



Fundação CECIERJ - Vice Presidência de Educação Superior a Distância

Curso de Tecnologia em Sistemas de Computação

Disciplina: Programação Orientada a Objetos

AP1 1º semestre de 2017.

Nome –

Assinatura –

Observações:

1. Prova sem consulta e sem uso de máquina de calcular.
 2. Use caneta para preencher o seu nome e assinar nas folhas de questões e nas folhas de respostas.
 3. Você pode usar lápis para responder as questões.
 4. Ao final da prova devolva as folhas de questões e as de respostas.
 5. Todas as respostas devem ser transcritas nas folhas de respostas. As respostas nas folhas de questões não serão corrigidas.
-

Questão 1) (4.0 pontos)

Implemente a classe concreta Data, que representa uma data. Esta classe deve:

- a) Armazenar a data em três atributos distintos de tipo inteiro primitivo, um para o dia, um para o mês e um para o ano. O valor assumido pelo atributo dia deve ir de 1 ao número máximo de dias do mês, o valor assumido pelo atributo mês deve ir de 1 a 12 e o valor assumido pelo atributo ano deve ser maior ou igual a 1900. Para simplificar a verificação de consistência, desconsidere a existência de anos bissextos.
- b) Declarar e implementar um construtor padrão que inicializa a data com o dia 1 de janeiro de 1900.
- c) Declarar e implementar um construtor de inicialização que recebe como argumentos os valores inteiros primitivos para o dia, mês e ano para o qual a data deve ser inicializada.
- d) Declarar e implementar o método void incrementar(), que incrementa a data armazenada em um dia. Fique atento para garantir que os atributos sejam atualizados de forma consistente.
- e) Sobrescrever o método String toString() para que retorne uma instância de String no formato “DD de MMMM de AAAA”, onde DD é a dia com um ou dois dígitos, MMMM é o nome do mês por extenso e AAAA é o ano com quatro dígitos.

Observação: não é permitida a inclusão de atributos além dos solicitados.

RESPOSTA:

```
final class Meses{
static final String[] NOME_DO_MES = {"janeiro", "fevereiro",
"março", "abril", "maio", "junho", "julho", "agosto", "setembro",
"outubro", "novembro", "dezembro"};
}
```

```
final class UltimosDiasMeses{
static final int[] DIAS_DO_MES = {31, 28, 31, 30, 31, 30, 31,
31, 30, 31, 30, 31};
}
```

//LETRA (a)

```
class Data {
int dia;
int mes;
int ano;
```

//LETRA (b)

```
public Data() { this(1, 1, 1900); }
```

//LETRA (c)

```
public Data(int dia, int mes, int ano) {
if ((ano < 1900) || (mes < 1 || 12 < mes) || (dia < 1 ||
UltimosDiasMeses.DIAS_DO_MES[mes - 1] < dia)) return;
this.dia = dia;
this.mes = mes;
this.ano = ano;
}
```

```
public int getDia() { return this.dia; }
```

```
public int getMes() { return this.mes; }
```

```
public int getAno() { return this.ano; }
```

//LETRA (e)

```
public void incrementar() {
this.dia++;
if (this.dia > UltimosDiasMeses.DIAS_DO_MES[this.mes - 1]) {
this.dia = 1; this.mes++;
if (this.mes > 12) {
this.mes = 1; this.ano++;
}
}
}
```

//LETRA (d)

```
public String toString() {
return this.dia + " de " + Meses.NOME_DO_MES[this.mes - 1] +
" de " + this.ano;
}
}
```

```
public class AP1_Q1_2017_1{
public static void main(String[] args){
```

```

        Data d = new Data(Integer.parseInt(args[0]), Integer.parseInt(
args[1]), Integer.parseInt(args[2]));
        System.out.println(d);
        d.incrementar();
        System.out.println(d);
    }
}

```

Questão 2) (3.0 pontos) Considere as classes abaixo referente a um sistema de inventário:

```

class Apto {
    int codigo, numQuartos;
    double metragem, preco;
    public Apto(int codigo, int numQuartos, double metragem, double preco) {
        this.codigo = codigo;
        this.numQuartos = numQuartos;
        this.metragem = metragem;
        this.preco = preco;
    }
}
class Carro {
    int codigo;
    String nome;
    int ano;
    double valor;
    public Carro(int codigo, String nome, int ano, double valor) {
        this.codigo = codigo;
        this.nome = nome;
        this.ano = ano;
        this.valor = valor;
    }
}

