

My Project

Generated by Doxygen 1.8.13

Contents

1	Project Name Source Code	1
2	Hierarchical Index	3
2.1	Class Hierarchy	3
3	Class Index	5
3.1	Class List	5
4	Class Documentation	7
4.1	Card Class Reference	7
4.1.1	Detailed Description	7
4.1.2	Constructor & Destructor Documentation	7
4.1.2.1	Card()	7
4.1.3	Member Function Documentation	8
4.1.3.1	getRank()	8
4.1.3.2	getSuit()	8
4.1.3.3	points()	8
4.1.3.4	rankString()	9
4.1.3.5	suitSymbol()	9
4.1.3.6	toString()	9
4.1.3.7	toSymbol()	9
4.2	Computer Class Reference	10
4.2.1	Detailed Description	10
4.2.2	Member Function Documentation	10
4.2.2.1	makeMove()	10

4.3	DiscardPile Class Reference	11
4.3.1	Detailed Description	11
4.3.2	Member Function Documentation	12
4.3.2.1	displayTopCard()	12
4.4	GameOps Class Reference	12
4.4.1	Detailed Description	12
4.4.2	Member Function Documentation	12
4.4.2.1	calculateScores()	12
4.4.2.2	createDiscardPile()	13
4.4.2.3	createStockPile()	13
4.4.2.4	distributeCards()	13
4.4.2.5	endGame()	14
4.4.2.6	playAgain()	14
4.4.2.7	processDecision()	14
4.4.2.8	resetEverything()	14
4.4.2.9	username()	15
4.5	GinRummy Class Reference	15
4.5.1	Detailed Description	15
4.5.2	Member Function Documentation	15
4.5.2.1	main()	15
4.5.2.2	playGame()	15
4.6	Hand Class Reference	16
4.6.1	Detailed Description	17
4.6.2	Constructor & Destructor Documentation	17
4.6.2.1	Hand()	17
4.6.3	Member Function Documentation	17
4.6.3.1	contains()	17
4.6.3.2	displayHand()	17
4.6.3.3	remove()	18
4.7	Meld Class Reference	18

4.7.1	Member Function Documentation	18
4.7.1.1	checkMelds()	18
4.8	Player Class Reference	19
4.8.1	Detailed Description	19
4.8.2	Constructor & Destructor Documentation	19
4.8.2.1	Player()	19
4.8.3	Member Function Documentation	20
4.8.3.1	addCardToHand()	20
4.8.3.2	addToTotalScore()	20
4.8.3.3	checkMelds()	20
4.8.3.4	discardFromHand()	20
4.8.3.5	displayHand()	21
4.8.3.6	extractDeadwood()	21
4.8.3.7	getDeadwoodScore()	21
4.8.3.8	getHand()	21
4.8.3.9	getMelds()	22
4.8.3.10	getName()	22
4.8.3.11	getTotalScore()	22
4.8.3.12	recalculateDeadwoodScore()	22
4.8.3.13	resetDeadwoodScore()	22
4.8.3.14	resetHand()	23
4.8.3.15	resetMelds()	23
4.8.3.16	resetTotalScore()	23
4.9	sortByRank Class Reference	23
4.9.1	Member Function Documentation	24
4.9.1.1	compare()	24
4.10	sortBySR Class Reference	24
4.10.1	Member Function Documentation	25
4.10.1.1	compare()	25
4.11	StockPile Class Reference	26

4.11.1 Detailed Description	26
4.11.2 Member Function Documentation	27
4.11.2.1 search()	27
4.12 Suit Enum Reference	27
4.12.1 Detailed Description	27
4.13 UserInputOps Class Reference	27
4.13.1 Detailed Description	28
4.13.2 Member Function Documentation	28
4.13.2.1 chooseDiscard()	28
4.13.2.2 closeScanner()	28
4.13.2.3 knock()	28
4.13.2.4 playAgain()	29
4.13.2.5 playerDecision()	29
4.13.2.6 username()	29
Index	31

Chapter 1

Project Name Source Code

The folders and files for this project are as follows:

...

