

# Carte

Una `Carta` è definita da:

- Un valore nel range [A=1,2,3,4,5,6,7,8,9,10,J=11,Q=12,K=13] (per comodità si possono considerare valori interi)
- Un seme nel range [cuori=C, quadri=Q, fiori=F, picche=P] (per comodità si possono usare caratteri o valori interi)

Le carte sono naturalmente ordinate prima per il loro valore e poi per il loro seme nell'ordine specificato sopra. Due carte differiscono se il loro seme o il loro valore sono diversi.

Un `Mazzo` appena creato contiene tutte le 52 carte possibili, ovvero 13 valori per tutti e 4 i semi, e fornisce le seguenti funzionalità:

- Verificare che una carta sia valida (tutte le carte non nulle e con le caratteristiche specificate sopra sono valide)
- Mischiare il mazzo, mettendo le carte in un ordine casuale. A tal fine è possibile utilizzare il metodo `shuffle` della classe `Collections`.
- Riordinare il mazzo, mettendo le carte nel loro ordine naturale
- Fornire le funzionalità di un iteratore standalone di carte, ovvero:
  - Verificare se vi sono ancora delle carte nel mazzo
  - Estrarre e restituire una carta dal mazzo (togliendola dal mazzo)
- Inserire una carta nel mazzo.
  - Se la carta è già presente nel mazzo viene restituita una nuova eccezione unchecked `CardExistsException`
  - Se la carta non è valida per il mazzo viene restituita una nuova eccezione checked `CardNotValidException`

Un `MazzoRidotto` appena creato contiene solo le carte con i valori [A=1,2,3,4,5,6,7,8,9,10] dei 4 semi e fornisce le seguenti funzionalità aggiuntive:

- Ridefinisce la verifica della validità di una carta (solo le carte disponibili alla creazione del `MazzoRidotto` sono valide)
- Ridefinisce il riordino del mazzo prima per il loro seme e poi per il loro valore (utilizzando un nuovo `Comparator`)

Progettare, specificare ed implementare le classi descritte sopra ed i loro metodi. Quando opportuno, le classi devono avere le loro Abstraction Function e Representation Invariant.

Scrivere una classe `Test` con un metodo `main()` per testare le classi create. A tal fine, leggere da **riga di comando** il tipo di mazzo che si intende creare. Dopo la creazione, il mazzo dovrà essere mischiato e stampato su **standard output**. Poi, da **standard input**, leggere una serie di righe nel formato:

- `estrai` se si desidera estrarre una carta dal mazzo
- `inserisci <CarattereValore> <CarattereSeme>` se si desidera inserire una carta. Svolgere le istruzioni fino a quando non viene inserito, con la combinazione `Ctrl+D` il comando `End-Of-File`.

Infine, stampare su **standard output** il mazzo riordinato.

## Esempio d'esecuzione (Mazzo):

```
$ java Test Mazzo
```

```
Mazzo:
```

```
Carta: 4 P
Carta: 9 P
Carta: 2 F
Carta: 13 C
Carta: 1 P
Carta: 6 Q
Carta: 11 F
Carta: 1 F
Carta: 7 F
Carta: 10 Q
Carta: 6 P
Carta: 8 Q
Carta: 12 F
Carta: 2 C
Carta: 5 C
Carta: 3 P
Carta: 9 C
Carta: 5 P
Carta: 1 C
Carta: 11 P
Carta: 12 C
Carta: 12 P
Carta: 13 P
Carta: 5 Q
Carta: 1 Q
Carta: 13 Q
Carta: 9 F
Carta: 11 C
Carta: 7 P
Carta: 11 Q
Carta: 4 Q
Carta: 9 Q
Carta: 7 C
Carta: 6 C
Carta: 8 F
Carta: 4 F
Carta: 3 F
Carta: 7 Q
Carta: 10 F
Carta: 3 C
Carta: 6 F
Carta: 2 Q
Carta: 3 Q
Carta: 8 C
Carta: 12 Q
Carta: 10 C
Carta: 8 P
Carta: 4 C
Carta: 13 F
Carta: 5 F
Carta: 2 P
Carta: 10 P
```

```
estrai
```

```
Estratto: Carta: 4 P
```

estrai

Estratto: Carta: 9 P

estrai

Estratto: Carta: 2 F

inserisci 2 F

Inserito: Carta: 2 F

inserisci 2 F

Mazzo - inserisci: carta già nel mazzo

Mazzo:

