



Lista de Exercícios 5 – Polimorfismo e Agregação

1- Dada a seguinte definição de classe:

```
public class Derivada extends Base {  
    int r,s,t;  
    public Derivada (int x, int y){  
        super(x,y);  
        r=x; s=y;  
    }  
    public Derivada (int x){  
        r=x;  
    }  
}
```

Quais as assinaturas dos construtores que devem existir na classe Base ? Justifique.

2- Suponha que duas classes sejam definidas como

```
class Professor extends Funcionario  
class Coordenador extends Professor
```

e duas variáveis são definidas:

```
Professor professor = new Professor();  
Object obj = professor;
```

Para cada atribuição abaixo, indique se há ou não erro, justificando a resposta.

- a) Funcionario func = obj;
- b) Funcionario func = (Funcionario) obj;
- c) Professor prof = (Professor) obj;

3- Para que membros de uma classe base possam ser usados por membros da classe derivada, eles devem ser:

- a) public;
- b) protected;
- c) private;
- d) visibilidade default;
- e) todas as anteriores.

4- Leia o trecho de código abaixo e responda o que se pede:

```
class A{
    void m1(){ System.out.println("A.m1");}
}
class B extends A{
    void m1(){ System.out.println("B.m1");}
    static void m1(String s){ System.out.println(s + ",");}
}
class Programa{
    public static void main (String[]args){
        B.m1("main");
        new B().m1();
    }
}
```

Ao tentar compilar e rodar, o programa:

- a) mostra: main,B.m1;
- b) dá erro de compilação;
- c) dá erro de execução;
- d) n.d.a.

5- Leia o trecho de código abaixo e responda o que se pede:

```
class A{
    static void m1(String s1){}
    void m1(char c){}
    void m1(boolean b1){}
    static void m1(byte b){}
}
class B extends A{
    static void m1(String s1){} //1
    void m1(char c){} //2
    static void m1(boolean b1){} //3
    void m1(byte b){} //4
}
```

Ao tentar compilar e rodar, o programa gera:

- a) Erro de compilação em 1;
- b) Erro de compilação em 2;
- c) Erro de compilação em 3;
- d) Erro de compilação em 4;
- e) N.D.A.

6- Leia o trecho de código abaixo e responda o que se pede:

```
class C {  
    protected void m1(String s1){}  
    protected void m1(char c){}  
    private static void m1(boolean b1){}  
    public static void m1(byte b){}  
}  
class D extends C {  
    public void m1(String s1){} //1  
    void m1(char c){} //2  
    static void m1(boolean b1){} //3  
    static void m1(byte b){} //4  
}
```

Ao tentar compilar e rodar, o programa gera:

- a) Erro de compilação em 1;
- b) Erro de compilação em 2;
- c) Erro de compilação em 3;
- d) Erro de compilação em 4;
- e) N.D.A.

7- Leia o trecho de código abaixo e responda o que se pede:

```
abstract class E {  
    void m1(String s1){} //1  
    abstract void m2(char c); //2  
}  
class F extends E {  
    void m1(String s1){} //3  
}
```

Ao tentar compilar e rodar, o programa gera:

- a) Erro de compilação em 1;
- b) Erro de compilação em 2;
- c) Erro de compilação em 3;
- d) Erro de execução;
- e) N.D.A.

8- Leia o trecho de código abaixo e responda o que se pede:

```
interface I {  
    void m1(String s1); //1  
    void m2(char c); //2  
}  
class C implements I {  
    void m1(String s1){} //3  
    protected void m2(char c){} //4  
}
```

Ao tentar compilar e rodar, o programa gera:

- a) Erro de compilação em 1;
- b) Erro de compilação em 2;
- c) Erro de compilação em 3;
- d) Erro de compilação em 4;
- e) Erro de execução;
- f) N.D.A.

9- Leia o trecho de código abaixo e responda o que se pede:

```
interface I {  
    void m1(String s1); //1  
    void m2(char c); //2  
    void m3(); //2  
}  
class C implements I {  
    public void m1(String s1){} //4  
    public void m2(char c){} //5  
}
```

Ao tentar compilar e rodar, o programa gera:

- g) Erro de compilação em 1;
- h) Erro de compilação em 2;
- i) Erro de compilação em 3;
- j) Erro de compilação em 4;
- k) Erro de compilação em 5;
- l) Erro de execução;
- m) N.D.A.

