

Encapsulamento	
Preparado por	Prontuário
José Roberto Claudino Ferreira	SP3024369

Exercícios de Encapsulamento

1 - O que é encapsulamento?

R: Encapsulamento é ocultar atributos internos do meio externo e acessá-los apenas com métodos públicos.

2 - Porque os atributos de uma classe, na maioria dos casos, devem ser private?

R: Porque assim só será possível alterar aquele atributo/estado com um método, evitando que outras classes mexam nos atributos das outras diretamente.

3 - Quais problemas podem ocorrer caso os atributos de uma classe sejam public?

R: Ao se usar o mesmo nome de variável em outra classe, os atributos de ambas serão modificados, porém se forem private somente serão modificadas através de um método público dentro da classe do atributo que quer modificar, além da falta de segurança.

4 - Para que servem os métodos getters e setters?

R: Os métodos getters servem para pegar um atributo de uma classe, já os setters eles modificam o atributo ao invés de pegá-lo.

5 - Como deve ser implementado um método get?

R: Da seguinte forma:

```
public TipoDoAtributo getNomeDoAtributo(){  
    return nomeDoAtributo;  
}
```

6 - Como deve ser implementado um método set?

R: Da seguinte forma:

```
public void setNomeDoAtributo(TipoDoAtributo nomeDoAtributo){  
    this.nomeDoAtributo = nomeDoAtributo;  
}
```

7 - É necessário ter sempre um método get e um método set para todos os atributos de uma classe? justifique sua resposta.

R: Não, somente para atributos private que você deseja pegar ou modificar seu valor, é necessário levar em conta a situação de cada programa para determinar se algum atributo terá get ou set ou ambos.

8 - Crie um Diagrama da classe Retângulo que tenha os atributos base e altura, e os seguintes métodos que calcula a área e outro que calcula o perímetro, além destes métodos criem os métodos getters e setters dos atributos.

R:

Retângulo
- base : int - altura : int
+ Retângulo() + Retângulo(base: int, altura: int) + Area() : int + Perimetro() : int + getBase() : int + setBase(base: int) : void + getAltura() : int + setAltura(altura: int) : void

9 - Implemente em Java a classe descrita no exercício 8, e implemente uma classe Forma que contenha o método main que manipula um objeto da classe retângulo.

R: **Class Retângulo**

```
public class Retangulo {  
  
    private int base;  
    private int lado;  
  
    public Retangulo(int base, int lado){  
        this.base = base;  
        this.lado = lado;  
    }  
    public int getArea(){  
        return base*lado;  
    }  
  
    public int getPerimetro() {  
        return ((base*2)+(lado*2));  
    }  
  
    public int getBase() {  
        return base;  
    }  
    public void setBase(int base){  
        this.base = base;  
    }  
    public int getLado() {  
        return lado;  
    }  
    public void setLado(int lado){  
        this.lado = lado;  
    }  
}
```

Class Forma

```
public class Forma {  
    public static void main(String[] args) {  
        Retangulo re1 = new Retangulo(10, 20);  
  
        System.out.println("Área: " + re1.getArea());  
        System.out.println("Perimetro: " + re1.getPerimetro());  
  
        System.out.println("Base: " + re1.getBase());  
        System.out.println("Lado: " + re1.getLado());  
  
        re1.setBase(35);  
        re1.setLado(25);  
  
        System.out.println("Base: " + re1.getBase());  
        System.out.println("Lado: " + re1.getLado());  
    }  
}
```