edu.gmu.cs.swe619.assignments.assignment06

Class Bag<E>

java.lang.Object
    edu.gmu.cs.swe619.assignments.assignment06.Bag<E>

```
public class Bag<E>
extends java.lang.Object
```

Generic Bag class representing a multiset for SWE-619, Assignment #6. See Liskov exercise 7.11. Example: A bag of 2 cats and a dog is map = {cat: 2, dog: 1}

Constructor Summary

Constructors

Constructor and Description
`Bag()` Public constructor to create a new Bag object.

Method Summary

All Methods | **Instance Methods** | **Concrete Methods**

Modifier and Type	Method and Description
`E`	`choose()` Choose an arbitrary element from the bag.
`void`	`insert(E e)` Add 1 occurrence of e to this
`boolean`	`isIn(E e)` Check if there is at least one occurrence of the given element in the Bag.
`void`	`remove(E e)` Remove 1 occurrence of e from this
`int`	`size()` Return cardinality of this.
`java.lang.String`	`toString()` Provide a string that represents the abstract state of the Bag.

Methods inherited from class java.lang.Object

`equals, getClass, hashCode, notify, notifyAll, wait, wait, wait`

Constructor Detail

Bag

```
public Bag()
```

Public constructor to create a new Bag object.

Method Detail

insert

```
public void insert(E e)
```

Add 1 occurrence of e to this

Parameters:

`e - the element to add to the Bag`

Throws:

`java.lang.RuntimeException - if the rep invariant is violated`

remove

```
public void remove(E e)
```

Remove 1 occurrence of e from this

Parameters:

`e - the element to add to the Bag`

Throws:

`java.lang.RuntimeException - if the rep invariant is violated`

isIn

```
public boolean isIn(E e)
```

Check if there is at least one occurrence of the given element in the Bag.

Parameters:

e - the element to find in the Bag

Returns:

true if there is at least one element of e in the Bag, false otherwise.

size

```
public int size()
```

Return cardinality of this. Not that the Bag can contain multiple elements of the same type, and these are counted separately. For example, a bag with map = {cat: 2, dog: 1} has a size of 3.

Returns:

the number of types of elements in the Bag.

choose

```
public E choose()
```

Choose an arbitrary element from the bag. If this is empty throw an ISE.

Returns:

arbitrary element of this

Throws:

java.lang.IllegalStateException - if the bag is empty

toString

```
public java.lang.String toString()
```

Provide a string that represents the abstract state of the Bag. We choose to use Map as our rep, so we can use that for the AF as well.

Overrides:

toString in class java.lang.Object

Returns:

a string representing the state of the bag.