

Progetto per l'esame del corso di Paradigmi (programmazione ad oggetti)

Il progetto va svolto o singolarmente o in gruppo di al più 2 persone. Per avere un gruppo da 3 si deve avere una buona motivazione e la mia autorizzazione.

Il progetto deve essere consegnato una settimana prima dell'esame scritto. Metterò un contenitore sul sito per la consegna.

Requisiti comuni

Lo svolgimento del progetto deve tenere in considerazione i seguenti requisiti:

- le classi del dominio del sistema devono essere corredate da classi di test;
- la documentazione deve essere prodotta con **javadoc** (trovate una sintetica descrizione di come annotare il codice sul sito del corso);
- ogni progetto deve essere accompagnato da una relazione in cui vengono illustrate le scelte adottate: le classi e le loro relazioni e metodi pubblici, a cosa servono, le eccezioni che sollevano e perché le sollevano;
- inoltre, per quanto riguarda l'interfaccia utente (che deve essere separata dalla classi che realizzano la logica del sistema), deve essere possibile eseguire tutte le operazioni catturando le eventuali eccezioni sollevate che vengono gestite facendo ripetere di nuovo l'operazione. Dovete avere sia:
 - una semplice interfaccia grafica che permetta di svolgere le principali operazioni;
 - una interfaccia da riga di comando.

Nello svolgere il progetto usare i principi visti nel corso. In particolare, buona strutturazione delle classi, uso di classi astratte e interfacce se necessario. Ogni classe ha un compito preciso (e fa solo quello), ha campi privati, uso di eccezioni e non "stampe" per segnalazioni di malfunzionamenti, metodi di dimensione limitata, non ripetizioni di codice, ecc... Mantenete separate le classi del dominio dall'interfaccia grafica/testuale

Valutazione

La discussione del progetto con i singoli studenti verterà sulle modalità di implementazione adottate e sulla padronanza di alcuni dei concetti necessari per realizzare il progetto e/o spiegati a lezione. La valutazione del progetto sarà fatta in base alla: conformità dell'implementazione scelta per risolvere il problema con il paradigma di programmazione a oggetti e alla conformità del codice presentato alle regole di buona programmazione.

Realizzare un'applicazione per gestire delle Liste della Spesa (o liste simili)

Realizzare un programma per mantenere varie **liste della spesa**, che hanno un nome e un contenitore con degli **articoli**. Gli articoli sono classificati secondo **categorie** definite dall'utente (rappresentate da una stringa).

Gli articoli hanno associato un costo e una **quantità**. L'associazione fra un articolo e una categoria come la sua **quantità** e il suo costo possono essere modificati. Quando inserite un articolo potete non specificare la categoria e la **quantità** (**ma** dovete sempre specificare nome e costo). Se non specificato all'articolo viene assegnata una categoria predefinita "Non Categorizzati" e **quantità 1**. Gli articoli possono essere rimossi da una lista. Deve essere possibile data una stringa e una lista della spesa trovare la lista degli articoli che hanno la stringa come prefisso. Quando una categoria viene cancellata dovete assegnare a tutti gli articoli in tutte le liste la categoria "Non Categorizzati".

Per una lista deve essere possibile calcolare il costo totale degli articoli (considerando costo e **quantità** degli articoli contenuti) e ritornare data una categoria gli articoli presenti nella lista che hanno associata quella categoria.

Le **liste** devono essere **iterabili** per permettere ulteriori elaborazioni da parte del codice cliente.

Infine, dovete poter **scrivere** una lista su un **file** e creare una lista **leggendo** le informazioni da file.

Quindi la struttura è molto simile a quella della Rubrica. La classe **ListaSpesa** è l'analogo della classe Rubrica. La classe **Articolo** è una classe interna che contiene: **nome**, **quantità** e categoria di un articolo. La categoria è una stringa. In più nella classe **GestioneListe** che contiene le liste dovete avere anche tutte le categorie perché queste sono comuni a tutte le liste.

Dovete realizzare tutti i punti descritti, ma siete liberi di strutturare le classi in modo diverso e/o realizzare funzionalità aggiuntive. In ogni modo dovete documentare le vostre scelte.