

Testing

Unit Test Case

Una volta conclusa la parte di implementazione si procede con la parte del testing in modo automatizzato con JUnit su NetBeans.

Unit Test Case: Contatto

UTC 1.1 Test Contatto-Contatto

Test items: classe Contatto, metodo Contatto()

Test :testContatto1

Case description: istanziare oggetto con entrambi i campi nomi e cognomi non vuoti

Location: ContattoTest.testContatto

Input : contatto1 = new Contatto("Giuseppe", "Russo");

Oracle: excpetedName=Giuseppe,
 excpetedSurname=Russo

Test:testContatto2

Case description: istanziare oggetto con solo campo nome (cogonme == null)

Location: ContattoTest.testContatto

Input : contatto2 = new Contatto("Andrea", null);

Oracle : expetedName="Andrea"
 excpetedSurname="null"

Test :testContatto3

Case description: istanzare oggetto con campo nome= null e campo cognome inizializzato

Location: ContattoTest.testContatto

Input: String expetedName=contatto3.getNome;

 String expected Surname=contatto 2.getCognome

Oracle :expetedName= "null"
 excpetedSurname="Bianchi"

UTC 1.2 Test Contatto-getNome

Test items: classe Contatto, metodo testGetNome

Test :testGetNome1

Case description:ottenuto il valore del campo nome

Location: ContattoTest.testGetNome

Input : String expName= contatto.getNome()

Oracle : String excpName:Giuseppe

Test:testGetNome2

Case description:

Location:ContattoTest.testContatto

Input : Stirng excpetedName=contatto2.getNome()

Oracle : String excpetedName=Andrea

UTC 1.3 Test Contatto-setNome

Test items: classe Contatto, metodo testSetNome

Test :testSetNome1

Case description:stabilito un valore al campo nome del contatto

Location: ContattoTest.testSetNome

Input : contatto1.setName("mario");

String expectedName=contatto1.getNome();

Oracle : String expectedName:mario

UTC 1.4 Test Contatto-getCognome

Test items: classe Contatto, metodo testGetCognome

Test :testGetCognome1

Case description:stabilito un valore al campo cognome

Location: ContattoTest.testGetCognome

Input : contatto1.getCognome("Russo);

String excpetedSurname=contatto1.getCognome();

Oracle : String expectedName=Russo

Test :testGetCognome2

Case description:stabilito un valore al campo cognome

Location: ContattoTest.testGetCognome

Input : contatto1.getCognome("Bianchi");

String excpetedSurname=contatto1.getCognome();

Oracle : String expectedName="Bianchi";

UTC 1.5 Test Contatto-setCognome

Test items: classe Contatto metodo testSetCognome

Test :testGetEmail1

Case description:ottenuto valore campo Email

Location: ContattoTest.testSetCognome

Input : String exSurname=contatto1.setCognome("rossi");

Oracle : exSurname="rossi"

UTC 1.6 Test Contatto-getEmail

Test items: classe Contatto metodo testGetEmail

Test :testGetEmail1

Case description:ottenuto valore campo Email

Location: List<String>email=new ArrayList<>();

email.add("mario@studenti.com");

contatto1.setEmail(email);

String expcetedEmail=contatto1.getEmail();

Oracle : exetedEmail=mario@studenti.com

UTC 1.7 Test Contatto-setEmail

Test items: classe Contatto metodo testSetEmail

Test :testGetEmail1

Case description:stabilito un valore campo Email

Location: List<String> email=new ArrayList<>();

email.add("mariorossi@gmail.com");

contatto1.setEmail(email);

String exEmail=contatto1.getEmail

Oracle : exEmail=mariorossi@gmail.com

UTC 1.8 Test Contatto-getNumTelefono

Test items: classe Contatto metodo testGetNumTelefono

Test :testGetNumTelefono

Case description: ottenuto il valore del campo numTelefono

Location: List<String> numeri=new ArrayList<>();
numeri.add("341204042");
contatto1.setNumTelefono(numeri);
String exNum=contatto1.getNumTelefono();

Oracle : exNum="341204042"

