**Test della classe Metodo:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Precondizioni** | **Descrizione** | **Azioni** | **Risultato Atteso** |
|  |  | **Test Costruttore** | |
|  | Test del  costruttore | Creare una attività m1:  Metodo m1=new Metodo()  Verificare il valore degli attributi | \*Nome: null  \*Classe di appartenenza: null  \*Indirizzo di ritorno: null  \*numero di ritorni:0  \*valore di ritorno: null |
|  |  | **Test Getter e Setter** | |
|  | Getter | Creare un metodo con nome “metodo1”, classe di appartenenza “classe1”, indirizzo di ritorno “123”,numero di ritorni 0 e valore di ritorno “int”. Visualizzare questi dati con gli appositi getter. | \*Nome:”metodo1”  \*Classe di appartenenza:”classe1”  \*Indirizzo di ritorno:”123”  \*numero di ritorni:0  \*valore di ritorno:”int” |
|  | Setter: | Creare un metodo utilizzando i setter, impostando come nome “metodo1”, classe di appartenenza “classe1”, indirizzo di ritorno “123”, numero di ritorni 0 e valore di ritorno “int”. Visualizzare questi dati con gli appositi getter. | \*Nome: ”metodo1”  \*Classe di appartenenza: ”classe1”  \*Indirizzo di ritorno:”123”  \*numero di ritorni:0  \*valore di ritorno:”int” |
| Test altri metodi | | | |
| Voglio immettere come nome “metodo1”, classe di appartenenza “classe1”, indirizzo di ritorno “123”,numero di ritorni 0 e valore di ritorno “int” | Inserisci metodo test | Invoco il metodo immettendo come nome “metodo1”, classe di appartenenza “classe1”, indirizzo di ritorno “123”, numero di ritorni 0 e valore di ritorno “int”. Invoco metodo1.visualizzaMetodo(). | \*Nome:”metodo1”  \*Classe di appartenenza:”classe1”  \*Indirizzo di ritorno:”123”  \*numero di ritorni:0  \*valore di ritorno:”int” |
|  | Test CheckStringEsa1() | Invoco  CheckStringEsa (“1gda”) | Restituisce FALSE |
|  | Test CheckStringEsa2() | Invoco  CheckStringEsa (“AG2R”) | Restituisce FALSE |
|  | Test CheckStringEsa3() | Invoco  CheckStringEsa (“0001”). | Restituisce TRUE |
|  | Test CheckStringEsa4() | Invoco  CheckStringEsa (“”). | Restituisce FALSE |
|  | Test CheckStringEsa5() | Invoco  CheckStringEsa (“FDA1”). | Restituisce TRUE |
|  | Test CheckStringEsa6() | Invoco  CheckStringEsa (“A D”). | Restituisce FALSE |
|  | Test CheckStringEsa7() | Invoco  CheckStringEsa (“123”). | Restituisce TRUE |
|  | Test checkTypeReturn 1 | Invoco  checkTypeReturn (“intimo”) | Restituisce FALSE |
|  | Test checkTypeReturn 2 | Invoco  checkTypeReturn (123) | Restituisce FALSE |
|  | Test checkTypeReturn 3 | Invoco  checkTypeReturn (“string”) | Restituisce TRUE |
|  | Test checkTypeReturn 4 | Invoco  checkTypeReturn (“STRING”) | Restituisce TRUE |
|  | Test checkTypeReturn 5 | Invoco  checkTypeReturn (“”) | Restituisce FALSE |
|  | Test 1 Inserisci numero parametri: | Invoco inserisciMetodo() e inserisco  numeroParametri=1 2 | Si verifica l’eccezione NumberFormatException |
|  | Test 2 Inserisci numero parametri: | Invoco inserisciMetodo() e inserisco numeroParametri=”a” | Si verifica l’eccezione NumberFormatException |
|  | Test 3 Inserisci numero parametri: | Invoco inserisciMetodo() e inserisco numeroParametri=”” | Si verifica l’eccezione NumberFormatException |

|  |  |  |  |
| --- | --- | --- | --- |
|  | Test 1 checkString | Invoco checkString(””). | Restituisce FALSE |
|  | Test 2 checkString | Invoco checkString(”a d”). | Restituisce FALSE |
|  | Test 3 checkString | Invoco checkString (”ciao”). | Restituisce TRUE |
|  | Test CheckNumber1() | Passo come parametro “ciao” | Restituisce true |
|  | Test CheckNumber2() | Passo come parametro “22” | Restituisce false |
|  | Test CheckNumber3() | Passo come parametro “” | Restituisce true |
|  | Test CheckNumber4() | Passo come parametro “metodo1” | Restituisce false |
|  | Test CheckNumber5() | Passo come parametro “metodo 1” | Restituisce false |
