

[OVERVIEW](#) [PACKAGE](#) [CLASS](#) [TREE](#) [DEPRECATED](#) [INDEX](#) [HELP](#)[ALL CLASSES](#)SEARCH: SUMMARY: [NESTED](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)    [DETAIL: FIELD](#) | [CONSTR](#) | [METHOD](#)**Package** [modele](#)

## Class Cheater

```
java.lang.Object
  modele.Occupant
    modele.Character
      modele.Cheater
```

**All Implemented Interfaces:**[Moveable](#), [Questionable](#)

```
public class Cheater
  extends Character
```

A character that cheat the hunters by giving them the worst direction to the treasure witch is the opposite of the best direction minus 1 mod 8 to make sure that the hunter does not go back to the treasure after hitting a border

### Constructor Summary

#### Constructors

Constructor	Description
<b>Cheater</b> ( <a href="#">Position</a> position, <a href="#">Position</a> treasure)	Create a cheater with a position and a treasure position

### Method Summary

All Methods	Instance Methods	Concrete Methods
<b>Modifier and Type</b>	<b>Method</b>	<b>Description</b>
int	<b>getWorstDirectionToTreasure</b> ( <a href="#">Position</a> treasure, <a href="#">Position</a> hunter)	Get the worst direction to the treasure.
void	<b>process</b> ( <a href="#">Character</a> m)	Process the interaction between the cheater and a character If the character is a hunter, it will give him the worst direction to the treasure Otherwise, it will give

[OVERVIEW](#) [PACKAGE](#) [CLASS](#) [TREE](#) [DEPRECATED](#) [INDEX](#) [HELP](#)[ALL CLASSES](#)SEARCH: [SUMMARY: NESTED](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)    [DETAIL: FIELD](#) | [CONSTR](#) | [METHOD](#)

`getDirection, getNextPosition, getWaitingTime, setDirection, setNextPosition, setTempDir, setWaitingTime, toString`

### Methods inherited from class `modele.Occupant`

`equals, getCol, getPosition, getRow, isStepEnabled, setPosition`

### Methods inherited from class `java.lang.Object`

`clone, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait`

## Constructor Detail

### Cheater

```
public Cheater(Position position,
               Position treasure)
```

Create a cheater with a position and a treasure position

**Parameters:**

`position` - the position of the cheater

`treasure` - the position of the treasure

## Method Detail

### getWorstDirectionToTreasure

```
public int getWorstDirectionToTreasure(Position treasure,
                                       Position hunter)
```

Get the worst direction to the treasure. It is used in the process method for hunters to guide them at the opposite of the best direction

**Parameters:**

[OVERVIEW](#) [PACKAGE](#) [CLASS](#) [TREE](#) [DEPRECATED](#) [INDEX](#) [HELP](#)[ALL CLASSES](#)SEARCH: SUMMARY: [NESTED](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)**process**

```
public void process(Character m)
```

Process the interaction between the cheater and a character. If the character is a hunter, it will give him the worst direction to the treasure. Otherwise, it will give him a random direction.

**Specified by:**

`process` in interface `Questionable`

**Specified by:**

`process` in class `Occupant`

**Parameters:**

`m` - the character to process

[OVERVIEW](#) [PACKAGE](#) [CLASS](#) [TREE](#) [DEPRECATED](#) [INDEX](#) [HELP](#)[ALL CLASSES](#)SUMMARY: [NESTED](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)