Quiz yourself: Access to enums

JAVA SE

Quiz yourself: Access to enums

What happens if an enum constructor is marked as public, protected, or private?

by Mikalai Zaikin and Simon Roberts

July 12, 2021

If you have worked on our quiz questions in the past, you know none of them is easy. They model the difficult questions from certification examinations. We write questions for the certification exams, and we intend that the same rules apply: Take words at their face value and trust that the questions are not intended to deceive you but to straightforwardly test your knowledge of the ins and outs of the language.

Given the enum

```
enum Type {
  BYTE(1), INT(4);
  final int size;
  public int getSizeInBytes() {
    return size;
  }
  Type(int i) {
    size = i;
  }
}
```

and the code fragment

```
System.out.println(/* add code here */);
```

Which expression when added to the `println()` above will print `1` to the console? Choose one.

A. `Type.values()[0]` The answer is A.

B. `Type.BYTE(1).getSizeInBytes()` The answer is B.

C. `Type.valueOf("Type.BYTE").getSizeInBytes()` The answer is C.

D.

```
Type.valueOf("BYTE").getSizeInBytes()
```

The answer is D.

Answer. Option A is incorrect. The `public static values()` method returns all values of the particular enum in an array. The element in position zero will be `Type.BYTE`, but when it's printed, it will produce the constant's string representation (unless you override `toString`), which is `BYTE`.

Option B demonstrates an attempt to call the constructor of the enum, which will fail at compile time. Java does not allow users to instantiate enum constants. This is because constructors of enums must always be `private`. The default constructor created by the compiler in the absence of a programmer-provided constructor is `private`, and any programmer-provided constructor must either be marked `private` or carry no access modifier. In either case, the result will be that the constructor is `private`. Note that if an enum constructor carries either of the modifiers `public` or `protected`, it will not compile.

Option C is incorrect and will fail at runtime. The compiler generates the `public static valueOf(String)` method for every enum. This method finds an enum object using its text representation. However, the text representation should simply be the short name of the constant (`BYTE`, in this case). The enum type is neither required nor permitted as part of that text; after all, the type is already known from the type on which the `valueOf` is called.

Based on the above (and by elimination of the three incorrect options), you can conclude that option D is correct. First, the code gets a reference to the `Type.BYTE` constant and then it calls the `getSizeInBytes()` method, which will return `1`.

The same result may be achieved by calling the `public static valueOf(...)` method on the `java.lang.Enum` class, because `Enum`, rather than `Object`, is the implicit and mandatory superclass for all enums. This method requires an additional argument indicating the enum type, but this argument is passed as a `java.lang.Class` object rather than in the textual representation. So, the following code would also evaluate to `1`:

```
Enum.valueOf(Type.class, "BYTE").getSizeInBytes();
```

Conclusion. The correct answer is option D.

Mikalai Zaikin

Mikalai Zaikin is a lead Java developer at IBA IT Park in Minsk, Belarus. During his career, he has helped Oracle with development of Java certification exams, and he has been a technical reviewer of several Java certification books, including three editions of the famous *Sun Certified Programmer for Java* study guides by Kathy Sierra and Bert Bates.

Simon Roberts

Simon Roberts joined Sun Microsystems in time to teach Sun's first Java classes in the UK. He created the Sun Certified Java Programmer and Sun Certified Java Developer exams. He wrote several Java certification guides and is currently a freelance educator who publishes recorded and live video training through Pearson InformIT (available direct and through the O'Reilly Safari Books Online service). He remains involved with Oracle's Java certification projects.

Share this Page