

JavaScript is disabled on your browser.

- [Prev Class](#)
- **Next Class**

- [Frames](#)
- [No Frames](#)

- [All Classes](#)

- [Summary:](#)
- [Nested |](#)
- [Field |](#)
- [Constr |](#)
- [Method](#)

- [Detail:](#)
- [Field |](#)
- [Constr |](#)
- [Method](#)

model.interfaces

Interface GameEngine

.

```
public interface GameEngine
```

Assignment interface for SADI providing main Card Game model functionality

Author:
Caspal Ryan

- *Field Summary*

Modifier and Type **Field and Description**

static final **JUST LEVEL**

- *Method Summary*

[All Methods](#) [Instance Methods](#) [Abstract Methods](#)

Modifier and Type

Method and Description

void **addGameEngineCallback**(GameEngineCallback gameEngineCallba
ck)

void **addPlayer**(Player player)

void **dealHouse**(int delay)
Same as dealPlayer() but deals for the house and calls the house
versions of the callback methods on GameEngineCallback, no
player parameter is required After the house deal has finished but
BEFORE calling houseResult() win/loss values are applied to all
players based on their bet

void **dealPlayer**(Player player, int delay)
Deal cards to the player, increments of delay are in milliseconds
(ms) 1.

java.util.Collection
<Player> **getAllPlayers**()

Player **getPlayer**(java.lang.String id)

java.util.Deque<Play **getShuffledDeck**()

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

<code>ingCard></code>	A debug method to return a deck of cards containing 52 unique cards in random/shuffled order
<code>boolean</code>	<code>placeBet</code> (<code>Player</code> player, <code>int</code> bet) the implementation should forward the call to the <code>Player</code> class to handle
<code>boolean</code>	<code>removeGameEngineCallback</code> (<code>GameEngineCallback</code> gameEngineCallback)
<code>boolean</code>	<code>removePlayer</code> (<code>Player</code> player)

- **Field Detail**

- **`BUST_LEVEL`**

`static final int BUST_LEVEL`

See Also:

[Constant Field Values](#)

- **Method Detail**

- **`dealPlayer`**

`void dealPlayer`(`Player` player, `int` delay)

Deal cards to the player, increments of delay are in milliseconds (ms) 1. deal a card to the player 2. call `GameEngineCallback.nextCard(...)` 3. continue looping until the player busts (default value of `GameEngine.BUST_TOTAL` is 104) 4. call `GameEngineCallback.bustCard(...)` 5. call `GameEngineCallback.result(Player, int, GameEngine)` with final result for player (the pre bust total) 6. update the player with final result so it can be retrieved later

Parameters:

- player - the current player who will have their result set at the end of the hand
- delay - the delay between cards being dealt

See Also:

[GameEngineCallback](#)

[dealHouse](#)

`void dealHouse`(`int` delay)

Same as `dealPlayer()` but deals for the house and calls the house versions of the callback methods on `GameEngineCallback`, no player parameter is required After the house deal has finished but BEFORE calling `houseResult()` win/loss values are applied to all players based on their bet

Parameters:

- delay - the delay between cards being dealt

See Also:

[GameEngineCallback](#), [dealPlayer](#)(`Player`, `int`)

- **`addPlayer`**

`void addPlayer`(`Player` player)

Parameters:

- player - to add to game

- **`getPlayer`**

`Player` [getPlayer](#)(`java.lang.String` id)

Parameters:

- id - id of player to retrieve (null if not found)

Returns:

the `Player` or null if `Player` does not exist

- **`removePlayer`**

`boolean` [removePlayer](#)(`Player` player)

Parameters:

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

player - to remove from game

Returns:

true if the player existed

- `addGameEngineCallback`

`void addGameEngineCallback(GameEngineCallback gameEngineCallback)`

Parameters:

gameEngineCallback - a client specific implementation of GameEngineCallback used to perform display updates etc. you will use a different implementation of the GameEngineCallback for GUI and console versions

- `removeGameEngineCallback`

`boolean removeGameEngineCallback(GameEngineCallback gameEngineCallback)`

Parameters:

gameEngineCallback - called when a player quits the game to remove no longer needed UI updates

Returns:

true if the gameEngineCallback existed

- `getAllPlayers`

`java.util.Collection<Player> getAllPlayers()`

Returns:

an unmodifiable collection (or a copy) of all Players

See Also:

`Player`

- `placeBet`

`boolean placeBet(Player player, int bet)`

the implementation should forward the call to the Player class to handle

Parameters:

player - the Player who is placing the bet
bet - the bet in points

Returns:

true if bet is greater than or equal to 0 and player had sufficient points to place the bet

- `getShuffledDeck`

`java.util.Deque<PlayingCard> getShuffledDeck()`

A debug method to return a deck of cards containing 52 unique cards in random/shuffled order

Returns:

a Deque (type of Collection) of PlayingCard

See Also:

`PlayingCard`

- [Prev Class](#)
- [Next Class](#)

- [Frames](#)
- [No Frames](#)

- [All Classes](#)

- [Summary:](#)
- [Nested |](#)
- [Field |](#)
- [Constr |](#)
- [Method](#)

- [Detail:](#)
- [Field |](#)
- [Constr |](#)
- [Method](#)

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder