

Roteiro 3

Parte 1 (roteiro3.parte1) – Funções Estáticas:

Funções estáticas ou **métodos estáticos** em Java são funções que pertencem à própria classe e não a uma instância específica dela. Isso permite que você as chame diretamente na classe, sem precisar criar um objeto. Funções estáticas são especialmente úteis para operações independentes do estado do objeto e oferecem uma maneira prática de invocação.

Abaixo você pode ver um exemplo de como se declara uma função estática em java:

```
modificador  tipo  nome      Parâmetros da função
static int soma(int valor1, int valor2) {
    return valor1 + valor2;
}
```

Em Java, **static** e **public** são palavras-chave usadas para modificar a declaração de uma função (ou método). Cada uma delas tem um propósito específico:

A palavra-chave **static** é usada para definir um método ou variável como pertencente à classe em vez de uma instância específica dessa classe.

A palavra-chave **public** é um modificador de acesso que indica que o método pode ser acessado de qualquer classe, seja ela na mesma classe, em uma classe diferente no mesmo pacote ou em um pacote diferente.

OBS.: Neste roteiro utilizaremos apenas funções estáticas. Os conceitos acima ficarão mais claros nos próximos roteiros.

1 – Crie o pacote **roteiro3.parte1** com a classe **Programa10** conforme o código abaixo. O cenário implementado abaixo diz respeito ao seguinte problema :

“Faça um programa que receba o salário-base de um funcionário, calcule e mostre o salário a receber. Sabendo-se que esse funcionário tem gratificação de 5% sobre o salário -base e paga imposto de 7% também sobre o salário-base.”

Faça os devidos testes para verificar se a solução implementada foi atendida.

```

import java.util.Scanner;

public class Programa10 {

    public static void main(String[] args) {

        Scanner entrada = new Scanner(System.in);

        double sb, grat, imp, sr;

        System.out.println("Informe o Salário Base :");
        sb = entrada.nextDouble();
        grat = sb * 5/100;
        imp = sb * 7/100;
        sr = sb + grat - imp;

        System.out.println("Salário a receber : " + sr);

    }

}

```

2 – No mesmo pacote crie a classe **Programa11** idêntica ao Programa10. Neste exemplo faremos o uso simples de uma função estática. Observe que foi criada uma função para calcular o valor da gratificação e esta função foi chamada na função main. Veja que a chamada da função foi feita com o nome da classe na frente, justamente por ser uma função estática : **Programa11.calcGrat(sb)** . Obs.: Como as funções foram criadas na mesma classe, a chamada da função poderia ser feita sem o nome da Classe : **calcGrat(sb)**

```

import java.util.Scanner;

public class Programa11 {

    public static void main(String[] args) {

        Scanner entrada = new Scanner(System.in);

        double sb, grat, imp, sr;

        System.out.println("Informe o Salário Base :");
        sb = entrada.nextDouble();
        grat = Programa11.calcGrat(sb);
        imp = sb * 7/100;
        sr = sb + grat - imp;

        System.out.println("Salário a receber : " + sr);

    }

    public static double calcGrat(double sb)
    {
        return sb*5/100;
    }

}

```

3 – Ainda no Programa11 crie uma função para calcular o imposto (**calcImp**) de forma semelhante ao que foi feito para a gratificação. Faça as adaptações necessárias no programa.

4 – Ao criar uma função para o cálculo da gratificação e outra para o cálculo do imposto, provavelmente terá criado 2 funções idênticas. Ter duas funções que fazem praticamente a mesma coisa não é adequado, pois caracteriza duplicidade de código. Crie agora uma classe **Programa12** no mesmo pacote, assim como uma nova função, onde seja possível o cálculo da gratificação e do imposto na mesma função. Ou seja, devemos ter neste programa uma função única capaz de calcular a gratificação e o imposto. Faça as adaptações necessárias no código.

Parte 2 (roteiro3.parte2) – Funções Estáticas:

1 - Crie o pacote **roteiro3.parte2** com a classe **Programa13** conforme o código abaixo. Este programa simula uma calculadora com operações básicas de soma, subtração, multiplicação e divisão.

```
import java.util.Scanner;

public class Programa13 {

    public static void main(String[] args) {

        Scanner entrada = new Scanner(System.in);

        System.out.println("Informe o valor de x ");
        double x = entrada.nextDouble();
        System.out.println("Informe a operação (+, -, *, /) ");
        String op = entrada.next();
        System.out.println("Informe o valor de y ");
        double y = entrada.nextDouble();

        switch (op) {
            case "+" : soma(x, y);
                break;
            case "-" : subtracao(x, y);
                break;
            case "*" : multiplicacao(x, y);
                break;
            case "/" : divisao(x, y);
                break;
            default:
                System.out.println("Operacao invalida");
        }
    }

    public static void soma(double x, double y) {
        System.out.println("Resultado : " + (x + y) );
    }

    public static void subtracao(double x, double y) {
        System.out.println("Resultado : " + (x - y) );
    }

    public static void multiplicacao(double x, double y) {
        System.out.println("Resultado : " + (x * y) );
    }

    public static void divisao(double x, double y) {
        System.out.println("Resultado : " + (x / y) );
    }
}
```

2 – Observe que no Programa13 as funções de soma, subtração, multiplicação e divisão, foram criadas com o retorno “void”. Ou seja, são funções que não geram resultado de retorno para quem as chama. Crie uma classe **Programa14** idêntico ao Programa13, de forma que as funções de soma, subtração, multiplicação e divisão gerem o retorno para a função main, e que o resultado da operação seja apresentado na função main (output) .