Enthuware Mobile Test Studio

Standard Tests - Test 2 : 2019-07-30 14:59

Q 37 of 84 ☐ Mark Advanced Class Design - Enumerated Types enthuware.ocpjp.v8.2.1437

Given the following code:

```
enum Title
{
    MR("Mr. "), MRS("Mrs. "), MS("Ms. ");
    private String title;
    private Title(String s){
    title = s;
    }
    public String format(String first, String last){
    return title+" "+first+" "+last;
    }
}

//INSERT CODE HERE
```

Identify valid code snippets ..

(Assume that Title is accessible wherever required.)

Answered Incorrectly You had to select 4 option(s)

☐
```
class TestClass{
 oid someMethod()
{
  System.out.println(Title.format("Rob", "Miller"));
 }
}
```

You cannot call format method directly on Title because it is not a static method. You must call it on Title instances, which are MR, MRS, and MS.

☑
```
class TestClass{
  void someMethod()
 {
     System.out.println(Title.MR.format("Rob", "Miller"));
 }
 }
```

☑
```
class TestClass{
 oid someMethod()
{
  System.out.println(MR.format("Rob", "Miller"));
 }
}
```

It must be `Title.MR.format("Rob", "Miller")`.

☐
```
enum Title2 extends Title
{
  DR("Dr. ");
}
```

An enum cannot extend another enum or class. It may implement an interface though.

```
☑                s{
 oid someMethod()

   Title.DR dr = new Title.DR("Dr. ");
  }
}
```

Enum constants cannot be instantiated/created using the new keyword.

```
☑ enum Title2
 {
    DR;
    private Title t;
 }
```

```
☐ enum Title2
 {
    DR;
    private Title t = Title.MR;
 }
```

```
☐ enum Title2
 {
    DR;
    private Title t = Title.MR;
    public String format(String s){ return t.format(s, s); };
 }
```

Previous Next Evaluate Finish Review

You need to know the following facts about enums:
1. Enum constructor is always private. You cannot make it public or protected. If an enum type has no constructor declarations, then a private constructor that takes no parameters is automatically provided.
2. An enum is implicitly final, which means you cannot extend it.
3. You cannot extend an enum from another enum or class because an enum implicitly extends `java.lang.Enum`. But an enum can implement interfaces.
4. Since enum maintains exactly one instance of its constants, you cannot clone it. You cannot even override the clone method in an enum because `java.lang.Enum` makes it final.
5. Compiler provides an enum with two public static methods automatically - `values()` and `valueOf(String)`. The `values` method returns an array of its constants and `valueOf` method tries to match the String argument exactly (i.e. case sensitive) with an enum constant and returns that constant if successful otherwise it throws `java.lang.IllegalArgumentException`.

The following are a few more important facts about `java.lang.Enum` which you should know:

1. It implements `java.lang.Comparable` (thus, an enum can be added to sorted collections such as SortedSet, TreeSet, and TreeMap).
2. It has a method `ordinal()`, which returns the index (starting with 0) of that constant i.e. the position of that constant in its enum declaration.
3. It has a method `name()`, which returns the name of this enum constant, exactly as declared in its enum declaration.

Add/Edit Note