Carta: 1 C

Carta: 1 Q

Carta: 1 F

Carta: 1 P

Carta: 2 C

Carta: 2 Q

Carta: 2 F

Carta: 2 P

Carta: 3 C

Carta: 3 Q

Carta: 3 F

Carta: 3 P

Carta: 4 C

Carta: 4 Q

Carta: 4 F

Carta: 5 C

Carta: 5 Q

Carta: 5 F

Carta: 5 P

Carta: 6 C

Carta: 6 Q

Carta: 6 F

Carta: 6 P

Carta: 7 C

Carta: 7 Q

Carta: 7 F

Carta: 7 P

Carta: 8 C

Carta: 8 Q

Carta: 8 F

Carta: 8 P

Carta: 9 C

Carta: 9 Q

Carta: 9 F

Carta: 10 C

Carta: 10 Q

Carta: 10 F

Carta: 10 P

Carta: 11 C

Carta: 11 Q

Carta: 11 F

Carta: 11 P

Carta: 12 C

Carta: 12 Q

Carta: 12 F

Carta: 12 P

Carta: 13 C

Carta: 13 Q

Carta: 13 F

Carta: 13 P

## Esempio d'esecuzione (MazzoRidotto):

```
$ java Test MazzoRidotto
```

```
Mazzo:
```

```
Carta: 2 F
Carta: 2 Q
Carta: 3 C
Carta: 1 F
Carta: 10 C
Carta: 7 P
Carta: 6 Q
Carta: 1 C
Carta: 3 P
Carta: 5 Q
Carta: 6 C
Carta: 7 F
Carta: 8 Q
Carta: 3 Q
Carta: 7 Q
Carta: 4 C
Carta: 4 F
Carta: 9 P
Carta: 5 P
Carta: 1 P
Carta: 1 Q
Carta: 6 P
Carta: 6 F
Carta: 5 C
Carta: 2 C
Carta: 10 P
Carta: 8 C
Carta: 9 Q
Carta: 4 Q
Carta: 10 Q
Carta: 8 P
Carta: 9 C
Carta: 3 F
Carta: 8 F
Carta: 7 C
Carta: 4 P
Carta: 2 P
Carta: 5 F
Carta: 10 F
Carta: 9 F
```

```
inserisci 12 P
```

```
Mazzo - inserisci: carta non valida
estrai
```

```
Estratto: Carta: 2 F
```

```
Mazzo:
```

```
Carta: 1 C
Carta: 2 C
Carta: 3 C
Carta: 4 C
Carta: 5 C
Carta: 6 C
Carta: 7 C
Carta: 8 C
Carta: 9 C
```

Carta: 10 C

Carta: 1 Q

Carta: 2 Q

Carta: 3 Q

Carta: 4 Q

Carta: 5 Q

Carta: 6 Q

Carta: 7 Q

Carta: 8 Q

Carta: 9 Q

Carta: 10 Q

Carta: 1 F

Carta: 3 F

Carta: 4 F

Carta: 5 F

Carta: 6 F

Carta: 7 F

Carta: 8 F

Carta: 9 F

Carta: 10 F

Carta: 1 P

Carta: 2 P

Carta: 3 P

Carta: 4 P

Carta: 5 P

Carta: 6 P

Carta: 7 P

Carta: 8 P

Carta: 9 P

Carta: 10 P