

Esame di Programmazione II, 5 luglio 2023 (si consegnano Main.java e PunishableSet.java)

Si crei un progetto Eclipse e il package `it.univr.sets`. Si copino al suo interno le classi del compito. Non si modifichino le dichiarazioni dei metodi e delle classi. Si possono definire altri campi, metodi, costruttori e classi, ma devono essere *private*. C'è solo un metodo *public* da aggiungere. La soluzione che verrà consegnata dovrà compilare, altrimenti non verrà corretta.

Un *insieme punibile* è un insieme di elementi, in cui ogni elemento ha associato un punteggio. Per default, il punteggio inizialmente è 1000. Per esempio il seguente codice:

```
String[] names = { "Fausto", "Samantha", "Giulio", "Giovanna" };
PunishableSet<String> set = new PunishableSet<>(names); // tutti partono da 1000 punti
System.out.println(set);
```

stampa:

```
Giovanna: 1000 points
Fausto: 1000 points
Samantha: 1000 points
Giulio: 1000 points
```

Esiste un altro costruttore che permette di specificare il punteggio iniziale per gli elementi appena aggiunti a un insieme, per esempio 500 per i nomi che terminano in o e 800 per gli altri (l'interfaccia di libreria `ToIntFunction<E>` è una funzione da E ad int):

```
ToIntFunction<String> init = new ToIntFunction<>() {
    public int applyAsInt(String s) { return s.endsWith("o") ? 500 : 800; }
};
set = new PunishableSet<>(names, init); // partono da 500 o 800 punti
set.add("Rocco"); // Rocco parte da 500 punti
System.out.println(set);
```

stampa:

```
Giovanna: 800 points
Fausto: 500 points
Rocco: 500 points
Samantha: 800 points
Giulio: 500 points
```

L'aggiunta di elementi a un insieme si fa con il metodo `add`, la rimozione con `remove`. La riduzione dei punti di un elemento si fa con `punish` e l'aumento con `pardon`. Se i punti di un elemento raggiungono zero o diventano negativi, quell'elemento viene eliminato. Ad esempio:

```
set = new PunishableSet<>(names); // tutti partono da 1000 punti
set.punish("Fausto", 400); // Fausto perde 400 punti
set.punish("Giulio", 1010); // Giulio ne perde 1010 e viene eliminato
System.out.println(set);
```

stampa:

```
Giovanna: 1000 points
Fausto: 600 points
Samantha: 1000 points
```

Esercizio 1 (7 punti). Si completino i tre costruttori di `PunishableSet`, cioè i due che abbiamo usato negli esempi precedenti e un altro che permette di specificare anche del codice che deve essere eseguito ogni volta che viene aggiunto o rimosso un elemento dall'insieme.

Esercizio 2 (14 punti). Si completino i metodi `add`, `remove`, `punish` e `pardon` di `PunishableSet`. Si legga bene la loro specifica nel commento che precede la dichiarazione dei metodi.

Esercizio 3 (3 punti). Si renda `PunishableSet` iterabile, in modo che iterando su un tale oggetto si ottengano gli elementi dentro l'insieme (si tratta di aggiungere un solo metodo `public`).

Esercizio 4 (7 punti). Si completi il `Main`, creando i `PunishableSet` come richiesto nel codice. Dovrebbe stampare qualcosa del tipo:

```
set1:
Aurora: 1000 points
Fausto: 500 points
Samantha: 1000 points

set2:
Aurora: 6 points
Samantha: 8 points

set3:
Aurora: 4 points
Samantha: 3 points

set4:
Giovanna: 1000 points
Aurora: 2000 points
Fausto: 1500 points
Samantha: 2000 points

adding Fausto
adding Samantha
adding Giulio
adding Giovanna
adding Aurora
removing Giovanna
removing Giulio
set5:
Aurora: 600 points
Fausto: 100 points
Samantha: 600 points

Exception in thread "main" java.lang.IllegalArgumentException
```