

Forum:[Hack Forums](#)

Thread: [Cant put object into hashmap connected to other object](#)

deadsh0t - January 9, 2019, 2:32 pm

So im making this consolebased game for my software development class as my assessment
But i cant understand this bit or why it is wrong to do it like this.

So basicly i have 10 rooms

so a room has the following things:

- A roomnumber
- A hashmap that contains a Room as value and a String as key
- An arraylist for items (Item has its own class ofcourse)

Now the problem, A room can be exited from 4 direction North,East,South,West so thats the String/key in the hashmap

It can also have 2 exits or 1 or 3.

So in that hashmap i save the direction as key and as value another room object so i can keep track wich exit goes to wich room.

so i have a method for adding exits. that looks like this:

```
Code public class Room {  
    private int RoomNo;  
    private HashMap<String, Room> Exits;  
  
    public Room(int roomNo){  
        Exits = new HashMap<String, Room>();  
        this.RoomNo = roomNo;  
    }  
  
    public void Addexits(Room room, String exitDirection){  
        Exits.put(exitDirection,room);  
    }  
}
```

And than in the game class i create 10 rooms and use call the addexits method to add exits that looks like this:

```
Code Rooms = new ArrayList<Room>();  
    //initialize all rooms  
    room1 = new Room(1);  
    room1.Addexits(room4,"West");  
    room1.Addexits(room3,"East");  
    room2 = new Room(2);  
    room2.Addexits(room2,"North");  
    room2.Addexits(room2,"South");  
    room3 = new Room(3);  
    room3.Addexits(room3,"West");  
    room3.Addexits(room3,"East");  
    room4 = new Room(4);  
    room4.Addexits(room4,"West");  
    room4.Addexits(room4,"East");
```

However as you can see with room1 i try to add 2 exits 1 in the west to room4 and 1 in the east to room3 however when i run this it shows this in the hashmap:

West -> null

East -> null

But when i add 2 exits both with object room1 it puts it like it should be like this:

West -> room1

East -> room1

So i dont really get what im doing wrong each room has its own hashmap on creation and the rooms are already created so that shouldnt be an issue.

GenericSkid - January 10, 2019, 2:37 am

Your rooms are null when you add them... initialize all the rooms first THEN add them as exits.

UpperCaseVariables is a bad convention in Java for field and variable names. Using camelCaseStyle is the typical convention.

You can make your fields final by adding it as a modifier.

Instead of instantiating exits in the constructor it can be done in-line.

The instantiation of exits can remove redundant generics: new HashMap<>();

The definition of exits can be made clearer by changing the field type to Map. This won't affect the instantiation / value. It takes less space, and is more abstract.

PHP Code

```
<?php
public class Room {
    private final int roomNo;
    private final Map<String, Room> exits new HashMap<>();

    public Room(int roomNo) {
        this.roomNo = roomNo;
    }

    public void addExit(Room room, String direction){
        exits.put(direction, room);
    }
}
```

deadsh0t - January 10, 2019, 6:30 am

(Jan 10, 2019 03:37 AM)GenericSkid Wrote: [\[CITE\]](#) Your rooms are null when you add them... initialize all the rooms first THEN add them as exits.

UpperCaseVariables is a bad convention in Java for field and variable names. Using camelCaseStyle is the typical convention.

You can make your fields final by adding it as a modifier.

Instead of instantiating exits in the constructor it can be done in-line.

The instantiation of exits can remove redundant generics: new HashMap<>();

The definition of exits can be made clearer by changing the field type to Map. This won't affect the instantiation / value. It takes less space, and is more abstract.

PHP Code

```
<?php
public class Room {
    private final int roomNo;
```

```
private final Map<String, Room> exits = new HashMap<>();
```

```
public Room(int roomNo) {  
    this.roomNo = roomNo;  
}
```

```
public void addExit(Room room, String direction){  
    exits.put(direction, room);  
}
```

Yeah i figured it out i felt so dumb haha

Yeah the camelCaseStyle is something that is a must in my assignment i really need to stick to it, but sometimes i forget to use it.

Thanks for the tip on the map and making it final :)