```
//conteúdo do arquivo ManipulaPacotes.java
  ...
      System.out.println ("5 - Imprimir o custo total
apenas dos pacotes que serao entregues a noite;");
        ...
        case 5:
          {
            double auxiliar=0;
            for (int i=0; i<numeroDeElementos; i++)
              if (vetor[i] instanceof PacoteNoite)
                auxiliar += vetor[i].calculaCusto();
            System.out.printf ("Custo total das %s %f\n",
            "postagens para entrega a noite: ", auxiliar);
          }
          break;
          ...
    }while(opcao!=6);
  }
}
```

Classe Object

Um detalhe, porém muito relevante, é que todas as classe de Java são subclasses da classe denominada ***Object***.

Ou seja, quando declaramos uma classe em Java e não especificamos que esta é derivada de nenhuma outra, mesmo assim implicitamente esta classe é uma subclasse da classe ***Object***.

A classe ***Object*** possui onze métodos e já estudamos alguns destes métodos que são herdados pelas subclasses da classe ***Object***, ou seja, pelas demais classe declaradas em Java.

Classe Object

Dentre os métodos da classe *Object* já estudamos:

- *toString* -> retorna uma representação String de um objeto;

- *finalize* -> é utilizado pelo coletor de lixo para realizar a limpeza de termino de um objeto antes da memória ocupada por este ser reinvidicada.

Estudaremos mais alguns métodos desta classe, para isto vamos analisaremos o seguinte código:

```
class Teste { // implicitamente tem-se “class Teste extends Object”
  private int v1;
  public void setV1(int novoV1) {
        v1 = novoV1;
  }
  public int getV1() {
        return v1;
  }
}
public class TesteObject {
  public static void main(String args[]) {
    Teste t1 = new Teste();
    t1.setV1(7);
   System.out.printf("\n%d\n", t1.getV1());
    Teste t2 = t1;
    if (t1 == t2)
      System.out.println("As referencias contidas em t1 e t2 sao
iguais.");
    t2 = t1.clone(); /*Este método é protected e, além disto,
problemas podem ocorrer quando os objetos clonados possuírem
variáveis com referências, para tal devemos sobrescrevê-lo. */
```

```
        if (t1 == t2)
            System.out.println("As referencias contidas em t1 e t2 sao iguais.");
        /* Para comparar objetos devemos sobrescrever o método equals herdado da classe Object */
        if (t1.equals(t2))
            System.out.println("Os objetos t1 e t2 possuem o mesmo estado.");
    }
}
```

```
class Teste {
  private int v1;
  public void setV1(int novoV1) {
        v1 = novoV1;
  }
  public int getV1() {
        return v1;
  }
  public boolean equals (Teste entrada) { //sobrescrevendo equals
    return (getV1()==entrada.getV1());
    //return (v1==entrada.v1);
  }
  public Teste clone () { //sobrepondo clone
        Teste aux = new Teste();
        aux.setV1(getV1());
        return aux;
  }
}
```

Fim