**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** |
| Creare un’oggetto della classe LocalDataTime. Ad esempio dataOra gg/mm/aa h:m:s | 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 h:m:s |
|  | Test Costruttore default | 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=accesso  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=a1  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=-->Matricola. 1 Data e ora: aa/mm/gg h:m:s Id Accesso:1 |
| 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 SO SE GIUSTO** |
| 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 | laboratorio=lcopia |
| File non presente | Test caricaLaboratorio | Invocare il metodo avente come parametro la data di cui si desidera caricare gli accessi | Eccezione IOException  **NON FAtTO** |
|  | Test esportaLaboratorioCSV | Costruire un Laboratorio. Invocare il metodo con la data con cui lo si vuole salvare. Creare una stringa dove inserire l’accesso informa CSV e confrontare con | “Operazione completata con successo”  **DA GUARDARE** |
|  | 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 |
| 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 |  | Digitare “a” | Visualizzare “formato dato inserito errato, reinserire” |  |  |
| 3 |  | Digitare 7 | Visualizzare “Opzione non disponibile, reinserire” |  |  |
| 4 |  | Digitare 0 “ESCI” | Visualizzare “SEI USCITO” |  |  |
| 5 | Azione 1 | Digitare 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” |  |  |
| 6 | Azione | Inserire la data: 19/20/2018 | Visualizzare “Errore nell’inserimento data, reinserirla” |  |  |
| 7 | Azione 2 | Inserire la data: 19/5/2018 | Visualizzare la data inserita “2018-05-19” |  |  |
| 8 |  | Inserire la matricola del dipendente: a | Visualizzare “Formato dato inserito errato, reinserire” |  |  |
| 9 | Azione 3 | Inserire la matricola del dipendente: 1 | Visualizzare ”Inserire la matricola del dipendente di cui si vogliono salvare gli accessi” |  |  |
| 10 | Azione 4 | Inserimento automatico dell’orario attuale in cui viene effettuata l’operazione | Visualizzazione orario attuale | Requisito 1(Registrare un accesso) |  |
| 11 | 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 2018-05-19?” |  |  |
| 12 | Azione 6 | Inserire la stringa: si | Ritorno ad azione 3. Visualizzare ”Inserire la matricola del dipendente di cui si vogliono salvare gli accessi” |  |  |
| 13 | Azione 7 | Inserire una stringa che sia diversa da si: no | Visualizzare: “scrittura su file avvenuta con successo”.  Visualizzazione Menù | Requisito 2(Serializzazione in file binario con il nome del giorno stesso: “19\_5\_2018.bin”) |  |
| 14 | Azione 8 | Digitare 2. “caricamento accessi per data” | Visualizzazione delle giornate se sono presenti file. Visualizzare “inserire la data di cui si vogliono caricare gli accessi” |  |  |
| 15 | Azione 9 | Inserire la data: 19/20/2018 | Visualizzare “Errore nell’inserimento data, reinserirla” |  |  |
| 16 |  | Inserire la data: 20/5/2018 | Visualizzare “Nessun accesso presente in data “2018-05-19.”  Visualizzazione Menu |  |  |
| 17 | Azione 10 | Inserire la data: 19/5/2018 | Visualizzare la data inserita “2018-05-19”.  Visualizzare gli accessi presenti in data “19/5/2018”. ->Matricola: 1. Data e ora: 2018-05-19T23:19:38.320 Id Accesso: 1  Visualizzazione menu. | Requisito 3(Deserializzazione) |  |
| 18 | Azione 11 | Digitare 3. “Verifica presenza dipendente in una determinata data” | Visualizzazione delle giornate se sono presenti file.  Visualizzare: “Inserire la data in cui si vuole verificare le presenza” |  |  |
| 19 | Azione 12 | Inserire la data: 19/20/2018 | Visualizzare “Errore nell’inserimento data, reinserirla” |  |  |
| 20 | Azione 13 | Inserire la data: 20/5/2018 | Visualizzare “Nessun accesso presente in data 2018-05-20” |  |  |
| 21 | Azione 14 | Inserire la data: 19/5/2018 | Visualizzare la data inserita “2018-05-19”.  Visualizzare gli accessi presenti in data 2018-05-19.  Visualizzare: “Inserire la matricola del dipendente di cui si vogliono verificare gli accessi” |  |  |
| 22 |  | Inserire la matricola “a” | Visualizzare “Formato dato inserito errato, reinserire” |  |  |
| 23 | Azione 15 | Inserire la matricola: 2 | Visualizzare “nessun accesso presente per il dipendente in data 2018-05-19”  Visualizzazione menu | Requisito 4(verificare se dipendente è presente o meno) |  |
| 24 | Azione 16 | Inserire la matricola 1 | Visualizzare “Sono presenti accessi del dipendente con matricola 1 in data 2018-05-19. PRESENZA VERIFICATA”  Visualizzazione menu | Requisito 4(verificare se dipendente è presente o meno) |  |
| 25 | Azione 17 | Digitare 4 “Salva accessi in un file di testo in ordine crescente di orario”. Quando non è presente alcuna data con accessi. | Visualizzazione “salva accessi in un file di testo in ordine crescente”  Visualizzare “nessuna giornata presente, effettuare prima l’operazione di caricamento sulla data desiderata” |  |  |
| 26 | Azione 18 | Digitare 4 “Salva accessi in un file di testo in ordine crescente di orario | Visualizzazione “salva accessi in un file di testo in ordine crescente”  Visualizzare gli accessi in ordine cronologico .  Visualizzare “Operazione completata con successo”  Visualizzazione menu. | Requisito 5(Salvataggio su file di testo con nome della data di cui si salvano gli accessi)  Requisito 6(Ordinamento degli accessi eseguito prima del salvataggio) |  |
| 27 | Azione 19 | Digitare 5 “Visualizza accessi di una data in ordine crescente di orario” | Visualizzazione delle giornate se sono presenti file. Visualizzare “inserire la data di cui si vogliono caricare gli accessi” |  |  |
| 28 |  | Inserire la data: 19/20/2018 | Visualizzare “Errore nell’inserimento data, reinserirla” |  |  |
| 29 |  | Inserire la data: 20/5/2018 | Visualizzare “Nessun accesso presente in data 2018-05-20” |  |  |
| 30 |  | Inserire la data: 19/5/2018 | Visualizzare la data inserita “2018-05-19”.  Visualizzare gli accessi presenti in data 2018-05-19 in ordine cronologico.  Visualizzazione menu | Requisito 3(deserializzazione)  Requisito 6(ordinare accessi cronologicamente) |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |