Package application

Class Summary

Card

Card

CardTest

CardTest

Deck

Deck

DeckTest

DeckTest

Game

Game

Hand

Hand

HandTest

Hand Test

Main

Main

Rank

Rank

Suit

Suit

---

application

Class Card

```
java.lang.Object
    |
    +--application.Card
```

< Fields > < Constructors > < Methods >

---

public class **Card**
extends java.lang.Object

Card

This is the card class. Each Deck is comprised of Card objects. Card uses the Enum Classes Suit and Rank. It's color is used described by a binary integer (color).

Author:
Derrik Fleming

Author:
Josh T

Author:
Jarrett Swales

Version:
1.0

2016 Fall

Fields

ACE_NO_TRUMP

`public static final int` **`ACE_NO_TRUMP`**
A round effectiveness score for non-trump ace.

ACE_TRUMP

`public static final int` **`ACE_TRUMP`**
A round effectiveness score for trump ace.

JACK_L_BOWER

`public static final int` **`JACK_L_BOWER`**
A round effectiveness score for the left bower.

JACK_NO_TRUMP

`public static final int` **`JACK_NO_TRUMP`**
A round effectiveness score for non-trump jack.

JACK_R_BOWER

`public static final int` **`JACK_R_BOWER`**
A round effectiveness score for the right bower.

KING_NO_TRUMP

`public static final int` **`KING_NO_TRUMP`**
A round effectiveness score for non-trump king.

KING_TRUMP

`public static final int` **`KING_TRUMP`**
A round effectiveness score for trump king.

NINE_NO_TRUMP

`public static final int` **`NINE_NO_TRUMP`**
A round effectiveness score for non-trump nine.

NINE_TRUMP

```
public static final int NINE_TRUMP
```
A round effectiveness score for trump nine.

QUEEN_NO_TRUMP

```
public static final int QUEEN_NO_TRUMP
```
A round effectiveness score for non-trump queen.

QUEEN_TRUMP

```
public static final int QUEEN_TRUMP
```
A round effectiveness score for trump queen.

TEN_NO_TRUMP

```
public static final int TEN_NO_TRUMP
```
A round effectiveness score for non-trump ten.

TEN_TRUMP

```
public static final int TEN_TRUMP
```
A round effectiveness score for trump ten.

Constructors

Card

```
public  Card(Suit cardSuit,
             Rank cardRank,
             boolean cardTrump)
```

This is the constructor for all Card class objects, the parameters suit, rank, trump are used to set the private.

Parameters:

cardSuit - is this Cards (Enum) Suit.
cardRank - is this Cards (Enum) Rank.
cardTrump - is this Card's trump value, true for trump.

Methods

getCardValue

`public final int` **`getCardValue`**`()`

This method returns a "effectiveness value" of the card. The "effectiveness" is a metric for the Card's value in the current round, generally for AI purposes.

Returns:

int The card's estimated value in the current round.

---

getColor

`public final int` **`getColor`**`()`

This method returns an integer (binary), that represents this Card's color (0 is black, 1 is red). Used to identify the rounds Left Bower.

Returns:

int Is 0 when Suit color is black, 1 when Suit color is red.

---

getFaceImage

`public final javafx.scene.image.Image` **`getFaceImage`**`()`

This method returns the Image faceImage of this Card.

Returns:

Image this card's faceImage, found in view/images/[RANK][SUIT].jpg.

---

getRank

`public final` Rank **`getRank`**`()`

This method returns this Card's rank.

Returns:

rank this Card's rank.

---

getSuit

`public final` Suit **`getSuit`**`()`

This method returns the field for this Card's Suit.

Returns:

Suit Is this Card's Suit.

getTrump

`public final boolean` **`getTrump`**`()`

This method returns true if this Card's suit is trump, false if not.

Returns:

trump is true when this Card's suit is trump.

setTrump

`public final void` **`setTrump`**`(boolean suitTrump)`

This method sets the trump field for this Card.

Parameters:

suitTrump - Is true, when this Card's suit is trump.

toString

`public final java.lang.String` **`toString`**`()`

Returns:

String

Overrides:

toString in class java.lang.Object

---

application

Class CardTest

```
java.lang.Object
    |
    +--application.CardTest
```

< Fields > < Constructors > < Methods >

public class **CardTest**
extends java.lang.Object

CardTest

This class tests the Card class.

