

Prática 3

Conteúdos que serão praticados:

- Criação de Classe
- **Definição** de métodos de Classe
- **Instanciação** de objetos
- **Chamada** de métodos de Classe através de objetos instanciados
- Entrada de dados com `Scanner`
- Saída de dados com `System.out.print`
- Passagem de parâmetros na chamada de métodos de Classe
- Retorno de valores após chamada de métodos de Classe

Atividade 1

1. Crie um projeto Java Padrão e dê o nome de `Atividade1`
2. Crie uma Classe e dê o nome de `Usuario`
3. Na Classe `Usuario`, defina um método que **NÃO** seja `static` e que irá apresentar a mensagem "Atividade 1" (`System.out.print` deve ser usado nesse método)
4. Na `main`, instancie um objeto da classe `Usuario` e, usando esse objeto, chame o método definido no item 3

OBSERVAÇÕES:

- O `System.out.print` deve ser usado no método da classe `Usuario` do item 3
- O método definido no item 3 **NÃO DEVE SER** `static`
- NÃO USE o `System.out.print` na `main`

Atividade 2

1. Crie um projeto Java Padrão e dê o nome de `Atividade2`
2. Crie uma Classe e dê o nome de `Usuario`
3. Na classe `Usuario`, defina um método que **NÃO SEJA** `static` e que receba um valor do tipo `String` como parâmetro e mostre esse valor `String` na tela (`System.out.println` deve estar nesse método).
4. Na `main`, instancie um objeto da classe `Usuario` e chame o método definido no item 3 de tal forma que, ao rodar o programa, a mensagem passada por parâmetro apareça na tela

OBS.:

- O método do item 3 **NÃO DEVE SER** `static`

- O `system.out.print` deve estar no método do item 3 (para mostrar os resultados)
 - NÃO use `system.out.print` na `main` (para mostrar os resultados)
-

Atividade 3

1. Crie um projeto Java Padrão e dê o nome de Atividade3
2. Crie uma Classe e dê o nome de `Usuario`
3. Na classe `Usuario`, defina um método que **NÃO SEJA** `static` e que receba um valor do tipo `String` como parâmetro e mostre esse valor `String` na tela (`System.out.println` deve estar nesse método).
4. Na `main`, peça para o usuário digitar o nome e armazene essa mensagem em uma variável local do tipo `String` (`Scanner` deve ser usado na `main`)
5. Na `main`, instancie um objeto da classe `Usuario` e chame o método definido no item 3 passando como parâmetro o nome que o usuário inseriu

OBSERVAÇÃO:

- O método do item 3 **NÃO DEVE SER** `static`
 - O `Scanner` deve ser usado na `main`
 - NÃO use o `Scanner` no método do item 3
 - O `system.out.print` deve ser usado no método definido no item 3
 - NÃO use o `System.out.print` na `main` (para mostrar os resultados)
-

Atividade 4

1. Crie um projeto Java Padrão e dê o nome de `Atividade4`
2. Crie uma Classe e dê o nome de `Usuario`
3. Na classe `Usuario`, defina um método que **NÃO SEJA** `static` (sem parâmetros) e que retorne um valor do tipo `String`; esse método deve pedir para o usuário digitar o nome, armazenar o nome em uma variável local do tipo `String` e retornar seu valor (`Scanner` deve ser usado nesse método)
4. Na `main`, crie uma variável local e armazene nela o retorno da chamada do método do item 3
5. Apresente, na `main`, o conteúdo dessa variável local na tela para o usuário (`System.out.print` deve ser usado na `main`)

OBSERVAÇÃO:

- O `System.out.print` deve ser usado na `main`
- NÃO USE o `System.out.print` no método do item 3 (para mostrar os resultados)
- O `Scanner` deve ser usado no método do item 3
- NÃO USE o `Scanner` na `main`

Atividade 5

1. Crie um projeto Java Padrão e dê o nome de `Atividade5`
2. Crie uma Classe e dê o nome de `Calculadora`
3. Na Classe `Calculadora`, defina os seguintes métodos que **NÃO SÃO** `static`
 - o `soma`: recebe dois parâmetros do tipo `int` e retorna um valor do tipo `int`
 - o `sub`: recebe dois parâmetros do tipo `int` e retorna um valor do tipo `int`
 - o `mult`: recebe dois parâmetros do tipo `int` e retorna um valor do tipo `int`
 - o `div`: recebe dois parâmetros do tipo `float` e retorna um valor do tipo `float`
4. Na `main`, peça para o usuário digitar valores do tipo `int` e armazene esses valores nas variáveis locais `x` e `y` (`Scanner` deve ser usado na `main`)
5. Na `main`, faça a chamada para cada um dos métodos da classe `Calculadora` definidos no item 3 passando os valores de `x` e `y` como parâmetro e apresente os valores retornados por cada um dos métodos na tela (`System.out.print` deve ser usado na `main`)

OBSERVAÇÃO:

- Os métodos do item 3 **NÃO DEVEM SER** `static`
- O `Scanner` deve ser usado na `main`
- NÃO USE o `Scanner` em nenhum dos métodos do item 3
- O `System.out.print` deve estar na `main` (para apresentar os resultados)
- NÃO USE o `System.out.print` nos métodos do item 2 (para apresentar os resultados)