Exercício prático “Constructor”

Um construtor é um bloco de código especial dentro de uma classe que é chamado quando um objeto dessa classe é criado. Ele tem o mesmo nome da classe e não tem tipo de retorno, nem mesmo void.

1. Construtor

- **Uso**: Inicializar novos objetos.
- **Exemplo**:

```
public class Carro {
public Carro() {
        // código para inicializar um novo objeto Carro
    }
}
```

Comandos Relacionados

Aqui estão mais 10 comandos/conceitos frequentemente usados em conjunto com construtores em Java:

- **new**: Usado para criar novos objetos.
- **this**: Referência ao objeto atual.
- **super**: Chama o construtor da superclasse.
- **public/private/protected**: Modificadores de acesso.
- **static**: Pertence à classe, não a uma instância específica.
- **void**: Tipo de retorno que indica que um método não retorna nada.
- **return**: Retorna um valor de um método.
- **if-else**: Estrutura de controle para testar condições.
- **for**: Loop para iterar.
- **class**: Define uma nova classe.

Exercício

Agora, vamos a um exercício com 10 perguntas, uma para cada comando/conceito. Cada pergunta será baseada em um snippet de código relacionado ao comando/conceito explicado.

Pergunta 1: Construtor

```
public class Pessoa {
    public Pessoa() {
        System.out.println("Uma nova pessoa foi criada!");
    }
}
```

Qual é a saída quando um objeto do tipo Pessoa é criado?

Pergunta 2: new

```
Pessoa pessoa = new Pessoa();
```

O que o comando new faz neste código?

Pergunta 3: this

```
public class Conta {
    private double saldo;
    public void depositar(double valor) {
        this.saldo += valor;
    }
}
```

Para que serve a palavra-chave this neste método?

Pergunta 4: super

```
public class Animal {
public Animal() {
        System.out.println("Um animal foi criado.");
    }
}

public class Cachorro extends Animal {
```

```
    public Cachorro() {
        super();
        System.out.println("Um cachorro foi criado.");
    }
}
```

O que acontece quando um objeto do tipo Cachorro é criado?

Pergunta 5: Modificadores de Acesso

```
public class Teste {
    private int valor;
}
```

Qual é o propósito do modificador private neste contexto?

Pergunta 6: static

```
public class Matematica {
    public static int soma(int a, int b) {
        return a + b;
    }
}
```

Como você chamaria o método soma de outra classe sem criar um objeto Matematica?

Pergunta 7: void

```
public void mostrarMensagem() {
System.out.println("Olá, Mundo!");
}
```

O que indica o tipo de retorno void neste método?

Pergunta 8: return

```
public int dobrar(int numero) {
    return numero * 2;
}
```

O que o comando return faz neste método?

Pergunta 9: if-else

```
if (saldo > 0) {
    System.out.println("Saldo positivo");
} else {
    System.out.println("Saldo negativo ou zero");
}
```

Como funciona a estrutura if-else neste snippet?

Pergunta 10: for

```
for(int i = 0; i < 5; i++) {
System.out.println(i);
}
```

O que faz o loop for neste código?

NOTAS:

Recordem-se das cores do Código,

Vermelho

• **Erro ou Aviso:** Alguns IDEs usam o vermelho para sublinhar erros de sintaxe ou avisos. Não foi especificamente usado nos
snippets fornecidos, mas é comum em muitos ambientes de desenvolvimento.

Azul

• **Palavras-chave:** Em muitos IDEs, as palavras-chave da linguagem (como **public**, **class**, **static**, **if**) são destacadas em azul. Isso indica
elementos da linguagem de programação que têm um significado especial e controlam o fluxo do programa.

Verde

• **Comentários:** Os comentários no código são geralmente mostrados em verde. Comentários são usados por desenvolvedores
para deixar notas ou explicar partes do código, e não são executados pelo computador.
Branco

• **Texto Padrão:** O texto que não se enquadra em outras categorias de sintaxe, como o nome de classes ou métodos definidos
pelo usuário, muitas vezes é exibido em branco ou em uma cor de texto padrão.

Rosa ou Magenta

• **Strings:** Textos dentro de aspas, conhecidos como strings, frequentemente aparecem em rosa, magenta ou vermelho. Strings são usadas para representar texto no código.

1.A saída é “Uma nova pessoa fopi criada!”
2.Cria um novo objecto

3. A palavra chave this. Referece a instancia da classe.
4.Vai receber características da classe mãe
5.aquela variável apenas esta disponível na class teste
6.Matematica.soma()
7.que é void ou seja vazio, não retorna nada
8.multiplica o numero por 2
9.se saldo for maior que 0 imprime saldo positivo senão imprime saldo negativo ou zero
10.0 1 2 3 4