Instituto Superior Técnico

Programação Avançada

Second Exam – 27/6/2013

Number: ______________________________

Name: ____

Write your number on every page. Your answers should not be longer than the available space. You can use the other side of the page for drafts. The exam has **5** pages and the duration is **2.0 hours**. The grade for each question is written in parenthesis. Good luck.

1. (1.0) Explain and exemplify the concept of *computational meta-system*.

2. (1.0) Explain and exemplify the concept of *reflective system*.

3. (1.0) Explain and exemplify the concept of *reification*.

4. (1.0) There are many languages providing introspection mechanisms but only some of them also provide intercession mechanisms. Why? Explain.

5. (1.0) The Lisp language invented the *backquote* syntax. For what purpose? Explain.

6. (1.0) What is the purpose of Javassist? Explain.

7. (1.0) The following fragment of code is part of a larger program that uses Javassist. Explain what the code does and its ultimate purpose.

```
static void mystery(CtClass ctClass, CtMethod ctMethod)
    throws NotFoundException, CannotCompileException {
    CtField ctField =
        CtField.make("static java.util.Hashtable cachedResults = " +
                     "    new java.util.Hashtable();",
                     ctClass);
    ctClass.addField(ctField);
    String name = ctMethod.getName();
    ctMethod.setName(name + "$original");
    ctMethod = CtNewMethod.copy(ctMethod, name, ctClass, null);
    ctMethod.setBody("{" +
                     "  Object result = cachedResults.get($1);" +
                     "  if (result == null) {" +
                     "    result = " + name + "$original($$);" +
                     "    cachedResults.put($1, result);" +
                     "  }" +
                     "  return ($r)result;" +
                     "}");
    ctClass.addMethod(ctMethod);
}
```

8. (1.0) Regarding method calls, Java uses *dynamic dispatch* for the receiver and *static dispatch* for the arguments. What are the advantages and disadvantages of this approach when compared to *multiple dispatch*? Explain.

9. (2.0) Describe the problems caused by *Cross-cutting Concerns*.

10. (2.0) Define the concepts of *Join Point*, *Pointcut*, *Advice*, and *Inter-Type Declaration*.

11. (1.0) In the context of CLOS, explain the concept of *effective method*.

12. (1.0) CLOS provides the concept of *metaclass*. Explain its purpose and responsibilities.

13. (1.0) The following Racket form defines the logical conjunction using a syntax that is more similar to what exists in many languages such as C, Java, Javascript, etc:

```
(define (&& x y) (and x y))
```

Can we use `&&` just like we used `and`? Explain.

14. (1.0) The meta-circular evaluator implemented in this course provided *macros*. Explain this concept.

15. (1.0) What is the difference between *direct style* and *continuation-passing style*? Explain.

16. (1.0) Suppose that you invented a tree-based data structure and you want to provide an iterator for its leaves. Do you prefer to provide an *internal iterator* or an *external interator*? Why?

17. (1.0) The solution of the quadratic equation $ax^2 + bx + c = 0$ is described by the formula

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

where the symbol $\pm$ represents, in fact, an operation with two possible results. Assuming that the `amb` operator is available, define the equivalent `+-` function in Racket.

18. (1.0) **Julia** is a recently proposed programming language that supports higher-order functions. As an example, the following fragment defines the `map` of a function over a list.

```
map(f, list::Nil) = list
map(f, list::Cons) = cons(f(head(list)), map(f, tail(list)))
```

Do you think Julia provides different namespaces for functions and variables or, instead, a common namespace? Explain.