Esame di Programmazione II, 24 febbraio 2016

Si intende programmare un registro delle iscrizioni in palestra. Per esempio, il seguente programma crea delle iscrizioni di tre utenti, le stampa in ordine cronologico e poi le proietta sul 2015, ristampando quelle del 2015:

```
public class Main {
  public static void main(String[] args)
      throws IscrizioneSovrappostaException, IscrizioneTroppoLungaException, IscrizioneVuotaException {

    Registro registro = new Registro();
    Utente u1 = new Utente("Fausto", "Spoto");
    Utente u2 = new Utente("Alan", "Turing");
    Utente u3 = new Utente("Noam", "Chomsky");
    registro.aggiungi(new Iscrizione(u1, Mese.SETTEMBRE, 2014, Mese.GENNAIO, 2015));
    registro.aggiungi(new Iscrizione(u1, Mese.SETTEMBRE, 2015, Mese.FEBBRAIO, 2016));
    registro.aggiungi(new Iscrizione(u1, Mese.MAGGIO, 2016, Mese.LUGLIO, 2016));
    registro.aggiungi(new Iscrizione(u3, Mese.AGOSTO, 2015, Mese.SETTEMBRE, 2015));
    registro.aggiungi(new Iscrizione(u2, Mese.DICEMBRE, 2015, Mese.NOVEMBRE, 2016));
    registro.aggiungi(new Iscrizione(u3, Mese.FEBBRAIO, 2016, Mese.AGOSTO, 2016));

    System.out.println(registro);

    Registro registro2015 = registro.proiettaSu(2015);

    System.out.println(registro2015);
  }
}
```

La sua esecuzione dovrà stampare:

```
Fausto Spoto: dall'inizio di SETTEMBRE 2014 alla fine di GENNAIO 2015: 200 euro
Noam Chomsky: dall'inizio di AGOSTO 2015 alla fine di SETTEMBRE 2015: 100 euro
Fausto Spoto: dall'inizio di SETTEMBRE 2015 alla fine di FEBBRAIO 2016: 210 euro
Alan Turing: dall'inizio di DICEMBRE 2015 alla fine di NOVEMBRE 2016: 360 euro
Noam Chomsky: dall'inizio di FEBBRAIO 2016 alla fine di AGOSTO 2016: 245 euro
Fausto Spoto: dall'inizio di MAGGIO 2016 alla fine di LUGLIO 2016: 120 euro

Fausto Spoto: dall'inizio di SETTEMBRE 2014 alla fine di GENNAIO 2015: 200 euro
Noam Chomsky: dall'inizio di AGOSTO 2015 alla fine di SETTEMBRE 2015: 100 euro
Fausto Spoto: dall'inizio di SETTEMBRE 2015 alla fine di FEBBRAIO 2016: 210 euro
Alan Turing: dall'inizio di DICEMBRE 2015 alla fine di NOVEMBRE 2016: 360 euro
```

Esercizio 1 [2 punti] Si scriva l'`enum Mese` con 12 costanti per i dodice mesi dell'anno, da `GENNAIO` a `DICEMBRE`.

Esercizio 2 [5 punti] Si completi la seguente classe che descrive un utente di una palestra (campi, costruttori e metodi aggiuntivi devono essere definiti `private`):

```
public class Utente implements Comparable<Utente> {
  ...
  public Utente(String nome, String cognome) { ... }
  public boolean equals(Object other) { devono avere stesso nome e stesso cognome }
  public int hashCode() { non deve essere banale }
  public int compareTo(Utente other) { li mette in ordine per cognome e poi per nome }
  public String toString() { ritorna il nome, uno spazio e il cognome }
}
```

Esercizio 3 [14 punti] Si completi la seguente classe che descrive un'iscrizione in palestra (campi, costruttori e metodi aggiuntivi devono essere definiti `private`). Il costruttore richiede di specificare l'utente che si iscrive, il mese a partire dal quale si iscrive (primo giorno del mese) e il mese fino a quando si iscrive (ultimo giorno del mese). Per esempio, iscriversi da febbraio 2015 a marzo 2015 significa iscriversi dal primo febbraio 2015 al 31 marzo 2015. Si scrivano le classi delle due eccezioni controllate lanciate nel costruttore.

```
public class Iscrizione implements Comparable<Iscrizione> {
  ...
  public Iscrizione(Utente utente, Mese meseInizio, int annoInizio, Mese meseFine, int annoFine)
      throws IscrizioneTroppoLungaException, IscrizioneVuotaException {
    ...
    se il mese di inizio viene cronologicamente dopo il mese di fine, lancia una IscrizioneVuotaException
    se l'iscrizione va oltre i 12 mesi, lancia una IscrizioneTroppoLungaException
  }

  public Utente getUtente() { ritorna l'utente che si e' iscritto }
  public boolean equals(Object other) { devono avere stesso utente, stesso inizio e stessa fine }
  public int hashCode() { non deve essere banale }
  public int compareTo(Iscrizione other) {
    li mette in ordine per inizio. A parita' di inizio, li mette in ordine per utente.
    A parita' di inizio e di utente, li mette in ordine per fine
  }
  public String toString() {
    ritorna una stringa del tipo: "Fausto Spoto: dall'inizio di SETTEMBRE 2014 alla fine di GENNAIO 2015"
  }
  public boolean sovrappostaCon(Iscrizione other) {
    determina se this e other sono sovrapposte nel tempo (anche parzialmente)
  }
  public int costo() {
    ritorna il costo dell'iscrizione: per 1 o 2 mesi costa 50 euro al mese,
    da 3 a 5 mesi costa 40 euro al mese, da 6 a 11 mesi costa 35 euro al mese, per 12 mesi costa 30 euro al mese
  }
  public boolean relativaAl(int anno) { determina se e' relativa all'anno indicato, anche parzialmente }
}
```

Esercizio 4 [10 punti, facoltativo per chi ha già fatto il progetto degli anni passati] Si completi la seguente classe, che rappresenta un registro di iscrizioni in palestra in ordine crescente, e si implementi l'eccezione controllata lanciata dal metodo `aggiungi` (campi, costruttori e metodi aggiuntivi devono essere definiti `private`):

```
import java.util.SortedSet;
import java.util.TreeSet;

public class Registro {
  private final SortedSet<Iscrizione> iscrizioni = new TreeSet<Iscrizione>();

  public void aggiungi(Iscrizione iscrizione) throws IscrizioneSovrappostaException {
    aggiunge l'iscrizione indicata al registro. Se ne e' gia' presente una
    sovrapposta e per lo stesso utente, lancia una IscrizioneSovrappostaException
  }

  public String toString() {
    ritorna la stringa delle iscrizioni di questo registro, in ordine, riportando il costo
    accanto a ciascuna iscrizione
  }

  public Registro proiettaSu(int anno) {
    ritorna un nuovo registro che contiene tutte e sole le iscrizioni di questo registro che
    riguardano l'anno indicato, anche parzialmente. Questo registro non viene modificato
  }
}
```