



Fundação CECIERJ - Vice Presidência de Educação Superior a Distância

Curso de Tecnologia em Sistemas de Computação

AD1 de Programação III

2º semestre de 2009

Nome:

Matrícula:

Polo:

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

Em uma lanchonete hipotética, alguns produtos podem ser vendidos avulsos, como hambúrgueres, batatas fritas, refrigerantes, fatias de queijo, fatias de presunto, fatias de bacon, ovos, enquanto outros (como x-búrguer, x-tudo, x-bacon, x-egg, por exemplo) são feitos a partir de produtos avulsos.

Todos os produtos têm um código sequencial que os identifica univocamente. Já qualquer produto deve ser capaz de informar qual é o seu preço. O preço de um produto é calculado da seguinte forma:

- todos os produtos avulsos devem ter seus preços informados na criação do objeto; e
- produtos criados a partir de outros têm o seu preço calculado pela soma dos preços dos produtos que os compõem.

Para que os produtos possam ser vendidos pelo telefone, implemente um sistema de compras no qual o usuário possa escolher diferentes produtos (isto é, avulsos e compostos) e possa, a qualquer momento, obter o total gasto. Este sistema também deve permitir que o usuário especifique um valor limite de tempo (fornecido na criação do objeto) de quanto tempo o usuário pode esperar, além do tempo mínimo de confecção dos produtos, para que o pedido chegue em sua casa, resultando, obviamente, em descontos para o consumidor. Para cada minuto esperado além do tempo de manufatura, o consumidor ganha 0.5% (meio por cento) de desconto. Se o pedido chegar depois do tempo combinado por telefone, o usuário ganha 1% de desconto para cada minuto esperado.

Crie as classes **ProdutoAvulso**, **ProdutoComposto** e **SistemaCompras**. Implemente uma classe de teste chamada **Lanchonete** onde possam ser criados alguns produtos avulsos e compostos, além do sistema de compras por telefone. Para cada compra por telefone, você deve criar um objeto da classe **SistemaCompras**, referente a este pedido. A classe **Lanchonete** deverá oferecer operações que permitam:

- incluir um tipo de produto a ser comercializado pela lanchonete;
- exibir os tipos de produtos que ele vende;
- remover um tipo de produto;
- aumentar o preço de todos os tipos de produtos de um percentual; e
- fazer operações referentes a pedidos por telefone.

RESPOSTA:

```
public class Lanchonete{
    ProdutoAvulso[] itens;
    int pos_livre;
    final int TAM_MAX;

    Lanchonete (int tam){
        TAM_MAX = tam;
        itens = new ProdutoAvulso[tam];
        pos_livre = 0;
    }

    boolean pertence (ProdutoAvulso e){
        for(int i = 0; i < pos_livre; i++){
            if(e.equals(itens[i]))
                return true;
        }
        return false;
    }

    void inclui (ProdutoAvulso i){
        if ((pos_livre != TAM_MAX) && (!pertence(i))) itens[pos_livre++] = i;
    }

    void retira (ProdutoAvulso e){
        int i = 0;
        while(i < pos_livre){
            if(e.equals(itens[i])) break;
            i++;
        }

        if(i == pos_livre) return; //nao tem

        if(i == (pos_livre - 1)) //ultimo
            pos_livre--;
        else{
            itens[i] = itens[pos_livre - 1];
            pos_livre--;
        }
    }

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

    void aumenta(float p){
        for(int i = 0; i < pos_livre; i++){
            itens[i].aumentaPreco(p);
        }
    }

    public static void main (String[] args) {
        //iniciando avulsos
        Hamburguer h = new Hamburguer (2.5F);
        Batata_Frita bf = new Batata_Frita (4.5F);
        Refrigerante r = new Refrigerante (1.5F);
        Queijo q = new Queijo(0.5F);
        Presunto p = new Presunto(0.5F);
        Bacon b = new Bacon(0.5F);
        Ovo o = new Ovo(0.5F);
    }
}
```

```

//imprimino avulsos
System.out.println(h);
System.out.println(bf);
System.out.println(r);
System.out.println(q);
System.out.println(p);
System.out.println(b);
System.out.println(o);

//iniciando compostos
ProdutoComposto xtudo = new ProdutoComposto();
xtudo.inclui(h);
xtudo.inclui(bf);
xtudo.inclui(q);
xtudo.inclui(p);
xtudo.inclui(b);
xtudo.inclui(o);

ProdutoComposto xbacon = new ProdutoComposto();
xbacon.inclui(h);
xbacon.inclui(q);
xbacon.inclui(b);

ProdutoComposto xburguer = new ProdutoComposto();
xburguer.inclui(h);
xburguer.inclui(q);

ProdutoComposto xegg = new ProdutoComposto();
xegg.inclui(h);
xegg.inclui(q);
xegg.inclui(o);

//imprimindo compostos
System.out.println(xtudo);
System.out.println(xbacon);
System.out.println(xburguer);
System.out.println(xegg);

Lanchonete l = new Lanchonete(100);
l.inclui(h);
l.inclui(bf);
l.inclui(r);
l.inclui(q);
l.inclui(p);
l.inclui(b);
l.inclui(o);
l.inclui(xtudo);
l.inclui(xbacon);
l.inclui(xburguer);
l.inclui(xegg);

// imprime cardapio
System.out.println(l);

l.retira(xburguer);

// imprime cardapio
System.out.println(l);

//usando o carrinho de compras
SistemaCompras car = new SistemaCompras (20);
if (l.pertence(xtudo)) car.inclui(xtudo);
if (l.pertence(r)) car.inclui(r);

