Guilherme de Souza Rodrigues – 1610147 – P1 – POOS3 – 12/05/17

1) (1,0) Analise as afirmações a seguir e assinale V caso a afirmação seja verdadeira e F caso a afirmação seja falsa:

I. ( F ) Variáveis estáticas em uma classe não são inicializadas automaticamente pela Máquina Virtual Java.

II. ( F ) Membros *final* de uma classe não inicializados em sua declaração devem ser inicializados em, pelo menos, um dos construtores da classe.

III. ( F ) O construtor deve possuir o mesmo nome da classe, pode especificar parâmetros e também pode especificar tipo de retorno.

IV. ( F ) Java permite que vários métodos com o mesmo nome sejam definidos. Para isso, basta que os métodos possuam tipos de retorno diferentes.

V. ( V ) O coletor de lixo fica sob controle da Máquina Virtual Java. É possível definir o instante em que ele será executado invocando o método System.gc().

VI. ( F ) Os construtores de superclasse não são herdados por subclasses.

VII. ( V ) Um relacionamento “tem um” é implementado via herança.

VIII. ( V ) Quando uma subclasse define um método utilizando a mesma assinatura de algum método da superclasse, diz-se que a subclasse sobrecarrega esse método da superclasse.

IX. ( F ) Suponha a existência de um método x() (método sem parâmetros) definido na classe Pessoa. Suponha que a classe PessoaFisica estenda a classe Pessoa e defina um método x(int a) (método com parâmetro inteiro). Nesse caso, existe uma sobrecarga de métodos.

X. ( V ) Considere a existência de uma classe chamada Pessoa e outra classe chamada PessoaFisica, que estende a classe Pessoa. Suponha a existência de um método x(), sem parâmetros, na classe Pessoa cujo tipo de retorno seja Pessoa, e um método x(), sem parâmetros, na classe PessoaFisica cujo tipo de retorno seja PessoaFisica. Nesse caso, existe sobreposição de métodos e todas as regras para se alcançar a sobreposição estão sendo respeitadas.

2) (1,0) Preencha as lacunas de cada uma das sentenças a seguir:

a. Os objetos têm a propriedade de \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ - embora objetos possam saber comunicar entre si por interfaces bem definidas, normalmente não têm permissão de saber como outros objetos são implementados.

b. Os programadores Java concentram-se na criação de classes, que contêm atributos e o conjunto de métodos que manipulam esses atributos e fornecem serviços para clientes.

c. O projeto orientado a objetos tira proveito de relacionamento de herança, em que novas classes de objetos são derivadas absorvendo características de classes existentes e, em seguida, adicionando características únicas dessas mesmas classes.

d. O tamanho, a forma, a cor e o peso de um objeto são considerados exemplos de atributos da classe do objeto.

e. Herança é uma forma de capacidade de reutilização de software em que novas classes adquirem os membros de classes existentes e aprimoram essas classes com novas capacidades.

f. Os membros default e protected de uma superclasse podem ser acessados (ter visibilidade) pela superclasse e também pelas subclasses.

g. Em um relacionamento \_\_\_\_\_\_\_\_\_ ou \_\_\_\_\_\_\_\_\_, um objeto de uma subclasse também pode ser tratado como um objeto de sua superclasse.

h. Em um relacionamento \_\_\_\_\_\_\_\_\_ ou \_\_\_\_\_\_\_\_, um objeto de classe tem referências a objetos de outras classes como membros.

i. Quando um objeto de uma subclasse é instanciado, um construtor de superclasse é chamado implicitamente ou explicitamente.

j. Os construtores de subclasse podem chamar construtores de superclasse via palavra-chave super.