

Gabarito proposto

P₁ – Introdução à POO (INE5603) – 03set2014
Sistemas de Informação – Universidade Federal de Santa Catarina

Estudante: _____

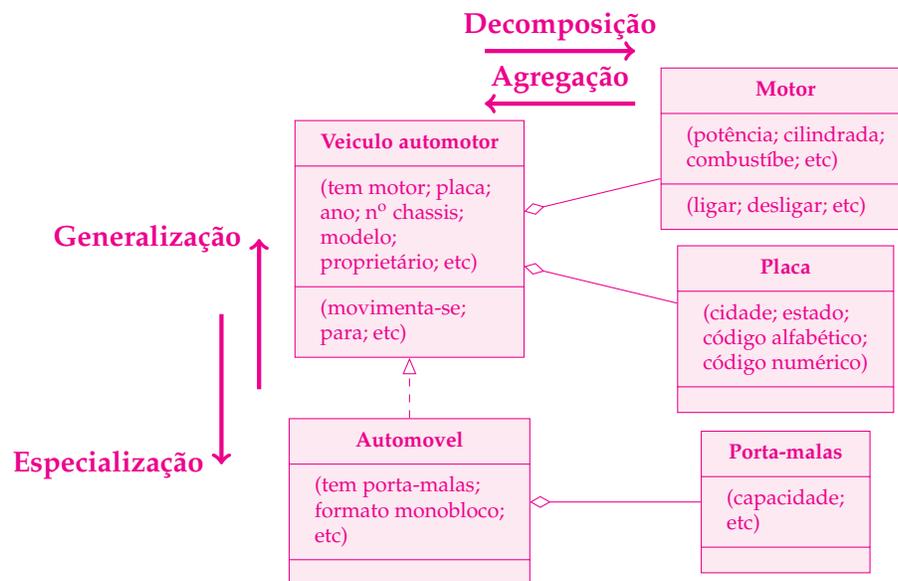
1. (1,0) Caracterize **classe** e **objetos**. Exemplifique.

Classe é uma abstração de atributos e serviços comum a um grupo de objetos. Exemplo: Abstração de livro. Objeto é uma instância de uma classe, ou seja, um exemplo real e utilizável de uma classe. Exemplo: Um exemplar físico de um livro.

2. (1,0) Explique as seguintes operações de abstração:

- **Classificação/Instanciação** *Classificação define a que classe um objeto pertence. Instanciação define (control) um objeto pertencente a uma dada classe.*
- **Generalização/Especialização** *Generalização define uma classe mais genérica, com características comuns a um conjunto de classes. Especialização identifica uma ou mais subclasses com características adicionais em relação à classe mais geral.*
- **Agregação/Decomposição** *Agregação é caracterizada por “é composto por” outros objetos. Decomposição é caracterizado por “é parte de” de um objeto. Tais operações também são chamadas de diagrama “todo-parte”.*
- **Associação** *Generalização é uma descrição genérica de uma ou mais ligações entre entidades independentes.*

3. (2,0) Escolha um pequeno problema qualquer, faça sua abstração e modelagem em classes, especifique os atributos e métodos de cada classe, e aplique **ao menos duas operações de abstração** da questão 2.



4. (2,0) Dadas as classes a seguir, em Java, para a implementação da fórmula de Bhaskara:

```

1 import java.lang.Math;
2
3 class Raizes {
4     protected double x1;
5     protected double x2;
6     public Raizes(double ax1, double ax2) {
7         x1 = ax1;
8         x2 = ax2;
9     }
10    public double retornaX1() {
11        return x1;
12    }
13    public double retornaX2() {
14        return x2;
15    }
16 }
17
18 class FormulaBhaskara {
19     protected Raizes r;
20     public FormulaBhaskara(double a, double b,
21         double c) {
22         double delta = b*b - 4*a*c;
23         double deltaraz = Math.sqrt(delta);
24         double ax1, ax2;
25         ax1 = (-b + deltaraz) / (2*a);
26         ax2 = (-b - deltaraz) / (2*a);
27         r = new Raizes(ax1, ax2);
28     }
29     public Raizes retornaRaizes() {
30         return r;
31     }
32 }

```

Pede-se:

- (a) (1,0) Explique o relacionamento existente entre `Raizes` e `FormulaBhaskara` e a troca de mensagens existente no construtor de `FormulaBhaskara`.

