Laboratório de Programação II
Primeira Prova — 2012

1. (2.0 pontos) No exemplo de herança, com as classes `Progressão`, `Aritmética`, etc, faria sentido usar um iterador (*Iterator*)? Como fazer isso?

2. (3.0 pontos) Determine se cada afirmação abaixo é verdadeira ou falsa, ***justificando.***

   (a) O operador `new` sempre aloca memória.

   (b) Em polimorfismo, a versão do método chamado é determinado *apenas* pelos tipos dos argumentos.

   (c) Um construtor nunca deve chamar outro construtor da mesma classe, para evitar o risco de recursão infinita.

   (d) Não faz sentido uma classe derivada conter um método que aparece em sua classe mãe como `private`.

   (e) Uma classe pode implementar diversas `interfaces` ao mesmo tempo, pois `interfaces` não possuem métodos implementados.

   (f) Todos os métodos declarados em uma `interface` são públicos.

3. (1.0 ponto) O construtor quase sempre recebe um parâmetro oculto, que se refere ao objeto chamado (`this`). Mas isto nem sempre é verdade, em qual situação a variável `this` não é criada? Por quê?

4. (1.0 ponto) Explique a diferença entre composição e herança.

5. (3.0 pontos) Escreva uma classe `Tempo` que armazena internamente uma quantidade inteira de segundos. Nenhum objeto deve manipular esta variável interna diretamente.

   (a) Escreva métodos para obter o número de horas, minutos e segundos completos contidos no valor armazenado. `Tempos`. Veja um exemplo de utilização:

```
import java.io.*;

class Main {
    public static void main(String[] args) {
        Tempo t = new Tempo(1729);
        System.out.println(t.horas());
        System.out.println(t.minutos());
        System.out.println(t.segundos());
    }
}
```

   Este código deve imprimir os valores **0**, **28** e **49** (já que $1729 = 49 + 60 \times 28 + 3600 \times 0$)

   (b) Escreva uma nova classe `TempoSomável` que extende `Tempo` permitindo a soma com outra variável `Tempo` e soma com inteiros representando minutos.

Divirta-se e Boa Sorte!

Clareza e concisão contam pontos