Author:
Derrik Fleming

Author:
Josh T

Author:
Jarrett Swales

Version:
1.0

2016 Fall

Fields

ACE_NO_TRUMP

`public static final int` **`ACE_NO_TRUMP`**
A round effectiveness score for non-trump ace.

ACE_TRUMP

`public static final int` **`ACE_TRUMP`**
A round effectiveness score for trump ace.

JACK_L_BOWER

`public static final int` **`JACK_L_BOWER`**
A round effectiveness score for the left bower.

JACK_NO_TRUMP

`public static final int` **`JACK_NO_TRUMP`**
A round effectiveness score for non-trump jack.

JACK_R_BOWER

`public static final int` **`JACK_R_BOWER`**
A round effectiveness score for the right bower.

KING_NO_TRUMP

`public static final int` **`KING_NO_TRUMP`**
A round effectiveness score for non-trump king.

KING_TRUMP

`public static final int` **`KING_TRUMP`**
A round effectiveness score for trump king.

NINE_NO_TRUMP

`public static final int` **`NINE_NO_TRUMP`**
A round effectiveness score for non-trump nine.

NINE_TRUMP

`public static final int` **`NINE_TRUMP`**
A round effectiveness score for trump nine.

QUEEN_NO_TRUMP

`public static final int` **`QUEEN_NO_TRUMP`**
A round effectiveness score for non-trump queen.

QUEEN_TRUMP

`public static final int` **`QUEEN_TRUMP`**
A round effectiveness score for trump queen.

TEN_NO_TRUMP

`public static final int` **`TEN_NO_TRUMP`**
A round effectiveness score for non-trump ten.

TEN_TRUMP

`public static final int` **`TEN_TRUMP`**
A round effectiveness score for trump ten.

Constructors

CardTest

```
public  CardTest()
```

Methods

testColorGetter

```
public final void testColorGetter()
```

This is a test for the color getter method. instantiates a Card for each possible Suit, and then checks that the Suit color of each Card matches.

---

testRankGetter

```
public final void testRankGetter()
```

This is a test for the Rank getter method. instantiates a Card for each possible Rank, and then checks that the Rank of each Card matches.

---

testSetTrump

```
public final void testSetTrump()
```

This is a test for the tremp setter method. instantiates 4 Card's for two different suits, sets trump to true/false for each suit and then checks that the trump value of each card matches.

---

testSuitGetter

```
public final void testSuitGetter()
```

This is a test for the Suit getter method. instantiates a card for each possible Suit, and then checks that the suit of each card matches.

---

testToString

```
public final void testToString()
```

This method tests the toString method of the Card class. It instantiates 4 Cards and then checks that the string of each is correct.

---

testValueGetter

`public final void` **`testValueGetter`**`()`

This is a test for the value getter method. instantiates a Card for a variety of values, and then checks that the value of each card matches.

---

application

Class Deck

```
java.lang.Object
    |
    +--application.Deck
```

< Constructors > < Methods >

---

public class **Deck**
extends java.lang.Object

Deck

The is the Deck class, it is used to hold the collection of cards used during game play. The cards are stored in an ArrayList, theDeck.

Author:
Derrik Fleming

Author:
Josh T

Author:
Jarrett Swales

Version:
1.0

2016 Fall

Constructors

Deck

`public` **`Deck`**`()`

Constructor, replenishes the deck and shuffles on instantiation.

Methods

getCardAt

`public final Card getCardAt(int index)`

This method is essentially a getter for a Card at a specified index.

Parameters:

index - is the index of the Deck.

Returns:

Card at index.

---

getCurrentSize

`public final int getCurrentSize()`

Getter for the current size of the Deck.

Returns:

int the size of the Deck.

---

getTrump

`public final Suit getTrump()`

Getter for trump.

Returns:

Suit is the set trump suit.

---

getTrumpColor

`public static int getTrumpColor()`

Getter for trump color.

Returns:

int 0=black 1=red.

---

removeCardAt

```
public final void removeCardAt(int index)
```

This method removes a Card from the Deck.

Parameters:

index - index of the Card to be removed.

---

replenishDeck

```
public final void replenishDeck()
```

This method replenishes the deck.

---

setTrump

```
public final void setTrump(Suit t)
```

Setter for trump.

Parameters:

t - is the parameter.

---

shuffle

```
public final void shuffle()
```

This method shuffles the Deck.

---

application

Class DeckTest

```
java.lang.Object
    |
    +--application.DeckTest
```

< Constructors > < Methods >

---

public class **DeckTest**
extends java.lang.Object

DeckTest

This class tests the Deck class.

