

1ª Avaliação Individual – 2016.1

Instruções:

- Todos os códigos-fonte produzidos, exceto arquivos produtos da compilação (ex: .class), devem ser enviados em um único arquivo .zip para sandro.andrade@gmail.com ao final da avaliação.
- O e-mail deve obrigatoriamente ter o subject: INF011-20161-P1.

Questão 1) (4,0) Implemente o N-ton, variação do pattern Singleton que permite a criação de, no máximo, N diferentes instâncias de uma classe. A implementação deve ser realizada de modo que o código abaixo funcione conforme esperado.

```
Nton.initialize(3);           // No máximo três instâncias serão criadas
Nton i1 = Nton.getInstance(); // Obtém a 1ª instância
Nton i2 = Nton.getInstance(); // Obtém a 2ª instância
Nton i3 = Nton.getInstance(); // Obtém a 3ª instância
Nton i4 = Nton.getInstance(); // Obtém a 1ª instância
if (i4 == i1) System.out.println("OK"); // Deve exibir "OK"
Nton i5 = Nton.getInstance(); // Obtém a 2ª instância
if (i5 == i2) System.out.println("OK"); // Deve exibir "OK"
```

OBS: note que a partir da (N+1)^a instância, o sistema deve retornar as instâncias já criadas em formato de rodízio, ou seja, na 4^a invocação do *getInstance()* será retornada a 1^a instância, na 5^a a 2^a, na 6^a a 3^a, na 7^a a 1^a, e assim sucessivamente.

- (1,0) – implementação correta do método *initialize*
- (1,0) – implementação correta do armazenamento das múltiplas instâncias
- (2,0) – implementação correta do método *getInstance*

Questão 2) (6,0) Implemente uma fábrica flexível que usa um registro de Prototypes para flexibilizar os produtos criados. A implementação deve ser realizada de modo que o código abaixo funcione conforme esperado.

```
FlexibleFactory ff = new FlexibleFactory();
ff.addPrototype("Pneu", new Pneu());
ff.addPrototype("Motor", new Motor());
Pneu p = (Pneu) ff.create("Pneu");
Motor m = (Motor) ff.create("Motor");
if (ff.create("Chassi") == null) System.out.println("NULL"); // Deve exibir NULL
```

- (2,0) – implementação correta da API de gerenciamento dos Prototypes
- (2,0) – implementação correta do método *create*
- (2,0) – implementação correta dos produtos

Boa sorte !