Chapter 2

Hierarchical Index

2.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

Card	7
Computer	10
GameOps	12
GinRummy	15
Meld	18
Player	19
Suit	27
UserInputOps	27
ArrayList	
Hand	16
Comparator	
sortByRank	23
sortBySR	24
Stack	
DiscardPile	11
StockPile	26

Chapter 3

Class Index

3.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

Card	A class to present Card data type	7
Computer	The computer opponent class to make the computer's move	10
DiscardPile	The discard pile class	11
GameOps	Class of operations for a game of Gin Rummy	12
GinRummy	Main operating class of the program	15
Hand	Hand of cards class	16
Meld	18
Player	Data type to represent a generic player in a Gin-Rummy game	19
sortByRank	23
sortBySR	24
StockPile	The stock pile class	26
Suit	Suit enum.H = Hearts, S = Spades, C = Clubs, D = Diamonds	27
UserInputOps	Class designated for user input and operations	27

Chapter 4

Class Documentation

4.1 Card Class Reference

A class to present `Card` data type.

Public Member Functions

- `Card (Suit s, int r)`
- `Suit getSuit ()`
- `int getRank ()`
- `int points ()`
- `String toString ()`
- `String toSymbol ()`
- `String rankString ()`
- `String suitSymbol ()`

4.1.1 Detailed Description

A class to present `Card` data type.

Author

Smita Singh

4.1.2 Constructor & Destructor Documentation

4.1.2.1 Card()

```
Card.Card (  
    Suit s,  
    int r ) [inline]
```

`Card` constructor

Parameters

<i>s</i>	- Suit type
<i>r</i>	- rank of card (1 to 13)

4.1.3 Member Function Documentation**4.1.3.1 getRank()**

```
int Card.getRank ( ) [inline]
```

Accessor for rank

Returns

This returns the card rank

4.1.3.2 getSuit()

```
Suit Card.getSuit ( ) [inline]
```

Accessor for suit

Returns

This returns the card suit

4.1.3.3 points()

```
int Card.points ( ) [inline]
```

This method checks if the card is lower than 10 if it is lower than 10 it returns the card rank. if it is not lower than 10, it returns the value of 10

Returns

Returns card points

4.1.3.4 rankString()

```
String Card.rankString ( ) [inline]
```

This method will create the string representation of the card rank

Returns

Returns a string representation of the card rank

4.1.3.5 suitSymbol()

```
String Card.suitSymbol ( ) [inline]
```

This method will create the string representation of the card suit symbol

Returns

Returns a string representation of the card suit symbol

4.1.3.6 toString()

```
String Card.toString ( ) [inline]
```

This method turns a card into a string value by calling two private methods that turn the rank and suit into string separately. Then it combines the two string

Returns

Returns a string representation of the card

4.1.3.7 toSymbol()

```
String Card.toSymbol ( ) [inline]
```

This method turns a card into a string value with the suit symbol by calling two private methods that turn the rank and suit into string separately. Then it combines the two string

Returns

Returns a string representation of the card with suit symbol

The documentation for this class was generated from the following file:

- Card.java

4.2 Computer Class Reference

The computer opponent class to make the computer's move.

Static Public Member Functions

- static boolean [makeMove](#) ([Player](#) player, [StockPile](#) stockPile, [DiscardPile](#) discardPile)

4.2.1 Detailed Description

The computer opponent class to make the computer's move.

Author

Joy Xiao

4.2.2 Member Function Documentation

4.2.2.1 makeMove()

```
static boolean Computer.makeMove (  
    Player player,  
    StockPile stockPile,  
    DiscardPile discardPile ) [inline], [static]
```

[Computer](#) opponent makes move

Parameters

<i>player</i>	the computer player
<i>stockPile</i>	the stock pile
<i>discardPile</i>	the discard pile

Returns

true when computer knocks, false when computer draws card

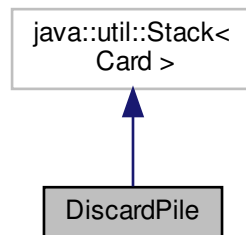
The documentation for this class was generated from the following file:

- Computer.java

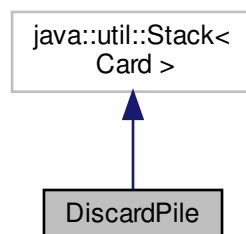
4.3 DiscardPile Class Reference

The discard pile class.