UTC 1.9 Test Contatto-getNumTelefono

Test items: classe Contatto metodo testSetNumTelefono

Test :testGetNumTelefono

Case description: stabilito un valore campo NumTelefono

Location: ContattoTest.testSetNumTelefono

Input List<String> numeri=new ArrayList<>();
numeri.add("010");
contatto1.setNumTelefono(numeri);
String exNum=contatto1.getNumTelefono(numeri);

Oracle : exNum="010"

UTC 1.10 Test Contatto-aggiungiNumTelefono

Test items: classe Contatto metodo testAggiungiNumTelefono

Test :aggiungiNumTelefono1

Case description:aggiunto un nuovo numTelefono

Location: ContattoTest.aggiungiNumTelefono;
contatto1.aggiungiNumeroTelefono("3276518923");
String exNum=contatto1.getNumTelefono.getIndex(0);

UTC 1.10 Test Contatto-aggiungiNumTelefono

Oracle : exNum= "3276518923";

Test :aggiungiNumTelefono2

Case description:aggiunto un nuovo numTelefono

Location: ContattoTest.testAggiungiNumTelefono;

Input : contatto1.aggiungiNumeroTelefono("333999666000");
String exNum=contatto1.getNumTelefono.getIndex(1);

Oracle : exNum= "333999666000"

Test :aggiungiNumTelefono3

Case description:aggiunto un nuovo numTelefono

Location: ContattoTest.testAggiungiNumTelefono;

Input: contatto1.aggiungiNumTelefono("1926");
String exMail=contatto1.getNumTelefono.getIndex(2);

Oracle : exMail= "1926";

UTC 1.11 Test Contatto-aggiungiEmail

Test items: classe Contatto metodo testAggiungiEmail

Test :aggiungiEmail1

Case description:aggiunto un nuovo numTelefono

Location: Contatto.Test.testaggiungiNumTelefono;

Input: contatto1.aggiungiEmail(example@email.com");
String exEmail=contatto1.getEmail.getIndex(0);

Oracle : exNum= example@email.com"

Test :aggiungiEmail2

Case description:aggiunto un nuovo numTelefono

Location: Contatto.Test.testAggiungiEmail

Input: contatto1.aggiungiEmail("example2@email.com");
String exMail=contatto1.getEmail.getIndex(1);

Oracle : exMail= "example2@email.com"

Test :aggiungiEmail3

Case description:aggiunto un nuovo numTelefono

Location: ContattoTest.testAggiungiNumTelefono;

Input: contatto1.aggiungiEmail("example3@gmai.com");
String exEmail=contatto1.getEmail.getIndex(2);

Oracle : exEmail= "example3@gmai.com"

UTC 1.12 Test Contatto-compareTo

Test items: classe Contatto metodo testCompareTo

UTC 1.12 Test Contatto-compareTo

Test :testCompareTo1

Case description: comparati due contatti

Location: ContattoTest.testCompareTo

Input : Contatto c1 = new Contatto("Mario", "");
Contatto c2 = new Contatto("Anna", "");

Oracle : (c1.compareTo(c2) > 0) = true

Test :testCompareTo2

Case description: comparati due contatti

Location: ContattoTest.testCompareTo

Input : Contatto c3 = new Contatto("Mario", "");
Contatto c4 = new Contatto("Anna", "Rossi");

Oracle : (c3.compareTo(c4) > 0) = true

Test :testCompareTo3

Case description: comparati due contatti

Location: ContattoTest.testCompareTo

Input : Contatto c5 = new Contatto("Anna", "Rossi");
Contatto c6 = new Contatto("Mario", "");

Oracle : (c5.compareTo(c6) < 0) = true

Test :testCompareTo4

Case description: comparati due contatti

Location: ContattoTest.testCompareTo

Input : Contatto c7 = new Contatto("Mario", "Rossi");
Contatto c8 = new Contatto("Anna", "Bianchi");

Oracle : (c7.compareTo(c8) > 0) = true

Test :testCompareTo5

Case description: comparati due contatti

Location: ContattoTest.testCompareTo

Input : Contatto c9 = new Contatto("Anna", "Rossi");
Contatto c10 = new Contatto("Mario", "Rossi");

Oracle : (c9.compareTo(c10) < 0) = true

Test :testCompareTo6

Case description: comparati due contatti

Location: ContattoTest.testCompareTo

Input : Contatto c11 = new Contatto("Anna", "Rossi");
Contatto c12 = new Contatto("Anna", "Rossi");

Oracle : c11.compareTo(c12) = 0

UTC 2.1 Test Rubrica-aggiungiContatto

Test items: classe Rubrica metodo aggiungiContatto

Test : testAggiungiContatto1

Case description: Aggiunta di un contatto valido.

