

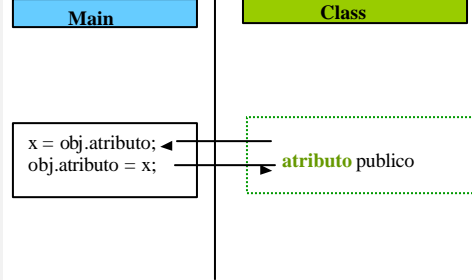
Programação III

Encapsulamento

Jocélio Passos

joceliopassos@bol.com.br

Sem Encapsulamento



7/22/2007

2

Encapsulamento : O que é

- Deitel Cap 8, páginas 342 e 351 [381 e 391]
- Sebesta Cap 10 pág 397, 407, 408
- O objetivo é ocultar atributos/métodos da classe
- Portanto, restringe e controla o acesso a classe
- Fornece apenas os métodos necessários para o uso da classe
- Os detalhes da implementação são omitidos
- Conhecido também como **data hiding, information hiding, dissimulação de informações**
- Dados e programas isolados como uma capsula ou caixa preta

7/22/2007

3

Encapsulamento : Vantagens

- Clareza do código
- Minimização de erros
- Facilidade de extensão
- Mas a **Facilidade na Manutenção** é a maior vantagem
- Por que a classe é conhecida apenas pela interface
- Pode-se mudar internamente sua estrutura sem que o usuário tome conhecimento disto

7/22/2007

4

Encapsulamento : private

- Formato geral :
- Na Classe
- **private tipo** atributo;
- **private tipo** metodo () { }

7/22/2007

5

Encapsulamento : private

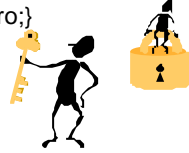
- **private** é uma palavra reservada, qualificador, modificador ou "specifiers"
- De forma contrária a **public**, permite que atributos e métodos sejam acessíveis somente nos métodos da própria classe.
- É o Nível mais rígido de encapsulamento.
- Pronome possessivos
 - public nosso (publico)
 - private meu (exclusivo)

7/22/2007

6

Encapsulamento

- Para alterar (assign) ou ler (access) o atributo encapsulado pode-se criar dois métodos
- Formato geral :
- `public void setAtributo (tipo parametro)`
- `{this.atributo = parametro;}`
- `public tipo getAtributo ()`
- `{return (this.atributo);}`



7/22/2007

7

Encapsulamento

Main

Class

`x = obj.getAtributo();`

`getAtributo ()`

`obj.setAtributo (x);`

`setAtributo (tipo p)`

~~`x = obj.atributo;`~~
~~`obj.atributo = x;`~~

atributo privado

7/22/2007

8

Encapsulamento

- Outras formas de encapsular são os modificadores :
 - protected (Herança)
 - default : “package” / “friendly” (pacotes)

7/22/2007

9

Exercícios

- 1) Encapsule a classe Circulo. Crie um método para alterar (set) e um método para ler (get) o raio. Depois crie um programa que crie o objeto, leia e use. Sair quando a base do for zero.

7/22/2007

10

Encapsulando : Classe Círculo

- `class Circulo {`
- `private float raio; // atributo encapsulado`
- `....`
- `public void setRaio (float raio)`
- `{this.raio = raio;}`
- `public float getRaio ()`
- `{return (this.raio);}`
- `...`
- `}`

7/22/2007

11

Encapsulando

- `class LerCirculo{`
- `public static void main(String[] args) throws Exception`
- `{ Circulo circulo1 = new Circulo (10);`
- `float raiotemp;`
- `System.out.println ("Dados do Circulo");`
- `do{`
- `circulo1.mostrarDados ();`
- `System.out.print ("\nInforme o raio > ");`
- `raiotemp = JUtil.readFloat ();`
- `circulo1.setRaio (raiotemp);`
- `} while (circulo1.getRaio () != 0);`
- `}}`

7/22/2007

12

Encapsulando

```
class LerCirculo{
    public static void main(String[] args) throws Exception
    {
        Circulo circulo1 = new Circulo (10);
        System.out.println ("Dados do Circulo");
        do{
            circulo1.mostrarDados ();
            System.out.print ("\nInforme o raio > ");
            circulo1.setRaio (JUtil.readFloat ());
        } while (circulo1.getRaio () != 0);
    }
}
```

7/22/2007

13

Encapsulando : Erro

- Acessar de forma direta um atributo causa um erro em tempo de compilação
- Exemplo :
 - `circulo1.raio = 10;`
 - "raio has private access in Circulo"

7/22/2007

14

Exercícios Propostos

- 1) Encapsule e use a classe Retângulo.
- 2) Encapsule e use a classe Cubo.
- 3) Para acessar (ler) um atributo encapsulado (peso, por exemplo) devemos criar um método chamado sempre de `getPeso`. Verdadeiro ou Falso ? Deitel 352
- 4) O que é um método acessor ? Deitel 343 [381]
- 5) O que é um método mutatório ? Deitel 343 [381]
- 6) Que vantagens temos no uso de métodos "set" ? Deitel 343, 352, 358 [381, 391-392]

7/22/2007

15

Exercícios Propostos

- 7) Todo atributo privado (ocultado ou encapsulado) deve obrigatoriamente ter métodos `get` / `set` para ele. Explique sua resposta . Deitel 343 [381]
- 8) O que é um método de consulta ? Deitel 352 [391-392]

7/22/2007

16

Exercícios Propostos

- 9) Após criar a classe, qual o resultado na tela do programa ?
- class Teste
- { float x;
- public void altera (float y)
- {float x;
- x = y;}
- }

7/22/2007

17

Exercícios Propostos

- class Teste2
- {public static void main(String[] args) throws Exception
- {Teste t = new Teste ();
- t.x = 20;
- System.out.println(t.x);
- t.altera (30);
- System.out.println(t.x);
- JUtil.pause ();
- }}

7/22/2007

18