Inheritance diagram for DiscardPile:



Collaboration diagram for DiscardPile:



Public Member Functions

- void [displayTopCard](#) ()

4.3.1 Detailed Description

The discard pile class.

Author

Joy Xiao

4.3.2 Member Function Documentation

4.3.2.1 displayTopCard()

```
void DiscardPile.displayTopCard ( ) [inline]
```

Display the top card of the discard pile

The documentation for this class was generated from the following file:

- DiscardPile.java

4.4 GameOps Class Reference

Class of operations for a game of Gin Rummy.

Static Public Member Functions

- static void [calculateScores](#) ([Player](#) p1, [Player](#) cpu)
- static [StockPile](#) [createStockPile](#) ()
- static [DiscardPile](#) [createDiscardPile](#) ()
- static void [distributeCards](#) ([Player](#) p1, [Player](#) cpu, [StockPile](#) stockPile, [DiscardPile](#) discardPile)
- static void [endGame](#) ()
- static char [playAgain](#) ()
- static boolean [processDecision](#) ([Player](#) p1, [StockPile](#) sp, [DiscardPile](#) dp)
- static void [resetEverything](#) ([Player](#) p)
- static String [username](#) ()

4.4.1 Detailed Description

Class of operations for a game of Gin Rummy.

Author

Benson Hall

4.4.2 Member Function Documentation

4.4.2.1 calculateScores()

```
static void GameOps.calculateScores (
    Player p1,
    Player cpu ) [inline], [static]
```

Calculate score: The absolute value of the difference of the deadwood scores is calculated, and the points are awarded to the player with the lowest deadwood score. If any of the players have 0 deadwood points and knocks, they get an additional 20 points.

Parameters

<i>p1</i>	- user player
<i>cpu</i>	- computer player

4.4.2.2 createDiscardPile()

```
static DiscardPile GameOps.createDiscardPile ( ) [inline], [static]
```

Create discard pile

Returns

new discard pile

4.4.2.3 createStockPile()

```
static StockPile GameOps.createStockPile ( ) [inline], [static]
```

Create initial stock pile and add cards to it

Returns

new stock pile

4.4.2.4 distributeCards()

```
static void GameOps.distributeCards (
    Player p1,
    Player cpu,
    StockPile stockPile,
    DiscardPile discardPile ) [inline], [static]
```

Distribute cards. 10 cards per player, next card on the discard pile

Parameters

<i>p1</i>	- user player
<i>cpu</i>	- computer player
<i>stockPile</i>	- stock pile
<i>discardPile</i>	- discard pile

4.4.2.5 endGame()

```
static void GameOps.endGame ( ) [inline], [static]
```

End the game, close scanners

4.4.2.6 playAgain()

```
static char GameOps.playAgain ( ) [inline], [static]
```

Play again? Ask the user.

Returns

user's input

4.4.2.7 processDecision()

```
static boolean GameOps.processDecision (
    Player p1,
    StockPile sp,
    DiscardPile dp ) [inline], [static]
```

Given the user's choice, process its choice and return true if the user knocks, false otherwise.

Parameters

<i>p1</i>	- user player
<i>sp</i>	- stock pile
<i>dp</i>	- discard pile

Returns

true if user knocks, false otherwise

4.4.2.8 resetEverything()

```
static void GameOps.resetEverything (
    Player p ) [inline], [static]
```

Reset all elements of the former deal

Parameters

<i>p</i>	- player being reset
----------	----------------------

4.4.2.9 username()

```
static String GameOps.username ( ) [inline], [static]
```

Call user operations to ask for a username

Returns

username from user operations

The documentation for this class was generated from the following file:

- GameOps.java

4.5 GinRummy Class Reference

Main operating class of the program.