Location: RubricaTest.testAggiungiContatto

Input : Rubrica r = new Rubrica();
Contatto c = new Contatto("Bruno", "Di Carluccio");
r.aggiungiContatto(c);

Oracle : rubrica.getTuttiContatti().get(0)=c;

Test : testAggiungiContatto2

Case description: Aggiunta di un contatto valido.

Location: RubricaTest.testAggiungiContatto

Input : Rubrica r = new Rubrica();
Contatto c = new Contatto("Bruno", "Di Carluccio");
r.aggiungiContatto(c);

Oracle : rubrica.getTuttiContatti().size() = 1

UTC 2.2 Test Rubrica-rimuoviContatto

Test items: classe Rubrica metodo rimuoviContatto

Test : testRimuoviContatto

Case description: Rimozione di un contatto presente nella rubrica.

Location: RubricaTest.testRimuoviContatto

Input : Rubrica r = new Rubrica();
Contatto c = new Contatto("Pasquale", "La Mura");
r.aggiungiContatto(c);
r.rimuoviContatto(c);

Oracle : rubrica.getTuttiContatti().size() = 0;
rubrica.getTuttiContatti().contains(c) = false;

UTC 2.3 Test Rubrica-cercaContatto

Test items: classe Rubrica metodo cercaContatto

Test : testCercaContatto

Case description: Ricerca contatto nella lista.

Location: RubricaTest.testCercaContatto

Input : Rubrica r = new Rubrica();

 Contatto c=new Contatto("Giovanni","Giachetta");
 rubrica.aggiungiContatto(c);

Oracle : rubrica.getTuttiContatti().size() = 1;
 rubrica.getTuttiContatti().get(0) = c;
 rubrica.getTuttiContatti().contains(c) = true;

UTC 2.4 Test Rubrica-ordinaRubrica

Test items: classe Rubrica metodo ordinaRubrica

Test : testOrdinaRubrica

Case description: Ordina la rubrica.

Location: RubricaTest.testOrdinaRubrica.

Input : Contatto c1 = new Contatto("Alessandro", "Di Carluccio");
 c1.aggiungiNumeroTelefono("3421113445");
 c1.aggiungiEmail("example1@gmail.com");
 Contatto c2 = new Contatto("Armando", "Di Cesare");
 rubrica.aggiungiContatto(c1);
 rubrica.aggiungiContatto(c2);
 rubrica.ordinaRubrica();

Oracle : rubrica.getTuttiContatti().get(0) = c1;
 rubrica.getTuttiContatti().get(1) = c2;

UTC 2.5 Test Rubrica-getTuttiContatti

Test items: classe Rubrica metodo getTuttiContatti

Test : testGetTuttiContatti

Case description: Ritorna la lista di tutti i contatti presenti in rubrica.

Location: RubricaTest.testGetTuttiContatti

Input : Contatto c1=new Contatto("Diego Armando","Maradona");
Contatto c2=new Contatto("Khvicha","Kvaratskhelia");
rubrica.aggiungiContatto(c1);
rubrica.aggiungiContatto(c2);
List<Contatto> list2=rubrica.getTuttiContatti();

Oracle : list2.size() = 2;
list2.get(0) = c1;
list2.get(1) = c2;

UTC 3.1 Test ValidatoreStandard - validaEmail

Test items: classe ValidatoreStandard metodo validaEmail

Test : testValidaEmail

Case description: Valida email.

