JavaScript is disabled on your browser.

[Skip navigation links](#1fob9te)

* [Overview](http://docs.google.com/overview-summary.html)
* [Package](http://docs.google.com/package-summary.html)
* Class
* [Use](http://docs.google.com/class-use/BlackjackEngine.html)
* [Tree](http://docs.google.com/package-tree.html)
* [Deprecated](http://docs.google.com/deprecated-list.html)
* [Index](http://docs.google.com/index-files/index-1.html)
* [Help](http://docs.google.com/help-doc.html)
* [Prev Class](http://docs.google.com/blackjack/Blackjack.html)
* [Next Class](http://docs.google.com/blackjack/Card.html)
* [Frames](http://docs.google.com/index.html?blackjack/BlackjackEngine.html)
* [No Frames](http://docs.google.com/BlackjackEngine.html)
* [All Classes](http://docs.google.com/allclasses-noframe.html)
* Summary:
* Nested |
* [Field](#3znysh7) |
* Constr |
* [Method](#2et92p0)
* Detail:
* [Field](#tyjcwt) |
* Constr |
* [Method](#35nkun2)

blackjack

## Interface BlackjackEngine

* All Known Implementing Classes: [Blackjack](http://docs.google.com/blackjack/Blackjack.html)  
    
  public interface BlackjackEngine  
  BlackJackEngine - interface defining the methods expected from a class implementing the Blackjack game logic.Author: cmsc132 2006 Copyright (C) 2006 University of Maryland See Also: [Card](http://docs.google.com/blackjack/Card.html)

### Field SummaryFields

|  |  |
| --- | --- |
| * + Modifier and Type | * + Field and Description |
| * + static int | * + [BLACKJACK](http://docs.google.com/blackjack/BlackjackEngine.html#BLACKJACK) |
| * + static int | * + [BUST](http://docs.google.com/blackjack/BlackjackEngine.html#BUST) |
| * + static int | * + [DEALER\_WON](http://docs.google.com/blackjack/BlackjackEngine.html#DEALER_WON) |
| * + static int | * + [DRAW](http://docs.google.com/blackjack/BlackjackEngine.html#DRAW) |
| * + static int | * + [GAME\_IN\_PROGRESS](http://docs.google.com/blackjack/BlackjackEngine.html#GAME_IN_PROGRESS) |
| * + static int | * + [HAS\_21](http://docs.google.com/blackjack/BlackjackEngine.html#HAS_21) |
| * + static int | * + [LESS\_THAN\_21](http://docs.google.com/blackjack/BlackjackEngine.html#LESS_THAN_21) |
| * + static int | * + [PLAYER\_WON](http://docs.google.com/blackjack/BlackjackEngine.html#PLAYER_WON) |

