Lista de Exercícios

Lista I

Programação Orientada a Objetos

Data de entrega: 29 - 07 - 2015

1) Defina programação orientada a objetos, diferenciando da programação estruturada.

É um paradigma de programação onde os elementos da vida real ou não são representados como classes e objetos, onde esses objetos tem vários atributos de tipos mais simples e métodos que representam as ações que o elemento modelado pode realizar.

Esse método de modelagem possibilita entre outras coisas o encapsulamento do código e herança entre classes. Diferentemente da programação estruturada onde o código é mais fragmentado. As variáveis e métodos muitas vezes são projetados para aquela aplicação.

2) Defina: classe, objeto, método e atributo.

3) Considerando os tipos de dados nativos em Java, escolha o tipo de dado ou classe mais

adequada para representar:

* O número de municípios de um estado do Brasil. int
* O nome de um estado do Brasil. String
* A população de um estado do Brasil. int
* A área do Brasil em quilômetros quadrados. float
* A população total do mundo. long int
* O CEP de um endereço no Brasil. int
* O nome de uma rua em um endereço no Brasil. String
* A altura de uma pessoa em metros. float
* O peso de uma pessoa em quilos. float
* A temperatura corporal de uma pessoa. float
* O sexo de uma pessoa. boolean
* A altura de uma pessoa em milímetros. float

4) Crie uma classe DVD, que tenha como características: marca, modelo, peso, se é slim ou

não, e o preço.

5) Quais os conceitos de orientação a objetos você consegue identificar no código abaixo

e onde:

**public** **class** Q1{ //Declaração de uma classe.

**private** **int** x,y; //Atributos da classe.

**public** Q1(**int** a, **int** b){ //Construtor com argumentos.

**if**(a < b){

x = a;

y = b;

}**else** {

x = b;

y = a;

}

}

**public** Q1(){ //Construtor sem argumentos(polimorfismo).

x = y = 0;

}

**public** **void** setX(**int** a){ //Metodos para setar os atributos privados.

x = a;

}

**public** **void** setY(**int** a){

y = a;

}

// ...

}

6) Descreva em no máximo 200 palavras o que é um automóvel e o que ele faz. Liste os

substantivos e verbos separadamente. Cada substantivo corresponde a um objeto que

precisará ser construído para implementar um sistema, nesse caso, um carro. Selecione

5 dos objetos que você listou e, para cada um, liste vários atributos e comportamentos.

Descreva brevemente como esses objetos interagem entre si e com outros objetos na

sua descrição. Estes passos que você seguiu são típicos do projeto orientado a objetos.

7) A fim de representar empregados em uma firma, crie uma classe chamada Empregado

que inclui as três informações a seguir como atributos:

* um primeiro nome,
* um sobrenome, e
* um salário mensal.

Sua classe deve ter um construtor que inicializa os três atributos. Forneça um método

set e get para cada atributo.

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

**public** **class** Empregado {

String primeiro\_nome;

String sobrenome;

**float** salário\_mensal;

**public** String getPrimeiro\_nome() {

**return** primeiro\_nome;

}

**public** **void** setPrimeiro\_nome(String primeiro\_nome) {

**this**.primeiro\_nome = primeiro\_nome;

}

**public** String getSobrenome() {

**return** sobrenome;

}

**public** **void** setSobrenome(String sobrenome) {

**this**.sobrenome = sobrenome;

}

**public** **float** getSalário\_mensal() {

**return** salário\_mensal;

}

**public** **void** setSalário\_mensal(**float** salário\_mensal) {

**this**.salário\_mensal = salário\_mensal;

}

**public** Empregado(String primeiro\_nome, String sobrenome,

**float** salário\_mensal) {

**super**();

**this**.primeiro\_nome = primeiro\_nome;

**this**.sobrenome = sobrenome;

**this**.salário\_mensal = salário\_mensal;

}

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

8) Represente, em Java, a seguinte afirmação: "A classe B é composta pelos objetos o1 e

o2 que pertencem, respectivamente, às classes A e C".

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

**public** **class** A{

**int** atributo1;

**char** atributo2;

**public** A(**int** atributo1,**char** atributo2){

**super**();

**this**.atributo1 = atributo1;

**this**.atributo2 = atributo2;

}

}

**public** **class** C{

String atributo1;

**char** atributo2;

**public** C(String atributo1,**char** atributo2){

**super**();

**this**.atributo1 = atributo1;

**this**.atributo2 = atributo2;

}

}

**public** **class** B{

A atributo1;

C atributo2;

**public** C(A atributo1,C atributo2){

**super**();

**this**.atributo1 = **new** A(atributo1);

**this**.atributo2 = **new** C(atributo2);

}

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

9) Considere a classe A abaixo. Implemente o método "m" conforme as seguintes regras:

o método retorna true se o valor proposto no parâmetro de m for maior que zero. O

método retorna false se o valor proposto no parâmetro de m for menor ou igual à zero.

Se o método retornar true então ele deve, também, alterar o valor do atributo a para 10 vezes o valor proposto no parâmetro do método m.

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

**import** java.util.\*;

**public** **class** A {

**static** **int** *a*;

**public** A() {

*a*=0;

}

**public** **static** **void** main(String[] args) {

**int** aux;

Scanner sc = **new** Scanner(System.***in***);

System.***out***.println("Entre com um inteiro");//Saida de dados.

aux = sc.nextInt();

**if**(*m*(aux)){

*a* = aux\*10;

System.***out***.println("a = "+*a*);

}

**else**{

System.***out***.println("Valor invalido!");

}

}

**public** **static** **boolean** m(**int** valor){

**if**(valor>0) **return** **true**;

**else** **return** **false**;

}

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

10) Qual é o valor da variável x após a invocação do método "m" dentro do método "t"?

10