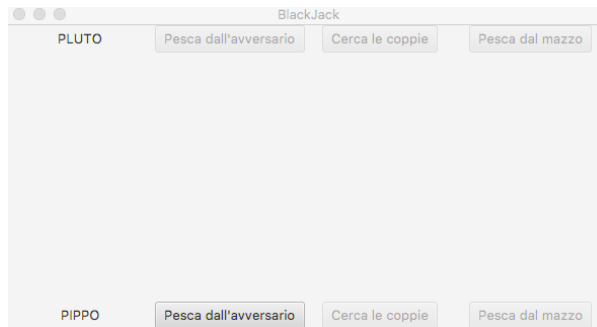


## Esame di Programmazione 2

### Prova al calcolatore – gennaio 23

Un gioco a carte vede opposti due giocatori, ciascuno dei quali è caratterizzato da un nome (es. Pippo e Pluto) che si assume predefinito all'interno del programma.



1) Ciascuna **carta** ha un valore compreso tra 1 e 4, ed un seme (C, Q, F, P). Vi è poi una carta speciale detta BlackJack, rappresentata dal valore 0 e seme J.

2) Una carta può essere stampata in console sotto forma di stringa composta di valore e seme (es. 4-P).

3) La classe che rappresenta le carte dovrà implementare il metodo `equals`. Due carte sono considerate uguali se hanno stesso valore indipendentemente dal loro seme.

4) La classe che rappresenta le carte dovrà implementare un criterio di ordinamento. Il criterio è numerico (sul valore) e, a parità di valore, alfabetico (sul seme). Es.: 1-C, 1-F, 3-C, 4-P, 4-Q.

5) Il **mazzo** è composto da una carta per seme e tipo più il Blackjack, per un totale di 17 carte ( $4 \times 4 + 1$ ). All'inizio di una partita è mescolato in ordine casuale. Il mazzo ha un metodo `main` di test che crea un mazzo, lo mescola e lo stampa in console, poi lo ordina e lo stampa in console.

6) All'inizio del gioco il mazzo viene mescolato, e a ciascun giocatore vengono date 4 carte, che vanno a costituire la sua **mano**. Le carte in una mano sono sempre disposte in ordine. Una mano viene stampata in console indicando il nome del giocatore ed in sequenza tutte le carte contenute.

7) Il gioco viene scandito da operazioni eseguite mediante un'interfaccia grafica (vedi figura iniziale) contenente tre bottoni per ogni giocatore, corrispondenti alle seguenti azioni:

- pesca una carta a caso dalla mano dell'avversario;
  - elimina le coppie di carte uguali nella propria mano se presenti;
  - se non hai eliminato almeno una coppia, pesca una carta dal mazzo, se questo non è vuoto.
- All'inizio del gioco i bottoni sono tutti disabilitati, eccetto quello che definisce la prossima azione eseguibile: "Pesca dall'avversario" del primo giocatore.

8) Premendo il bottone "**Pesca dall'avversario**":

- una carta a caso viene presa dalla mano dell'avversario e aggiunta alla propria mano
- la propria mano viene stampata
- il bottone si disattiva e si attiva il bottone successivo ("Cerca le coppie" per lo stesso giocatore).

9) Il bottone "**Cerca le coppie**":

- trova e scarta le coppie presenti (una coppia è composta di due carte “uguali” secondo il punto 3). Si crei una classe `ListWithRemovablePairs` come estensione di `ArrayList` in modo che fornisca il metodo `scartaCoppie()` che elimina tutte le coppie. Questo deve restituire `true` se almeno una coppia è stata eliminata e `false` altrimenti.
- la propria mano viene stampata
- il bottone si disattiva e si attiva il bottone successivo (“Pesca dal mazzo” per lo stesso giocatore).

