

### Note sullo svolgimento della prova

- Non è consentito l'uso di vecchi testi d'esame, libri, o appunti.
- Utilizzare NetBeans per editare, compilare ed eseguire i programmi.
- Creare un file **Cognome\_matricola.zip** contenente i soli file *.java* da consegnare.
- Al termine della prova, dopo il caricamento degli elaborati, il docente mostrerà una possibile soluzione. Dopo aver visto la soluzione, gli studenti avranno la possibilità di riprendere il proprio elaborato.

### Esercizio

Un mazzo è composto da 40 carte. I valori delle carte vanno da 1 a 10. I semi sono cuori, quadri, fiori e picche. I giocatori possono comprare carte a loro scelta. Quando tutte le carte sono state vendute, viene estratta una carta a caso da un mazzo gemello. Il giocatore che possiede la carta estratta vince.

Scrivere una classe *Partita* dotata almeno dei seguenti costruttori e metodi:

- *Partita(int n)*: crea una nuova partita con *n* giocatori.
- *Carta compraCarta(int v, Seme s)*: compra la carta con valore *v* e seme *s*. Restituisce un oggetto *Carta* che rappresenta la carta appena acquistata. Lancia *CartaNonDisponibileException* se la carta è già stata acquistata da qualcun altro o *IllegalArgumentException* se gli argomenti non sono validi.
- *Carta[] carteDisponibili()*: restituisce le carte non ancora vendute.
- *boolean attendiEstrazione(Carta[] bb)*: chi chiama questo metodo passa l'array di carte in suo possesso. Il metodo è bloccante fino a quando tutte le carte non sono state vendute e tutti i giocatori non si sono messi in attesa. L'ultimo giocatore che si mette in attesa causa l'estrazione della carta vincente e sblocca gli altri. Il metodo restituisce *true* al vincitore, *false* a tutti gli altri.

Definire anche le classi ausiliarie *Seme* e *Carta* utili a realizzare quanto sopra. Infine, scrivere una classe *Giocatore* che si comporta nel seguente modo:

- quando un giocatore viene attivato compra 10 carte a caso;
- si mette in attesa dell'estrazione;
- stampa un messaggio che riporta l'esito dell'estrazione.

### Esempio di esecuzione:

```
Thread-0: ho queste carte [5C, 6P, 1F, 1C, 3Q, 6Q, 7P, 8F, 5P, 8C]
Thread-1: ho queste carte [9Q, 6C, 3F, 5F, 1Q, 2P, 7C, 0C, 4F, 2Q]
Thread-2: ho queste carte [5Q, 0F, 4P, 8P, 7Q, 2C, 9F, 8Q, 0P, 4C]
Thread-3: ho queste carte [9P, 3P, 9C, 6F, 3C, 4Q, 0Q, 7F, 1P, 2F]
Carta estratta: 3C
Thread-2: Non ho vinto
Thread-3: Ho vinto
Thread-0: Non ho vinto
Thread-1: Non ho vinto
```