# Class BlackjackGame

java.lang.Object
    BlackjackGame

---

```
public class BlackjackGame
extends java.lang.Object
```

This is the a game class for blackjack. It uses the edu.gvsu.* package. It is only used by the GUI class to run the program. One player will be able to play against the dealer in blackjack.

**Since:**

2016-10-18

**Version:**

1.0

**Author:**

Nathan Lindenbaum

## *Field Summary*

### Fields

| Modifier and Type | Field and Description |
|---|---|
| private static int | **ACE**<br>The Constant ACE. |
| private static int | **ACE_DIFF**<br>The Constant ACE_DIFF. |
| private int | **bet**<br>The bet. |
| private static java.lang.String | **BET_MESS**<br>The Constant BET_MESS. |
| private static int | **BLACKJACK**<br>The Constant BLACKJACK. |
| private edu.gvsu.GVcard | **c**<br>The c. |
| private static int | **CARD_INDENT**<br>The Constant CARD_INDENT. |

| | |
|---|---|
| private int | **creditBalance**<br>The bet. |
| private static int | **CREDITS**<br>The Constant CREDITS. |
| private static int | **DEALER_MIN_HAND**<br>The Constant DEALER_MIN_HAND. |
| private edu.gvsu.GVpile | **dealerCards**<br>The dealer temp cards. |
| private int | **dealerCount**<br>The player count. |
| private edu.gvsu.GVpile | **dealerTempCards**<br>The dealer temp cards. |
| private edu.gvsu.GVpile | **deck**<br>The player cards. |
| private static int | **FACECARD**<br>The Constant FACECARD. |
| private static int | **JACK**<br>The Constant JACK. |
| private static int | **KING**<br>The Constant KING. |
| private static int | **MAX_BET**<br>The Constant MAX_BET. |
| private java.lang.String | **message**<br>The message. |
| private static int | **MIN_BET**<br>The Constant MIN_BET. |
| private static int | **MIN_DECK_CARDS**<br>The Constant MIN_DECK_CARDS. |
| private static int | **NEWACE**<br>The Constant NEWACE. |
| private static int | **NOT_BLACKJACK**<br>The Constant NOT_BLACKJACK. |
| private static java.lang.String | **NOT_ENOUGH**<br>The Constant NOT_ENOUGH. |
| private int | **numberOfAcesd**<br>The number of acesd. |

| | |
|---|---|
| private int | **numberOfAcesp**<br>The number of acesd. |
| private edu.gvsu.GVpile | **playerCards**<br>The player cards. |
| private int | **playerCount**<br>The player count. |
| private edu.gvsu.GVpile | **playerTempCards**<br>The dealer temp cards. |
| private static int | **QUEEN**<br>The Constant QUEEN. |
| private static java.lang.String | **TABLE_MAX**<br>The Constant TABLE_MAX. |
| private static java.lang.String | **TABLE_MIN**<br>The Constant TABLE_MIN. |
| private static java.lang.String | **VALID_INT**<br>The Constant VALID_INT. |

## *Constructor Summary*

### Constructors

| Constructor and Description |
|---|
| **BlackjackGame**()<br>Instantiates a new blackjack game. |

## *Method Summary*

### All Methods Instance Methods Concrete Methods

| Modifier and Type | Method and Description |
|---|---|
| private void | **adjustHandValueAce**(java.lang.String hand)<br>This method will be called after each new card is dealt. |
| private boolean | **checkWinner**()<br>This method will check to see if the player has a winning hand. |
| private void | **clearTable**()<br>This method uses the GVpile command of .pop(). |
| private void | **createHandPiles**()<br>This method will update all the piles that are used in the game to be empty. |