### Method SummaryAll Methods Instance Methods Abstract Methods

|  |  |
| --- | --- |
| * + Modifier and Type | * + Method and Description |
| * + void | * + [createAndShuffleGameDeck](http://docs.google.com/blackjack/BlackjackEngine.html#createAndShuffleGameDeck--)() Creates and shuffles the card deck(s) using a random number generator. |
| * + void | * + [deal](http://docs.google.com/blackjack/BlackjackEngine.html#deal--)() Creates a new deck of cards, and assigns cards to the dealer and player. |
| * + int | * + [getAccountAmount](http://docs.google.com/blackjack/BlackjackEngine.html#getAccountAmount--)() Returns the player's account amount |
| * + int | * + [getBetAmount](http://docs.google.com/blackjack/BlackjackEngine.html#getBetAmount--)() Returns an integer representing the bet amount. |
| * + [Card](http://docs.google.com/blackjack/Card.html)[] | * + [getDealerCards](http://docs.google.com/blackjack/BlackjackEngine.html#getDealerCards--)() Returns dealer's cards. |
| * + int | * + [getDealerCardsEvaluation](http://docs.google.com/blackjack/BlackjackEngine.html#getDealerCardsEvaluation--)() Returns an integer value that can assume the values LESS\_THAN\_21 if the dealer's cards have a value less than 21, BUST if the dealer's cards have a value greater than 21, and BLACKJACK if the dealer has an Ace along with a "10", Jack, Queen, or King. |
| * + int[] | * + [getDealerCardsTotal](http://docs.google.com/blackjack/BlackjackEngine.html#getDealerCardsTotal--)() Returns an array representing the possible value(s) associated with the dealer's cards if the cards represent a value less than or equal to 21. |
| * + [Card](http://docs.google.com/blackjack/Card.html)[] | * + [getGameDeck](http://docs.google.com/blackjack/BlackjackEngine.html#getGameDeck--)() Returns the current deck of cards. |
| * + int | * + [getGameStatus](http://docs.google.com/blackjack/BlackjackEngine.html#getGameStatus--)() Returns an integer representing the game status. |
| * + int | * + [getNumberOfDecks](http://docs.google.com/blackjack/BlackjackEngine.html#getNumberOfDecks--)() Returns the number of decks being used. |
| * + [Card](http://docs.google.com/blackjack/Card.html)[] | * + [getPlayerCards](http://docs.google.com/blackjack/BlackjackEngine.html#getPlayerCards--)() Returns player's cards. |
| * + int | * + [getPlayerCardsEvaluation](http://docs.google.com/blackjack/BlackjackEngine.html#getPlayerCardsEvaluation--)() Returns an integer value that can assume the values LESS\_THAN\_21 if the player's cards have a value less than 21, BUST if the players's cards have a value greater than 21, and BLACKJACK if the player has an Ace along with a "10", Jack, Queen, or King. |
| * + int[] | * + [getPlayerCardsTotal](http://docs.google.com/blackjack/BlackjackEngine.html#getPlayerCardsTotal--)() Returns an array representing the possible value(s) associated with the player's cards if the cards represent a value less than or equal to 21. |
| * + void | * + [playerHit](http://docs.google.com/blackjack/BlackjackEngine.html#playerHit--)() Retrieves a card from the deck and assigns the card to the player. |
| * + void | * + [playerStand](http://docs.google.com/blackjack/BlackjackEngine.html#playerStand--)() Flips the dealer's card that is currently face down and assigns cards to the dealer as long as the dealer doesn't bust and the cards have a value less than 16. |
| * + void | * + [setAccountAmount](http://docs.google.com/blackjack/BlackjackEngine.html#setAccountAmount-int-)(int amount) Updates the player's account with the parameter value. |
| * + void | * + [setBetAmount](http://docs.google.com/blackjack/BlackjackEngine.html#setBetAmount-int-)(int amount) Updates the bet amount to the provided value |

### Field Detail

#### DRAW static final int DRAWSee Also: [Constant Field Values](http://docs.google.com/constant-values.html#blackjack.BlackjackEngine.DRAW)

#### LESS\_THAN\_21 static final int LESS\_THAN\_21See Also: [Constant Field Values](http://docs.google.com/constant-values.html#blackjack.BlackjackEngine.LESS_THAN_21)

#### BUST static final int BUSTSee Also: [Constant Field Values](http://docs.google.com/constant-values.html#blackjack.BlackjackEngine.BUST)

#### BLACKJACK static final int BLACKJACKSee Also: [Constant Field Values](http://docs.google.com/constant-values.html#blackjack.BlackjackEngine.BLACKJACK)

#### HAS\_21 static final int HAS\_21See Also: [Constant Field Values](http://docs.google.com/constant-values.html#blackjack.BlackjackEngine.HAS_21)

#### DEALER\_WON static final int DEALER\_WONSee Also: [Constant Field Values](http://docs.google.com/constant-values.html#blackjack.BlackjackEngine.DEALER_WON)

#### PLAYER\_WON static final int PLAYER\_WONSee Also: [Constant Field Values](http://docs.google.com/constant-values.html#blackjack.BlackjackEngine.PLAYER_WON)

#### GAME\_IN\_PROGRESS static final int GAME\_IN\_PROGRESSSee Also: [Constant Field Values](http://docs.google.com/constant-values.html#blackjack.BlackjackEngine.GAME_IN_PROGRESS)

### Method Detail

#### getNumberOfDecks int getNumberOfDecks() Returns the number of decks being used.Returns: number of decks

#### createAndShuffleGameDeck void createAndShuffleGameDeck() Creates and shuffles the card deck(s) using a random number generator.

#### getGameDeck [Card](http://docs.google.com/blackjack/Card.html)[] getGameDeck() Returns the current deck of cards.Returns: Card array representing deck of cards.

#### deal void deal() Creates a new deck of cards, and assigns cards to the dealer and player. A total of four cards are dealt in the following order: Player (face up), Dealer (face down), Player (face up), Dealer (face up). Once the cards have been dealt, the game's status will be GAME\_IN\_PROGRESS. Delete the bet amount from the account.

#### getDealerCards [Card](http://docs.google.com/blackjack/Card.html)[] getDealerCards() Returns dealer's cards.Returns: Card array representing the dealer's cards.

