

Black Jack assignment

Create a BlackJack application to simulate a simplified version of the game of "21".

Game requirements:

- 1) You should have at least 4 classes.
- 2) One of the classes should have only static variables and static methods. This class should provide jokes for a player and a dealer. Every time a player wins – display a joke. This class should have a method nextJoke() which will check if the ArrayList of jokes is empty. If it is empty it should call another method loadJokes() that opens a file, loads jokes into an ArrayList and shuffles them. Every time nextJoke() is called a joke is returned and is removed from the ArrayList. You can use your own method names and own structure for this class, however your class should have a similar functionality.
- 3) You should have at least one class that extends the other class.
- 4) A card should be represented by an integer. This is a card ID. Only card ID should be passed from the deck to a player.
- 5) You should use encapsulation, inheritance and polymorphism.
- 6) When totals are calculated consider that Ace values 1 or 11. Make sure you consider the situation with more than one Ace.



Game playing:

- 7) Print a deck of cards at the beginning of the game.
- 8) Shuffle and print the deck again.
- 9) Show player's cards and card's total all the time.
- 10) Show dealer's cards face down. (or one card and other face down).
- 11) Show dealer cards and card's total when the game is over.
- 12) Display the deck after every round.
- 13) Ask if the user would like to play again.
- 14) You should continue playing with the same deck. Introduce a new deck if less than 10 cards are left in a deck of card.
- 15) Show dealer cards and card's total when the game is over.
- 16) Every time a player or dealer wins – display a joke from the Joke class. (7, 9 and 12 are for Ms. Strelkovska – usually you don't display it in a game)

Possible additions.

When you create a dealer randomly pick a dealer's character or give the user a choice to pick For example, a dealer could be risky, conscious, with ability to count cards, or other.

The assignment will be mainly evaluated on the Object Oriented Structure.

Example running:

```
Enter your name:
Bob
K♥ A♥ 2♥ 3♥ 4♥ 5♥ 6♥ 7♥ 8♥ 9♥ 10♥ J♥ Q♥ K♠ A♠ 2♠ 3♠ 4♠ 5♠ 6♠ 7♠ 8♠ 9♠ 10♠ J♠ Q♠ K♣ A♣ 2♣ 3♣ 4♣ 5♣
6♣ 7♣ 8♣ 9♣ 10♣ J♣ Q♣ K♦ A♦ 2♦ 3♦ 4♦ 5♦ 6♦ 7♦ 8♦ 9♦ 10♦ J♦ Q♦
3♠ A♠ Q♠ 3♥ 7♠ 4♥ 6♥ J♠ 7♠ 5♥ 2♠ 4♠ 8♦ 7♥ A♦ 6♠ 10♦ 3♦ 4♠ Q♠ 6♦ K♥ Q♦ 9♠ K♠ 5♠ 8♠ 10♥ 10♠ J♥ 9♥ 2
♠ 8♠ A♥ 4♦ 2♦ J♠ K♠ 5♦ Q♥ 5♠ 2♥ 10♠ 8♥ K♦ A♠ 3♠ J♦ 9♦ 9♠ 6♠ 7♦
-----

dealer's hand: J♦ ?    sum: > 10
Bob's hand: 7♦ 6♠    sum: 13
Hit or stand?
Hit
-----

dealer's hand: J♦ ?    sum: > 10
Bob's hand: 7♦ 6♠ A♠    sum: 14
Hit or stand?
hit
Bob busted with 24!
Bob's hand: 7♦ 6♠ A♠ K♦
computer player 1 busted with 26!
-----

Dealer says: "Your dog ate your coding assignment? It took him a couple bytes"
=====
The dealer wins!
Results:
Bob's hand: 7♦ 6♠ A♠ K♦    sum: 24
computer player 1's hand: 9♠ 9♦ 8♥    sum: 26
dealer's hand: J♦ 3♠ 10♠    sum: 23
Play again? [Y/N]
```

(Ms.S: In this case if both players busted, dealer wins even if the dealer busted as well.)

Second Example:

```
ame>java BlackJack
K♥ A♥ 2♥ 3♥ 4♥ 5♥ 6♥ 7♥ 8♥ 9♥ 10♥ J♥ Q♥ K♥ A♠ 2♠ 3♠ 4♠ 5♠ 6♠ 7♠ 8♠ 9♠ 10♠ J♠ Q♠ K♠ A♠ 2♠
3♠ 4♠ 5♠ 6♠ 7♠ 8♠ 9♠ 10♠ J♠ Q♠ K♠ A♠ 2♠ 3♠ 4♠ 5♠ 6♠ 7♠ 8♠ 9♠ 10♠ J♠ Q♠
K♠ 10♥ 5♥ 6♠ 5♠ A♠ A♠ 8♠ 3♠ J♠ 6♠ 7♠ 10♠ 5♠ 6♥ 2♠ 7♥ 8♥ 3♠ K♠ Q♠ 2♠ 4♠ 8♠ 9♠ 4♠ 2♥ 9♠ J♠
4♠ A♠ 10♠ 6♠ Q♥ 5♠ J♠ 9♠ 3♠ 7♠ 10♠ K♥ 9♥ 4♥ 2♠ Q♠ A♥ J♥ Q♠ 7♠ 3♥ K♠ 8♠

How many players are there? [MAX: 6] 1

For player 1:
  What is their name? Bob
  How much money do they have? 1000

-----
ROUND 1:

Bob's wager [$1000.0]: 10

Dealer's hand: 3♥ ?
Bob's hand: 8♠ K♠ [total: 18]

Bob's hand: 8♠ K♠ [total: 18]
Hit or stand? [1 or 2] 2

Dealer's hand: 3♥ 7♠ [total: 10] Dealer hit!
Dealer's hand: 3♥ 7♠ Q♠ [total: 20] Dealer stands

-----
ROUND 1 RESULTS:

Hands:
Dealer's hand: 3♥ 7♠ Q♠ [total: 20]
Bob's hand: 8♠ K♠ [total: 18] Lost!

Bob: $990.00

Do you want to play another round? [Y/N]

K♠ 10♥ 5♥ 6♠ 5♠ A♠ A♠ 8♠ 3♠ J♠ 6♠ 7♠ 10♠ 5♠ 6♥ 2♠ 7♥ 8♥ 3♠ K♠ Q♠ 2♠ 4♠ 8♠ 9♠ 4♠ 2♥ 9♠ J♠
4♠ A♠ 10♠ 6♠ Q♥ 5♠ J♠ 9♠ 3♠ 7♠ 10♠ K♥ 9♥ 4♥ 2♠ Q♠ A♥ J♥

-----
ROUND 2:

Bob's wager [$990.0]: 600000
You don't have enough money for that.
Bob's wager [$990.0]: 90

Dealer's hand: Q♠ ?
Bob's hand: J♥ A♥ [total: 21]

Bob's hand: J♥ A♥ [total: 21]
Bob has Blackjack!

Dealer's hand: Q♠ 2♠ [total: 12] Dealer hit!
Dealer's hand: Q♠ 2♠ 4♥ [total: 16] Dealer hit!
Dealer's hand: Q♠ 2♠ 4♥ 9♥ [total: 25] Dealer bust!

-----
ROUND 2 RESULTS:

Hands:
Dealer's hand: Q♠ 2♠ 4♥ 9♥ [total: 25]
Bob's hand: J♥ A♥ [total: 21] Winner!

Bob: $1080.00

Do you want to play another round? [Y/N]
```

(Ms.S: In this example the student forgot to display jokes.)