| | |
|---|---|
| void | **deal**()<br>This method will deal the initial 4 cards. |
| int | **dealerCountTotal**()<br>This method will return the present value of the dealers cards. |
| void | **dealerDraw**()<br>This method will pop a card from the deck. |
| void | **doubleDown**()<br>This method is called by the GUI when a player wants to double their bet and only draw one card. |
| int | **getCreditBalance**()<br>This method will return the present amount of credits. |
| edu.gvsu.GVpile | **getDealerCards**()<br>This method will return the GVpile of dealer cards. |
| java.lang.String | **getMessage**()<br>This method is a getter method for an updating message. |
| edu.gvsu.GVpile | **getPlayerCards**()<br>This method will return the GVpile of the players cards. |
| private void | **newShoe**()<br>This method will update the deck (1 single deck). |
| void | **placeBet**()<br>This method will update the players bet. |
| int | **playerCountTotal**()<br>This method will return the present value of the players cards. |
| void | **playerDraw**()<br>This method will pop a card from the deck. |
| private int | **realCardValue**(edu.gvsu.GVcard cardCheck, java.lang.String hand)<br>This method will take two parameters. |
| void | **stand**()<br>First, this method will flip the first dealer card by adding it to a new pile and then popping it off. |

## Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

## *Field Detail*

### FACECARD

```
private static final int FACECARD
```

The Constant FACECARD.

**See Also:**

Constant Field Values

### JACK

```
private static final int JACK
```

The Constant JACK.

**See Also:**

Constant Field Values

### QUEEN

```
private static final int QUEEN
```

The Constant QUEEN.

**See Also:**

Constant Field Values

### KING

```
private static final int KING
```

The Constant KING.

**See Also:**

Constant Field Values

### ACE

```
private static final int ACE
```

The Constant ACE.

**See Also:**

Constant Field Values

### NEWACE

```
private static final int NEWACE
```

The Constant NEWACE.

**See Also:**

Constant Field Values

### ACE_DIFF

```
private static final int ACE_DIFF
```

The Constant ACE_DIFF.

**See Also:**

Constant Field Values

### MIN_DECK_CARDS

```
private static final int MIN_DECK_CARDS
```

The Constant MIN_DECK_CARDS.

**See Also:**

Constant Field Values

### DEALER_MIN_HAND

```
private static final int DEALER_MIN_HAND
```

The Constant DEALER_MIN_HAND.

**See Also:**

Constant Field Values

### CARD_INDENT

```
private static final int CARD_INDENT
```

The Constant CARD_INDENT.

**See Also:**

Constant Field Values

### BLACKJACK

```
private static final int BLACKJACK
```

The Constant BLACKJACK.

**See Also:**

Constant Field Values

### NOT_BLACKJACK

```
private static final int NOT_BLACKJACK
```

The Constant NOT_BLACKJACK.

**See Also:**

Constant Field Values

### MAX_BET

```
private static final int MAX_BET
```

The Constant MAX_BET.

**See Also:**

Constant Field Values

### MIN_BET

```
private static final int MIN_BET
```

The Constant MIN_BET.

**See Also:**

Constant Field Values

### CREDITS

```
private static final int CREDITS
```

The Constant CREDITS.

**See Also:**

Constant Field Values

### dealerCount

```
private int dealerCount
```

The player count.

### playerCount

```
private int playerCount
```

The player count.

### creditBalance

```
private int creditBalance
```

The bet.

### bet

```
private int bet
```

The bet.

### numberOfAcesp

```
private int numberOfAcesp
```

The number of acesd.

### numberOfAcesd

```
private int numberOfAcesd
```

The number of acesd.

### c

```
private edu.gvsu.GVcard c
```

The c.

### deck

```
private edu.gvsu.GVpile deck
```

The player cards.

### playerCards

```
private edu.gvsu.GVpile playerCards
```

The player cards.

### playerTempCards

```
private edu.gvsu.GVpile playerTempCards
```

The dealer temp cards.

### dealerCards

```
private edu.gvsu.GVpile dealerCards
```

The dealer temp cards.

### dealerTempCards

```
private edu.gvsu.GVpile dealerTempCards
```

The dealer temp cards.

### message

```
private java.lang.String message
```

The message.

### NOT_ENOUGH

```
private static final java.lang.String NOT_ENOUGH
```

The Constant NOT_ENOUGH.

