Questions

Bookmark this page

variable scope

4/4 points (graded)

Consider the following code (which is missing some variable declarations):

```
1  class Apartment {
2      Apartment(String newAddress) {
3          this.address = newAddress;
4          this.roommates = new HashSet<Person>();
5      }
6
7      String getAddress() {
8          return address;
9      }
10
11     void addRoommate(Person newRoommate) {
12         roommates.add(newRoommate);
13         if (roommates.size() > MAXIMUM_OCCUPANCY) {
14             roommates.remove(newRoommate);
15             throw new TooManyPeopleException();
16         }
17     }
18
19     int getMaximumOccupancy() {
20         return MAXIMUM_OCCUPANCY;
21     }
22 }
```

Which of these lines are within the scope of the `newRoommate` variable? Check all that apply.

- [ ] line 3
- [ ] line 8
- [x] line 12
- [x] line 15
- [ ] line 20

✔

Explanation
The scope of a parameter is its function body, so lines 10-16 are in scope.

What is the appropriate scope for the (currently undeclared) `address` variable?

- (•) lines 2-21
- ( ) lines 3-4
- ( ) line 8
- ( ) lines 12-16

✔

Explanation
`address` is an instance variable, also called a field in Java, so its scope should be the entire class.

Out of the choices below, what is the best declaration for the `roommates` variable?

- ◯ List roommates;
- ◯ Set roommates;
- ◉ final Set roommates;
- ◯ HashSet roommates;

✔

Explanation
The `roommates` variable is a `Set`, not a `List`, and it's best to use interfaces like `Set` rather than concrete types like `HashSet` when declaring a variable. Include `final` because the `roommates` variable should always point to the same `Set` object. It never needs to be reassigned, only mutated.

Out of the choices below, what is the best declaration for the `MAXIMUM_OCCUPANCY` variable?

- ◯ int MAXIMUM_OCCUPANCY = 8;
- ◯ final int MAXIMUM_OCCUPANCY = 8;
- ◯ static int MAXIMUM_OCCUPANCY = 8;
- ◉ static final int MAXIMUM_OCCUPANCY = 8;

✔

Explanation
`MAXIMUM_OCCUPANCY` is a constant, so it should be declared static (not an instance variable, but a class variable) and final (so that it can never be reassigned).

Submit

Show Answer

Answers are displayed within the problem

Previous Next