Author:
Derrik Fleming

Author:
Josh T

Author:
Jarrett Swales

Version:
1.0

2016 Fall

Constructors

DeckTest

`public` **`DeckTest`**`()`

Methods

testGetAndRemoveCardAt

`public final void` **`testGetAndRemoveCardAt`**`()`

Tests the get and remove card at methods.

---

testGetCurrentSize

`public final void` **`testGetCurrentSize`**`()`

Test the get current size methods.

---

testGetTrumpColor

`public final void` **`testGetTrumpColor`**`()`

Tests the getter for trump color.

---

testReplenishAndShuffleDeck

`public final void` **`testReplenishAndShuffleDeck`**`()`

Tests the replenish and shuffle deck methods.

---

testSetAndGetTrump

`public final void` **`testSetAndGetTrump`**`()`

Tests the setters and getters for trump.

---

application

Class Game

```
java.lang.Object
    |
    +--application.Game
```

< Constructors > < Methods >

---

public class **Game**
extends java.lang.Object

Game

This is the class where all the action happens most of the hard game logic is within this class.

Author:
Derrik Fleming

Author:
Josh T

Author:
Jarrett Swales

Version:
1.0

2016 Fall

Constructors

Game

`public` **`Game`**`()`

This is the Game default constructor for the Game class, it sets the attributes for the Game class for a new Game.

Methods

clearTable

`public final void` **`clearTable`**`()`

This method clears the Array cards, which holds the played Cards for the round.

---

getGameScore

`public final int` **`getGameScore`**`()`

Getter for gameScore.

Returns:

the game's score.

---

getPlayedCard

`public final` Card **`getPlayedCard`**`(int index)`

This method returns a Card at the passed value, index.

Parameters:

index - the index of the array cards, (the played cards).

Returns:

the Card at the passed index.

---

getPlayerHand

`public final` Hand **`getPlayerHand`**`(int playerIndex)`

Getter for the Hand of the the player at the passed index value.

Parameters:

playerIndex - the index of the player whose Hand to get.

Returns:

Hand is the players Hand at playerIndex.

---

getScore1

`public final int` **`getScore1`**`()`

Getter for team1's score.

Returns:

the score of team1.

---

getScore2

`public final int` **`getScore2`**`()`

Getter for team2's score.

Returns:

the score of team2.

---

getTrump

`public final` `Suit` **`getTrump`**`()`

This method returns the Suit of the trump suit.

Returns:

Suit is the trump suit.

---

getTurn

`public final int` **`getTurn`**`()`

Getter for the turn counter, turnCounter.

Returns:

the index of the player whose turn it is.

---

getUpCard

`public final` `Card` **`getUpCard`**`()`

Getter for upCard, the face up card.

Returns:

the face up Card.

---

nextTurn

```
public final void nextTurn()
```

This sets the turn to the next player.

---

orderUpAI

```
public final void orderUpAI(Hand hand,
                            Card faceUpCard)
```

This method affords an AI player the ability to call trump during the trump call rounds.

Parameters:

hand - is the Hand of the AI whose turn is is to call trump.
faceUpCard - is the Card that is turned face up.

---

playCard

```
public final void playCard(Hand crnt,
                           int index)
```

This method adds the selected card to the Array cards and removes it from the player's Hand, and then calls nextTurn() to set the turn to the next player.

Parameters:

crnt - is the current player's Hand.
index - is the index of the selected card.

---

playCardAI

```
public final void playCardAI(int index)
```

This plays a card from the AI players Hand.

Parameters:

index - the index of the AIs Card to be played.

---

removeUpCard

```
public final void removeUpCard()
```

This method sets upCard to null, in effect removing it.

---

roundComplete

`public final boolean roundComplete()`

This method returns true when the round is complete and false otherwise.

Returns:

true when the round is complete.

---

roundTaker

`public final int roundTaker()`

This method compares each player's played card, and determines who won the hand.

Returns:

the index of the highest valued played card.

---

scoreRound

`public final void scoreRound()`

This method adds points to the round winning team's game score.

---

selectTrumpAI

`public final void selectTrumpAI()`

Shell for R2 method.

---

selectTrumpAI

`public final Suit selectTrumpAI(Hand hand)`

This method helps the AI determine how to select trump.

Parameters:

hand - is the Hand of the AI player.

Returns:

Suit of the selected trump suit.

