

ESERCIZIO SULLE INTERFACCE

Parte I

Creare un'interfaccia *OperazioniAnimale*, che esponga i seguenti metodi:

- *public Date getProssimoAppuntamento()*, che restituisce la data di un prossimo appuntamento
- *public void calcolaProssimoAppuntamento()*, che calcola la data di un prossimo appuntamento

Parte II

Definire una classe astratta *Animale* che contenga i seguenti dati:

- codice (stringa)
- nome (stringa)
- razza (stringa)
- età (intero)
- giorni che devono trascorrere prima della prossima visita (intero)
- data prossimo appuntamento (una data, inizialmente nulla)
- data ultimo appuntamento (data)

La classe espone il seguente metodo

- *public abstract void stampaDatiAnimale()*, che stampa tutti i dati dell'animale, ad eccezione delle date degli appuntamenti e dei giorni che devono trascorrere prima della prossima visita.

Parte III

Definire due classi che estendono *Animale*, avendo cura che ciascuna abbia campi propri oltre a quelli ereditati dalla classe astratta, che dovranno essere stampati dalla funzione che fa overriding, assieme ai dati generali.

Parte IV

Definire una classe *ClinicaVeterinaria*, con i seguenti campi:

- 1) Nome (stringa)
- 2) Indirizzo (stringa)
- 3) Lista animali in cura (array di animali)

La classe espone i seguenti metodi:

- a. *public int getNumeroAnimali()*, che restituisce il numero di animali seguiti dalla clinica

- b. *public void aggiungiAnimale(Animale animale)*, che aggiunge un nuovo animale alla lista se non è già presente uno con lo stesso codice, altrimenti lancia un'eccezione. Se la lista è piena e bisogna aggiungere un nuovo animale, creare una nuova lista lunga il doppio della precedente
- c. *public void eliminaAnimale(String codice)*, che elimina l'animale col codice fornito dalla lista, se presente, altrimenti lancia un'eccezione.
In caso di eliminazione, far scorrere la lista in avanti per eliminare eventuali buchi.
- d. *public void visitaAnimale(String codice)*, che registra la visita all'animale indicato dal codice, aggiornando la data dell'ultimo appuntamento con quella odierna
- e. *public void stampaListaAnimali()*, che stampa i dati completi di ogni animale nella lista (ad eccezione delle date degli appuntamenti e del numero di giorni tra due visite)
- f. *public void prenotaAppuntamento(String codice)*, che aggiorna la data del prossimo appuntamento per l'animale indicato da *codice*. Se il codice non è nella lista, si lancia un'eccezione.

Parte V

Definire una classe *ServiziVeterinari*, che esponga il seguente metodo:

- *public static void stampaVisite(ClinicaVeterinaria clinica)*, che stampa tutti gli animali della clinica, scrivendo su ogni riga il codice di ogni animale, la data dell'ultimo appuntamento (se non presente, stampare "ND"), la data del prossimo appuntamento (se non presente, stampare "ND")

Parte VI

Realizzare una classe di test per verificare il funzionamento delle classi create ai punti IV e V.

Hints

Vi sarà utile sapere che una data (un oggetto della classe *java.util.Date*) può essere convertita in una stringa in un certo formato, per esempio il classico gg-mm-aaaa. Allo stesso modo, una stringa che contiene una data in un certo formato, può essere trasformata in oggetto *Date*:

```
/* trasformare una data in una stringa con un formato specificato */  
SimpleDateFormat formatter = new SimpleDateFormat("dd-MM-yyyy HH:mm");  
Date today = new Date(); // crea un oggetto con la data corrente
```

```
/* costruire una data a partire da una stringa in un formato specifico
*/
String formatoData = "25-04-2021 12:15";
Date formattedDate;
try
{
    formattedDate = formatter.parse(formatoData);
}

catch (ParseException e)
{
    e.printStackTrace();
}
```

Per svolgere l'esercizio, vi sarà necessario aggiungere dei giorni alle date. Ecco il codice che può farlo:

```
Calendar c = Calendar.getInstance();
c.setTime(today);
c.add(Calendar.DATE, 10); // esempio di aggiunta di dieci giorni
System.out.println(formatter.format(c.getTime()));
```

Per poter utilizzare le due classi *Date* e *Calendar* sarà necessario aggiungere al vostro codice i seguenti import:

```
import java.util.Calendar;
import java.util.Date;
```