10) Il bottone **"Pesca dal mazzo"**:

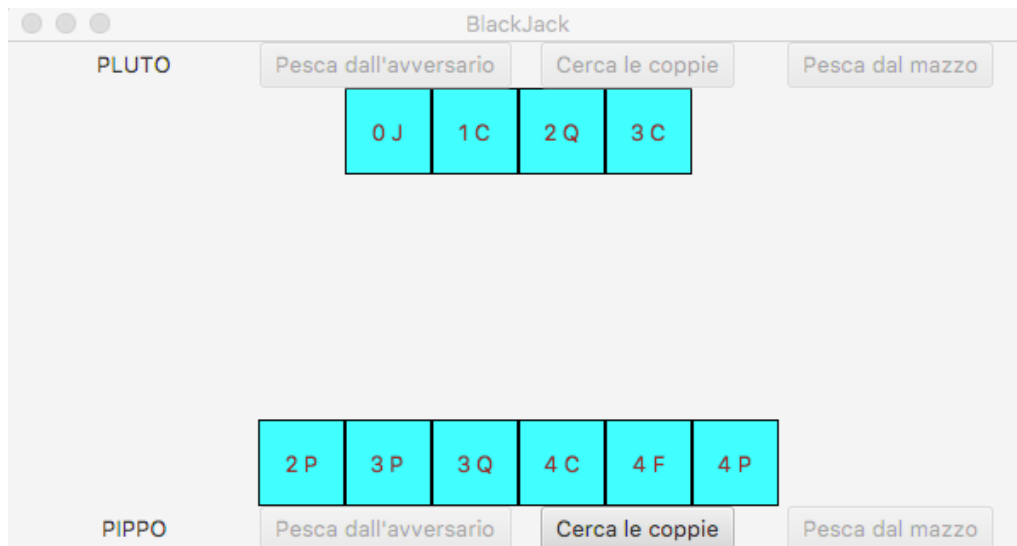
- se è stata scartata almeno una coppia, stampa il nome dell’utente e la scritta “ho scartato, non pesco”; altrimenti pesca una carta dal mazzo e la aggiunge alla propria mano
- stampa la propria mano
- il bottone si disattiva e si attiva il bottone successivo (“Pesca dall'avversario” per l’altro giocatore).

11) Il gioco prosegue iterando i passi 8-10, alternati fra i due giocatori. Vince il giocatore che, in una qualsiasi fase del gioco, rimane senza carte. In questo caso:

- in console viene stampato “<nome del giocatore> ha vinto”
- si apre una finestra nella quale appare la stessa scritta.

**Nota:** si consiglia di implementare in prima battuta la funzionalità grafica di base dei bottoni (disabilitazione e abilitazione del successivo) senza tutta la business logic, ma semplicemente con una stampa “Bottone *tag* premuto”, e solo successivamente passare alla business logic.

12) Si dia una rappresentazione grafica delle mani (oltre a quella testuale in console), nella quale le carte sono rappresentate da rettangoli contenenti una scritta che indica valore e seme. La rappresentazione grafica viene aggiornata ogni volta che il contenuto della mano cambia.



Esempio di output in console per una partita:

PIPPPO: 0-J 1-C 1-Q 3-Q 4-C

PLUTO: 1-P 2-C 2-F 3-F 4-F

===

PIPPPO: 0-J 1-C 1-Q 3-Q 4-C 4-F

PLUTO: 1-P 2-C 2-F 3-F

===

PIPPPO: 0-J 3-Q

===

PIPPPO: ho scartato, non pesco

===

PIPPPO: 3-Q

PLUTO: 0-J 1-P 2-C 2-F 3-F

===

PLUTO: 0-J 1-P 3-F

===

PLUTO: ho scartato, non pesco

===

PIPPPO: 1-P 3-Q

PLUTO: 0-J 3-F

===

PIPPPO: 1-P 3-Q

===

PIPPPO: 1-P 3-Q 4-Q

===

PIPPPO: 1-P 4-Q

PLUTO: 0-J 3-F 3-Q

===

PLUTO: 0-J

===

PLUTO: ho scartato, non pesco

===

PIPPPO: 0-J 1-P 4-Q

PLUTO:

PLUTO ha vinto!