**IL TEST DEL SOFTWARE**

**Analisi**

Si vuole realizzare un’applicazione software denominata Porto Navale che si basa

sul funzionamento di un porto, concentrandosi quindi sulla gestione degli accessi e inoltre degli accessi previsti.

1. Aggiungere un nuovo arrivo previsto.

2- Simulare l’arrivo di una nave, gestendo quindi l’arrivo eliminando l’oggetto (Barca) dalla lista e salvandone i dati su file.

3-Modificare l’orario di arrivo previsto.

1. Visualizzare i dati di tutti gli arrivi previsti in ordine cronologico.
2. Simulare un’emergenza in modo da permettere alla nave in “difficoltà” di accedere con precedenza (quindi per prima).
3. Visualizza i dati di tutti gli arrivi provenienti dalla città che sarà inserita dall’utente.

Test della classe Barca: i setter vengono testati in contemporanea ai getter.

|  |  |  |
| --- | --- | --- |
| Descrizione | Azioni | Risultato |
| Test costruttori | | |
| Test del costruttore | Si crea un oggetto inserendo ovviamente dei valori arbitrari, per poi verificare coi getter che risultino corretti. | Ogni attributo dovrà essere stato settato correttamente. |
| Test del costruttore di copia | Si passa una Barca già costruita e si verifica la giusta copia di tutto. | Sempre coi getter verifico che sia tutto corretto. |
| Test del costruttore vuoto | Si crea un oggetto vuoto che avrà quindi tutti i parametri settati a 0 o null (valore “vuoto”) | Si verifica coi getter che sia davvero tutto vuoto. |
| Test getter | | |
| Test get codice | Sempre con un oggetto creato da noi verifichiamo che i dati inseriti siano veramente stati settati. Poi li risettiamo (usando così i setter) e andiamo a rifare il controllo. | Si verifica che tutti i cambiamenti vengano apportati e corrispondano a ciò che ci aspettiamo. |
| Test get porto provenienza | Sempre con un oggetto creato da noi verifichiamo che i dati inseriti siano veramente stati settati. Poi li risettiamo (usando così i setter) e andiamo a rifare il controllo. | Si verifica che tutti i cambiamenti vengano apportati e corrispondano a ciò che ci aspettiamo. |
| testGetOrarioArrivo | Sempre con un oggetto creato da noi verifichiamo che i dati inseriti siano veramente stati settati. Poi li risettiamo (usando così i setter) e andiamo a rifare il controllo. | Si verifica che tutti i cambiamenti vengano apportati e corrispondano a ciò che ci aspettiamo. |
| testGetSos | Sempre con un oggetto creato da noi verifichiamo che i dati inseriti siano veramente stati settati. Poi li risettiamo (usando così i setter) e andiamo a rifare il controllo. | Si verifica che tutti i cambiamenti vengano apportati e corrispondano a ciò che ci aspettiamo. |

Test della classe nodo: i setter vengono testati in contemporanea ai getter.

|  |  |  |
| --- | --- | --- |
| Descrizione | Azioni | Risultato |
| Test costruttori | | |
| Test del costruttore | Si crea un Nodo avendo come unico parametro necessario un oggetto (Barca) per poi verificare coi getter che risultino corretti. | Ogni attributo (sono 2: Link e Info) dovrà corrispondere a ciò che abbiamo inserito. (Es: Info dovrà essere uguale al puntatore della Barca in quanto punta proprio all’oggetto che gli viene passato). |
| Test getter | | |
| Test get Link | Servendosi di un Nodo già creato da noi verifichiamo che i dati inseriti siano veramente stati settati. Poi li risettiamo (usando così i setter) e andiamo a rifare il controllo. | Si verifica che tutti i cambiamenti vengano apportati e che i dati di ritorno siano consistenti e quindi reali. |
| Test get Info | Servendosi di un Nodo già creato da noi verifichiamo che i dati inseriti siano veramente stati settati. Poi li risettiamo (usando così i setter) e andiamo a rifare il controllo. | Si verifica che tutti i cambiamenti vengano apportati e che i dati di ritorno siano consistenti e quindi reali. |

Test della classe Porto: i setter vengono testati in contemporanea ai getter.

