**Ministero dell’Istruzione, dell’Università e della Ricerca**

Istituto d’Istruzione Superiore

“OLIVELLI PUTELLI”

ANNO SCOLASTICO 2017-18

**SOFTWARE PER LA GESTIONE**

**DEGLI ACCESSI DI UN LABORATORIO**

**Tabelle di Test**

Carizzoni Davide

**PIANO DI TEST DELLA CLASSE: Accesso**

|  |  |  |  |
| --- | --- | --- | --- |
| **PRECONDIZIONI** | **DESCRIZIONE** | **AZIONI** | **RISULTATO ATTESO** |
|  | Test Costruttore Accesso | Costruire un oggetto di classe Accesso con il costruttore principale. Invocare i metodi getMatricola, getIdAccesso,getDataOra | getMatrciola=1;  getIdAccesso=1;  getDataOra= gg/mm/aa o:m:s |
|  | Test Costruttore vuoto | Costruire un oggetto di classe Accesso utilizzando il costruttore di default.  Invocare i metodi getMatricola, getIdAccesso,getDataOra | getMatrciola=0;  getIdAccesso=0;  getDataOra= null |
|  | Test setIdAccesso() | Costruire un oggetto di classe Accesso utilizzando il costruttore di default. Invocare setIdAccesso(1) e dopo getIdAccesso () | getIdAccesso=1; |
|  | Test setMatricola() | Costruire un oggetto di classe Accesso utilizzando il costruttore di default. Invocare setMatricola (1) e dopo getMatricola() | getMatrciola=1; |
| Creare un’oggetto della classe LocalDataTime. Ad esempio dataOra gg/mm/aa o:m:s | Test setDataOra () | Costruire un oggetto di classe Accesso utilizzando il costruttore di default. Invocare setDataOra(dataOra) e dopo getDataOra() | getDataOra= gg/mm/aa o:m:s |

**PIANO DI TEST DELLA CLASSE: Nodo**

|  |  |  |  |
| --- | --- | --- | --- |
| **Precondizioni** | **Descrizione** | **Azioni** | **Risultato** **Atteso** |
|  | Test costruttore Nodo | Costruire un oggetto di classe Accesso, per passarlo come parametro al costruttore Nodo.  Invocare getInfo() e  getLink() | getInfo=a1  getLink=null |
|  | Test setInfo(a2) | Costruire 2 oggetti della classe Accesso.  Il primo(a) passarlo come parametro alla costruttore del nodo il secondo(a2) si utilizza come parametro del metodo setInfo(). | getInfo=a2  getLink=null; |
|  | Test setLink(nodo2) | Costruire 2 oggetti di classe Accesso e 2 oggetti di classe Nodo.  Settare alla componente link del primo nodo il reference del secondo nodo | getInfo=a1  getLink=a2 |