Location: ValidatoreStandardTest.testValidaEmail

Input : validatore=new ValidatoreStandard();

Oracle : validatore.validaEmail("alessandro@gmail.com") = true;
validatore.validaEmail("bro.ciao@hotmail.com") = true;
validatore.validaEmail("") = false;

```
validatore.validaEmail("marioorris@gmailcom") = false;  
validatore.validaEmail("plainaddress") = false;  
validatore.validaEmail("ciao@.com") = false;
```

UTC 3.2 Test ValidatoreStandard - validaNumTelefono

Test items: classe ValidatoreStandard metodo validaNumTelefono

Test : testValidaNumTelefono

Case description: Valida numero di telefono.

Location: ValidatoreStandardTest.test.ValidaNumTelefono

Input : validatore=new ValidatoreStandard();

Oracle : validatore.validaNumTelefono("") = true;
validatore.validaNumTelefono("123456789") = true;
validatore.validaNumTelefono("12345c") = false;

UTC 3.3 Test ValidatoreStandard - validaNome

Test items: classe ValidatoreStandard metodo validaNome

Test : testValidaNome

Case description: Valida il nome.

Location: ValidatoreStandardTest.test.ValidaNome

Input : validatore=new ValidatoreStandard();

Oracle : validatore.validaNome(" alessandro") = true;
validatore.validaNome(null) = false;
validatore.validaNome(" ") = false;

UTC 3.4 Test ValidatoreStandard - validaCognome

Test items: classe ValidatoreStandard metodo validaCognome

Test : testValidaCognome

Case description: Valida il cognome.

Location: ValidatoreStandardTest.test.ValidaCognome

Input : validatore=new ValidatoreStandard();

Oracle : validatore.validaCognome(null) = false;
validatore.validaCognome(" ") = false;
validatore.validaCognome("di carluccio") = true;

Matrice di tracciabilità

Nome requisito	ID requisito	Design	Codice	Test	Requisiti correlati
Dati contatto	1	Modulo 6 Modulo 8	Contatto()	1.1, 1.2, 1.3, 1.4, 1.5, 1.6, 1.7, 1.8, 1.9, 1.10, 1.11, 3.1, 3.2, 3.3, 3.4	2,3,4,5,6,8
Creazione contatto	2	Modulo 3 Modulo 4 Modulo 5 Modulo 6 Modulo 7 Modulo 8 Modulo 9	ContattoView() ContattoViewController() Rubrica() Contatto() ValidatoreStandard() GestoreContatti	1.1, 3.3, 3.4	1,8
Modifica contatto	3	Modulo 3 Modulo 4 Modulo 5 Modulo 6 Modulo 7 Modulo 8 Modulo 9	ContattoView() ContattoViewController() Rubrica() Contatto() ValidatoreStandard() GestoreContatti	1.1, 1.3, 1.5, 1.7, 1.10, 1.11,2.1, 3.1, 3.2, 3.3, 3.4	1,8

Nome requisito	ID requisito	Design	Codice	Test	Requisiti correlati
Eliminazione contatto	4	Modulo 3 Modulo 4 Modulo 5 Modulo 6 Modulo 9	ContattoView() ContattoViewController() Rubrica() Contatto() GestoreContatti	2.2	8
Salvataggio rubrica su file	5	Modulo 1 Modulo 2 Modulo 5 Modulo 9	RubricaView() RubricaViewController() Rubrica() GestoreContatti()		1
Carica rubrica da file	6	Modulo 1 Modulo 2 Modulo 5 Modulo 9	RubricaView() RubricaViewController() Rubrica() GestoreContatti()		1
Cerca contatto	7	Modulo 1 Modulo 2 Modulo 5 Modulo 6	RubricaView() RubricaViewController() Rubrica() Contatto()	2.3, 1.2, 1.4, 1.12	8,9
Interfaccia utente	8	Modulo 1 Modulo 2 Modulo 3 Modulo 4	RubricaView() RubricaViewController() ContattoView() ContattoViewController()		1,2,3,4,5,6,7,9,10
Ordine rubrica	9	Modulo 1 Modulo 2 Modulo 5	RubricaView() RubricaViewController() Rubrica()	2.4, 1.12	7,8
Semplicità d'uso	10	Modulo 1 Modulo 3	RubricaView() ContattoView()		8