

Esame di Programmazione II, 3 febbraio 2016

Si consideri un'enumerazione di tutti i partiti che si presentano alle elezioni:

```
public enum Partito {
    PENSIONATI,
    FELICE,
    FLOREALE,
    CAOTICO,
    BASSOTTI;

    public final static int NUMERO_PARTITI = elementi().length;

    public static Partito[] elementi() {
        return values();
    }

    public int indice() {
        for (int pos = 0; pos < NUMERO_PARTITI; pos++)
            if (elementi()[pos] == this)
                return pos;

        throw new RuntimeException("impossibile");
    }
}
```

Si noti che questa enumerazione definisce una costante per il numero totale dei partiti, un metodo `elementi()` che permette di ottenere l'array di tutti i partiti e un metodo `indice()` che restituisce l'indice di un dato partito dentro l'array di tutti i partiti.

Esercizio 1 [11 punti] Si completi la classe che rappresenta un'elezione. Essa permette di registrare i voti per i vari partiti:

```
public class Elezioni {
    ...
    public void vota(Partito partito) { ...registra un altro voto per il partito indicato }

    @Override
    public String toString() {
        ...sulla base dei voti registrati per ciascun partito, restituisce una stringa del tipo:

        Risultati elezioni:

        1      PENSIONATI:  7762 voti (33.03%)
        2      FELICE:     3216 voti (13.68%)
        3      FLOREALE:   5695 voti (24.23%)
        4      CAOTICO:    5731 voti (24.39%)
        5      BASSOTTI:   1098 voti (4.67%)
    }

    protected final int numeroVotiEspressi() {
        ...restituisce il numero totale di voti registrati
    }

    protected final int votiPer(Partito partito) {
        ...restituisce il numero totale di voti registrati per il partito indicato
    }
}
```

Esercizio 2 [10 punti] Si definisca una sottoclasse `ElezioniIstogramma` di `Elezioni`, la cui unica differenza rispetto alla sua superclasse è nel metodo `toString()` che, *oltre a generare la stessa stringa che nella superclasse*, deve aggiungere un istogramma di 50 caratteri che indicano in proporzione i voti ottenuti per ciascun partito. Per esempio deve restituire una stringa del tipo:

Risultati elezioni:

```
1     PENSIONATI: 7762 voti (33.03%)
2     FELICE: 3216 voti (13.68%)
3     FLOREALE: 5695 voti (24.23%)
4     CAOTICO: 5731 voti (24.39%)
5     BASSOTTI: 1098 voti (4.67%)
```

11111111111111111111222222333333333333333344444444444444445

Esercizio 3 [10 punti, facoltativo per chi ha già fatto il progetto degli anni passati] Si consideri la seguente classe che rappresenta una coppia di un partito e dei voti che ha ottenuto a un'elezione:

```
public class VotiPerPartito {
    private final Partito partito;
    private final int voti;

    public VotiPerPartito(Partito partito, int voti) {
        this.partito = partito;
        this.voti = voti;
    }

    @Override
    public String toString() {
        return partito + ": " + voti;
    }
}
```

Scrivete le modifiche al codice fin qui realizzato in modo da rendere gli oggetti di classe `Elezioni` ed `ElezioniIstogramma` iterabili su `VotiPerPartito` (questo significa che iterando su tali oggetti si deve ottenere oggetti `VotiPerPartito` che rappresentano i voti per ciascun partito, uno dopo l'altro).