Esame di Programmazione II, 16 giugno 2022

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

Esercizio 1 (5 punti)
(si consegni `Dado.java`)

Si completi la classe astratta `Dado.java`. Essa rappresenta un dado con un numero prefissato di facce, che può essere lanciato ottenendo un numero tra uno e il suo numero di facce (inclusi).

Esercizio 2 (5 punti)
(si consegnino `D6.java`, `D8.java` e `D10.java`)

Si creino tre sottoclassi concrete di `Dado.java`, che rappresentano rispettivamente un dado a sei facce (`D6.java`), un dado a otto facce (`D8.java`) e un dado a dieci facce (`D10.java`).

Esercizio 3 (15 punti)
(si consegni `Lanci.java`)

Si completi la classe `Lanci.java`, che rappresenta l'esecuzione di più lanci con dei dadi. Il numero dei lanci da effettuare e i dadi da usare sono forniti al costruttore. Tale costruttore dovrà lanciare i dadi per il numero di lanci richiesto, tenendo traccia dei numeri ottenuti (quando si lanciano più dadi, il numero ottenuto è la somma dei numeri ottenuti da ciascun dado). Il metodo `toString()` restituisce i numeri ottenuti dal costruttore, tra parentesi quadre e separati da virgole. Per esempio, lanciando dieci volte due dadi con sei facce si potrebbe ottenere la stringa `[4, 2, 6, 12, 2, 9, 8, 8, 7, 11]` chiamando `toString()`. Il metodo `frequenze()` restituisce una stringa che descrive i numeri ottenuti dal costruttore come istogrammi di asterischi, di lunghezza proporzionale alla frequenza del numero ottenuto, seguiti dalla frequenza percentuale con cui il numero è stato ottenuto. Si vedano gli esempi alla pagina seguente.

Esercizio 4 (6 punti)
(si consegni `LanciBarreDiverse.java`)

Si completi la sottoclasse `LanciBarreDiverse.java` di `Lanci.java`, che si comporta in modo identico a `Lanci.java` ma stampa le barre degli istogrammi con tre caratteri diversi, come nell'ultimo esempio della pagina seguente.

Se tutto è corretto, l'esecuzione di `Main.java` deve stampare qualcosa del tipo:

```
Lanciamo 20 volte due dadi a sei facce
Lanci ottenuti: [6, 6, 8, 5, 8, 8, 8, 8, 9, 7, 6, 7, 9, 7, 4, 4, 8, 3, 2, 3]
  2: ***** (5.0%)
  3: ********** (10.0%)
  4: ********** (10.0%)
  5: ***** (5.0%)
  6: *************** (15.0%)
```

```
  7: *************** (15.0%)
  8: ****************************** (30.0%)
  9: ********** (10.0%)
 10:  (0.0%)
 11:  (0.0%)
 12:  (0.0%)

Lanciamo 10000 volte un dado a sei facce e uno a dieci facce
  2: * (1.5%)
  3: *** (3.4%)
  4: **** (5.0%)
  5: ******* (7.3%)
  6: ******** (8.2%)
  7: ********* (9.7%)
  8: ********** (10.3%)
  9: ********* (9.9%)
 10: ********* (9.6%)
 11: ********** (10.1%)
 12: ******** (8.5%)
 13: ****** (6.3%)
 14: ***** (5.2%)
 15: *** (3.4%)
 16: * (1.7%)

Lanciamo 10000 volte un dado a otto facce
  1: ************ (12.5%)
  2: ************ (12.4%)
  3: ************ (12.2%)
  4: ************ (12.3%)
  5: ************ (12.3%)
  6: ************* (13.4%)
  7: ************ (12.8%)
  8: ************ (12.1%)

Lanciamo 10000 volte tre dadi a sei facce
  3:  (0.5%)
  4: * (1.5%)
  5: ** (2.7%)
  6: **** (4.7%)
  7: ****** (7.0%)
  8: ********* (9.5%)
  9: *********** (11.5%)
 10: ************ (12.8%)
 11: ************ (12.6%)
 12: *********** (11.6%)
 13: ********* (9.4%)
 14: ****** (6.8%)
 15: **** (4.6%)
 16: ** (2.9%)
 17: * (1.4%)
 18:  (0.5%)

Lanciamo 10000 volte tre dadi a sei facce, usando barre diverse
  3:  (0.4%)
  4: @ (1.4%)
  5: ++ (3.0%)
  6: **** (4.6%)
  7: @@@@@@@ (7.3%)
  8: +++++++++ (9.6%)
  9: *********** (11.6%)
 10: @@@@@@@@@@@@ (12.5%)
 11: ++++++++++++ (12.1%)
 12: ************ (12.0%)
 13: @@@@@@@@@ (9.4%)
 14: +++++++ (7.1%)
 15: **** (4.5%)
 16: @@ (2.7%)
 17: + (1.4%)
 18:  (0.5%)
```