

**Curso** : Engenharia de Software  
**Disciplina** : Algoritmos e Estruturas de Dados II  
**Professora** : Eveline Alonso Veloso

### Exercícios sobre Programação Orientada por Objetos

- 1) Crie uma classe *Retangulo* com os atributos privados: *base* (double) e *altura* (double). Sua classe também terá, pelo menos, dois construtores, e os métodos *gets*, *sets*, *clone* e *imprimir*. O método *imprimir* exibe o valor de todos os atributos do objeto. Implemente também o método *double getArea()*, que retorna o valor da área do retângulo; e o método *double getPerimetro()*, que retorna o valor do perímetro do retângulo.

Desenvolva também uma classe *Aplicacao* contendo um método *main* para testar os métodos da classe *Retangulo*.

- 2) Crie uma classe *CartaoValeTransporte* com os atributos privados: *numero* (int), *saldo* (double) e *bloqueado* (boolean). Sua classe também terá, pelo menos, dois construtores, e os métodos *gets*, *sets*, *clone* e *imprimir*. O método *imprimir* exibe o valor de todos os atributos do objeto.

Implemente também os métodos da classe *CartaoValeTransporte* descritos abaixo, obedecendo às seguintes descrições:

- *carregar(double credito)*: Se o cartão vale-transporte estiver desbloqueado e o valor a ser creditado, passado como parâmetro para esse método, for positivo: incrementa o saldo do cartão vale-transporte com o valor de crédito informado;
- *pagar(double tarifa)*: Se o cartão vale-transporte estiver desbloqueado, o valor da tarifa, passado como parâmetro para esse método, for positivo, e houver saldo suficiente: decrementa, do saldo atual do cartão vale-transporte, o valor da tarifa informada;
- *bloquear(CartaoValeTransporte destino)*: bloqueia o cartão vale-transporte, transfere seu saldo restante para o outro cartão vale-transporte que foi informado como parâmetro desse método, e zera seu saldo.

Após a implementação da classe *CartaoValeTransporte*, a função *main*, a seguir, deverá funcionar:

```
public static void main(String[] args) {  
  
    CartaoValeTransporte cartao1, cartao2;  
  
    cartao1 = new CartaoValeTransporte(21074, 50.00);  
    System.out.println("Saldo atual do cartão vale-transporte 1: R$ " + cartao1.getSaldo()); // R$  
50,00  
  
    cartao1.pagar(2.80);  
    System.out.println("Saldo atual do cartão vale-transporte 1: R$ " + cartao1.getSaldo()); // R$  
47,20  
  
    cartao1.pagar(1.40);  
    System.out.println("Saldo atual do cartão vale-transporte 1: R$ " + cartao1.getSaldo()); // R$
```

45,80

```
cartao1.carregar(5.00);
```

```
System.out.println("Saldo atual do cartão vale-transporte 1: R$ " + cartao1.getSaldo()); // R$
```

50,80

```
cartao2 = new CartaoValeTransporte(390176, 25.00);
```

```
System.out.println("Saldo atual do cartão vale-transporte 2: R$ " + cartao2.getSaldo()); // R$
```

25,00

```
cartao1.bloquear(cartao2);
```

```
System.out.println("Saldo atual do cartão vale-transporte 1: R$ " + cartao1.getSaldo()); // R$
```

0,00

```
System.out.println("Saldo atual do cartão vale-transporte 2: R$ " + cartao2.getSaldo()); // R$
```

75,80

```
}
```