Player.Player Class Reference

Public Member Functions

def **__init__** (self, player_id, name, property_sets)
constructor, takes in an id, a name, and a dictionary describing the property groups More...

def **__str__** (self)
describes the player in plain text More...

def **addMoney** (self, money)
add cash directly to current players balance More...

def **takeMoney** (self, money)
take cash away from current players balance and update bankruptcy status More...

def **getName** (self)
Accessor: returns the players name. More...

def **getId** (self)
Accessor: returns the id of the current player. More...

def **getBalance** (self)
Accessor: returns the cash balance for the current player. More...

def **getPosition** (self)
Accessor: returns the current position of the player on board. More...

def **setPosition** (self, position)
move the player to a specified position More...

def **getProperties** (self)
Accessor: retuns a dictionary where key = name of property group, value = a list where first element is a list of property spaces and second element is the max number of properties in that group. More...

def **addProperty** (self, space)
sets owner on space and adds it to players collection of properties, if player owns all properties in group activates set bonus More...

def **removeProperty** (self, space)
resets owner on space and removes it from players collection of properties. More...

def **isJailed** (self)
check players jail status More...

def **updateJailed** (self, jail_status)
Takes in a boolean which states if player is jailed or not. More...

def **hasBail** (self)
check players jail status More...

def **updateBail** (self, bail_status)
Takes in a boolean which states if player is jailed or not. More...

def **isBankrupt** (self)
check if the current player is bankrupt More...

Private Attributes

- **_id**
- **_name**
- **_position**
- **_balance**
- **_properties**
- **_bankrupt**
- **_jailed**
- **_on_bail**

Detailed Description

```
Class Player describes each player including how much money
and property they currently own
```

Constructor & Destructor Documentation

◆ __init__()

```
def Player.Player.__init__ (  self,
                              player_id,
                              name,
                              property_sets
                           )
```

constructor, takes in an id, a name, and a dictionary describing the property groups

Parameters

String	player_id
String	name
int	position
int	balance
boolean	bankrupt
boolean	jailed
boolean	on_bail

Member Function Documentation

◆ __str__()

def Player.Player.__str__ (self)

describes the player in plain text

Parameters

self the object pointer

◆ addMoney()

def Player.Player.addMoney (self,
money
)

add cash directly to current players balance

Parameters

self the object pointer

int money the value involved in the transaction

◆ addProperty()

def Player.Player.addProperty (self,
space
)

sets owner on space and adds it to players collection of properties, if player owns all properties in group activates set bonus

Parameters

self the object pointer

int space the position on the board

◆ getBalance()

def Player.Player.getBalance (self)

Accessor: returns the cash balance for the current player.

Parameters

self the object pointer

getId()

def Player.Player.getId (self)

Accessor: returns the id of the current player.

Parameters

self the object pointer

getName()

def Player.Player.getName (self)

Accessor: returns the players name.

Parameters

self the object pointer

getPosition()

def Player.Player.getPosition (self)

Accessor: returns the current position of the player on board.

Parameters

self the object pointer

getProperties()

def Player.Player.getProperties (self)

Accessor: retuns a dictionary where key = name of property group, value = a list where first element is a list of property spaces and second element is the max number of properties in that group.

Parameters

self the object pointer

hasBail()

def Player.Player.hasBail (self)

check players jail status

Parameters

self the object pointer

◆ isBankrupt()

def Player.Player.isBankrupt (self)

check if the current player is bankrupt

Parameters

self the object pointer

◆ isJailed()

def Player.Player.isJailed (self)

check players jail status

Parameters

self the object pointer

◆ removeProperty()

def Player.Player.removeProperty (self,
space
)

resets owner on space and removes it from players collection of properties.

Also removes set bonus for owning a full set

Parameters

self the object pointer

int space the position on the board

◆ setPosition()

```
def Player.Player.setPosition ( self,
                                position
                              )
```

move the player to a specified position

Parameters

self the object pointer

int position the position on the board

◆ takeMoney()

```
def Player.Player.takeMoney ( self,
                              money
                            )
```

take cash away from current players balance and update bankruptcy status

Parameters

self the object pointer

int money the value involved in the transaction

◆ updateBail()

```
def Player.Player.updateBail ( self,
                               bail_status
                             )
```

Takes in a boolean which states if player is jailed or not.

Parameters

self the object pointer boolean bail_status

◆ updateJailed()

```
def Player.Player.updateJailed (  self,
                                  jail_status
                               )
```

Takes in a boolean which states if player is jailed or not.

Parameters

self the object pointer

boolean jail_status

Member Data Documentation

_balance

Player.Player._balance private

_bankrupt

Player.Player._bankrupt private

_id

Player.Player._id private

_jailed

Player.Player._jailed private

_name

Player.Player._name private

_on_bail

Player.Player._on_bail private

◆ _position

Player.Player._position

◆ _properties

Player.Player._properties private

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

- D:/Documents/CS3306/monopoly/Monopoly-master/Monopoly-master/**Player.py**