Static Public Member Functions

- static void [playGame](#) ([Player](#) p1, [Player](#) cpu)
- static void [main](#) (String[] args)

4.5.1 Detailed Description

Main operating class of the program.

Author

Benson

4.5.2 Member Function Documentation

4.5.2.1 main()

```
static void GinRummy.main (
    String [] args ) [inline], [static]
```

Main method

4.5.2.2 playGame()

```
static void GinRummy.playGame (
    Player p1,
    Player cpu ) [inline], [static]
```

Play a game of Gin Rummy

Parameters

<i>p1</i>	- user player
<i>cpu</i>	- computer player

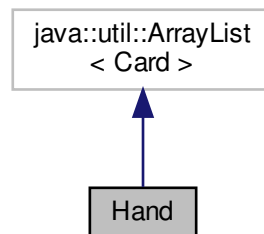
The documentation for this class was generated from the following file:

- GinRummy.java

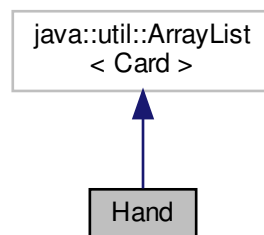
4.6 Hand Class Reference

[Hand](#) of cards class.

Inheritance diagram for Hand:



Collaboration diagram for Hand:

**Public Member Functions**

- [Hand](#) ()
- void [displayHand](#) ()
- boolean [contains](#) (String playerInput)
- [Card remove](#) (String playerInput)

4.6.1 Detailed Description

[Hand](#) of cards class.

Author

Joy Xiao

4.6.2 Constructor & Destructor Documentation

4.6.2.1 Hand()

```
Hand.Hand ( ) [inline]
```

[Hand](#) constructor

4.6.3 Member Function Documentation

4.6.3.1 contains()

```
boolean Hand.contains (
    String playerInput ) [inline]
```

Checks if hand contains the card

Parameters

<i>playerInput</i>	String input of the card to check
--------------------	-----------------------------------

Returns

true when the card is in the hand or else return false

4.6.3.2 displayHand()

```
void Hand.displayHand ( ) [inline]
```

Displays the player's hand to the console

4.6.3.3 remove()

```
Card Hand.remove (
    String playerInput ) [inline]
```

Discards the specific card from the player's hand

Parameters

<i>playerInput</i>	String input of card to discard
--------------------	---------------------------------

Returns

the card that has been removed

The documentation for this class was generated from the following file:

- Hand.java

4.7 Meld Class Reference

Static Public Member Functions

- static ArrayList< ArrayList< Card > > [checkMelds](#) (Hand H)

4.7.1 Member Function Documentation

4.7.1.1 checkMelds()

```
static ArrayList<ArrayList<Card> > Meld.checkMelds (
    Hand H ) [inline], [static]
```

Finds the sequence and group melds of a given hand of cards

Parameters

<i>Hand</i>	H hand of cards of the player
-------------	-------------------------------

Returns

2d arraylist of the the list of melds

The documentation for this class was generated from the following file:

- Meld.java

4.8 Player Class Reference

Data type to represent a generic player in a Gin-Rummy game.

Public Member Functions

- [Player](#) (String name)
- String [getName](#) ()
- [Hand](#) [getHand](#) ()
- int [getTotalScore](#) ()
- int [getDeadwoodScore](#) ()
- ArrayList< ArrayList< [Card](#) > > [getMelds](#) ()
- void [addCardToHand](#) ([Card](#) c)
- [Card](#) [discardFromHand](#) (String input) throws IllegalArgumentException
- void [addToTotalScore](#) (int points)
- ArrayList< [Card](#) > [extractDeadwood](#) ()
- void [recalculateDeadwoodScore](#) ()
- void [checkMelds](#) ()
- void [displayHand](#) ()
- void [resetHand](#) ()
- void [resetDeadwoodScore](#) ()
- void [resetMelds](#) ()
- void [resetTotalScore](#) ()

4.8.1 Detailed Description

Data type to represent a generic player in a Gin-Rummy game.

Author

Benson Hall

4.8.2 Constructor & Destructor Documentation

4.8.2.1 Player()

```
Player.Player (
    String name ) [inline]
```

[Player](#) constructor

Parameters

<i>name</i>	- name of player
-------------	------------------

4.8.3 Member Function Documentation

4.8.3.1 addCardToHand()

```
void Player.addCardToHand (
    Card c ) [inline]
```

Add a card to the player's hand

Parameters

<i>c</i>	- card to be added
----------	--------------------

4.8.3.2 addToTotalScore()

```
void Player.addToTotalScore (
    int points ) [inline]
```

Add points earned in the round to the total score

Parameters

<i>points</i>	- points earned that round
---------------	----------------------------

4.8.3.3 checkMelds()

```
void Player.checkMelds ( ) [inline]
```

Check for melds at the beginning of each round

4.8.3.4 discardFromHand()

```
Card Player.discardFromHand (
    String input ) throws IllegalArgumentException [inline]
```

Given the user's input, discard a card from the player's hand if it is there and return its [Card](#) representation.

Parameters

<i>input</i>	- user's input
--------------	----------------

Returns

card that was discarded

Exceptions

<i>IllegalArgumentException</i>	if the card does not exist in the hand
---------------------------------	--

4.8.3.5 displayHand()

```
void Player.displayHand ( ) [inline]
```

Interfacing method to display the hand

4.8.3.6 extractDeadwood()

```
ArrayList<Card> Player.extractDeadwood ( ) [inline]
```

Extract deadwood cards from the hand, given the melds

4.8.3.7 getDeadwoodScore()

```
int Player.getDeadwoodScore ( ) [inline]
```

Accessor for the player's current deadwood score

Returns

player's current deadwood score

4.8.3.8 getHand()

```
Hand Player.getHand ( ) [inline]
```

Accessor for player's hand

Returns

player's hand

4.8.3.9 getMelds()

```
ArrayList<ArrayList<Card> > Player.getMelds ( ) [inline]
```

Get the melds in the player's hand

Returns

melds in player's hand

4.8.3.10 getName()

```
String Player.getName ( ) [inline]
```

Accessor for player's name.

Returns

name of player

4.8.3.11 getTotalScore()

```
int Player.getTotalScore ( ) [inline]
```

Accessor for the player's total game score

Returns

player's total game score

4.8.3.12 recalculateDeadwoodScore()

```
void Player.recalculateDeadwoodScore ( ) [inline]
```

Recalculate the deadwood score of the hand

4.8.3.13 resetDeadwoodScore()

```
void Player.resetDeadwoodScore ( ) [inline]
```

Reset the deadwood score at the end of a new deal

4.8.3.14 resetHand()

```
void Player.resetHand ( ) [inline]
```

Reset the player's hand at the end of a deal

4.8.3.15 resetMelds()

```
void Player.resetMelds ( ) [inline]
```

Reset melds

4.8.3.16 resetTotalScore()

```
void Player.resetTotalScore ( ) [inline]
```

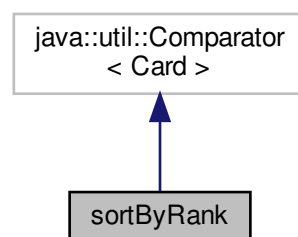
Reset the total score at the start of a new game

The documentation for this class was generated from the following file:

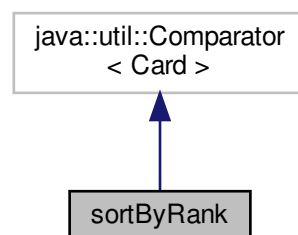
- Player.java

4.9 sortByRank Class Reference

Inheritance diagram for sortByRank:



Collaboration diagram for sortByRank:



Public Member Functions

- `int compare (Card c1, Card c2)`

4.9.1 Member Function Documentation

4.9.1.1 compare()

```
int sortByRank.compare (  
    Card c1,  
    Card c2 ) [inline]
```

