UNIDADE 4 – PROGRAMAÇÃO ORIENTADA A OBJETOS EM JAVA (PARTE II)

1.[FCC - 2012 - TRE/CE] Sobre orientação a objetos, é INCORRETO afirmar:

(a)os conceitos de generalização e especialização da orientação a objetos estão diretamente associados ao conceito de herança
(b)um objeto pode existir mesmo que não exista nenhum evento a ele associado
(c)um construtor visa inicializar os atributos e pode ser executado automaticamente sempre que um novo objeto é criado
(d)polimorfismo é o princípio pelo qual duas ou mais classes derivadas de uma mesma superclasse podem invocar métodos que têm a mesma assinatura e mesmo comportamento
(e)uma classe define o comportamento dos objetos através de seus métodos, e quais estados ele é capaz de manter através de seus atributos

2.[CESGRANRIO - 2012 - PETROBRÁS] Ao escrever o código da Classe PortaDeCofre em Java para que ela atenda a interface Porta, como um programador deve começar a declaração da classe?

(a)public class Porta:PortadeCofre {
(b)public class PortadeCofre :: Porta {
(c)public class PortadeCofre inherits Porta {
(d)public class PortadeCofre extends Porta {
(e)public class PortadeCofre implements Porta {

3.[FCC - 2012 - TCE/SP] Em um programa Java, considere a existência de uma variável do tipo long chamada cod contendo o valor 1234. Para passar o valor contido nessa variável para uma variável do tipo byte chamada codNovo, deve-se fazer casting. Para isso, utiliza-se a instrução: byte codNovo =

(a)Byte.valueOf(cod);
(b)(long) cod;
(c)Byte.pasreByte(cod);
(d)(byte) cod;
(e)(cast) cod;

4.[FCC - 2011 - TRT/RS] Existem circunstâncias onde métodos específicos não devem ser implementados em uma classe, porém apenas fornecidas suas especificações. O emprego de um modificador em Java permite a declaração de um protótipo (método sem bloco de código), adiando sua implementação para subclasses. Trata-se do modificador

(a)interface
(b)object
(c)abstract
(d)upcasting
(e)downcasting

5.[CESGRANRIO - 2010 PETROBRÁS] Ao tentar compilar e executar o código abaixo, o resultado será

```
1  package javaapplication2;
2
3  abstract class A {
4      public A() {
5          System.out.print("A");
6      }
7      public abstract void metodo();
8  }
9
10 class B extends A {
11     public void metodo() {
12         System.out.print("B");
13     }
14 }
15
16 class C extends A {
17     public void metodo() {
18         System.out.print("C");
19     }
20 }
21
22 public class Main {
23     public static void main(String[] args) {
24         A obj=new B();
25         obj.metodo();
26         obj=new C();
27         obj.metodo();
28     }
29 }
```

(a)a correta compilação e execução do código, com a exibição na saída padrão da sequência BC
(b)a correta compilação e execução do código, com a exibição na saída padrão da sequência ABAC
(c)um erro de compilação, pois A é uma classe abstrata e não pode ter instâncias, como obj
(d)um erro de compilação, pois obj é da classe A e tentase instanciá-lo como sendo um objeto da classe B
(e)um erro de execução, pois uma vez feito o binding de obj com a classe B, não se pode mudar a classe do mesmo

6.[CESGRANRIO - 2010 PETROBRÁS] Analisando o código abaixo, verifica-se que o programa

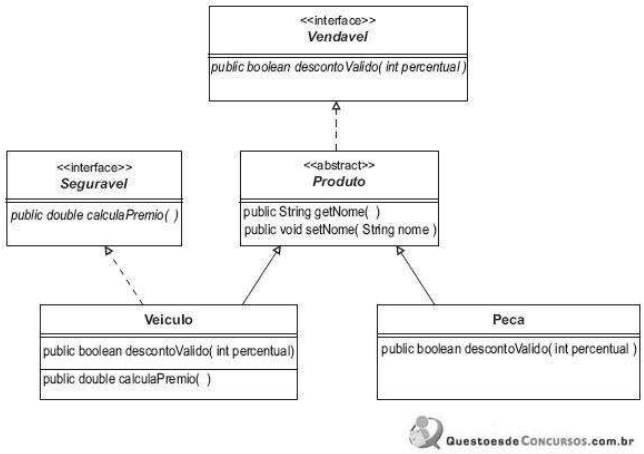

```
abstract class C1 {
    void f() {
        System.out.println("C1");
    }
}

class C2 extends C1 {
    void f() {
        System.out.println("C2");
    }
}

class C3 extends C1 {
    void f() {
        System.out.println("C3");
    }
}

public class Main {
    public static void main(String[] args) {
        C1 a, b, c[];
        a = new C2();
        b = new C3();
        c = new C1[] {a,b};
        for(int i=0;i<c.length;i++) {
            c[i].f();
        }
    }
}
```

(a)compila e executa imprimindo na saída padrão C1 duas vezes
(b)compila e executa imprimindo na saída padrão C2 e C3
(c)não compila, pois classes abstratas não podem ser instanciadas
(d)não compila, pois há incompatibilidade de tipos em atribuição
(e)não compila, pois um vetor foi construído de forma incorreta

7.[CESGRANRIO - 2010 BNDES] Qual das afirmações a seguir faz uma apreciação correta a respeito da linguagem de programação Java?

(a)O conceito de herança múltipla é implementado nativamente
(b)Uma classe pode implementar somente uma interface ao mesmo tempo
(c)Uma classe pode implementar uma interface ou ser subclasse de outra classe qualquer, mas não ambos simultaneamente
(d)A construção de um método que pode levantar uma exceção, cuja instância é uma subclasse de java.lang.RuntimeException, não exige tratamento obrigatório por parte do programador dentro daquele método
(e)Objetos da classe java.lang.String têm comportamento otimizado para permitir que seu valor seja alterado sempre que necessário, liberando imediatamente a memória usada pelo conteúdo anterior

8.[CESGRANRIO - 2008 PETROBRÁS] Com base no diagrama abaixo, analise os trechos de código Java a seguir.




```
I -  Produto p = new Produto();
     p.setNome("Carro");

II -  Seguravel s = new Veiculo();
      s.setNome("Carro");
      double p = s.calculaPremio();

III -  Seguravel s = new Veiculo();
       ((Veiculo) s).setNome("Carro");

IV -  Vendavel v = new Peca();
      Produto p = (Produto) v;
      p.setNome("Pneu");

V -  Vendavel v = new Veiculo();
     Produto p = (Peca)((Produto)((Veiculo) v));
```

Estão corretos APENAS os trechos de código

(a)I e III
(b)I e IV
(c)II e III
(d)III e IV
(e)IV e V

9.[CESGRANRIO - 2008 CAPES] Em que porção da JVM (Java Virtual Machine) são armazenados objetos instanciados em um programa JAVA?

(a)Heap
(b)GUnit
(c)Stack Pool
(d)Dump Buffer
(e)Text Segment

10.[CESGRANRIO- 2006 DECEA] Em Java, a palavra-chave que implementa uma relação de herança de classes é:

(a)isFatherOf
(b)isChildOf
(c)inherits
(d)derives
(e)extends

GABARITO

1 - D; 2 - E; 3 - D; 4 - C; 5 - B; 6 - B; 7 - D; 8 - D; 9 - A; 10 - E