

Fundação CECIERJ - Vice Presidência de Educação Superior a Distância  
Curso de Tecnologia em Sistemas de Computação  
AD1 de Programação OO  
2º semestre de 2019

**EXERCÍCIO (ENTREGAR OS ARQUIVOS EM MÍDIA, PARA FINS DE TESTE):**

Suponha que sua empresa tenha sido contratada para desenvolver um software para gerenciar uma rede de tintas. Os produtos nessa rede podem ser simples (pincel, broxa, tintas prontas, entre outros) e compostos (tintas misturadas em máquinas, com suas bases e seus respectivos corantes, e kit pintura, como, por exemplo, pincel, rolo e bandeja. É importante salientar que existem vários kits pintura nessa rede). Cada produto tem um código único que o identifica nessa rede. Escreva um programa em Java, usando os conceitos de orientação a objetos e sem utilizar coleções, para gerenciar essa rede. Seu programa deve ser capaz de:

- imprimir os vários tipos de produtos, precificando-os. Se o produto é simples, seu preço é obtido de maneira trivial. Se o produto é composto, seu programa deve calcular o seu preço por meio da soma dos preços dos demais produtos;
- gerar um orçamento para um cliente; e
- implementar um carrinho de compras, caso o cliente queira obter os produtos pela internet.

**RESPOSTA:**

```
abstract class Item{
    private int cod;
    private static int proxCod=0;
    protected float preco;

    Item(){ this.cod = ++proxCod; }
    abstract float getPreco();
    int getCodigo(){ return this.cod; }
    public String toString(){ return "(" + cod + ")"; }
}
```

```
class Pincel extends Item{
    int tam;
    Pincel(int t, float p){ tam = t; preco = p; }
    float getPreco () { return this.preco; }
```

```

        public String toString(){
            return "Pincel [" + tam + " cm] " + super.toString() + ": R$ " +
getPreco() + "\n";
        }
    }
}

```

```

class Broxa extends Item{
    Broxa(float p){ preco = p; }
    float getPreco () { return this.preco; }
    public String toString(){
        return "Broxa " + super.toString() + ": R$ " + getPreco()+ "\n";
    }
}

```

```

class Bandeja extends Item{
    Bandeja(float p){ preco = p; }
    float getPreco () { return this.preco; }
    public String toString(){
        return "Bandeja " + super.toString()+ ": R$ " + getPreco()+ "\n";
    }
}

```

```

class Rolo extends Item{
    int tam;
    Rolo(int t, float p){ tam = t; preco = p; }
    float getPreco () { return this.preco; }
    public String toString(){
        return "Rolo [" + tam + " cm] " + super.toString() + ": R$ " +
getPreco() + "\n";
    }
}

```

```

class Tinta extends Item{
    int cor;
    Tinta(int c, float p){ cor = c; preco = p; }
    float getPreco () { return this.preco; }
    public String toString(){
        return "Tinta [" + cor + "]" + super.toString() + ": R$ " +
getPreco() + "\n";
    }
}

```

```

class Corante extends Item{
    int cor;
    Corante(int c, float p){ cor = c; preco = p; }
    float getPreco () { return this.preco; }
    public String toString(){
        return "Corante [" + cor + "]" + super.toString() + ": R$ " +
getPreco() + "\n";
    }
}

```

```

class ItemComposto extends Item {
    int TAM_MAX = 5;
    private Item[] partes;
    private int pos_livre;

    ItemComposto () {
        partes = new Item[TAM_MAX];
        pos_livre = 0;
    }

    void inclui (Item e) {
        if(pos_livre == TAM_MAX){
            TAM_MAX *= 2;
            Item[] tmp = new Item[TAM_MAX];
            System.arraycopy (partes, 0, tmp, 0, Math.min (partes.length,
tmp.length));
            // o comando supracitado copia o array antigo sobre o novo
            partes = tmp; // isto desaloca o array antigo
            tmp = null;
        }
        partes[pos_livre++] = e;
    }

    float getPreco () {
        float total = 0;
        for (int i = 0; i < pos_livre; i++)
            total += partes[i].getPreco();
        return total;
    }

    public String toString() {
        String texto = super.toString() + ":\n";
        for (int i = 0; i < pos_livre; i++) texto += partes[i];
        return texto;
    }
}

class TintaMisturada extends ItemComposto {
    void inclui (Item e) {
        if ((e instanceof Tinta) || (e instanceof Corante))
            super.inclui(e);
    }

    public String toString() {
        return "Tinta Misturada [R$ " + getPreco() + "]" + " " +
super.toString();
    }
}

```

```

class KitPintura extends ItemComposto {
    void inclui (Item e) {
        if (!(e instanceof Tinta)) && (!(e instanceof Corante)))
            super.inclui(e);
    }

    public String toString() {
        return "KitPintura [R$ " + getPreco() + "]" + super.toString();
    }
}

class CarrinhoCompras {
    int TAM_MAX = 5;
    private Item[] partes;
    private int pos_livre;

    CarrinhoCompras () {
        partes = new Item[TAM_MAX];
        pos_livre = 0;
    }

    void inclui (Item e) {
        if(pos_livre == TAM_MAX){
            TAM_MAX *= 2;
            Item[] tmp = new Item[TAM_MAX];
            System.arraycopy (partes, 0, tmp, 0, Math.min (partes.length,
tmp.length));
            // o comando supracitado copia o array antigo sobre o novo
            partes = tmp; // isto desaloca o array antigo
            tmp = null;
        }
        partes[pos_livre++] = e;
    }

    float getPreco () {
        float total = 0;
        for (int i = 0; i < pos_livre; i++)
            total += partes[i].getPreco();
        return total;
    }

    public String toString() {
        String texto = "Carrinho de Compras [R$ " + getPreco() + "]\n";
        for (int i = 0; i < pos_livre; i++) texto += partes[i];
        return texto;
    }
}

```

```

public class AD1_P00_2019_2 {
    public static void main (String[] args) {
        Pincel p1 = new Pincel(5, 2.22F);
        Pincel p2 = new Pincel(10, 4.44F);
        Broxa b1 = new Broxa (10.99F);
        Broxa b2 = new Broxa (5.55F);
        Bandeja bj1 = new Bandeja(3.33F);
        Bandeja bj2 = new Bandeja(1.17F);
        Rolo r1 = new Rolo(5, 3.33F);
        Rolo r2 = new Rolo(9, 3.99F);
        Tinta t1 = new Tinta(1, 30.99F);
        Tinta t2 = new Tinta(2, 39.11F);
        Corante c1 = new Corante(1, 3.99F);
        Corante c2 = new Corante(2, 5.55F);
        KitPintura kp1 = new KitPintura();
        kp1.inclui(p1);
        kp1.inclui(p2);
        kp1.inclui(b1);
        kp1.inclui(b2);
        kp1.inclui(bj1);
        kp1.inclui(bj2);
        kp1.inclui(r1);
        kp1.inclui(r2);
        kp1.inclui(t1);
        kp1.inclui(t2);
        System.out.print(kp1);
        System.out.println();

        TintaMisturada tm1 = new TintaMisturada();
        tm1.inclui(p1);
        tm1.inclui(p2);
        tm1.inclui(t1);
        tm1.inclui(t2);
        tm1.inclui(c1);
        tm1.inclui(c2);
        System.out.print(tm1);
        System.out.println();

        CarrinhoCompras cc = new CarrinhoCompras();
        cc.inclui(kp1);
        cc.inclui(tm1);
        System.out.print(cc);
        System.out.println();
    }
}

```