**See Also:**

Constant Field Values

### TABLE_MIN

```
private static final java.lang.String TABLE_MIN
```

The Constant TABLE_MIN.

**See Also:**

Constant Field Values

### TABLE_MAX

```
private static final java.lang.String TABLE_MAX
```

The Constant TABLE_MAX.

**See Also:**

Constant Field Values

#### VALID_INT

```
private static final java.lang.String VALID_INT
```

The Constant VALID_INT.

**See Also:**

Constant Field Values

#### BET_MESS

```
private static final java.lang.String BET_MESS
```

The Constant BET_MESS.

**See Also:**

Constant Field Values

### *Constructor Detail*

#### BlackjackGame

```
public BlackjackGame()
```

Instantiates a new blackjack game.

### *Method Detail*

#### getPlayerCards

```
public final edu.gvsu.GVpile getPlayerCards()
```

This method will return the GVpile of the players cards. This pile is painted in the GUI.

**Returns:**

playerCards

#### getDealerCards

```
public final edu.gvsu.GVpile getDealerCards()
```

This method will return the GVpile of dealer cards. This pile is painted in the GUI.

**Returns:**

`dealerCards`

### createHandPiles

```
private void createHandPiles()
```

This method will update all the piles that are used in the game to be empty.

### getMessage

```
public final java.lang.String getMessage()
```

This method is a getter method for an updating message.

**Returns:**

`message`

### newShoe

```
private void newShoe()
```

This method will update the deck (1 single deck). Then it will shuffle the deck and remove the first card. Uses GVcard and GVpile.

### clearTable

```
private void clearTable()
```

This method uses the GVpile command of .pop(). It will clear out all the values that are currently in each of the piles.

### doubleDown

```
public final void doubleDown()
```

This method is called by the GUI when a player wants to double their bet and only draw one card. It will update the current bet to double what it was originally.

### deal

```
public final void deal()
```

This method will deal the initial 4 cards. 2 to the player and 2 to the dealer. It will then check to see if the player has a Blackjack.

### playerDraw

```
public final void playerDraw()
```

This method will pop a card from the deck. Then it will update the player count and check to see if the player busted.

### dealerDraw

```
public final void dealerDraw()
```

This method will pop a card from the deck. Then it will update the dealer count and

### stand

```
public final void stand()
```

First, this method will flip the first dealer card by adding it to a new pile and then popping it off. Then, it will continue to call `dealerDraw()` until the dealer count is greater than 17. Last, it will check if player is a winner `checkWinner()`

### dealerCountTotal

```
public final int dealerCountTotal()
```

This method will return the present value of the dealers cards.

**Returns:**

`dealerCount`

### playerCountTotal

```
public final int playerCountTotal()
```

This method will return the present value of the players cards.

**Returns:**

`playerCount`

### placeBet

```
public final void placeBet()
```

This method will update the players bet. It will open a JOptionPane and ask the player to input an integer greater than 0. It will check to make sure it is a valid choice, then update the bet.

### getCreditBalance

```
public final int getCreditBalance()
```

This method will return the present amount of credits.

**Returns:**

`creditBalance`

### realCardValue

```
private int realCardValue(edu.gvsu.GVcard cardCheck,
                          java.lang.String hand)
```

This method will take two parameters. cardCheck is the GVcard the needs to have the correct value associated to it. hand is just a String to indicate which player's ace count will be updated. the local variable x is the cards value and will be updated if it is a face card.

**Parameters:**

`cardCheck - GVcard associated with player card`

`hand - String to indicate which player's ace count is updated`

**Returns:**

`x`

### checkWinner

```
private boolean checkWinner()
```

This method will check to see if the player has a winning hand.

**Returns:**

`playerBetterScore || dealerBustScore || playerBJ`

### adjustHandValueAce

```
private void adjustHandValueAce(java.lang.String hand)
```

This method will be called after each new card is dealt. It will check to see if the player or dealer has an ace. Then it will update the playerCount or dealerCount accordingly.

**Parameters:**

`hand - See if player has ace`