Java Lab 21: Reflection

This lab practices with reflection and RTTI.

1. Create a new Java project called Lab21. Download the file Employee.class. ***Do not copy*** Employee.class to the src directory – see #2.

2. Create a class Lab21Main with two methods: main( ) and void classFun(Class<?> c) { } – that is, an empty method (for now). In main, new up a Lab21Main object. Declare the variable Class<?> c = Class.forName("Employee"). IntelliJ will likely complain that this requires a try-catch block, so choose the option that surrounds it with one. Call classFun( ) with c as the parameter. This will fail – you should get something like "No such class" error message. This is just to make sure that the production directory with the .class files is created. Now copy Employee.class into the production directory out/production/Lab21 (where Lab21Main.class is stored)

3. In classFun():
- This, and the rest of this method, need to be in a try-catch block. Let IntelliJ create this for you, and later, let it add the new Exception types to the catch. Then, use this class to get and display:
- its canonical name
- all the member data
- all the local constructors, then all the constructors (getDeclaredConstructors versus getConstructors)
- all the local methods, then all the methods – number the output for clarity

4. Construct an Object instance of type Employee using its default constructor from the constructor set (***don't*** do this: Employee e = new Employee();) and the newInstance( ) method. Then print out whether the thing is an enum, is an interface, and print its toString().

Next, use the find( ) method, given below, to search through the methods for setSalary. Then use invoke(object, 1000.0) to set the salary to 1000.0. Then find the method getSalary – display what it returns (it should be 1000.0).

```
    private static Method find(Method[] methods, String what) {
        for (Method m: methods) {
            if (m.toString().contains(what)) {
                return m;
            }
        }
        return null;
    }
```