IDEA interface with the project navigation bar at the top. Below it is the code editor containing the `TC_SuggersterUserIntegration.java` file. The code defines a test method `testSuggersterUser()` that uses Mockito annotations to set up a search parameter and session, and then asserts that the resulting string writer buffer is not null. The run tool window at the bottom shows a successful test run with one test passed in 1 second.

```
39
40
41     @BeforeEach
42     public void setUp() throws Exception {
43         MockitoAnnotations.initMocks(testClass);
44     }
45
46     @Test //TCS
47     public void testSuggersterUser() throws Exception {
48         when(request.getParameter("srch")).thenReturn("rosalia@libero.it");
49         when(request.getSession()).thenReturn(session);
50
51         StringWriter stringWriter = new StringWriter();
52         PrintWriter writer = new PrintWriter(stringWriter);
53         when(response.getWriter()).thenReturn(writer);
54         servlet.init();
55         servlet.service(request, response);
56         String result = stringWriter.getBuffer().toString().trim();
57         assertNotEqual(result, "");
58     }
59 }
```

Run: TC_SuggersterUserIntegration x
Tests passed: 1 of 1 test – 1s 249ms

Test Results 1s 249 ms /Library/Java/JavaVirtualMachines/jdk-15.0.2.jdk/Contents/Home/bin/java ...
TC_SuggersterUser 1s 249 ms
testSuggersterUser 1s 249 ms
Process finished with exit code 0

TC_USERMANAGERINTEGRATION

The screenshot shows the IntelliJ IDEA interface with the project navigation bar at the top. Below it is the code editor containing the `TC_UserManagerIntegration.java` file. The code defines a test method `test2()` that sets up a registered user bean and performs a series of assertions related to session parameters and response URLs. The run tool window at the bottom shows a successful test run with two tests passed in 2 seconds.

```
75
76
77
78     @Test
79     public void test2() throws Exception {
80         Registered userBean= new Registered();
81
82         when(request.getSession()).thenReturn(session);
83         when(request.getParameter("op")).thenReturn("viewFatture");
84         when(request.getParameter("pg")).thenReturn("1");
85         when(session.getAttribute("user")).thenReturn(userBean);
86         when(response.encodeURL("/OrdiniUtente.jsp")).thenReturn("/OrdiniUtente.jsp");
87         when(request.getRequestDispatcher("/OrdiniUtente.jsp")).thenReturn(rd);
88
89         ArgumentCaptor<String> captor= ArgumentCaptor.forClass(String.class);
90         servlet doGet(request, response);
91         verify(response).encodeURL(captor.capture());
92         assertEquals(expected: "/OrdiniUtente.jsp", captor.getValue());
93     }
94 }
```

Run: TC_UserManagerIntegration x
Tests passed: 2 of 2 tests – 2s 342 ms

Test Results 2s 342 ms C:\Users\utente\.jdks\openjdk-15.0.2\bin\java.exe ...
TC_UserManagerIntegration 2s 342 ms
test10 2s 307 ms
test20 35 ms
Process finished with exit code 0

KATALON

GESTIONE UTENTE

LOGIN

login

- TCS4
- TCS5
- TCS3
- TCS2
- TCS1

⋮

REGISTRAZIONE

⋮ **Registrazione**

- TC7
- TC6
- TC10
- TC9
- TC8

⋮

CAMBIO PASSWORD

AGGIUNGI INDIRIZZO

The screenshot shows a mobile application interface with a dark background. At the top, there is a header labeled "CambioPassword" with three vertical dots on the right side. Below the header is a list of items, each preceded by a green circular bullet point. The items are: TC15, TC17, and TC16. Underneath this list is another header labeled "AggiungiIndirizzo*" with three vertical dots on the right side. Below this second header is another list of items, also preceded by green circular bullet points. The items are: TC18 *, TC21 *, TC20 *, and TC19 *. The text "TC19 *" is partially obscured by a vertical line on the left.

- TC15
- TC17
- TC16

AggiungiIndirizzo*

- TC18 *
- TC21 *
- TC20 *
- TC19 *

CAMBIO NOME

CAMBIO COGNOME

The screenshot shows a mobile application interface with a dark background. At the top, there is a header labeled "CambioNome" with three vertical dots on the right side. Below the header is a list of items, each preceded by a green circular bullet point. The items are: TC11 and TC12. Underneath this list is another header labeled "CambioCognome" with three vertical dots on the right side. Below this second header is another list of items, also preceded by green circular bullet points. The items are: TC14 and TC13. A vertical red bar is visible on the right side of the screen.

CambioNome

- TC11
- TC12

CambioCognome

- TC14
- TC13

GESTIONE PRODOTTO

AGGIUNGI RECENSIONE

MODIFICA PRODOTTO

AGGIUNGI NUOVO PRODOTTO

Aggiunta recensione

...

- TCS23
- TCS22

Modifica prodotto

...

- TCS24
- TCS25

Aggiunta nuovo prodotto

...

- TCS26
- TCS27