#### getDealerCardsTotal int[] getDealerCardsTotal() Returns an array representing the possible value(s) associated with the dealer's cards if the cards represent a value less than or equal to 21.Returns: Integer array representing the possible value(s) or null if cards represent a value higher than 21. The array will have a size of 1 if only one value is associated with the set of cards, and a size of two if two values are possible. For the case of an array of size two, the smaller value must appear in the first array entry.

#### getDealerCardsEvaluation int getDealerCardsEvaluation() Returns an integer value that can assume the values LESS\_THAN\_21 if the dealer's cards have a value less than 21, BUST if the dealer's cards have a value greater than 21, and BLACKJACK if the dealer has an Ace along with a "10", Jack, Queen, or King. If the dealer's cards have a value equivalent to 21 and the hand does not correspond to a blackjack, HAS\_21 will be returned.Returns: Integer value that corresponds to one of the following: LESS\_THAN\_21, BUST, BLACKJACK, HAS\_21

#### getPlayerCards [Card](http://docs.google.com/blackjack/Card.html)[] getPlayerCards() Returns player's cards.Returns: Card array representing the player's cards.

#### getPlayerCardsTotal int[] getPlayerCardsTotal() Returns an array representing the possible value(s) associated with the player's cards if the cards represent a value less than or equal to 21.Returns: integer array representing the possible value(s) or null if cards represent a value higher than 21. The array will have a size of 1 if only one value is associated with the set of cards, and a size of two if two values are possible. For the case of an array of size two, the smaller value must appear in the first array entry.

#### getPlayerCardsEvaluation int getPlayerCardsEvaluation() Returns an integer value that can assume the values LESS\_THAN\_21 if the player's cards have a value less than 21, BUST if the players's cards have a value greater than 21, and BLACKJACK if the player has an Ace along with a "10", Jack, Queen, or King. If the players' cards have a value equivalent to 21 and the hand does not correspond to a blackjack, HAS\_21 will be returned.Returns: Integer value that corresponds to one of the following: LESS\_THAN\_21, BUST, BLACKJACK, HAS\_21

#### playerHit void playerHit() Retrieves a card from the deck and assigns the card to the player. The new sets of cards will be evaluated. If the player busts, the game is over and the games's status will be updated to DEALER\_WON. Otherwise the game's status is GAME\_IN\_PROGRESS.

#### playerStand void playerStand() Flips the dealer's card that is currently face down and assigns cards to the dealer as long as the dealer doesn't bust and the cards have a value less than 16. Once the dealer has a hand with a value greater than or equal to 16, and less than or equal to 21, the hand will be compared against the player's hand and whoever has the hand with a highest value will win the game. If both have the same value we have a draw. The game's status will be updated to one of the following values: DEALER\_WON, PLAYER\_WON, or DRAW. The player's account will be updated with a value corresponding to twice the bet amount if the player wins. If there is a draw the player's account will be updated with the only the bet amount.

#### setBetAmount void setBetAmount(int amount) Updates the bet amount to the provided value

#### getBetAmount int getBetAmount() Returns an integer representing the bet amount.Returns: bet amount.

#### setAccountAmount void setAccountAmount(int amount) Updates the player's account with the parameter value.Parameters: amount -

#### getAccountAmount int getAccountAmount() Returns the player's account amountReturns: account amount

#### getGameStatus int getGameStatus() Returns an integer representing the game status.Returns: DRAW, PLAYER\_WON, DEALER\_WON OR GAME\_IN\_PROGRESS

[Skip navigation links](#ihv636)

* [Overview](http://docs.google.com/overview-summary.html)
* [Package](http://docs.google.com/package-summary.html)
* Class
* [Use](http://docs.google.com/class-use/BlackjackEngine.html)
* [Tree](http://docs.google.com/package-tree.html)
* [Deprecated](http://docs.google.com/deprecated-list.html)
* [Index](http://docs.google.com/index-files/index-1.html)
* [Help](http://docs.google.com/help-doc.html)
* [Prev Class](http://docs.google.com/blackjack/Blackjack.html)
* [Next Class](http://docs.google.com/blackjack/Card.html)
* [Frames](http://docs.google.com/index.html?blackjack/BlackjackEngine.html)
* [No Frames](http://docs.google.com/BlackjackEngine.html)
* [All Classes](http://docs.google.com/allclasses-noframe.html)
* Summary:
* Nested |
* [Field](#3znysh7) |
* Constr |
* [Method](#2et92p0)
* Detail:
* [Field](#tyjcwt) |
* Constr |
* [Method](#35nkun2)

[Web Accessibility](https://www.umd.edu/web-accessibility)