Public method that returns integer based on which card is greater in rank

Parameters

<i>c1</i>	card one
<i>c2</i>	card two

Returns

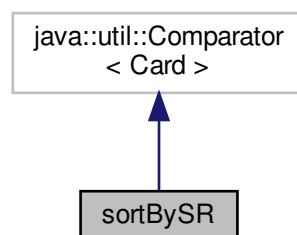
integer number, if *c1* is greater it returns a positive int else it returns a negative int

The documentation for this class was generated from the following file:

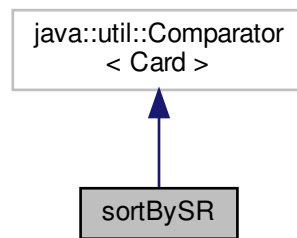
- `sortByRank.java`

4.10 sortBySR Class Reference

Inheritance diagram for sortBySR:



Collaboration diagram for sortBySR:



Public Member Functions

- int `compare` (`Card` c1, `Card` c2)

4.10.1 Member Function Documentation

4.10.1.1 `compare()`

```
int sortBySR.compare (  
    Card c1,  
    Card c2 ) [inline]
```

Public method that returns integer based on which card is greater in suit and rank

Parameters

<i>c1</i>	card one
<i>c2</i>	card two

Returns

integer number, if c1 is greater it returns a positive int else it returns a negative int

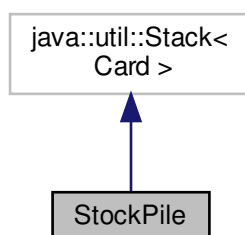
The documentation for this class was generated from the following file:

- `sortBySR.java`

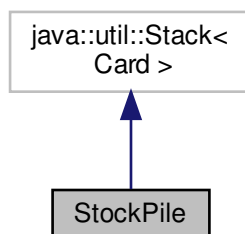
4.11 StockPile Class Reference

The stock pile class.

Inheritance diagram for StockPile:



Collaboration diagram for StockPile:



Public Member Functions

- int [search](#) (Object e)

4.11.1 Detailed Description

The stock pile class.

Author

Joy Xiao

4.11.2 Member Function Documentation

4.11.2.1 search()

```
int StockPile.search (
    Object e ) [inline]
```

Overrides search method since information should not be given

Returns

-1 always

The documentation for this class was generated from the following file:

- StockPile.java

4.12 Suit Enum Reference

[Suit](#) enum.H = Hearts, S = Spades, C = Clubs, D = Diamonds.

Public Attributes

- H
- S
- C

4.12.1 Detailed Description

[Suit](#) enum.H = Hearts, S = Spades, C = Clubs, D = Diamonds.

Author

Smita Singh

The documentation for this enum was generated from the following file:

- Suit.java

4.13 UserInputOps Class Reference

Class designated for user input and operations.

Static Public Member Functions

- static String [chooseDiscard](#) ()
- static void [closeScanner](#) ()
- static char [knock](#) ()
- static char [playAgain](#) ()
- static int [playerDecision](#) ()
- static String [username](#) ()

4.13.1 Detailed Description

Class designated for user input and operations.

Author

Benson Hall

4.13.2 Member Function Documentation

4.13.2.1 [chooseDiscard](#)()

```
static String UserInputOps.chooseDiscard ( ) [inline], [static]
```

Discard from hand and put on top of discard pile

4.13.2.2 [closeScanner](#)()

```
static void UserInputOps.closeScanner ( ) [inline], [static]
```

Close scanner

4.13.2.3 [knock](#)()

```
static char UserInputOps.knock ( ) [inline], [static]
```

Does the user want to knock?

Returns

y if yes, n if no

4.13.2.4 playAgain()

```
static char UserInputOps.playAgain ( ) [inline], [static]
```

Does the user want to play a new game?

