

Lista 1 – Introdução à POO (INE5603) – 2017s2
Sistemas de Informação – Universidade Federal de Santa Catarina

1. Sobre conceitos preliminares:
 - (a) Explique e exemplifique cada operação de abstração:
 - Classificação/Instanciação
 - Generalização/Especialização
 - Agregação/Decomposição
 - Associação
 - (b) Defina classe e objetos. Caracterize suas diferenças. Exemplifique.
 - (c) Caracterize os elementos envolvidos no envio de mensagem.
2. Faça a representação visual da classe de um cliente de banco, seus atributos e métodos. Traduza sua modelagem para a linguagem Java (não é necessária a implementação dos métodos, apenas suas definições).
3. Defina instanciação em uma linguagem orientada a objetos?
4. Explique a finalidade do método de mesmo nome da classe? O que ocorre quando há vários destes métodos com assinaturas distintas no momento da instanciação de um objeto?
5. Descreva o processo de compilação (em *bytecode*) e interpretação (pela Máquina Virtual Java) de código em Java. Diferencie erro de compilação e erro de execução de programas.
6. Por que é desejável a separação de aspectos de interface das demais classes?
7. Indique o valor verdade das seguintes expressões, escritas em Java, para as variáveis booleanas A, B, C, D, E e F:
 - (a) `A = (1 > 2) // exemplo: A=false`
 - (b) `B = (8 == 8) // exemplo: B=true`
 - (c) `C = ((12 - 5) > 6)`
 - (d) `D = (0 < 3) and (8 < 9)`
 - (e) `E = ((i++) > i)`
 - (f) `F = ((10 * 90 / 50 - 2) == 16)`
8. Escreva, em Java, trechos de códigos para representar as seguintes expressões matemáticas. Defina um valor qualquer para cada variável e mostre o resultado com `System.out.println`. Utilize `Math.sqrt(x)` para \sqrt{x} , ou `Math.pow(x, a)` para x^a
 - (a) $a - b(c + d^2)/e$
 - (b) x^y
 - (c) $m * k^n$
 - (d) $\sqrt{\frac{1}{a+b}}$
9. Inclua linhas de comando para imprimir os conteúdos das variáveis desejadas do código abaixo. Entenda cada atribuição, expressão e saída gerada:

```

1 | int a,b,c;
2 | a = -125;
3 | b = 2003;
4 | long x, y = 96;
5 | x = 14L;
6 | byte b1 = 127;
7 | byte b2 = -128;
8 | short s1;
9 | s1 = Short.MAX_VALUE;
10 | short s2 = Short.MIN_VALUE;
11 | System.out.println("s1 = "+s1);
12 | float valorSalario, outra;
13 | valorSalario = 240.00f;
14 | outra = 3.056E-5f;
15 | System.out.println("Float valorSalario= "+
16 |     valorSalario+" \tFloat outra="+outra);
17 | double d1 = 2.56e2;
18 | double d2, d3;
19 | d2 = 24.56d;
20 | d3 = 40.50;
21 | char letra;
22 | letra = 'c';
23 | boolean adulto, crianca;
24 | int idade = 21;
25 | adulto = idade > 18;
26 | crianca = false;
27 | System.out.println("Adulto = "+adulto+" \nCrianca
28 |     = "+crianca);
29 | // int,byte,short,long - podem ser representados
30 | // em base 8 e base 16
31 | int a = 010; // Base 8 --> representacao deve
32 | // iniciar por zero
33 | int b = 0xA; // Base 16 --> representacao deve
34 | // iniciar por 0x ou 0X
35 | int c = a+b;
36 | System.out.println("Soma "+a+" com "+b+" = "+c);

```

10. Modele uma classe (sugestão: utilize BlueJ) para representar um retângulo no espaço, alinhado com os eixos do plano cartesiano (ou seja, sem rotação), implementando os seguintes atributos e métodos:

Atributos

- Coordenada do canto esquerdo superior;
- Coordenada do canto direito inferior.

Métodos

- Definição da coordenada do canto esquerdo superior;
- Definição da coordenada do canto direito inferior;
- Informação da coordenada x mais à esquerda;
- Informação da coordenada x mais à direita;
- Informação da coordenada y mais acima;
- Informação da coordenada y mais abaixo;
- (opcional) Determinação da altura;
- (opcional) Determinação da largura;
- (opcional) Determinação da área.

