

ESERCIZIO 1

- Si crei un progetto Java che mappi un semplice sistema medico. Il progetto deve contenere **tre classi**: una per i pazienti, una per le patologie e una per il main.
- Inoltre si definiscano le seguenti eccezioni: **MoltoGraveException** e **GuaritaException**.

1. La classe delle patologie deve avere come **attributi** con le dovute visibilità:

1. nome della patologia (stringa);
2. grado di gravità (intero) da 1(guarito) a 10 (molto grave).

e come metodi:

- **costruttore**
- **getter di nome e gravità**
- **void aggrava() e void attenua()**, che aumentano o

ESERCIZIO 1

- diminuiscono la gravità di 1 grado se è possibile, o altrimenti lanciano la MoltoGraveException (gravità 10) e la GuaritaException (gravità 1) rispettivamente.
- **boolean equals(Object o)**, che controlla se l'oggetto passato (una patologia) è uguale a quello su cui è stato invocato (per nome)
- **String toString()**, un metodo che restituisce i valori degli attributi.

1. La classe dei pazienti deve avere come **attributi** con le dovute visibilità:

1. cognome (stringa);
2. patologie, un ArrayList di patologie.

e come metodi:

ESERCIZIO 1

- **costruttore**
- **getter del cognome**
- **void aggiungiPatologia(Patologia p)**, che sfrutta il metodo **contains** di ArrayList per verificare che la patologia non sia già presente, ed eventualmente la aggiunge
- **boolean rimuoviPatologia(Patologia p)**, che sfrutta il metodo **contains** di ArrayList per verificare che la patologia sia presente e in tal caso la rimuove (restituendo true), altrimenti restituisce false
- **String toString()**, che restituisce una stringa formattata per la stampa chiamando il metodo toString() delle patologie

ESERCIZIO 1

3. La classe del main deve:

- creare 2 patologie: asma con gravità 10 e allergia alle graminacee con gravità 2;
- creare il paziente Rossi;
- aggiungere al paziente le 2 patologie e stampare a video le informazioni sul paziente;
- aggravare l'asma del paziente e gestire l'eccezione **MoltoGraveException** (facendone stampare il messaggio);
- attenuare l'allergia fino a portarla a 0, gestendo l'eccezione **GuaritaException** in modo da togliere la patologia al paziente;
- stampare la situazione aggiornata sul paziente.

ESERCIZIO 2

- Si realizzi una classe **Call**, con attributi di tipo stringa **sender**, **receiver** e di tipo intero **start** e **stop**. Tale classe, che rappresenta una singola chiamata telefonica, codifica un metodo costruttore a 4 argomenti e implementa il metodo **public int getDuration()**. Il metodo restituisce un intero che rappresenta la durata della chiamata, che è calcolata come la differenza tra **stop** e **start**.
- Si realizzi poi un metodo **main** in una classe **Prova** che:
 1. Legga, riga per riga, il file di testo *data.txt* che è così strutturato: ogni riga contiene il numero del chiamante (**sender**), il numero del chiamato (**receiver**),

ESERCIZIO 2

l'istante iniziale della chiamata (**start**) e quello finale (**stop**), separati da “;”.

Esempio del contenuto:

```
0351 3194181;0341 6859980;420336000;422142826  
0362 6516225;0382 4413043;421977600;422224970
```

2. Crei, per ogni riga del file, un oggetto **Call** usando il metodo **split** della classe String (vedi API) per ottenere i 4 valori (sender, receiver, start, stop) da passare al costruttore. Il metodo **split** suddivide la stringa su cui è invocato in tante stringhe separate in base ad un carattere specificato e le restituisce sotto forma di vettore di oggetti di tipo **String**.

ESERCIZIO 2

3. Inserisca in una lista (scegliere la migliore implementazione da JCF) tutte le istanze delle chiamate la cui durata è superiore a 550000 millisecondi. Non si conosce a priori quanti elementi dovrà contenere la lista.
4. Stampi a video la dimensione della lista.

Esempio di output a video:

Totale: 100 chiamate lunghe.