10- Crie a classe Veiculo (abstrata) contendo placa (String), chassi (long), ano (int), peso em quilos (int), velocMax em km/h (float), preço em R\$ (float) e capacidade em Kg (int). Inclua um construtor sem argumentos que inicializa os dados com seus respectivos valores defaults (padrão) e um construtor que inicializa os dados com os valores recebidos como argumento. Acrescente métodos set e get para mudar e acessar cada atributo da classe e um método exibir() para a visualização dos valores das variáveis de instância. Acrescente também um método para aumentar o preço do veículo, aumentarPreco (double aumento) e outro para reduzir o preço do veículo, reduzirPreco (double redução).

Crie uma interface Exibível que possui um único método chamado exibir(). Todas as classes que desejam exibir informações na tela devem implementar essa interface.

Crie a classe Motor. Inclua dados como potência e cilindrada. Inclua um método exibir() para a visualização dos valores das variáveis de instância.

Crie uma classe Peça. Inclua dados como código, descrição e marca e um método exibir() para a visualização dos valores das variáveis de instância. Acrescente uma variável de classe (estática) para contar o número de objetos da classe Peça criados, bem como um método estático para retornar seu valor.

A classe Veiculo deve agregar uma instância da classe Motor e diversas instâncias da classe Peça, estas últimas sendo guardadas em uma lista (ArrayList) específica de Peças. Crie métodos em Veiculo para incluir e excluir Peças e consultar por código na lista.

Crie a classe CarroPasseio usando a classe Veiculo como Base. Inclua cor (String) e modelo (String). Inclua um construtor sem argumentos que inicializa os dados com seus respectivos valores defaults (padrão) e um construtor que inicializa os dados dessa classe e da classe Base com os valores recebidos como argumento. Acrescente métodos set e get para mudar e acessar cada atributo da classe e um método exibir() para a visualização dos valores das variáveis de instância dessa classe e da classe Base.

Crie a classe Caminhão derivada da classe Veiculo. Inclua toneladas (float), alturaMax (float) e comprimento (float). Inclua um construtor sem argumentos que inicializa os dados com seus respectivos valores defaults (padrão) e um construtor que inicializa os dados dessa classe e da classe Base com os valores recebidos como argumento. Acrescente métodos set e get para mudar e acessar cada atributo da classe e um método exibir() para a visualização dos valores das variáveis de instância dessa classe e da classe Base.

11- Faça um programa que cria um ArrayList do tipo Veículo. Inclua um laço que pergunta ao usuário sobre o tipo de veículo e use o operador new para criar objetos do tipo escolhido (CarroPasseio ou Caminhão). Quando o usuário terminar a entrada dos dados de todos os veículos mostre os resultados usando outro laço.

12- Faça um programa que cria duas Hashtable, uma do tipo CarroPasseio e outra do tipo Caminhao, onde as chaves das mesmas será o atributo placa. Inclua um laço que pergunta ao usuário sobre o tipo de veículo e use o operador new para criar objetos do tipo escolhido (CarroPasseio ou Caminhão). Quando o usuário terminar a entrada dos dados de todos os veículos realize buscas nas Hashtables.

13- Desenhe o diagrama de classes relativo às questões 10 e 11.

14- Escreva uma classe Frota que é uma agregação de Veículos, utilizando uma lista dinamicamente redimensionável de Veículos (ArrayList<Veiculo>). Acrescente métodos para:

- incluir veículos na frota,
- excluir a partir da placa,
- excluir a partir do chassi,
- excluir vários a partir do ano,
- consultar a partir da placa,
- consultar a partir do chassi,
- aumentar o preço de um veículo da lista a partir da placa – aumentarPreco(placa, aumento) – chamar o método aumentarPreço(aumento) do veículo que tem o parâmetro placa,
- aumentar o preço de um veículo da lista a partir do chassi – idem,
- reduzir o preço de um veículo da lista a partir do chassi – idem, reduzirPreco(chassi, redução),
- reduzir o preço de um veículo da lista a partir da placa – idem,
- listar todos os dados de todos os veículos da frota, chamando os métodos exibir() de cada veículo da lista.