---

setTrump

```
public final void setTrump(Suit s)
```

This method sets any Card of the trump suit in the players Hand to True.

Parameters:

s - is the Suit to be trump.

---

setTurn

```
public final void setTurn(int turn)
```

Setter for the current players turn.

Parameters:

turn - the index of the player whose turn it is.

---

updateScore

```
public final void updateScore()
```

This method updates the score depending on who took the round.

---

application

Class Hand

```
java.lang.Object
    |
    +--application.Hand
```

< Constructors > < Methods >

---

public class **Hand**
extends java.lang.Object

Hand

This class contains the attributes of a Euchre player's hand.

Author:

Derrik Fleming

Author:

Josh T

Author:

Jarrett Swales

Version:

1.0

2016 Fall

Constructors

Hand

`public` **`Hand`**`(`[`Deck`]`crnt)`

Default constructor for class, replenishes hand, called after each round of play.

Parameters:

crnt - is the parameter.

Methods

get

`public final` [`Card`] **`get`**`(int index)`

Gets the Card at the parameter index.

Parameters:

index - is the parameter.

Returns:

Card

---

getHandValue

`public final int` **`getHandValue`**`()`

Returns the combined value of all cards in this Hand.

Returns:

int This is the estimated "effectiveness" value for this hand.

---

getHighCard

`public final int` **`getHighCard`**`()`

Returns the value of the high Card in this Hand.

Returns:

int value per the Card's value.

---

getLowCard

`public final int` **`getLowCard`**`()`

Returns the value of the lowest Card in this Hand.

Returns:

int value per the Card's value.

---

getSize

`public final int` **`getSize`**`()`

This is a getter for the size of the ArrayList theHand.

Returns:

int The hand's size.

---

removeCardAt

`public final void` **`removeCardAt`**`(int index)`

This method removes the card at index in theHand.

Parameters:

index - is the Card's index to remove.

---

replenishHand

`public final void` **`replenishHand`**`(Deck crnt)`

Adds 5 new Cards from the Deck crnt to the this Hand.

Parameters:

crnt - Is the the Deck that Card's are being taken from.

---

set

```
public final void set(int index,
                      Card c)
```

This method adds Card's to the ArrayList theHand at the passed index value.

Parameters:

index - Index for the ArrayList in theHand.
c - is the Card to be added to the ArrayList the Hand.

---

toString

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

Appends a string which describes the players Hand.

Returns:

String handDesc

Overrides:

toString in class java.lang.Object

---

application

Class HandTest

```
java.lang.Object
    |
    +--application.HandTest
```

< Constructors > < Methods >

---

public class **HandTest**
extends java.lang.Object

Hand Test

This class tests the Hand class.

Author:

Derrik Fleming

Author:

Josh T

Author:

Jarrett Swales

Version:

1.0

2016 Fall

Constructors

HandTest

`public` **`HandTest`**`()`

Methods

testGetAndSetCardAt

`public final void` **`testGetAndSetCardAt`**`()`

This method tests the set/get Card at method in the Hand class. It instantiates a Deck, Card, Hand sets a Card in the Hand, and verifies it at the specified index.

---

testGetHandValue

`public final void` **`testGetHandValue`**`()`

This method tests the getHandValue method in Hand. It instantiates a deck, and a hand, and verifies the Hand's value.

---

testGetHighCard

`public final void` **`testGetHighCard`**`()`

This method tests the getHighCard method in Hand. It instantiates a Deck, and a Hand, and it fills the hand, then attempts to get the Hand's high Card

---

testGetSize

`public final void` **`testGetSize`**`()`

This method tests the getSize method of the Hand class. It instantiates a Deck, and a Hand and checks the size of the Hand.

---

testRemoveCardAt

```
public final void testRemoveCardAt()
```

This method tests the removeCardAt method of the Hand class. It instantiates a Deck, a Hand, and removes a card from the Hand.

---

testToString

```
public final void testToString()
```

This method tests the toString method of the Hand class. It instantiates a Deck, and a Hand. It manually sets Cards to each available index in the Hand. and then compares the string description.

---

application

Class Main

```
java.lang.Object
    |
    +--javafx.application.Application
        |
        +--application.Main
```

< Constructors > < Methods >

---

public class **Main**
extends javafx.application.Application

Main

The main class's only purpose, as an extension of Application, is essentially is to load the JavaFX UI, and set the "Game Play Scene".

