MOPS APCS1 pd3
ARYAN BHATT & HAIYAO LIU 5/1/2017

FINAL PROJECT PROPOSAL
BLACKJACK

Project will allow users to play a typical shoe-game of blackjack from the command line (singleplayer) either 1 on 1 with the dealer or with a chosen number of AI's. For example, the player can choose to play alongside 3 other AI players, or by themselves.

Game basics:
The game will allow for all of the typical blackjack commands/actions (bet, hit, stand, split, double down, insurance, surrender)
Cards will be represented by their number (name for face cards) and a letter {c,d,h,s} to represent suits (clubs, diamonds, hearts, spades, respectively)
Player can request for the current state of the game at any time (a summary will be printed out)
New players will start with $100

Minimum Viable Product:
The game will be between only the player and the dealer, with no additional AI players. It will use a 1 cardThe only two options for the player will be to hit or to stand. The dealer will have play using an extremely simple algorithm in with it will hit once (no matter what) , and then stand (no matter what). There will also be betting, and the player will start with $100.

Player-changeable options:
The player will be able to set how many decks to be used (1-6), how many AI's will play in the game, and how the dealer plays (stand on all 17 or soft 17)
AI's will have difficulty levels that change how well they can count cards (how many played cards the AI can remember)

Turns:
On a player turn, the player will enter actions into the command line and they will be executed until they stand or exceed 21 (lose).
On all turns, player or AI, a summary of their actions and current states will be printed after the turn is completed, including their hand(s) and bet value(s).

Example:

AI 3: status
Jh 10h - 20
bet - 15
total - 75

AI 3: hit
Jh 10h 3s - 23
bust

AI 4 status
etc.

Tools and Concepts from class that we will implement:

- We will create an abstract superclass called Player that will have subclasses User, AI, and Dealer. In doing this, we will implement class hierarchy, which we learned about in class.
- We will also create class Deck, which will have an instance variable called _cards. This instance variable will contain an ArrayList of Strings that represent cards. Deck will have methods (such as shuffle(), firstCard(), deal(), etc.) that implement ArrayList functionality, which we also learned about in class.
- User will have methods that require user input, which we also discussed in class.

Additional features to add if time allows:

- saved player accounts so multiple people can play on the same terminal without affecting others
- multiplayer games so more than one player can play on the same machine at the same time
- a more aesthetic way to print out the current game status