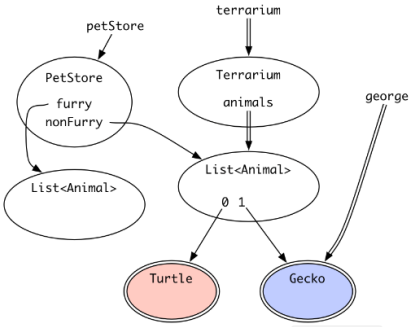
Questions

Bookmark this page

follow me

3/3 points (graded)



Is it possible that a client with the variable `terrarium` could modify the `Turtle` in red?

- ○ Yes, because all the references between "terrarium" and the "Turtle" are mutable
- ○ Yes, because of some reference between "terrarium" and the "Turtle" that is mutable
- ○ Yes, because the "Turtle" is mutable
- ◉ No, because the "Turtle" is immutable
- ○ No, because of some reference between "terrarium" and the "Turtle" that is immutable
- ○ No, because all the references between "terrarium" and the "Turtle" are immutable

✔

Explanation
The double circle means it's immutable. Whether it can be reached via mutable or immutable references is irrelevant.

Is it possible that a client with the variable `george` could modify the `Gecko` in blue?

- ○ Yes, because all the references between "george" and the "Gecko" are mutable
- ○ Yes, because of some reference between "george" and the "Gecko" that is mutable
- ○ Yes, because the "Gecko" is mutable
- ◉ No, because the "Gecko" is immutable
- ○ No, because of some reference between "george" and the "Gecko" that is immutable
- ○ No, because all the references between "george" and the "Gecko" are immutable

✔

Explanation
The double circle means it's immutable. Whether it can be reached via mutable or immutable references is irrelevant.

Is it possible that a client with the variable `petStore` could do something such that a client with the variable `terrarium` could no longer access the `Gecko` in blue?

- ◯ Yes, because all the references between "petStore" and the "Gecko" are mutable
- ◉ Yes, because of some reference between "petStore" and the "Gecko" that is mutable
- ◯ Yes, because the "Gecko" is mutable
- ◯ No, because the "Gecko" is immutable
- ◯ No, because of some reference between "petStore" and the "Gecko" that is immutable
- ◯ No, because all the references between "petStore" and the "Gecko" are immutable

✔

Explanation
In particular, there is a mutable reference that is also between `terrarium` and the `Gecko`, which a client with `petStore` might be able to reassign.
That's a 99% correct answer. If the mutable reference was inside an *immutable* object, then that object must be preventing that reference from being reassigned.
Here the mutable reference is in a mutable object (a `List`), so it's possible that the reference could be reassigned, breaking `terrarium`'s path to the `Gecko`.

Submit

Show Answer

Answers are displayed within the problem

Previous Next