Author:
Derrik Fleming

Author:
Josh T

Author:
Jarrett Swales

Version:
1.0

2016 Fall

Constructors

Main

```
public  Main()
```

Methods

main

```
public static void main(java.lang.String[] args)
```

Standard method call to start JavaFX UI.

Parameters:

args - Default.

---

showSceneView

```
public final void showSceneView()
                  throws java.io.IOException
```

This method sets the stage to the main scene (ie really just the background).

Throws:

java.io.IOException - If loading game play scene fails.

---

start

```
public final void start(javafx.stage.Stage s)
```

Overrides:

start in class javafx.application.Application

application

Class Rank

```
java.lang.Object
    |
    +--java.lang.Enum
        |
        +--application.Rank
```

All Implemented Interfaces:
java.io.Serializable, java.lang.Comparable

< Fields > < Methods >

---

public final class **Rank**
extends java.lang.Enum

Rank

An Enum describing the Cards rank.

Author:
Derrik Fleming

Author:
Josh T

Author:
Jarrett Swales

Version:
1.0

2016 Fall

Fields

ACE

`public static final Rank` **`ACE`**
Card Rank, ace.

JACK

`public static final Rank` **`JACK`**
Card Rank jack.

KING

```
public static final Rank KING
```
Card Rank, king.

NINE

```
public static final Rank NINE
```
Card Rank nine.

QUEEN

```
public static final Rank QUEEN
```
Card Rank, queen.

TEN

```
public static final Rank TEN
```
Card Rank ten.

Methods

valueOf

```
public static Rank valueOf(java.lang.String name)
```

values

```
public static application.Rank[] values()
```

application

Class Suit

```
java.lang.Object
    |
    +--java.lang.Enum
        |
        +--application.Suit
```

All Implemented Interfaces:
java.io.Serializable, java.lang.Comparable

< Fields > < Methods >

---

public final class **Suit**
extends java.lang.Enum

Suit

An Enum describing the Card's suit.

Author:
Derrik Fleming

Author:
Josh T

Author:
Jarrett Swales

Version:
1.0

2016 Fall

Fields

CLUBS

`public static final Suit` **`CLUBS`**
Card Suit Clubs.

DIAMONDS

`public static final Suit` **`DIAMONDS`**
Card Suit Diamonds.

HEARTS

```
public static final Suit HEARTS
```
Card Suit Hearts.

SPADES

```
public static final Suit SPADES
```
Card Suit Spades.

Methods

valueOf

```
public static Suit valueOf(java.lang.String name)
```

values

```
public static application.Suit[] values()
```

Package application.view

Class Summary

GameViewController

GamePlayController

---

application.view

Class GameViewController

```
java.lang.Object
    |
    +--application.view.GameViewController
```

All Implemented Interfaces:
javafx.fxml.Initializable

< Constructors > < Methods >

---

public class **GameViewController**
extends java.lang.Object
implements javafx.fxml.Initializable

GamePlayController

The controller class for the GameView.fxml Some gameplay control logic is found in this class, It works in congruence with the instantiated Game object to controll the game play.

Author:
Derrik Fleming

Author:
Josh T

Author:
Jarrett Swales

Version:
1.0

2016 Fall

Constructors

GameViewController

```
public  GameViewController()
```

Methods

buildCardSwapDialog

```
public final void buildCardSwapDialog()
```

---

buildTrumpr1Dialog

```
public final void buildTrumpr1Dialog()
```

---

buildTrumpr2Dialog

```
public final void buildTrumpr2Dialog()
```

---

initialize

```
public final void initialize(java.net.URL location,
                             java.util.ResourceBundle resources)
```

---

nextPlayer

```
public final void nextPlayer()
```

This method is called after the user plays a card, it sets the next turn, or determines the round is over and disables/enables the appropriate buttons.

---

nextRound

```
public final void nextRound(javafx.event.ActionEvent event)
```

This method is called when the next round button is clicked. It starts the next round.

Parameters:

event - An event from the user on the next round button.

---

playThisCard

```
public final void playThisCard(javafx.scene.input.MouseEvent event)
```

This method is called whenever a card in the user's hand is clicked on. It plays the card that the user clicked on.

Parameters:

event - An event from the user on users Cards displayed.

---

refresh

```
public final void refresh()
```

This method is used to refresh the table top to reflect the Game object, after game events occur.

---

startGame

```
public final void startGame(javafx.event.ActionEvent event)
```

This method is called when the start game button is clicked, and starts a new game.

Parameters:

event - An event from the user on the start button.

INDEX

R

S

T

U

V