

POO – Lezione 1

Esercizio 1:

Si realizzi un software Java per la gestione delle informazioni degli Studenti universitari. In particolare, tramite la decomposizione in classi, si garantiscano i seguenti requisiti:

- a) Caratterizzazione degli studenti tramite informazioni anagrafiche di base ed elenco degli esami superati.
- b) Discretizzazione con studenti DSA, tipicamente tramite una descrizione della patologia principale.
- c) Fornire il metodo **public List<Studente> getStudentiByEsame(Esame e)** che restituisce una collection di Studenti che hanno superato l'esame e.
- d) Fornire il metodo **public void printInfoDSA()** che stampa a video le informazioni inerenti ai soli studenti DSA.
- e) Fornire un metodo che permetta di ottenere un elenco di studenti DSA caratterizzati da una caratteristica patologica comune.
- f) Il gestore **StudentiManager** dovrebbe implementare una funzione di interfacciamento per l'inserimento di un esame superato.
- g) Il gestore, inoltre, dovrebbe garantire un'altra funzione di interfacciamento per la memorizzazione su disco di un elenco di studenti DSA e non DSA, rispettivamente.

Esercizio 2:

Si realizzi un software Java per la gestione del personale di un'azienda. In particolare, tramite la decomposizione in classi, si garantiscano i seguenti requisiti:

- a) Informazioni di base di un dipendente (Nome,Cognome,ID,salario)
- b) Discretizzazione tra: Determinati (caratterizzati da una data di fine contratto), Indeterminati (caratterizzati da una costante bonus del 30% sul proprio salario e da una qualifica settoriale) e Stagisti (che sono caratterizzati da un supervisore)
- c) Fornire il metodo **public List<Dipendente> getStagisti()** che restituisca un elenco di tutti i dipendenti esclusivamente stagisti (Suggerimento: si veda l'utilizzo di **instanceof**).
- d) Fornire un metodo che calcoli lo stipendio mensile. Per i dipendenti a tempo indeterminato esso è calcolato come il salario base incrementato del bonus. Per gli stagisti è calcolato come il salario base meno una ritenuta di 300 euro. Invariato per i dipendenti a tempo determinato.
- e) Implementare l'interfaccia **FileManager** che fornisca i metodi **void readLavoratori(String filename)** e **void saveLavoratori(String filename)** che garantisca la lettura e la memorizzazione di un elenco di lavoratori da file, rispettivamente.
- f) Stampa di tutti i lavoratori con un salario mensile maggiore di una certa soglia (ad esempio maggiore strettamente di 1000 euro).
- g) Fornisca una GUI elementare con dei pulsanti dedicati per: stampare su console l'elenco dei lavoratori con un certo salario e gestire da file l'elenco dei lavoratori.

Esercizio 3:

Si realizzi un software Java per la gestione di locali. In particolare, tramite la decomposizione in classi, si garantiscano i seguenti requisiti:

- a) Informazioni di base di un locale (Nome,Indirizzo,Posti disponibili)
- b) Discretizzazioni tra Pizzerie (caratterizzate da un menù di pizze preparate con il relativo prezzo) e Paninoteche (caratterizzate dal numero di panini e birre disponibili).
- c) Implementare l'interfaccia **LocaleOptions** che fornisca i metodi per: prenotare per un certo numero di persone (Controllare per le pizzerie il numero di posti disponibili e per le paninoteche il numero di posti e panini disponibili); stampare tutte le informazioni dei locali, e registrare un nuovo locale.
- d) Gestire il menù delle pizzerie e l'elenco dei locali memorizzati a sistema tramite un'HashMap.
- e) Organizzare propriamente le classi nei package core, gui, interfacce, e tester.
- f) Implementare il metodo **public static LocaliManager createManager()** che popola e restituisca un'istanza di LocaliManager.
- g) Implementare il metodo **public void printPizzerieByPizza(String Pizza)** che stampi tutte le pizzerie che preparano una certa Pizza (con il relativo prezzo).
- h) Implementare una GUI modularizzata per la la funzione printPizzerieByPizza precedentemente implementata.