

**Non è consentito usare libri o appunti.**

Implementare un sistema per gestire le registrazioni ad una conferenza.

1. **[10 punti]** Definire la classe `Registrazione` che le registrazioni effettuate per una conferenza. Ogni registrazione è caratterizzata da nome, cognome, data di registrazione. Nel caso in cui la data di registrazione è susseguente al 30 aprile 2020 lanciare l'eccezione controllata `BadDateException`. Corredare la classe con opportuni metodi.

Inoltre le registrazioni possono essere di 2 tipi a seconda se sono effettuate da studenti o professionisti.

- 1) `Studente` caratterizzato dalla variabile `nomeUniversità`, `Tipologia` (`Triennale`, `Magistrale`, `Dottorato`), e che fornisce i seguenti metodi:
  - `isInRitardo()` che restituisce `true` se la registrazione è stata effettuata dopo il 31 Gennaio 2020.
  - `calcolaCosto()` che restituisce il costo dell'iscrizione alla conferenza secondo il seguente criterio: nel caso in cui la registrazione è stata effettuata prima del 31 Gennaio 2020 allora il costo è di 250€, altrimenti il costo è di 300€.
- 2) `Professionista` caratterizzato da `nomeAzienda` e `partitaIVA`, e che fornisce i seguenti metodi:
  - `isInRitardo()` che restituisce `true` se la registrazione è stata effettuata dopo il 3 Febbraio 2020.
  - `calcolaCosto()` che restituisce il costo dell'iscrizione alla conferenza secondo il seguente criterio: nel caso in cui la registrazione è stata effettuata prima del 3 Febbraio 2020 allora il costo è di 500€, altrimenti il costo è di 550€.

2. **[8 punti]** Scrivere la classe `Registrazioni` che modella un insieme di registrazioni ricevute da una conferenza. La classe è corredata dai seguenti metodi:

- `void aggiungiRegistrazione(Registrazione r)` che inserisce la registrazione `r` nella lista in modo ordinato rispetto al cognome. Nel caso in cui nella lista è già presente una registrazione con lo stesso cognome e nome di `r` viene lanciata l'eccezione non controllata `RegistrationException`.
- `double dammiTotale()` che restituisce l'incasso totale ottenuto da tutte le registrazioni.
- `ArrayList<Registrazione> dammiRegistrazionePerTipoPartecipante(int x)`, che restituisce l'elenco delle registrazioni di una certa tipologia. In particolare, `x=0` indica tipologia studente, `x=1` professionista. Per valori di `x` diversi da 0, 1, viene la lanciata l'eccezione controllata `InvalidParameterException`.
- `double dammiTotalePerTipoRegistrazione(int x)`, che restituisce l'incasso ottenuto dalle registrazioni di una certa tipologia. In particolare, `x=0` indica tipologia studente, `x=1` professionista. Per valori di `x` diversi da 0, 1, viene la lanciata l'eccezione controllata `InvalidParameterException`.
- `ArrayList<Registrazione> cerca(GregorianCalendar data)`, che cerca e restituisce la lista di registrazioni con data di registrazione uguale all'argomento ricevuto.

3. **[12 punti]** Considerando le classi ai punti precedenti, scrivere un programma Java che realizzi un'interfaccia grafica per

- a) caricare da un file una lista di registrazioni (il file deve essere stato istanziato in precedenza con 20 registrazioni in modo random tra studenti e professionisti),

Il diagramma mostra un'interfaccia grafica con un rettangolo principale. All'interno, in alto a sinistra, c'è un titolo "Carica". Sotto il titolo, a sinistra, c'è l'etichetta "Nome file:" seguita da un rettangolo vuoto per l'input del nome del file. In basso a destra, c'è un pulsante con la scritta "Carica".

- b) visualizzare gli studenti o i professionisti registrati alla conferenza (insieme all'incasso totale e l'incasso per tipologia di registrazione selezionata)

Menù principale

Visualizza studenti      Visualizza professionisti

Paolo Rossi, Università di Salerno, Magistrale, 20/01/2020  
Mario Giordano, Università di Trento, Triennale, 25/01/2020

Totale incasso: 1500€      Totale incasso studenti: 500€

Ricerca

- c) ricercare le registrazioni in base alla data di registrazione

Ricerca registrazione

Data registrazione: 02/02/2020

19/01/2020  
20/01/2020  
22/01/2020  
27/01/2020  
02/02/2020

Paolo Rossi, Studente, 300€  
Marco Bianchi, Professionista, 500€

**Ogni violazione delle regole enunciate ai punti sotto elencati comporta l'annullamento della prova (l'elaborato viene valutato 0).**

1. Prima di eseguire eclipse assicurarsi che non ci siano file Java (sorgenti, bytecode, workspace, progetti, pacchetti) sul desktop.
2. Eseguire eclipse specificando un workspace sul desktop.
3. Durante la prova d'esame è vietato usare:
  - a. libri e appunti sia in forma cartacea che in forma digitale
  - b. supporti di memoria esterni
  - c. un font di dimensione maggiore di 10 punti.

4. Non è consentito modificare i file allegati alla traccia.
5. Il nome del progetto consegnato deve cominciare con COGNOME seguito dal carattere underscore e quindi dal NOME (tutto in maiuscole). Ad esempio, il nome del progetto di Marco Rossi può essere ROSSI\_MARCO, ROSSI\_MARCO\_P2, ROSSI\_MARCO\_ESERCIZIO, ROSSI\_MARCO\_549449384, etc.
6. Il file da consegnare deve essere creato da eclipse seguendo i passi:
  - a. Seleziona “export...” nel menu file
  - b. Seleziona “Archive File” in “General”
  - c. Pressa “Next”
  - d. Seleziona progetto da esportare
  - e. Controllare il percorso del file (nell’area di testo con etichetta “To archive file:”)
  - f. Assicurarsi che i pulsanti radio nel pannello Options siano selezionati su “Save in zip format” e “Create directory structure for files”
  - g. Pressa “Finish”

Assicurarsi che i progetti consegnati possono essere importati in eclipse come:  
General → Existing Projects into Workspace