

## Fondamenti di Informatica 2 – Prova Scritta

Riportare su Eventuale Foglio : Nome, Cognome, Matricola e Numero di Postazione

Una azienda che si occupa del commercio di scarpe, con 50 negozi sparsi per il Nord Italia, vuole automatizzare parte della gestione dei magazzino. Ogni scarpa ha un modello ed una taglia. L'azienda dispone di un magazzino centrale, a partire dal quale ogni giorno partono i furgoni addetti al trasporto, che portano stock di scarpe (più paia di scarpe dello stesso modello e taglia) ai negozi. Nel magazzino centrale e nei magazzini di tutti i negozi si tiene un registro di carico/scarico dove vengono segnati gli stock di scarpe in uscita (magazzino centrale) e in entrata (magazzini dei negozi), accompagnato dal codice dell'operatore (vettura) addetto al trasporto. Sappiamo che ogni magazzino verrà dotato di una moderna base di dati, con accesso via web.

### TEMA A-----

Si realizzi un componente software che consenta di estrarre quotidianamente (a fine giornata) un report che descriva per ogni vettura quante scarpe non siano ancora state consegnate di quelle caricate al mattino. Il componente deve produrre inoltre un secondo report che descriva gli stock di scarpe fermi in vettura da più di 3 giorni consecutivi.

ESEMPIO: la vettura "ANGELO-ROSSI-4723" la mattina carica 50 "Spike-Air(43)" e 30 "Sneaker-Plus(42)" dal magazzino centrale. In giornata, il magazzino 'Brescia01' registra uno scarico di 20 "Spike-Air(43)" e 5 "Sneaker-Plus(42)", e il magazzino 'Cremona02' registra uno scarico di 10 "Spike-Air(43)", 6 "Sneaker-Plus(42)" e 11 "Conserve-Classic-Yellow(40)" (caricate in un giorno precedente...). A fine giornata il sistema deve capire dai registri dei magazzini che nel furgone ci sono sicuramente ancora 20 "Spike-Air(43)" e 19 "Sneaker-Plus(42)". Il Sistema ripete la stessa verifica per i 2 giorni precedenti, e conclude ad esempio che ci sono 15 "Conserve-Classic-Yellow(40)" caricate 3 giorni prima e mai consegnate.

NOTA: per non rendere le cose troppo complesse si escluda il caso in cui la stessa vettura carichi stock di scarpe dello stesso Modello-Taglia in giorni differenti.

### TEMA B-----

Si realizzi una interfaccia grafica da utilizzare presso il magazzino Centrale. L'interfaccia deve contenere un Menu dove l'utente seleziona il tipo di informazioni da visualizzare e un pannello dove verranno presentate tali informazioni. L'interfaccia mostra:

- (1) DETTAGLIO : l'elenco dettagliato di tutti gli stock di scarpe in uscita dal magazzino degli ultimi 3 giorni. (Solo il dettaglio degli stock, non vengono riportate le vetture)
- (2) SINTESI : l'elenco sintetico per Modello di Scarpa e Taglia, che mostra cioè la somma di tutti gli stock appartenenti allo stesso modello e scarpa negli ultimi 3 giorni.
- (3) VETTURA #NOME\_VETTURA# : L'elenco giornaliero di tutte le operazioni di carico e scarico per ciascuna vettura (nel Menu compare una voce aggiuntiva per ogni vettura...).

ESEMPIO punto (2): la vettura "ANGELO-ROSSI-4723" carica 50 "Spike-Air(43)" e 30 "Sneaker-Plus(42)", mentre la vettura "MARIO-BIANCHI-5321" carica 70 "Spike-Air(42)" e 18 "Sneaker-Plus(42)". Il rapporto sintetico riporta: 70 "Spike-Air(42)", 50 "Spike-Air(43)" e 48 "Sneaker-Plus(42)".

A TITOLO DI TEST. Si riduca il numero dei negozi a 3 (invece che 50), si ipotizzino solo 2 vetture, e si utilizzino dei File al posto delle (ancora inesistenti) Basi di Dati e WEB.[Tema A: gli stessi report, per ogni vettura, devono essere scritti all'interno di opportuni file].Sia chiaro che il formato dei File e dei dati rappresentati è al contempo irrilevante e temporaneo.

1. Creare un Workspace **Eclipse**. Creare un Progetto **esame**. Dopo aver studiato il problema, implementare in **Java** una possibile soluzione modulare e ad oggetti alle richieste del Tema, ed effettuare sui componenti realizzati opportuni Tests.
2. Su foglio protocollo, **a titolo di documentazione e ai fini della valutazione**, si realizzi uno schema UML **sintetico** che metta in luce le relazioni che intercorrono tra i moduli implementati. E' possibile utilizzare ObjectAID UML, ma in quel caso è obbligatorio esportare gli schemi UML in formato immagine png. E' inoltre necessario aver chiaro che non basta buttare qualsiasi cosa dentro al diagramma solo perché è comodo e semplice fare così . E' necessario inoltre utilizzare la documentazione Javadoc nel codice dove lo si ritenga opportuno.
3. Lo studente può accedere al percorso **/home/etc/FD12** per recuperare la documentazione **Javadoc**, i cosiddetti **esempi forniti** e altro materiale utile. E' inoltre possibile consultare qualsiasi testo scritto.
4. Alla fine dell'esame, esportare un file zip attraverso la funzionalità **Export...** di eclipse (vedi le **istruzioni di salvataggio dati**) e salvarlo come **/home/esm/esame\_N/esame\_N.zip** (ad esempio **/home/esm/esame\_20/esame\_20.zip**)

#### **Punteggio (Totale 15+ punti)**

- **6+ punti** per l'**architettura** del progetto, con particolare enfasi alla struttura del Modello.
- **3 punti** per la corretta **implementazione** in **Java** delle funzionalità del programma.
- **3 punti** in merito alla **validità** di implementazione interna ad **ogni singola classe**.
- **3 punti** sono assegnati sulla base dello sviluppo dei Tests.