



Fundação CECIERJ - Vice Presidência de Educação Superior a Distância

**Curso de Tecnologia em Sistemas de Computação**

**AD1 de Programação III**

**1º semestre de 2010**

**Nome:**

**Matrícula:**

**Pólo:**

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

Considere que sua empresa seja contratada para desenvolver o software que trata o RioCard. O RioCard é um cartão pré-pago que é debitado do valor da passagem, caso o saldo do cartão seja suficiente, toda vez que o usuário o utiliza em qualquer meio de transporte público (tais como, metrô, trens, vans regularizadas, ônibus e barcas). Existem algumas informações que são relevantes no uso deste cartão, tais como, o saldo e o código do usuário para a realização de recargas.

Existe uma especialização deste cartão, chamado de BilheteUnico, que é mais econômico que o RioCard porque possibilita, em viagens intermunicipais de duração de, no máximo, duas horas, e realizadas em, no máximo, dois meios de transporte quaisquer, que o valor único debitado deste cartão seja pré-fixado em R\$ 4,40. Quando se tratar de serviço intermunicipal com valor superior a R\$ 4,40, será debitado do cartão o valor máximo de R\$ 4,40 mesmo que não haja integração. Neste cartão é importante indicar qual foi a última viagem realizada até o momento.

O tempo de viagem no BilheteUnico só começa a ser contado após o primeiro contato entre este cartão e o equipamento eletrônico de cobrança (chamado de validador) no primeiro embarque. Após o segundo embarque, o passageiro também passa este cartão no validador, mas não será computado o preço inteiro da viagem.

A economia do BilheteUnico pode ser observada nas seguintes situações:

**(1) Em viagens diretas:**

Linha	Itinerário	Tarifa	Economia
118T	XXX-YYY	R\$ 12,35	R\$ 7,95
601I	NNN-NI	R\$ 17,30	R\$ 12,90
425D	AA-CG	R\$ 10,85	R\$ 6,45
496B	MC-PM	R\$ 10,65	R\$ 6,25

**(2) Em viagens com integração:**

Trem (R\$ 2,50) + ônibus (R\$ 2,25) = R\$ 4,75	Economia: R\$ 0,35
Trem (R\$ 2,50) + metrô (R\$ 2,80) = R\$ 5,30	Economia: R\$ 0,90
Ônibus (R\$ 2,20) + ônibus (R\$ 3,40) = R\$ 5,60	Economia: R\$ 1,20
Ônibus (R\$ 2,20) + barcas (R\$ 2,80) = R\$ 5,00	Economia: R\$ 0,60

Considere que algumas informações sobre a viagem devem ser mantidas, como, por exemplo, código (único para todos os meios de transporte), tarifa e uma informação se o trajeto é intermunicipal.

Escreva as seguintes classes RioCard, BilheteUnico, Viagem e as demais classes que você achar conveniente. Desenvolva também uma classe de Teste para verificar o uso correto de suas classes. Nesta última classe é interessante constar a criação de vários cartões e diversas viagens para testar a funcionalidade descrita no enunciado.