```

```

        if (l.pertence(xburguer)) car.inclui(xburguer);

        car.alteraTempoChegada(10);
        System.out.println(car);
        System.out.println(car.obtemPreco());

        //aumenta preco lanchonete
        l.retira(xtudo);
        l.retira(xbacon);
        l.retira(xburguer);
        l.retira(xegg);
        l.aumenta(0.1F);
        System.out.println(l);
    }
}

```

```

abstract class ProdutoAvulso {
    private int cod;
    private static int proxCod=0;

    ProdutoAvulso () {
        this.cod=++proxCod;
    }

    abstract float obtemPreco();

    abstract void aumentaPreco(float p);

    int obtemCodigo() { return this.cod; };

    public String toString() {
        return "(" + cod + ")";
    }
}

```

```

class Hamburguer extends ProdutoAvulso{
    private float preco;

    Hamburguer (float p){
        super();
        preco = p;
    }

    void aumentaPreco(float p){
        preco = (1 + p) * preco;
    }

    float obtemPreco () {
        return this.preco;
    }

    public String toString() {
        return "Hamburguer " + preco + " " + super.toString() + " ";
    }
}

```

```

class Batata_Frita extends ProdutoAvulso{
    private float preco;

    Batata_Frita (float p){
        super();
        preco = p;
    }

    void aumentaPreco(float p){
        preco = (1 + p) * preco;
    }

    float obtemPreco () {
        return this.preco;
    }

    public String toString() {
        return "Batata Frita " + preco + " " + super.toString() + " ";
    }
}

class Refrigerante extends ProdutoAvulso{
    private float preco;

    Refrigerante (float p){
        super();
        preco = p;
    }

    void aumentaPreco(float p){
        preco = (1 + p) * preco;
    }

    float obtemPreco () {
        return this.preco;
    }

    public String toString() {
        return "Refrigerante " + preco + " " + super.toString() + " ";
    }
}

class Queijo extends ProdutoAvulso{
    private float preco;

    Queijo (float p){
        super();
        preco = p;
    }

    void aumentaPreco(float p){
        preco = (1 + p) * preco;
    }

    float obtemPreco () {
        return this.preco;
    }

    public String toString() {
        return "Fatia de Queijo " + preco + " " + super.toString() + " ";
    }
}

```

```

class Presunto extends ProdutoAvulso{
    private float preco;

    Presunto (float p){
        super();
        preco = p;
    }

    void aumentaPreco(float p){
        preco = (1 + p) * preco;
    }

    float obtemPreco () {
        return this.preco;
    }

    public String toString() {
        return "Fatia de Presunto " + preco + " " + super.toString() + " ";
    }
}

class Bacon extends ProdutoAvulso{
    private float preco;

    Bacon (float p){
        super();
        preco = p;
    }

    void aumentaPreco(float p){
        preco = (1 + p) * preco;
    }

    float obtemPreco () {
        return this.preco;
    }

    public String toString() {
        return "Fatia de Bacon " + preco + " " + super.toString() + " ";
    }
}

class Ovo extends ProdutoAvulso{
    private float preco;

    Ovo (float p){
        super();
        preco = p;
    }

    void aumentaPreco(float p){
        preco = (1 + p) * preco;
    }

    float obtemPreco () {
        return this.preco;
    }

    public String toString() {
        return "Ovo " + preco + " " + super.toString() + " ";
    }
}

```

```

class ProdutoComposto extends ProdutoAvulso {
    private final int TAM_MAX = 100;
    private ProdutoAvulso[] partes;
    private int pos_livre;

    ProdutoComposto () {
        partes = new ProdutoAvulso[TAM_MAX];
        pos_livre = 0;
    }

    void inclui (ProdutoAvulso e) {
        if (pos_livre != TAM_MAX) partes[pos_livre++] = e;
    }

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

    void aumentaPreco (float p) {
        for (int i = 0; i < pos_livre; i++)
            partes[i].aumentaPreco(p);
    }

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

class SistemaCompras{
    private final int TAM_MAX = 100;
    private ProdutoAvulso[] partes;
    private int pos_livre;
    private int limite_tempo;
    private int tempo_alem;

    SistemaCompras (int lt) {
        partes = new ProdutoAvulso[TAM_MAX];
        pos_livre = 0;
        limite_tempo = lt;
        tempo_alem = 0;
    }

    void alteraTempoChegada (int t){
        tempo_alem = t;
    }

    void inclui (ProdutoAvulso e) {
        if (pos_livre != TAM_MAX) partes[pos_livre++] = e;
    }

    float obtemPreco () {
        float total=0;

        for (int i = 0; i < pos_livre; i++)
            total += partes[i].obtemPreco();
    }
}

```

```
        total = total - ((0.5F * limite_tempo * total)/100F) - ((tempo_alem *
total)/100F);

        return total;
    }

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