|  | Test possibileErrore1() | Inserisco “sio” e poi “si” | prima richiede il dato e poi restituisce true |
|  | Test possibileErrore2() | Inserisco “123” e poi “si” | prima richiede il dato e poi restituisce true |
|  | Test possibileErrore3() | Inserisco “noo” e poi “no” | prima richiede il dato e poi restituisce false |

|  |  |  |  |
| --- | --- | --- | --- |
| **TEST CLASSE STACK** | | | |
| **TEST GETTER** | | | |
| Lista vuota | Test 1 getElementi() | Invoco getElementi() | Elementi=0 |
| **ALTRI METODI** | | | |
| Invoco creaNodo() 3 volte | Test 2 getElementi() | Invoco getElementi() | Elementi=3 |
| Creo due metodi con nome metodo1 e metodo2 | Test 1 push() | Visualizzo il risultato con toString() | Inseriti correttamente.  Mostra:  Head🡪metodo2;metodo1; |
| Creo cinque metodi con nome metodo1, metodo2, metodo3, metodo4 e metodo5 | Test 2 push() | Visualizzo il risultato con toString() | Inseriti correttamente.  Mostra:  Head🡪 metodo5;metodo4;metodo3;  metodo2;metodo1; |
| Creo 1 metodo con nome metodo2 e richiamo il metodo push() | Test toString() | Visualizzo il risultato con toString() | Mostra:  Head🡪metodo2; |
| Creo due metodi con nome metodo1 e metodo2 e invoco  Metodo1.push()  Metodo2.push() | Test 1 pop | Invoco pop() una volta e visualizzo il risultato con toString() | Mostra:  Head🡪metodo2; |
| Creo due metodi con nome metodo1 e metodo2 e invoco  Metodo1.push()  Metodo2.push() | Test 2 pop | Invoco pop() due volta e visualizzo il risultato con toString() | Mostra:  Head🡪 |
| Creo due metodi con nome metodo1 e metodo2 e invoco  Metodo1.push()  Metodo2.push() | Test 3 pop | Invoco pop() tre volte | Eccezione stackException |
| Creo due metodi con nome metodo1 e metodo2 e invoco  Metodo1.push()  Metodo2.push() | Test convertiStack() | Invoco convertiStack(array) | Array contiene info dei nodi dello stack |
| Creo un nuovo array di metodi di due elementi, metodo1 e metodo2 | Test convertiMetodo() | Invoco stack.convertiMetodo() | Nuovo nodo con 2 nodi contenenti metodi dell’array in info |
| Creo due metodi con nome “ca” e “ci” e invoco  Metodo1.push()  Metodo2.push(),  dopodichè lo converto in array con convertiStack() | Test 1 selectionSortCrescente() | Invoco selectionSortCrescente(array) | L’array ottenuto è ordinato a seconda del nome, quindi “ca;ci" |
| Creo due metodi con nome “dia” e “cano” e invoco  Metodo1.push()  Metodo2.push(),  dopodichè lo converto in array con convertiStack() | Test 2 selectionSortCrescente() | Invoco selectionSortCrescente(array) | L’array ottenuto è ordinato a seconda del nome, quindi “cano;dia" |
| Creo due metodi con nome “anno” e “cano” e invoco  Metodo1.push()  Metodo2.push(),  dopodichè lo converto in array con convertiStack() | Test 3 selectionSortCrescente() | Invoco selectionSortCrescente(array) | L’array ottenuto è ordinato a seconda del nome, quindi “anno;cano " |
| Creo due metodi con nome metodo1 e metodo2 e invoco  Metodo1.push()  Metodo2.push() | Test Serializzazione1() | Invoco  Stack.Serializzazione  ("c:\\percorsoNonEsistente") | Eccezione IOException |
|  | Test deserializzazione() | Provo a inserire un carattere nella scelta file(la path deve essere giusta) | Mostra errore inserimento |

|  |  |  |  |
| --- | --- | --- | --- |
| **TEST CLASSE NODO** | | | |
|  | Test getInfo() | Provo a creare un nodo con come paramero un nodo e a mostrarne le info | Il metodo e correttamente inserito nelle info del nodo e correttamente mostrato |
|  | Test setInfo() | Modifico le info di un nodo con setInfo() | Modifica avvenuta correttamente |
| Creo due nodi | Test getLink() | Mostro il link del head | Il link del head è uguale al secondo nodo |
| Creo un nodo | Test getLinkNull() | Mostro il link del nodo | Il link del head è null |