|  |  |  |
| --- | --- | --- |
| Descrizione | Azioni | Risultato |
| Test costruttori | | |
| Test del costruttore | Si crea un Porto sapendo che verrà sempre creato vuoto (quindi con zero elementi). | Usiamo il metodo con cui otteniamo il numero degli elementi in lista e verifichiamo che siano 0. |
| Test getter | | |
| Test get elementi | Ogni Porto ha un attributo che si auto incrementa ed indica il numero di oggetti presenti nella lista. | Si verifica che aggiunto un elemento o tolto un elemento questo contatore si aggiorni dandoci quindi il risultato che ci aspettiamo. |
| Test Inserisci In Testa | Servendosi di un Porto già creato permette di inserire una Barca (oggetto) come primo oggetto (quindi in cima alla lista) | Si verifica l’oggetto sia veramente in prima posizione estraendolo con l’apposito metodo. |
| Test Inserisci In Coda | Servendosi di un Porto già creato permette di inserire una Barca (oggetto) come ultimo oggetto (quindi in fondo alla lista) | Si verifica l’oggetto sia veramente in ultima posizione estraendolo con l’apposito metodo. |
| Test Inserisci In Posizione | Servendosi di un Porto già creato permette di aggiungere in qualsivoglia posizione una Barca; sfrutta anche i 2 metodi precedenti in caso la posizione sia 1 o sia maggiore degli elementi. | Si verifica l’oggetto sia veramente nella posizione desiderata estraendolo con l’apposito metodo. |
| Test Elimina In Testa | Servendosi di un Porto già creato permette di eliminare una Barca (oggetto) in prima posizione (quindi in cima alla lista) | Si verifica l’oggetto sia veramente stato eliminato e che ora in prima posizione sia presente l’oggetto che prima era in posizione 2 |
| Test Elimina In Coda | Servendosi di un Porto già creato permette di eliminare una Barca (oggetto) in ultima posizione (quindi in fondo alla lista) | Si verifica l’oggetto sia veramente stato eliminato e che ora in quella posizione non ci sia niente (che non sia accessibile) |
| Test Elimina In Posizione | Servendosi di un Porto già creato permette di aggiungere in qualsivoglia posizione una Barca; sfrutta anche i 2 metodi precedenti in caso la posizione sia 1 o sia maggiore degli elementi. | Si verifica l’oggetto sia stato veramente eliminato e che nella posizione desiderata ora ci sia l’elemento che prima era in quella posizione + 1; |
| Test Get Barca | Permette di ottenere una singola Barca (Oggetto) indicando la posizione in cui si trova. | Ottenuta la Barca si verifica che si possono utilizzare i metodi della classe Barca e che tutto funzioni correttamente. |

**TEST DI INTEGRAZIONE**

Il test di integrazione è detto anche “test di accettazione” in quanto la sua esecuzione con esito positivo deve stabilire la soddisfazione con esito positivo dei requisiti e quindi consente la distribuzione del software ed il suo e ed il conseguente passaggio alla fase di manutenzione. Il test è costituito da una serie di azioni mirate a verificare il soddisfacimento dei requisiti.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| N | Precondizioni | Azioni | Risultato atteso | Soddisfatto | Non soddisfatto |
| 1 | Avere un porto già creato. | Aggiungere un nuovo arrivo previsto. | Avere una nuova barca nella lista. | FINITO |  |
| 2 | Avere nell’archivio una nave che deve attraccare | Simulare l’arrivo di una nave | Stampa su file, eliminazione dalla lista | Riconoscimento della nave nel porto, esportazione su file | Eliminazione dalla lista (in lavorazione) |
| 3 | Avere già una Barca che quindi ha un orario d’arrivo | Modificare l’orario di arrivo previsto | Modifica l’orario della Barca. | FINITO |  |
| 4 | Avere una lista; per verificare se il metodo funziona ovviamente che non sia vuota | Visualizzare in ordine cronologico. | Stampa in ordine cronologico le Barche presenti in lista non agendo direttamente sulla Lista. | FINITO |  |
| 5 | Avere una lista popolata da Barche. | Visualizza i dati degli arrivi di una città | Scorrere la lista e stampare a schermo le Barche che rispondono ai requisiti. | Parzialmente | Problemi nel visualizzare le barche; ora e possibile vedere solamente la prima (in lavorazione) |

**DIAGRAMMA DELLE CLASSI**

