

**LABORATORIO DI PROGRAMMAZIONE 2009–2010**  
**EDIZIONE 2 – TURNI A E B**  
**EDIZIONE 3 (SERALE)**  
**PROVA D'ESAME**  
**20.IX.2010**

VINCENZO MARRA

*AVVERTENZA. Non è ammesso l'uso delle classi del package `prog.io` allegato al libro di testo del corso.*

ESERCIZIO 1

*La classe Dado.*

Scrivete una classe di nome `Dado`, una cui istanza rappresenti un dado a 6 facce, numerate da 1 a 6.

Scegliete se dotare la classe uno o più costruttori espliciti, o se adottare il solo costruttore d'ufficio implicito. La classe deve essere dotata almeno dei seguenti attributi e metodi. Potrete aggiungere altri attributi e metodi, se lo riterrete utile ai fini della risoluzione del tema d'esame.

- Un attributo privato di nome `risultati` che permetta di tenere traccia dei risultati di un numero non determinato a priori di lanci del dado. Usate il tipo parametrico `Vector<E>`, con tipo parametro `E` appropriato.
- Un metodo di nome `lancia`, di prototipo appropriato, che esegua un lancio del dado, e ne memorizzi il risultato nell'attributo di cui al punto precedente. Per estrarre a caso il valore del risultato del lancio, usate la classe `java.util.Random`, il suo costruttore `Random()`, e il suo metodo `nextInt(int n)`. Consultate la documentazione della classe. Prestate attenzione a quale sia il punto migliore della classe `Dado` in cui istanziare la classe `Random`.
- Un metodo di prototipo `int risultato(int n)`, che restituisca il risultato del lancio numero `n` del dado, se tale lancio è stato eseguito tramite il metodo `lancia` di cui al punto precedente, e restituisca `-1` altrimenti.
- Un metodo di nome `numeroLanci`, che restituisca il numero di lanci eseguiti (tramite il metodo `lancia` di cui al punto precedente) fino al momento dell'invocazione del metodo. Se non è stato eseguito nessun lancio, il metodo restituisce `-1`.

ESERCIZIO 2

*La classe CoppiaDiDadi.*

Scrivete una classe di nome **CoppiaDiDadi**, una cui istanza rappresenti una coppia di istanze della classe **Dado** sviluppata nell'Esercizio 1.

Scegliete innanzitutto se prevedere o no una relazione di ereditarietà fra la classe **CoppiaDiDadi** e la classe **Dado**. Scegliete poi se dotare la classe di uno o più costruttori espliciti, o se adottare il solo costruttore d'ufficio implicito. La classe deve essere dotata almeno dei seguenti attributi e metodi. Potrete aggiungere altri attributi e metodi, se lo riterrete utile ai fini della risoluzione del tema d'esame.

- Due attributi privati di nome **dado1** e **dado2**, di tipo appropriato.
- Un metodo di nome **lancia**, di prototipo appropriato, che esegua un lancio di ciascuno dei due dadi.
- Un metodo di prototipo **int risultato(int n)**, che restituisca la somma dei risultati del lancio numero **n** di ciascun dado, se tali lanci sono stati eseguiti tramite il metodo **lancia** di cui al punto precedente, e restituisca **-1** altrimenti.
- Un metodo **toString**, che ridefinisce l'omonimo metodo ereditato da **Object**. Il metodo restituisce una rappresentazione sotto forma di stringa dei risultati

### ESERCIZIO 3

*La classe **PartitaDadi**.*

Scrivete una classe di nome **PartitaDadi**, contenente il solo metodo **main**, una cui esecuzione simuli una partita a dadi fra due giocatori. Ciascun giocatore lancia, a turno, una coppia di dadi. Il punto ottenuto in un singolo lancio è dato dalla somma dei punti ottenuti con ciascun dado. Una partita consiste di  $n$  lanci consecutivi da parte del giocatore A, seguiti da  $n$  lanci consecutivi da parte del giocatore B. Vince il giocatore che ottiene il punteggio *totale* più alto dopo gli  $n$  lanci di ciascun giocatore. Se il punteggio ottenuto dai due giocatori è uguale, la partita è patta. L'intero  $n > 0$  può cambiare per ciascuna esecuzione del programma.

- Il metodo **main** accetta un unico argomento dalla riga di comando. Se sulla riga di comando è presente più di un argomento, o non è presente alcun argomento, il programma termina con un messaggio d'errore appropriato.
- Se l'unico argomento presente sulla riga di comando non è interpretabile come un numero intero  $n$  che soddisfa la condizione  $n > 0$ , il programma termina con un messaggio d'errore appropriato.
- Altrimenti, il programma usa la classe **CoppiaDiDadi** per simulare la partita a dadi di  $n$  lanci fra A e B.
- Il programma visualizza quindi le informazioni seguenti.
  - L'esito della partita (vince A, vince B, è patta) con i relativi punteggi totali.
  - I punti ottenuti da A e da B lancio per lancio.
  - Le due medie aritmetiche dei punti ottenuti (separatamente) da A e da B negli  $n$  lanci. Si ricorda che la media aritmetica di  $n > 0$  numeri  $r_1, \dots, r_n$  è data da  $\frac{r_1 + \dots + r_n}{n}$ .
  - Il minimo e massimo punto ottenuto da A, e il minimo e massimo punto ottenuto da B, negli  $n$  lanci.

CLASSI DA CONSEGNARE

- (1) Dado
- (2) CoppiaDiDadi
- (3) PartitaDadi

Per la consegna, eseguite l'upload dei singoli file sorgente dalla pagina web:  
<http://upload.dico.unimi.it>.

(V. Marra) DIPARTIMENTO DI INFORMATICA E COMUNICAZIONE, UNIVERSITÀ DEGLI STUDI DI  
MILANO, VIA COMELICO, 39-41, I-20135 MILAN, ITALY

*E-mail address:* `marra@dico.unimi.it`