Java Reflection API

The goal of this survey is to identify the perception of applications developers about the Java Reflection API. It consists of four parts:

i. Participant Background (3 questions)
ii. Java Compilers and IDEs (5 questions)
iii. The Java Reflection API (6 questions)
iv. Additional Comments (3 questions)

Answering the survey should take around 5-7 minutes of your time. All the data collected from the survey is anonymous. The results of the survey may be reported in academic publications. If you have any questions or concerns, please contact <blinded>.

Thanks,

<blinded>

* Required

Participant Background

In this section, we will ask questions about your profile.

1. **How many years of Java programming experience do you have?** *

   Mark only one oval.

   ○ I don't have experience developing Java applications *After the last question in this section, stop filling out this form.*

   ○ Less than one year

   ○ 1-3 years

   ○ 4-6 years

   ○ 7-10 years

   ○ More than 10 years

2. **Rate your background/knowledge about the Java Reflection API.** *

   Mark only one oval.

   ○ Not knowledgeable - I do not know anything about it *After the last question in this section, stop filling out this form.*

   ○ Somewhat knowledgeable - I have a vague idea about it

   ○ Knowledgeable - I am familiar with it

   ○ Very knowledgeable - I know all/most classes and methods of it

3. **How often do you need to use the Java Reflection API in your software applications?** *

Mark only one oval.

○ Never *Stop filling out this form.*

○ Sometimes - I need reflection for less than 33% of the software applications I develop

○ Occasionally - I use reflection in more than 33% but less than 66% of the software applications I develop

○ Frequently - I need reflection for more than 66% of the software applications I develop

Java Compilers and IDEs

In this section, we present questions about Java compilers.

4. **Which compiler do you use when developing Java applications?** *

Mark only one oval.

○ Oracle javac

○ It does not matter

○ Eclipse for Java Compiler (ECJ)

○ OpenJDK javac

○ Other: ______________________

5. **Which IDE do you use when developing Java programs?** *

Mark only one oval.

○ Eclipse

○ Intellij IDEA

○ NetBeans

○ Other: ______________________

6. **Do you use the same Java compiler during development and during deployment proccess?** *

Mark only one oval.

○ I don' know

○ Yes

○ No

○ It does not matter

7. **Why?** *

____

____

____

____

____

8. **It is common to see Eclipse and IntelliJ IDEA being used in the same Java development team.** *

Mark only one oval.

○ Strongly disagree

○ Disagree

○ Neither disagree nor agree

○ Agree

○ Strongly agree

The Java Reflection API

In this section, we present questions about the Java Reflection API.

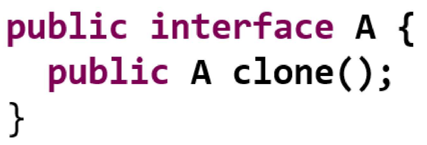
9. **Do you know what a synthetic field or method is in the Java Reflection API context?** *

Mark only one oval.

○ No

○ Yes

Class.getDeclaredMethods() returns an array containing Method objects reflecting all the declared methods of the class or interface represented by this Class object, including public, protected, default (package) access, and private methods, but excluding inherited methods.



```
public interface A {
  public A clone();
}
```

10. **What is the result of getDeclaredMethods for interface "A"?** *

Mark only one oval.

○ No declared methods

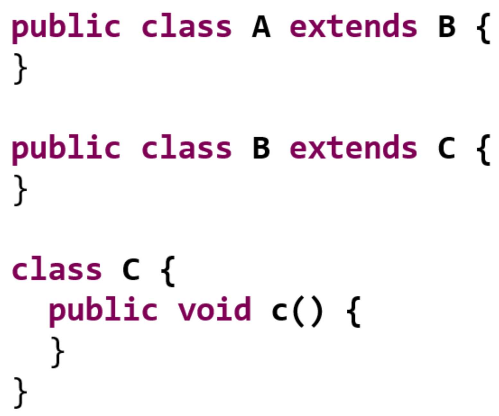
○ public abstract A A.clone() and public default Object A.clone()

○ public default Object A.clone()

○ public abstract A A.clone()

○ Other: ______________________________

Class.getMethod(String methodName) returns a Method object that reflects the specified public member method of the class or interface represented by this Class object.



```
public class A extends B {
}

public class B extends C {
}

class C {
  public void c() {
  }
}
```

11. **What is the result of invoking getMethod("c") for class "A"?** *

Mark only one oval.

○ public void A.c

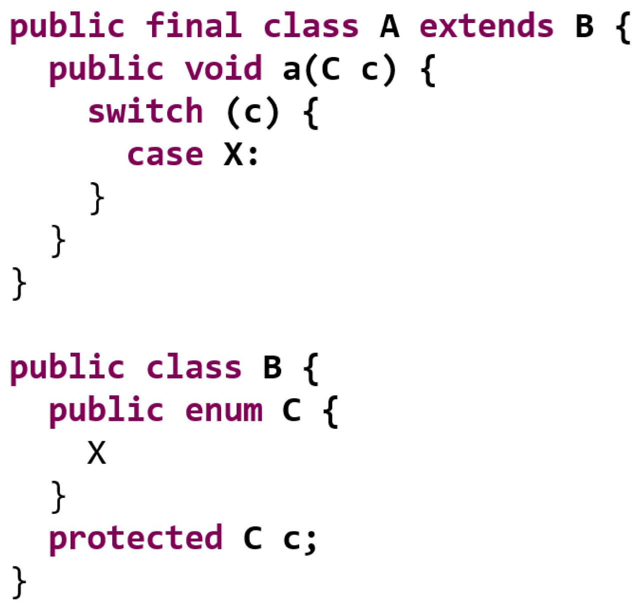
○ public void Object.c

○ public void B.c

○ public void C.c

○ Other: ______________________________

Class.getDeclaredFields() returns an array of Field objects reflecting all the fields declared by the class or interface represented by this Class object. This includes public, protected, default (package) access, and private fields, but excludes inherited fields.



```
public final class A extends B {
  public void a(C c) {
    switch (c) {
      case X:
    }
  }
}

public class B {
  public enum C {
    X
  }
  protected C c;
}
```

12. **What is the result of invoking getDeclaredFields for class "A"?** *

Mark only one oval.

○ No declared fields

○ private static int[] A.$SWITCH_TABLE$B$C

○ protected int[] A.$SWITCH_TABLE

○ protected C B.c

○ Other: ______________________________

13. **Do you think that the Java Reflection API methods can present different results when using binaries of a program compiled using two different Java compilers?** *

Mark only one oval.

○ I do not know *Skip to question 15.*

○ No *Skip to question 15.*

○ Yes

14. **What do you do to handle different results presented by the Java Reflection API?** *

____

____

____

____

____

Additional Comments

15. **Please, let us know if you have any additional comments about the Java Reflection API.**

____

____

____

____

____

16. **Do you have any doubt or suggestion?**

____

____

____

____

____

17. **If you would like to receive the results of our survey, please leave your email address.**

____