**TEST D’INTEGRAZIONE**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **N** | **PRECONDIZIONI** | **AZIONI** | **RISULTATO ATTESO** | **REQUISITI VERIFICATI** | **NON OK** |
| 1 |  | Avviare programma | Visualizzare menu |  |  |
| 2 | Azione 1 | Digitazione del valore 1(push) | Richiesto inserimento dati per effettuare il push | Requisito 1 |  |
| 3 | Azione 1 | Digitazione del valore 2(pop) | lista vuota | Requisito 2 |  |
| 4 | Azione 1 | Digitazione del valore 3(interrompi esecuzione) | Mostra “lista vuota” e interrompe l’esecuzione | Requisito 3, requisito 4 e requisito 6 |  |
| 5 | Azione 1 | Digitazione del valore 4(carica stack) con path giusta e con file presenti | Mostra le possibili scelte e richiede l’inserimento dello stack da caricare | Requisito 5 |  |
| 6 | Azione 1 | Digitazione del valore 5(visualizza stack) | Visualizza lo stack con il nome dei suoi metodi |  |  |
| 7 | Azione 1 | Digitazione del valore 6 | Ripropone il menu |  |  |
| 8 | Azione 1 | Digitazione di un carattere | Mostra errore e ripropone il menu |  |  |
| 9 | Azione 1 | Digitazione numero negativo | Ripropone il menu |  |  |
| 10 | Azione 2 | Inserire carattere numerico nel campo “nome”, ”classe di appartenenza” o ”nome parametro” | Richiedere se continuare nonostante inserimento numerico |  |  |
| 11 | Azione 2 | Inserire carattere e numero nel campo “nome”, ”classe di appartenenza” o ”nome parametro” | Richiedere se continuare nonostante inserimento numerico |  |  |
| 12 | Azione 2 | Inserire carattere nel campo tipo ritorno diverso dal nome dei dati nativi | Mostrare errore inserimento e richiedere reinserimento |  |  |
| 13 | Azione 2 | Inserire numero nel campo tipo ritorno | Mostrare errore inserimento e richiedere reinserimento |  |  |
| 14 | Azione 2 | Inserire carattere successivo ad ‘f’ nel campo “stringa esadecimale” | Mostrare errore inserimento e richiedere reinserimento |  |  |
| 15 | Azione 2 | Inserire un numero maggiore di 9 in “numero parametri” | Mostrare errore inserimento e richiedere reinserimento |  |  |
| 16 | Azione 2 | Inserire un carattere in “numero parametri” | Mostrare errore inserimento e richiedere reinserimento |  |  |
| 17 | Azione 2 | Inserire nulla in uno dei campi richiesti | Mostrare errore inserimento e richiedere reinserimento |  |  |
| 18 | Azione 2 | Inserire “ “ in uno dei campi richiesti | Mostrare errore inserimento e richiedere reinserimento |  |  |
| 19 | Azione 2 | Inserire tutti i dati correttamente | Mostrare dati metodo e il menu |  |  |
| 20 | Azione 19 | Digitare 2 | Mostrare simulazione esecuzione metodo e eliminarlo dallo stack |  |  |
| 21 | Azione 19 due volte | Digitare 2 2 volte e digitare 5 | Mostrare “head🡪” |  |  |
| 22 | Azione 19 con nome “metodo1” e digitare di nuovo 1 con metodo di nome “metodo2” | Digitare 2 1 volta e digitare 5 | Mostrare “head🡪metodo1” |  |  |
| 23 | Azione 19 tre volte | Digitare 2(pop) 3 volte | Mostrare errore alla terzo pop |  |  |
| 24 | Azione 19 | Digitare 3 | Salvare stack su file txt e bin e terminare progetto |  |  |
| 25 | Azione 1 | digitare 4(non sono presenti stack caricabili) | All’inserimento di un qualsiasi valore viene mostrato “errore scelta” |  |  |
| 26 | Azione 5(sono presenti 2 file) | Digitare 1 | Caricare primo file |  |  |
| 27 | Azione 5(sono presenti 2 file) | Digitare 2 | Caricare secondo file |  |  |
| 28 | Azione 5(sono presenti 2 file) | Digitare 3 | Mostrare errore scelta e menu |  |  |
| 29 | Azione 5(sono presenti 2 file) | Digitare “” | Mostrare errore scelta e menu |  |  |
| 30 | Azione 5(sono presenti 2 file) | Digitare “a” | Mostrare errore scelta e menu |  |  |
| 31 | Azione 19 con nome “metodo1” | Digitare 5 | Mostrare “head🡪metodo1;” |  |  |
| 32 | Azione 19 2 volte con nomi “metodo1” e “metodo2” | Digitare 5 | Mostrare “head🡪metodo2; metodo1;” |  |  |