### RESPOSTA:

```
public class Teste{
    public static void main(String[] args){
        RC rc1 = new RC(50.0);
        BU bu1 = new BU(20.0), bu2 = new BU(200.0);
        boolean I = true;

        rc1.criaViagem(2.70, !I, new Horario(9, 0, 0));
        rc1.criaViagem(2.80, I, new Horario(9, 30, 0));
        rc1.criaViagem(2.70, !I, new Horario(10, 0, 0));
        rc1.criaViagem(2.80, I, new Horario(10, 30, 0));
        rc1.criaViagem(12.50, I, new Horario(11, 0, 0));
        rc1.criaViagem(17.50, I, new Horario(12, 0, 0));
        rc1.criaViagem(11.00, I, new Horario(13, 0, 0));
        System.out.println(rc1);

        bu1.criaViagem(2.70, !I, new Horario(9, 0, 0));
        bu1.criaViagem(2.80, I, new Horario(9, 30, 0));
        bu1.criaViagem(3.00, !I, new Horario(10, 0, 0));
        bu1.criaViagem(3.00, I, new Horario(12, 0, 1));
        bu1.criaViagem(2.50, !I, new Horario(12, 30, 0));
        bu1.criaViagem(2.50, !I, new Horario(22, 0, 0));
        System.out.println(bu1);

        bu2.criaViagem(12.50, I, new Horario(9, 0, 0));
        bu2.criaViagem(2.50, !I, new Horario(10, 30, 0));
        bu2.criaViagem(2.70, !I, new Horario(22, 0, 0));
        bu2.criaViagem(2.80, I, new Horario(23, 59, 59));
        bu2.criaViagem(12.50, I, new Horario(7, 0, 0));
        bu2.criaViagem(17.50, I, new Horario(8, 0, 0));
        bu2.criaViagem(11.10, I, new Horario(9, 0, 0));
        bu2.criaViagem(2.50, !I, new Horario(9, 30, 0));
        bu2.criaViagem(2.40, !I, new Horario(10, 0, 0));
        System.out.println(bu2);
    }
}

class Horario{
    private int horas;
    private int minutos;
    private int segundos;

    Horario(int h, int m, int s){
        horas = h;
        minutos = m;
        segundos = s;
    }
}
```

```

    int pegaHoras(){ return horas; }

    int pegaMinutos(){ return minutos; }

    int pegaSegundos(){ return segundos; }

    int calculaDiferenca(Horario x){
        int h = x.pegaHoras(), m = x.pegaMinutos(), s = x.pegaSegundos(),
        difx, dif;

        dif = segundos + (minutos * 60) + (horas * 3600);
        difx = s + (m * 60) + (h * 3600);

        if(dif >= difx) return (dif - difx);
        return (difx - dif);
    }

    public String toString(){ return horas + "hora(s) " + minutos +
    "minuto(s) " + segundos + "segundo(s)\n"; }
}

class Viagem{
    private static int prox_cod = 0;
    private int codigo;
    private boolean intermunicipal;
    private double tarifa;
    private Horario hora;

    Viagem(double t, boolean i, Horario h){
        codigo = ++prox_cod;
        tarifa = t;
        intermunicipal = i;
        hora = h;
    }

    Horario pegaHorario(){ return hora; }

    double pegaTarifa(){ return tarifa; }

    boolean eIntermunicipal(){ return intermunicipal; }

    public String toString(){
        String r = "Viagem: " + codigo + " R$ = " + tarifa + " ";
        if(intermunicipal) r += "(Intermunicipal) ";
        r += hora.toString();
        return r;
    }
}

class RC{
    final int NUM_MAX = 1000;
    protected int prox_viagem;
    protected Viagem[] vet;
    private static int prox_cod = 0;
    protected int codigo;
    protected double saldo;

```

```

RC(double s){
    vet = new Viagem[NUM_MAX];
    codigo = ++prox_cod;
    prox_viagem = 0;
    saldo = s;
}

void efetuaRecarga(double s){ saldo += s; }

double pegaSaldo(){ return saldo; }

Viagem criaViagem(double t, boolean i, Horario h){
    if((saldo < t) || (prox_viagem == NUM_MAX))
        return null;

    Viagem v = new Viagem(t, i, h);
    vet[prox_viagem++] = v;
    saldo -= t;
    return v;
}

public String toString(){
    String r = "Rio Card (" + codigo + "): R$ = " + saldo + "\n";
    for(int i = 0; i < prox_viagem; i++)
        r += vet[i].toString();

    return r;
}
}

class BU extends RC{
    private int conta_viagem;
    private double vuv;

    BU(double s){
        super(s);
        conta_viagem = 0;
        vuv = 0.0;
    }

    Viagem criaViagem(double t, boolean i, Horario h){
        if((saldo < t) || (prox_viagem == NUM_MAX))
            return null;

        Viagem v;

        if((conta_viagem == 0) && (i)){
            v = new Viagem(t, i, h);
            vet[prox_viagem++] = v;
            if(t <= 4.4){
                saldo -= t;
                vuv = t;
            }
            else{
                vuv = 4.40;
                saldo -= vuv;
            }
        }
    }
}

```

```

        conta_viagem = 1;
    }

    else if((conta_viagem == 1) && (vet[prox_viagem - 1].pegaHorario().calculaDiferenca(h) <= 7200) && (i) && (!vet[prox_viagem - 1].eIntermunicipal())){
        v = new Viagem(t, i, h);
        vet[prox_viagem++] = v;
        saldo = saldo - (4.40 - vuv);
        conta_viagem = 0;
        vuv = 0.0;
    }

    else if((conta_viagem == 1) && (vet[prox_viagem - 1].pegaHorario().calculaDiferenca(h) <= 7200) && (!i) && (vet[prox_viagem - 1].eIntermunicipal())){
        v = new Viagem(t, i, h);
        vet[prox_viagem++] = v;
        saldo = saldo - (4.40 - vuv);
        conta_viagem = 0;
        vuv = 0.0;
    }

    else if(i){
        v = new Viagem(t, i, h);
        vet[prox_viagem++] = v;
        if(t <= 4.4){
            saldo -= t;
            vuv = t;
        }
        else{
            vuv = 4.40;
            saldo -= vuv;
        }
        conta_viagem = 1;
    }

    else{
        v = super.criaViagem(t, i, h);
        vuv = t;
        conta_viagem = 1;
    }

    return v;
}

public String toString(){
    String r = "Bilhete Unico (" + codigo + "): R$ = " + saldo + "\n";
    for(int i = 0; i < prox_viagem; i++){
        r += vet[i].toString();
    }

    return r;
}
}

```