

Set 09 Answers

1. Define the terms mutable and immutable.

The state of a mutable object may change over time. If an object is immutable, its state is set when it is constructed and will never vary after that.

2. Are String objects mutable?

No, Strings are immutable.

3. Are Integer objects mutable? (If you're not sure, inspect the online API documentation for the Integer class and find out!)

No, Integer objects are immutable.

4. True/False _ Aliasing can lead to problems if the object is mutable.

True.

5. True/False _ Aliasing can lead to problems if the object is immutable.

False.

6. Suppose you are passing (or returning) an array of primitives to/from a method. Is it safe to make a reference copy only?

No.

7. Suppose you are passing (or returning) an array of references to immutable objects to/from a method. Is it safe to make a reference copy only? Is it safe to make a shallow copy?

Reference copy is unsafe. Shallow is fine.

8. Suppose you are passing or returning an array of references to mutable objects to/from a method. Is it safe to make a reference copy only? Is it safe to make a shallow copy?

Neither one is safe. Only a deep copy is safe in this case.

9. How many `ints` are created by the statement: `int[] a = new int[5];`

5

10 How many Strings are created by the statement: `String[] a = new String[5];`

(Hint: The answer to this question and the previous question are different!)
None. What is created is an array of null references.

11. Are the elements of an array of primitives automatically initialized? If so, to what values?

Yes, to 0 or false.

12. Are the elements of an array of references to objects initialized? If so, to what values?

Yes, to null.

13. Draw the memory diagram for each of the following code fragments:

a. `int[] a = new int[4];`

Should have a variable on the call stack which points to an array on the heap of size 4. Each box in the array contains a zero.

b. `String[] b = new String[4];`

`for (int i = 0; i < b.length; i++)`

`b[i] = _value_ + i;`

Should have a variable on the call stack which points to an array on the heap of size 4. Each box in the array points to a String. The strings say `_value0_`, `_value1_`, `_value2_`, and `_value3_`.

14. Write a class that has an instance variable which is an array of Cat objects, called `kitties`. Write a method that returns a reference copy of `kitties`. Write a method that returns a shallow copy of `kitties`. Write a method that returns a deep copy of `kitties`.

```
public class Circus {
    private Cat[] kitties;

    public Cat[] referenceCopy() {
        return kitties;
    }

    public Cat[] shallowCopy() {
        Cat[] copy = new Cat[kitties.length];
        for (int i = 0; i < copy.length; i++) {
            copy[i] = kitties[i];
        }
    }
}
```

```
        }  
        return copy;  
    }  
  
    public Cat[] deepCopy() {  
        Cat[] copy = new Cat[kitties.length];  
        for (int i = 0; i < copy.length; i++) {  
            copy[i] = new Cat(kitties[i]);  
        }  
        return copy;  
    }  
}
```