A classe `FormulaBhaskara` utiliza o atributo `r` da classe `Raizes` (composição) para registrar as duas possíveis raízes da fórmula. A troca de mensagens ocorre entre o método construtor da `FormulaBhaskara` e o construtor da classe `Raizes`, especificamente na instânciação "`r = new Raizes(ax1, ax2);`", onde os argumento `ax1` e `ax2` são enviados e, como resposta, é obtido o objeto `r`.

- (b) (0,5) Indique os possíveis erros em tempo de execução deste programa.

Se `a` for igual a zero, ocorrerá uma divisão por zero não permitida. Se `delta` for negativo, o cálculo da raiz não é determinada dentro conjunto dos reais.

- (c) (0,5) Determine as saídas na tela para o seguinte trecho de código:

```

1     public static void main(String[] args) {
2         FormulaBhaskara r = new FormulaBhaskara(3.0, -2.0, -1.0);
3         System.out.println("x1 = " + r.retornaRaizes().retornaX1());
4         System.out.println("x2 = " + r.retornaRaizes().retornaX2());
5     }

```

x1 = 1.0

x2 = -0.3333333333333333

5. (2,0) Mostre o valor de cada variável após as seguintes atribuições em Java:

```

1 int a, b;
2 double c, d;
3 boolean e, f;
4
5 a = 5;                               System.out.println(a); // a = _____
6
7 a += 2;                               System.out.println(a); // a = _____
8
9 b = --a + 5;                          System.out.println(b); // b = _____
10
11 b = 2 * b / a;                        System.out.println(b); // b = _____
12
13 c = 0.5e2;                            System.out.println(c); // c = _____

```

```

14
15 c /= 2; System.out.println(c); // c = _____
16
17 d = a / b; System.out.println(d); // d = _____
18
19 d = a + c / (b + 2) * d; System.out.println(d); // d = _____
20
21 e = (a > b) && (c > d); System.out.println(e); // e = _____
22
23 f = (c % 2 == 0) || (2 * b == a); System.out.println(f); // f = _____

```

a = 5 a = 7 b = 11 b = 3 c = 50.0 c = 25.0 d = 2.0 d = 16.0 e = true f = true

6. (2,0) Escreva uma classe, em Java, para representar uma **progressão aritmética** (PA). Exemplo: 1, 4, 7, 10, 13, ... é uma PA com primeiro termo igual a 1 e razão (diferença entre números consecutivos) igual a 3.

Considere os seguintes atributos:

- a_1 é o primeiro termo
- r é a razão

E implemente os seguintes métodos, onde n deve um parâmetro de entrada:

- cálculo do n -ésimo termo:

$$a_n = a_1 + (n - 1) \cdot r$$

- soma dos n primeiros termos:

$$S_n = \frac{n \cdot (a_1 + a_n)}{2}$$

```

class PA {
    protected double a1;
    protected double r;
    public PA(double va1, double vr) {
        a1 = va1;
        r = vr;
    }
    public double nesimo(int n) {
        double an;
        an = a1 + (n - 1) * r;
        return an;
    }
    public double soma(int n) {
        double Sn;
        double an;
        an = nesimo(n);
        Sn = n * (a1 + an) / 2;
        return Sn;
    }
}

public class PAteste {
    public static void main(String[] args) {
        PA pa;
        pa = new PA(4,3);
        //4, 7, 10, 13, ...
        System.out.println("an = " +
            pa.nesimo(4));
        //an = 13
        System.out.println("Sn = " +
            pa.soma(4));
        //Sn = 4+7+10+13 = 34
    }
}

```

Boa prova!