Returns

y if yes, n if no

4.13.2.5 playerDecision()

```
static int UserInputOps.playerDecision ( ) [inline], [static]
```

[Player](#) makes decisions

Returns

player's decision, integer between 1 and 4

4.13.2.6 username()

```
static String UserInputOps.username ( ) [inline], [static]
```

Get user to print name and return result

Returns

name - user's name

The documentation for this class was generated from the following file:

- UserInputOps.java

Index

- addCardToHand
 - Player, [20](#)
- addToTotalScore
 - Player, [20](#)
- calculateScores
 - GameOps, [12](#)
- Card, [7](#)
 - Card, [7](#)
 - getRank, [8](#)
 - getSuit, [8](#)
 - points, [8](#)
 - rankString, [8](#)
 - suitSymbol, [9](#)
 - toString, [9](#)
 - toSymbol, [9](#)
- checkMelds
 - Meld, [18](#)
 - Player, [20](#)
- chooseDiscard
 - UserInputOps, [28](#)
- closeScanner
 - UserInputOps, [28](#)
- compare
 - sortByRank, [24](#)
 - sortBySR, [25](#)
- Computer, [10](#)
 - makeMove, [10](#)
- contains
 - Hand, [17](#)
- createDiscardPile
 - GameOps, [13](#)
- createStockPile
 - GameOps, [13](#)
- discardFromHand
 - Player, [20](#)
- DiscardPile, [11](#)
 - displayTopCard, [12](#)
- displayHand
 - Hand, [17](#)
 - Player, [21](#)
- displayTopCard
 - DiscardPile, [12](#)
- distributeCards
 - GameOps, [13](#)
- endGame
 - GameOps, [14](#)
- extractDeadwood
 - Player, [21](#)
- GameOps, [12](#)
 - calculateScores, [12](#)
 - createDiscardPile, [13](#)
 - createStockPile, [13](#)
 - distributeCards, [13](#)
 - endGame, [14](#)
 - playAgain, [14](#)
 - processDecision, [14](#)
 - resetEverything, [14](#)
 - username, [15](#)
- getDeadwoodScore
 - Player, [21](#)
- getHand
 - Player, [21](#)
- getMelds
 - Player, [21](#)
- getName
 - Player, [22](#)
- getRank
 - Card, [8](#)
- getSuit
 - Card, [8](#)
- getTotalScore
 - Player, [22](#)
- GinRummy, [15](#)
 - main, [15](#)
 - playGame, [15](#)
- Hand, [16](#)
 - contains, [17](#)
 - displayHand, [17](#)
 - Hand, [17](#)
 - remove, [17](#)
- knock
 - UserInputOps, [28](#)
- main
 - GinRummy, [15](#)
- makeMove
 - Computer, [10](#)
- Meld, [18](#)
 - checkMelds, [18](#)
- playAgain
 - GameOps, [14](#)
 - UserInputOps, [28](#)
- playGame
 - GinRummy, [15](#)

- Player, 19
 - addCardToHand, 20
 - addToTotalScore, 20
 - checkMelds, 20
 - discardFromHand, 20
 - displayHand, 21
 - extractDeadwood, 21
 - getDeadwoodScore, 21
 - getHand, 21
 - getMelds, 21
 - getName, 22
 - getTotalScore, 22
 - Player, 19
 - recalculateDeadwoodScore, 22
 - resetDeadwoodScore, 22
 - resetHand, 22
 - resetMelds, 23
 - resetTotalScore, 23
- playerDecision
 - UserInputOps, 29
- points
 - Card, 8
- processDecision
 - GameOps, 14
- rankString
 - Card, 8
- recalculateDeadwoodScore
 - Player, 22
- remove
 - Hand, 17
- resetDeadwoodScore
 - Player, 22
- resetEverything
 - GameOps, 14
- resetHand
 - Player, 22
- resetMelds
 - Player, 23
- resetTotalScore
 - Player, 23
- search
 - StockPile, 27
- sortByRank, 23
 - compare, 24
- sortBySR, 24
 - compare, 25
- StockPile, 26
 - search, 27
- Suit, 27
- suitSymbol
 - Card, 9
- toString
 - Card, 9
- toSymbol
 - Card, 9
- UserInputOps, 27
 - chooseDiscard, 28
 - closeScanner, 28
 - knock, 28
 - playAgain, 28
 - playerDecision, 29
 - username, 29
- username
 - GameOps, 15
 - UserInputOps, 29