```

- Crie uma classe Principal e instancie um objeto para cada classe fornecida.
- Considere que queiramos somar os valores dos bens no cálculo de um inventário. Adicione os objetos (item a) a um vetor e calcule a soma destes objetos usando uma única chamada de método (método polimórfico). Altere o mínimo possível as classe Apto e Carro de forma que essa implementação seja possível. Adicione qualquer outra classe ou método, se for necessário.

RESPOSTA:

```

// item b)
interface Bem {
    double retornaValor();
}

class Apto /* item b) */ implements Bem {
    int codigo, numQuartos;
    double metragem, preco;

    public Apto(int codigo, int numQuartos, double metragem, double preco) {
        this.codigo = codigo;
        this.numQuartos = numQuartos;
        this.metragem = metragem;
    }
}

```

```

        this.preco = preco;
    }

    // item b)
    public double retornaValor() {
        return preco;
    }
}

class Carro /* item b) */ implements Bem {
    int codigo;
    String nome;
    int ano;
    double valor;

    public Carro(int codigo, String nome, int ano, double valor) {
        this.codigo = codigo;
        this.nome = nome;
        this.ano = ano;
        this.valor = valor;
    }

    // item b)
    public double retornaValor() {
        return valor;
    }
}

public class AP1_2017_1_Q2 {
    public static void main(String[] args) {
        //item a)
        Apto apto = new Apto(1, 3, 100, 500);
        Carro carro = new Carro(2, "Lamborghini", 2016, 1000);

        //item b)
        Bem[] shapes = new Bem[2];
        shapes[0] = apto;
        shapes[1] = carro;
        double soma = 0;
        for (int i=0; i<shapes.length; i++)
            soma = soma + shapes[i].retornaValor();
        System.out.println("O valor da soma é: " + soma);
    }
}

```

Questão 3) (3.0 pontos) Dada a classe abaixo que representa um ponto em 2 dimensões,

```

class Ponto {
    private double x, y;
    public Ponto(double x, double y) {
        this.x = x;
        this.y = y;
    }
}

```

defina uma classe Ponto3D que permita a criação de um ponto em 3 dimensões, ou seja XYZ. Além disso, implemente:

- a) Dado um outro ponto como argumento (um outro objeto Ponto3D ou as coordenadas x, y e z deste outro ponto), retorne o objeto Ponto3D referente à diferença entre as coordenadas.
- b) Calcule a distância entre 2 pontos. Supondo ponto P com dimensões px, py, pz, Q com dimensões qx, qy e qz, a distância é calculada com a seguinte fórmula:
$$\text{distancia} = \text{raiz_quadrada} ((px - qx)^2 + (py - qy)^2 + (pz - qz)^2)$$

Obs.: 1) Utilize os conceitos de OO vistos sempre que possível; 2) A raiz quadrada pode ser calculada com o método Math.sqrt()

RESPOSTA:

```
class Ponto {
    ...

    // Necessário para acessar os campos na classe Ponto3D
    public double getX() { return x; };
    public double getY() { return y; };
}

class Ponto3D extends Ponto {
    private double z;

    public Ponto3D(double x, double y, double z) {
        super(x, y);
        this.z = z;
    }

    public double getZ() { return z; };

    // item a) OU ...
    public Ponto3D diferenca (Ponto3D p) {
        return new Ponto3D (p.getX() - this.getX(), p.getY() - this.getY(),
p.getZ() - this.getZ());
    }
    // ... item a)
    public Ponto3D diferenca (double x, double y, double z) {
        return new Ponto3D (x - this.getX(), y - this.getY(), z - this.getZ());
    }
    // item b)
    public double distancia (Ponto3D p) {
        return Math.sqrt(Math.pow(p.getX() - this.getX(), 2) +
Math.pow(p.getY() - this.getY(), 2) +
Math.pow(p.getZ() - this.getZ(), 2));
    }
}
```