



Fundação CECIERJ - Vice Presidência de Educação Superior a Distância

**Curso de Tecnologia em Sistemas de Computação**

**AD1 de Programação III**

**2º semestre de 2010**

**Nome:**

**Matrícula:**

**Pólo:**

**Exercício (ENTREGAR OS ARQUIVOS EM MÍDIA, PARA FINS DE TESTE, JUNTAMENTE COM A AD IMPRESSA):**

Em um supermercado, alguns itens podem ser comprados avulsos (como 1kg de arroz, feijão, óleo, entre outros) enquanto outros (como cestas básicas e 5kg de arroz) são montados a partir de itens avulsos e/ou itens montados. Por exemplo, uma cesta de natal pode ser formada por uma cesta básica, um item montado, seguido de guloseimas de natal, tais como nozes, castanhas, panetone, vinho, entre outros.

Todos os itens têm um código sequencial que os identifica univocamente. Qualquer item, independente de ser um item avulso ou um item montado, deve ser capaz de informar qual é o seu nome e o seu preço. O preço de um item de compra é calculado da seguinte forma:

- 1) se o item é avulso, o preço dele é o valor passado, como parâmetro de entrada, pelo construtor;
- 2) se o item é um item montado, o preço dele é calculado pela soma dos preços de suas partes.

Projete a classe item avulso e a classe item montado. Pense em uma solução genérica que possa servir para qualquer produto do supermercado que possa ser montado a partir de outros já existentes. Você pode usar para o programa de teste a seguir para testar seu programa.

```
class Teste {
    public static void main (String[] args) {
        /* Cria diversos itens avulsos */
        item a1 = new item("Arroz de 1kg", 1.5F);
        item f1 = new item("Feijao de 1kg", 2.5F);
        item o1 = new item("Oleo", 3.5F);
        item acu1 = new item("Acucar de 1kg", 2.0F);
        item sal1 = new item("Sal de 1kg", 1.0F);

        /* Cria diversos itens montados */
        item a5 = new Montado("Arroz de 5kg");
        ((Montado) a5).inclui(a1);
        ((Montado) a5).inclui(a1);
```

```

        ((Montado) a5).inclui(a1);
        ((Montado) a5).inclui(a1);
        ((Montado) a5).inclui(a1);

        item acu5 = new Montado("Acucar de 5kg");
        ((Montado) acu5).inclui(acu1);
        ((Montado) acu5).inclui(acu1);
        ((Montado) acu5).inclui(acu1);
        ((Montado) acu5).inclui(acu1);
        ((Montado) acu5).inclui(acu1);

        item cb1 = new Montado("Cesta basica simples");
        ((Montado) cb1).inclui(a1);
        ((Montado) cb1).inclui(f1);
        ((Montado) cb1).inclui(o1);
        ((Montado) cb1).inclui(acu1);
        ((Montado) cb1).inclui(sal1);

        item cb2 = new Montado("Cesta basica complexa");
        ((Montado) cb2).inclui(a5);
        ((Montado) cb2).inclui(f1);
        ((Montado) cb2).inclui(o1);
        ((Montado) cb2).inclui(acu5);
        ((Montado) cb2).inclui(sal1);

        /* informar todos os produtos que fazem parte do item Montado */
        System.out.println (a5);
        System.out.println (acu5);
        System.out.println (cb1);
        System.out.println (cb2);
    }
}

```

### RESPOSTA:

```

class item{
    private int cod;
    private static int proxCod=0;
    private float preco;
    private String desc;
    item(){ this.cod=++proxCod; }

    item(String d){
        this();
        desc = d;
    }

    item(String d, float p){
        this(d);
        preco = p;
    }

    float obterPreco(){ return preco; }
    int obterCodigo(){ return this.cod; };
    public String toString(){ return "item: " + desc + " (" + cod + ")="
R$" + preco; }
}

```

```

class Montado extends item{
    private final int MAX = 100;
    private item[] partes;

    Montado(String d){
        super(d, 0.0F);
        partes = new item[MAX];
    }

    void inclui(item e){
        int i;
        for(i = 0; i < partes.length; i++) if(partes[i] == null) break;
        if(i < partes.length) partes[i] = e;
        else return;
    }

    float obtemPreco () {
        float total=0;
        for(int i = 0; i < partes.length; i++)
            if(partes[i] != null) total += partes[i].obtemPreco();
        return total;
    }

    public String toString() {
        String texto="Montado composto de:\n";
        for (int i = 0; i < partes.length; i++)
            if(partes[i] != null) texto = texto + partes[i].toString() + "\n";
        return texto;
    }
}

```