SUMMARY: NESTED | FIELD | CONSTR | METHOD   DETAIL: FIELD | CONSTR | METHOD

SEARCH:

Package edu.fau.COT4930

Class TablePile

java.lang.Object
    edu.fau.COT4930.TablePile

```
public class TablePile
extends Object
```

Constructor Summary

Constructors

Constructor	Description
TablePile(int x, int y)	

Method Summary

All Methods   Instance Methods   Concrete Methods

Modifier and Type	Method	Description
void	**addCardToPile** (**Card** newCard)	Pushes a new card onto the table stack assigns new y location to where last card is added
void	**addStackToPile** (**Stack**<**Card**> cards)	Takes a user stack cards and pushes stack on to table pile before card is pushed, a new x and y location is assigned to it to represent its position in the stack
boolean	**canAdd**(**Card** c)	given a card c, function tests if card (and resulting pile) can be added to stack Returns true if: input card is a different color than top card of the table stack and the input card is the next logical value to to the stack

int	getSize()	return size of the stack
Stack	getSubStack(int i)	given an index location on the stack i function removes cards in stack up to index i removed cards are returned to the user to push onto a different table stack
int	inStack(int x, int y)	given an x and y coordinate function parses through table pile and searches for card clicked on by user the index i of clicked card is return to the user if the clicked element is not in the stack, then -1 is returned
Card	popCard()	pops the top card off the stack and returns it
Card	showCard(int i)	returns card at index i without altering the stack
void	topCardFaceUp()	checks the top card of the stack if the top card has its back facing then function flips card

Methods inherited from class java.lang.Object

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

Constructor Detail

TablePile

```
public TablePile(int x,
                 int y)
```

Method Detail

`public int inStack(int x, int y)`

given an x and y coordinate function parses through table pile and searches for card clicked on by user the index i of clicked card is return to the user if the clicked element is not in the stack, then -1 is returned

Parameters:

x -

y -

Returns:

---

getSubStack

`public Stack getSubStack(int i)`

given an index location on the stack i function removes cards in stack up to index i removed cards are returned to the user to push onto a different table stack

Parameters:

ind -

Returns:

---

showCard

`public Card showCard(int i)`

returns card at index i without altering the stack

Parameters:

ind -

Returns:

---

popCard

`public Card popCard()`

pops the top card off the stack and returns it

Returns:

SEARCH: [   ]

---

addStackToPile

```
public void addStackToPile(Stack<Card> cards)
```

Takes a user stack cards and pushes stack on to table pile before card is pushed, a new x and y location is assigned to it to represent its position in the stack

Parameters:

cards -

---

addCardToPile

```
public void addCardToPile(Card newCard)
```

Pushes a new card onto the table stack assigns new y location to where last card is added

Parameters:

newCard -

---

canAdd

```
public boolean canAdd(Card c)
```

given a card c, function tests if card (and resulting pile) can be added to stack Returns true if: input card is a different color than top card of the table stack and the input card is the next logical value to to the stack

Parameters:

c -

Returns:

topCardFaceUp

public void topCardFaceUp()

checks the top card of the stack if the top card has its back facing then function flips card

getSize

public int getSize()

return size of the stack

Returns: