

## Quarto Appello – Prova di Livello Base-Intermedio

Cognome e Nome: \_\_\_\_\_ Matricola: \_\_\_\_\_

### Tempo a disposizione: 3 ore

È necessario scrivere un'applicazione che simula il frammento di un sistema informativo per la gestione dei cesti natalizi. Ogni cesto ha un codice, un nome, un prezzo e una tipologia (che può essere piccolo, medio e grande). Ad esempio un possibile cesto è {"NAT01", "Natale con i fiocchi", 23€, "piccolo"}.

Ad ogni cesto natalizio sono associati dei prodotti. Un cesto piccolo può contenere massimo 3 prodotti, uno medio 5 e uno grande 7.

Ogni prodotto ha un nome, una tipologia (che può essere "dolci", "salumi", "pasta" oppure "bevanda") ed un peso in grammi e la data della scadenza. Ad esempio alcuni possibili prodotti sono:

- Panettone Classico, dolce, 150 gr, scadenza 30/02/2022
- Prosecco Valdobbiadene, bevanda, 583 gr, scadenza 01/01/2030

L'applicazione deve consentire di svolgere i seguenti casi d'uso:

#### "Utente carica un archivio"

- l'utente carica l'archivio *NOTA: per velocizzare le operazioni, si suggerisce di utilizzare un "mock object" per simulare il funzionamento del DAO che carica i dati dal disco*

#### "Utente cerca cesti natalizi"

- l'utente sceglie una tipologia (*tramite JComboBox*) e un criterio di ordinamento (*tramite JRadioButton*), che può essere "prezzo crescente", "prezzo decrescente" o "nome crescente".
- il sistema mostra in una tabella la lista di cesti della tipologia scelta, ordinati secondo il criterio scelto dall'utente. Nella tabella bisogna mostrare codice, nome, tipologia e prezzo

#### "Utente inserisce nuovo prodotto"

- l'utente seleziona un cesto natalizio dai risultati
- il sistema mostra, in un nuovo pannello, i dati del cesto e i prodotti attualmente registrati. Per ogni prodotto bisogna mostrare nome, tipologia, peso e data scadenza
- l'utente può decidere di inserire i dati di un nuovo prodotto, specificando tutti i dati.

#### Scenario alternativo

- dati scorretti: il sistema mostra un messaggio di errore all'utente
- prodotto scaduto: non è possibile inserire un prodotto con una data di scadenza nel passato
- scadenza non valida: un prodotto di tipo "dolce" non può avere una scadenza più lontana di 6 mesi dalla data odierna.

#### "Utente verifica archivio"

- selezionando una voce di menu, il sistema conta i cesti che hanno prodotti duplicati, ovvero prodotti con lo stesso nome e stessa tipologia
- il sistema mostra un messaggio con il risultato calcolato al passo precedente

#### "Utente ricerca mesi preferiti" – Per la prova di livello intermedio

- il sistema calcola, per ogni tipologia di spesa, qual è la scadenza media in giorni
- i dati saranno mostrati in una tabella ordinata per tipologia crescente. Un esempio di tabella è la seguente

*Nota: è necessario sviluppare i relativi test di regressione*

Tipologia	Mese
dolci	35 giorni
salumi	120 giorni

Sviluppare l'applicazione che implementa i casi d'uso elencati, seguendo il processo di sviluppo descritto a lezione, e in particolare le seguenti operazioni:

- costruire e documentare il modello concettuale dell'applicazione
- sviluppare il frammento di applicazione che implementa i casi d'uso descritti
- sviluppare l'interfaccia grafica in **Java Swing** utilizzando l'**architettura MVC** presentata a lezione, rispettando i requisiti per l'accessibilità e l'usabilità.
- sviluppare i test di regressione utilizzando la libreria JUnit
- per effettuare operazioni di logging utilizzare le librerie SLF4J e LogBack
- come sistema di costruzione del codice usare **NetBeans standard**

E' possibile trovare il materiale necessario nelle seguenti cartelle

- librerie jar: **c:\lib**
- materiale didattico: **c:\documentazione\POO1 e POO2**
- file di esempio di logback.xml: **c:\documentazione\POO1 - materiale software aggiuntivo\java**