

## EXERCÍCIO SOBRE THREADS

Construir uma pequena aplicação que simule uma entidade emissora de bilhetes para um dado evento e duas entidades compradoras que concorrem entre si e com a emissora, para aquisição de todos os bilhetes disponíveis, (neste caso são emitidos 20 bilhetes).

Sempre que um dos compradores tenta comprar um bilhete e não há disponíveis é enviada uma mensagem sugestiva.

Quando um bilhete é adquirido é visualizada uma mensagem, indicando o comprador e o número do bilhete comprado.

<b>Classes envolvidas:</b>	Bilhete
	Bilheteira
	Distribuidor
	Comprador

```
public static void main(String []args){
    Bilheteira b = new Bilheteira();
    Distribuidor d = new Distribuidor(b);
    Comprador c1= new Comprador(b,"Comprador1");
    Comprador c2= new Comprador(b,"Comprador2");
    d.start();
    c1.start();
    c2.start();
}
}
```

```
public class Bilhete {
    int numero;

    public Bilhete(int i) {
        numero = i;
    }

    public int getNum(){
        return numero;
    }

    public void setNum(int n){
        numero=n;
    }
}
```

```

import java.util.*;
public class Bilheteira{

    private Vector conjBilhetes=new Vector();

    public synchronized int getBilhete(){
        if(conjBilhetes.size()==0){
            System.err.println(Thread.currentThread
            ().getName()+" Nao ha bilhetes");
            return -1;
        }
        else{
            System.err.println(Thread.currentThread().g
            etName()+" Bilhete comprado numero:"
            +((Bilhete)conjBilhetes.get(0)).getNum());
            Bilhete b=(Bilhete)conjBilhetes.remove(0);
            return b.getNum();
        }
    }

    public synchronized void setBilhete(int val) {

        System.err.println(Thread.currentThread().
        getName()+" Bilhete emitido num: "+val);
        //junta bilhete à bilheteira
        conjBilhetes.add(new Bilhete(val));
    }
}

```

```
public class Comprador extends Thread  
{  
    private Bilheteira b;  
    private static boolean fim;  
  
    public Comprador(Bilheteira b,String nome){  
        super(nome);  
        this.b=b;  
        fim=false;  
    }  
  
    public void run(){  
        int valor=0;  
        while(!fim)  
        {  
            try{  
                Thread.sleep((int)(Math.random()*3000));  
            }  
            catch(InterruptedException e){  
                System.err.println("\n"+e.toString());  
            }  
            if(fim==false)  
            {  
                valor =b.getBilhete();  
                if(valor==20)  
                    fim=true;  
            }  
            System.err.println("\n"+getName()+  
                               " Bilhetes comprados!!!");  
        }  
    }  
}
```

```

public class Distribuidor extends Thread{

    private Bilheteira b;

    public Distribuidor(Bilheteira b){
        super("Distribuidor");
        this.b=b;
    }

    public void run(){
        for(int i=1;i<=20;i++)
        {
            try{
                Thread.sleep((int)(Math.random()*3000));
            }
            catch(InterruptedException e){
                System.err.println("\n"+e.toString());
            }
            b.setBilhete(i);
        }
        System.err.println("\n"+getName()+
                            "Terminei!!!");
    }
}

```

### RESULTADO DA EXECUÇÃO

```

Comprador2    Nao ha bilhetes
Comprador1    Nao ha bilhetes
Distribuidor  Bilhete emitido num: 1
Comprador2    Bilhete comprado numero: 1
Comprador1    Nao ha bilhetes
Comprador1    Nao ha bilhetes
Comprador2    Nao ha bilhetes

```

Comprador1 Nao ha bilhetes  
Distribuidor Bilhete emitido num: 2  
Comprador2 Bilhete comprado numero: 2  
Comprador2 Nao ha bilhetes  
Comprador1 Nao ha bilhetes  
Comprador1 Nao ha bilhetes  
Comprador1 Nao ha bilhetes  
Distribuidor Bilhete emitido num: 3  
Comprador2 Bilhete comprado numero: 3  
Comprador1 Nao ha bilhetes  
Distribuidor Bilhete emitido num: 4  
Comprador2 Bilhete comprado numero: 4  
Distribuidor Bilhete emitido num: 5  
Comprador1 Bilhete comprado numero: 5  
Comprador2 Nao ha bilhetes  
Comprador1 Nao ha bilhetes  
Comprador2 Nao ha bilhetes  
Comprador2 Nao ha bilhetes  
Distribuidor Bilhete emitido num: 6  
Comprador1 Bilhete comprado numero: 6  
Comprador2 Nao ha bilhetes  
Distribuidor Bilhete emitido num: 7  
Comprador2 Bilhete comprado numero: 7  
Comprador1 Nao ha bilhetes  
Comprador2 Nao ha bilhetes  
Comprador1 Nao ha bilhetes  
Distribuidor Bilhete emitido num: 8  
Comprador2 Bilhete comprado numero: 8  
Distribuidor Bilhete emitido num: 9  
Comprador1 Bilhete comprado numero: 9  
Comprador1 Nao ha bilhetes  
Comprador2 Nao ha bilhetes  
Distribuidor Bilhete emitido num: 10  
Comprador2 Bilhete comprado numero: 10

Comprador2 Nao ha bilhetes  
Comprador1 Nao ha bilhetes  
Distribuidor Bilhete emitido num: 11  
Comprador2 Bilhete comprado numero: 11  
Comprador1 Nao ha bilhetes  
Comprador1 Nao ha bilhetes  
Comprador2 Nao ha bilhetes  
Comprador2 Nao ha bilhetes  
Comprador1 Nao ha bilhetes  
Distribuidor Bilhete emitido num: 12  
Distribuidor Bilhete emitido num: 13  
Distribuidor Bilhete emitido num: 14  
Comprador1 Bilhete comprado numero: 12  
Comprador2 Bilhete comprado numero: 13  
Distribuidor Bilhete emitido num: 15  
Comprador1 Bilhete comprado numero: 14  
Comprador1 Bilhete comprado numero: 15  
Comprador1 Nao ha bilhetes  
Comprador2 Nao ha bilhetes  
Comprador1 Nao ha bilhetes  
Distribuidor Bilhete emitido num: 16  
Comprador1 Bilhete comprado numero: 16  
Distribuidor Bilhete emitido num: 17  
Comprador2 Bilhete comprado numero: 17  
Distribuidor Bilhete emitido num: 18  
Distribuidor Bilhete emitido num: 19  
Comprador1 Bilhete comprado numero: 18  
Comprador1 Bilhete comprado numero: 19  
Distribuidor Bilhete emitido num: 20  
Distribuidor Terminei!!!  
Comprador2 Bilhete comprado numero: 20  
Comprador2 Bilhetes comprados!!!  
Comprador1 Bilhetes comprados!!!