

1) O que é um construtor em Java ?

R: É um método: obrigatório para toda classe; utilizado para inicializar as variáveis da classe; que pode chamar outros métodos; não possui retorno; aceita parâmetros e sobrecarga.

2) Qual o conceito de se usar um Construtor, para que serve ?

R: Basicamente para inicializar as variáveis de uma classe, uma forma de inicializar as variáveis com um determinado valor.

3) O que é encapsulamento em programação orientada a objetos ?

R: É um mecanismo utilizado para reunir os dados e os métodos em uma estrutura que pode esconder a implementação do objeto, para garantir uma integridade dos dados contidos no objeto.
O encapsulamento permite definir os níveis de visibilidade dos elementos da classe.

4) Para que servem os getter e setters na programação orientada a objetos ?

R: Um método que permite pegar e alterar (respectivamente) os atributos da classe de forma protegida, sendo o get um método que retorna um valor e o set um método tipo void, ou seja, sem retorno.

5) Quais são os tipos de modificadores de acesso existentes no Java ?

R: Os modificadores de acesso existentes no Java são: *private*, *protected* e *public*.

6) Para que serve cada um dos modificadores existentes no Java ?

R: *Private*: faz com que os atributos e métodos sejam exclusivo da classe, e não são transmitidos por herança.

Protected: faz com que os atributos e métodos sejam visíveis por todas as classes do mesmo pacote.

Public: faz com que os atributos e métodos não possuam restrição de visibilidade.

7) O que é Herança no Java ?

R: Um mecanismo que permite que uma classe herde todos os atributos, métodos e relacionamentos de uma outra classe.

8) Por que utilizar Herança na programação orientada a objetos ?

R: Ela cria uma vantagem de reutilização de código, já que você pode reaproveitar os atributos e métodos de uma classe.

```
package exercicio.pkg9;
import java.util.Scanner;
public class Exercicio9
{
    public static void main(String[] args)
    {
        double val;
        Conta edu = new Conta(01, "Eduardo", 0.0);
        for(int i = 0; i < 3; i++)
        {
            System.out.println("Digite o valor do depósito: ");
            System.out.print("R$: ");
            Scanner n = new Scanner(System.in);
            val = n.nextDouble();
            edu.depositar(val);
        }
    }
}
```

```

for(int i = 0; i < 3; i++)
{
    System.out.println("Digite o valor do saque: ");
    System.out.print("R$: ");
    Scanner n = new Scanner(System.in);
    val = n.nextDouble();
    edu.sacar(val);
}
}

```

```

public class Conta
{
    private int numeroConta;
    private String nomeDono;
    private double saldo;
    public Conta(int numConta, String nomeConta, double saldoConta)
    {
        this.numeroConta = numConta;
        this.nomeDono = nomeConta;
        this.saldo = saldoConta;
    }

    public void depositar(double valorDeposito)
    {
        this.saldo = this.saldo + valorDeposito;
        System.out.println("Saldo atual R$: " + this.saldo);
    }

    public void sacar(double valorSaque)
    {
        if(valorSaque < this.saldo)
        {
            this.saldo = this.saldo - valorSaque;
            System.out.println("Saldo atual R$: " + this.saldo);
        }
        else
        {
            System.out.println("Saldo insuficiente !!");
        }
    }
}
}

```