**PIANO DI TEST DELLA CLASSE: Laboratorio**

|  |  |  |  |
| --- | --- | --- | --- |
| **Precondizioni** | **Descrizione** | **Azioni** | **Risultato** **Atteso** |
|  | Test costruttore  Laboratorio() | Costruire un oggetto di classe Laboratorio.  Invocare i metodi getElementi() e getHead() | getHead=null  getElementi=0 |
| Dato che è un metodo privato lo si fa diventare pubblico per il test dopo di che verrà commentato e portato ancora a privato | Test creaNodo() | Costruire 1 oggetto della classe Laboratorio e un oggetto della classe Nodo. | getInfo=a1  getLink=null; |
|  | Test registraAccesso() | Costruire un oggetto della classe Laboratorio e uno della classe Accesso.  Registrare l’accesso al Laboratorio. | getElementi=1; |
|  | Test toString() | Costruire un oggetto della classe Laboratorio e uno della classe Accesso. Aggiungere l’accesso al Laboratorio, creare un string con il risultato atteso e invocare il metodo toString. | toString=Head->Matricola. 1 Data e ora: aa/mm/gg h:m:s idAccesso:1 **NON VA** |
| Laboratorio vuoto | Test toString() | Costruire un oggetto di tipo Laboratorio. | toString= “” |
|  | Test getAccesso() | Costruire un oggetto della classe Laboratorio e uno della classe Accesso. Aggiungere l’accesso al laboratorio e invocare il getAccesso in posizione 1, ossia dove è stato aggiunto l’accesso. | getAccesso=a1  **NON VA** |
| Laboratorio vuoto | Test getAccesso() | Costruire un oggetto della classe Laboratorio. Aggiungere l’accesso al laboratorio e invocare il getAccesso. | Eccezione LaboratorioException |
| Posizione non valida | Test getAccesso() | Costruire un oggetto della classe Laboratorio e uno della classe accesso. Aggiungere l’accesso al laboratorio e invocare il getAccesso in una posizione non valida. | Eccezione LaboratorioException |
|  | Test EliminaInTesta() | Costruire un oggetto della classe Laboratorio e uno della classe accesso. Aggiungere l’accesso al laboratorio e invocare il metodo EliminaInTesta | getHead=null; |
| Laboratorio vuoto | Test EliminaInTesta() | Costruire un oggetto della classe Laboratorio. Invocare il metodo EliminaInTesta | Eccezione LaboratorioException |
|  | Test EliminaInCoda() | Costruire un oggetto della classe Laboratorio e uno della classe accesso. Aggiungere l’accesso al laboratorio e invocare il metodo EliminaInCoda | getHead=null; |
| Laboratorio vuoto | Test EliminaInCoda() | Costruire un oggetto della classe Laboratorio. Invocare il metodo EliminaInCoda | Eccezione LaboratorioException |
|  | Test salvaLaboratorio e caricaLaboratorio | Costruire un Laboratorio ed eseguire la serializzazione. Invocare il metodo con la data con cui lo si vuole salvare. Sucessivamente creare un altro oggetto di tipo laboratorio dove caricare gli | l=lcopia |
| File non presente | Test caricaLaboratorio | Invocare il metodo avente come parametro la data di cui si desidera caricare gli accessi | Eccezione IOException  **NON VA** |
|  | Test esportaLaboratorioCSV | Costruire un Laboratorio. Invocare il metodo con la data con cui lo si vuole salvare | “Operazione completata con successo” |
|  | Test verificaPresenza | Costruire un oggetto della classe Laboratorio e uno della classe accesso(con matricola 1). Aggiungere l’accesso al laboratorio e invocare il metodo verificaPresenza passando come parametro 1(matricola cercata) | verificaPresenza=true  **NON VA** |
| Dipendente con matricola cercata non presente nel laboratorio | Test verificaPresenza | Costruire un oggetto della classe Laboratorio e uno della classe accesso(con matricola 1). Aggiungere l’accesso al laboratorio e invocare il metodo verificaPresenza passando come parametro 2(matricola cercata) | Eccezione AccessoMatricolaNotFoundException |

PIANO TEST DI INTEGRAZIONE

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| N | PRECONDIZIONI | AZIONI | RISULTATO ATTESO | REQUISITI VERIFICATI | ERRORI |
| 1 | Azione 0 | Avviare l’esecuzione | Visualizzazione del menù |  |  |
| 2 | Azione 1 | Digitare tasto 1 “Registra Accesso.. richiesta salvataggio su file” | Visualizzazione delle giornate se sono presenti file. Visualizzare “inserire la data in cui si vogliono registrare gli accessi” |  |  |
| 3 | Azione 2 | Inserire la data: 19/5/2018 | Visualizzare la data inserita “19/5/2018” |  |  |
| 4 | Azione 3 | Inserire la matricola del dipendente: 1 | Visualizzare ”Inserire la matricola del dipendente di cui si vogliono salvare gli accessi” |  |  |
| 5 | Azione 4 | Inserimento automatico dell’orario attuale in cui viene effettuata l’operazione | Visualizzazione orario attuale |  |  |
| 6 | Azione 5 | Visualizzazione accessi inseriti | Visualizzare: ->Matricola: 1. Data e ora: 2018-05-19T23:19:38.320 Id Accesso: 1  Visualizzare: ”registrare un altro accesso in data 19/5/2018?” |  |  |
| 7 | Azione 6 | Inserire la scelta: si | Ritorno ad azione 3. Visualizzare ”Inserire la matricola del dipendente di cui si vogliono salvare gli accessi” |  |  |
| 8 | Azione 7 | Inserire la scelta: no | Visualizzare: “scrittura su file avvenuta con successo”.  Visualizzazione Menù | Serializzazione in file binario con il nome del giorno stesso: “19\_5\_2018.bin” |  |
| 9 | Azione 8 |  |  |  |  |
| 10 | Azione 9 |  |  |  |  |
| 11 | Azione 10 |  |  |  |  |
| 12 | Azione 11 |  |  |  |  |
| 13 | Azione 12 |  |  |  |  |
| 14 | Azione 13 |  |  |  |  |
| 15 | Azione 14 |  |  |  |  |
| 16 | Azione 15 |  |  |  |  |
| 17 | Azione 16 |  |  |  |  |
| 18 | Azione 17 |  |  |  |  |
| 19 | Azione 18 |  |  |  |  |
| 20 | Azione 19 |  |  |  |  |