Sugestão de estruturação do código:

```

1 | class Retangulo {
2 |
3 |     protected double x1, y1; //ponto esquerdo-superior
4 |     protected double x2, y2; //ponto direito-inferior
5 |
6 |     public Retangulo() {
7 |         // COLOCAR SEU CODIGO AQUI...
8 |     }
9 |     public void defineCoordenadaEsquerdaSuperior(double x, double y) {
10 |         // COLOCAR SEU CODIGO AQUI...
11 |     }
12 |     public void defineCoordenadaDireitaInferior(double x, double y) {

```

```

13     // COLOCAR SEU CODIGO AQUI...
14 }
15 public double informaXmaisEsquerda() {
16     // COLOCAR SEU CODIGO AQUI...
17 }
18 public double informaXmaisDireita() {
19     // COLOCAR SEU CODIGO AQUI...
20 }
21 public double informaYmaisAcima() {
22     // COLOCAR SEU CODIGO AQUI...
23 }
24 public double informaXmaisAbaixo() {
25     // COLOCAR SEU CODIGO AQUI...
26 }
27 }

```

Procure alterá-lo para usar um objeto ponto (como [Ponto2D](#)).

11. Sobre o código em Java a seguir para a representação de um ponto no espaço bidimensional:

| | |
|--|--|
| <pre> 1 import java.lang.Math; 2 3 class Ponto2D { 4 protected double x, y; //atributos 5 public Ponto2D() { 6 this.x = 0.0; 7 this.y = 0.0; 8 } 9 public void atualizaCoordenada(double x_, 10 double y_) { 11 this.x = x_; 12 this.y = y_; 13 } 14 public double distanciaOrigem() { 15 double d; 16 d = Math.sqrt(this.x*this.x + this. 17 y*this.y); 18 return d; 19 } </pre> | <pre> 18 } 19 20 public class PontoEx { 21 public static void main(String[] args) { 22 Ponto2D P; 23 P = new Ponto2D(); 24 P.atualizaCoordenada(3.0, 4.0); 25 double d = P.distanciaOrigem(); 26 System.out.println("Distancia: "+d); 27 } 28 } </pre> |
|--|--|

(a) Explique os métodos [Ponto2D](#), [atualizaCoordenada](#) e [distanciaOrigem](#)?

(b) Qual o significado do comando da linha 23?

(c) Qual o valor das coordenadas do ponto **P** após a execução da linha 23 e da linha 24?

12. Crie uma classe Java que tenha métodos para calcular a áreas ou volumes. Considere que as variáveis necessárias são disponibilizadas por argumentos/parâmetros:

(a) Triângulo: $A = \frac{base \cdot altura}{2}$

(b) Trapézio: $A = altura \cdot \frac{base_menor \cdot base_maior}{2}$

(c) Paralelepípedo: $V = comprimento \cdot largura \cdot altura$

(d) Cilindro: $V = \pi \cdot raio^2 \cdot altura$

13. Dadas as seguintes informações de um funcionário: nome, idade, cargo e seu salário bruto. Considere que:

(a) O funcionário recebe uma gratificação de 20% sobre o salário bruto.

(b) Há um desconto de 15% sobre o salário total.

Escreva uma classe `Funcionario` que tenha os métodos necessários para executar um programa que imprime o nome, o salário bruto e o salário líquido de um funcionário.

14. Crie uma classe Java que tenha métodos para realizar a conversão de:

- (a) Conversão cm/pol – o método deve receber um valor em centímetros por parâmetro e deve retornar o seu valor correspondente em polegadas. Uma polegada equivale a 2,54 centímetros.
- (b) Conversão Celsius/Fahrenheit – o método deve receber um valor de temperatura em graus Celsius (de 0 a 100) e deve retornar o valor correspondente em Fahrenheit (de 32 a 212). Seguem as fórmulas:

$$C = \frac{5}{9}(F - 32) \qquad F = \frac{9}{5}C + 32$$

- (c) Conversão Real/Dólar – o método deve receber um valor em reais e deve retornar o valor correspondente em dólares. Para isto, considere que há um atributo que registra a cotação atual, e implemente um método que a atualize.

15. Escreva um método, em Java, que calcule a quantidade de litros de combustível gasta em uma viagem por um automóvel. Deve-se ter, como parâmetros, o tempo gasto na viagem (*TEMPO*), a velocidade média (*VELOCIDADE*) e o consumo médio (por exemplo, 12km por litro) (*CONSUMO*). A distância percorrida é:

$$DISTANCIA = TEMPO \cdot VELOCIDADE$$

Tendo o valor da distância, a quantidade de litros de combustível utilizada na viagem é:

$$LITROS_COMBUSTIVEL = \frac{DISTANCIA}{CONSUMO}$$

O programa deve apresentar a quantidade de litros de combustível aproximadamente utilizada na viagem.