CSL202-Assignment-4

CSL202 | Assignment#4 | 12 Mar 2014 | Due on 20 Mar 2014

Important Instructions

1. Plagiarism and sharing code will lead to F grade (and possibly disciplinary action for repeat cases). You are responsible for ensuring that you code is not copied by others.
2. Please document the code properly; there is partial credit for clean and well written code.
3. Your score for an assignment solution will be assessed by looking at:
    1. Implementation approach and its correctness.
    2. Readability of the code and associated manual/readme etc.
    3. Correct functioning of the code. That is, it should produce correct results for various input scenarios.
    4. Quality of your design and code. For example, if changing one input value requires you to recompile your program then it is a bad design.

Description

We need to design and implement a program which scans a given Java .class file and produces the list of APIs used in that code. API usage here means the use of functions from various classes etc. A Java .class files is generated by the Java compiler as an output of compiling a Java source file. A .class file contains the bytecode.

You should make use of the `javap` program (included in the Java SDK) to determine the fully qualified names of the APIs being used in the input file (Java compiled .class files).

Example

For the .class file generated from the following input Java source (see listing 1 below), your program should print the output in a tabular form as below:

S. No.	Class	Method	Occurances
1.	java.util.Random	nextInt(int)	1
2.	java.lang.StringBuilder	append(char)	1
3.	java.lang.StringBuilder	toString()	1

4.	java.io.PrintStream	println(String)	1

Listing 1: Input code for you program will be the compiled version of this Java code:

```
package org.csl202;

import java.util.Random;

public class MyWorker {

        private static Random rand = new Random();
        public static void main(String[] args) {
                System.out.println("Value:
"+generateRandomString(10));
        }

        private static String generateRandomString(int length) {
                StringBuilder sb = new StringBuilder();
                for (int i=0; i<length; i++) {
                        int ascii = 65 + rand.nextInt(26);
                        sb.append((char)ascii);
                }
                return sb.toString();
        }

}
```

Output produced by `javap` for the class file corresponding to the above Java source code is shown in listing 2 below.

Listing 2:
theuser@goldfish:~/workspace2/CLS202A4/bin$ javap -c -classpath . org.csl202.MyWorker

```
Compiled from "MyWorker.java"
public class org.csl202.MyWorker {
  static {};
    Code:
       0: new         #10             // class java/util/Random
       3: dup
       4: invokespecial #12               // Method java/util/Random."<init>":()V
       7: putstatic     #15             // Field rand:Ljava/util/Random;
      10: return
```

```
  public org.csl202.MyWorker();
    Code:
      0: aload_0
      1: invokespecial #19                 // Method java/lang/Object."<init>":()V
      4: return

  public static void main(java.lang.String[]);
    Code:
      0: getstatic     #24                 // Field java/lang/System.out:Ljava/io/PrintStream;
      3: new           #30                 // class java/lang/StringBuilder
      6: dup
      7: ldc           #32                 // String Value:
      9: invokespecial #34                 // Method java/lang/StringBuilder."<init>":
(Ljava/lang/String;)V
     12: bipush        10
     14: invokestatic  #37                 // Method generateRandomString:(I)Ljava/lang/String;
     17: invokevirtual #41                 // Method java/lang/StringBuilder.append:
(Ljava/lang/String;)Ljava/lang/StringBuilder;
     20: invokevirtual #45                 // Method java/lang/StringBuilder.toString:
()Ljava/lang/String;
     23: invokevirtual #49                 // Method java/io/PrintStream.println:(Ljava/lang/String;)V
     26: return
}
```

theuser@goldfish:~/workspace2/CLS202A4/bin$