**Encapsulamento (data hiding) :**

• É definido como uma técnica para minimizar as interdependências entre módulos, através da definição de interfaces externas.

• “Caixa preta” - não é necessário saber como funciona internamente, mas sim como utilizar.

**Encapsulamento :** interface (pública) de um objeto declara todas as operações permitidas (métodos).

**Encapsulamento - benefícios**

**• Segurança:** protege os objetos de terem seus atributos corrompidos por outros objetos.

**• Independência:** “escondendo” seus atributos, um objeto protege outros de complicações de dependência da sua estrutura interna.

**Mensagens e Métodos**

• Para invocar um método, deve-se enviar uma mensagem para o objeto desejado.

• Para enviar uma mensagem, deve-se

– identificar o objeto que receberá a mensagem

– identificar o método que o objeto deverá executar

– passar os argumentos requeridos pelo método

**Abstração**

• Focalizar o essencial, ignorar propriedades acidentais.

Classes

• Uma classe determina um conjunto de objetos com :

– propriedades semelhantes

– comportamentos semelhantes

– relacionamentos comuns com outros objetos

Precisamos instanciá-la, criar um objeto bolo a partir dessa especificação (a classe) para utilizá-la. Podemos criar centenas de bolos a partir dessa classe (a receita, no caso), eles podem ser bem semelhantes, alguns até idênticos, mas são objetos diferentes. Podemos fazer milhares de analogias semelhantes. A planta de uma casa é uma casa? Definitivamente não. Não podemos morar dentro da planta de uma casa, nem podemos abrir sua porta ou pintar suas paredes. Precisamos, antes, construir instâncias a partir dessa planta. Essas instâncias, sim, podemos pintar, decorar ou morar dentro. Pode parecer óbvio, mas a dificuldade inicial do paradigma da orientação a objetos é justo saber distinguir o que é classe e o que é objeto. É comum o iniciante utilizar, obviamente de forma errada, essas duas palavras como sinônimos.