Fundamentos de Java
7-2: Parâmetros e métodos sobrecarregados
Atividades Práticas

Objetivos da Lição:

- Usar modificadores de acesso
- Passar objetos para métodos
- Retornar objetos de métodos
- Usar métodos de argumento variáveis
- Sobrecarregar construtores
- Sobrecarregar métodos
- Escrever uma classe com arrays, construtores e métodos específicos

Vocabulário:

Identifique a palavra do vocabulário para cada definição a seguir.

	Um tipo de modificador de acesso. Permite o acesso de qualquer lugar.
	Usado para atribuir valores iniciais a variáveis da instância de uma classe.
	Uma forma de chamar um método com um número de argumentos variável.
	Ter mais de um construtor ou método com o mesmo nome, porém argumentos diferentes.
	Um tipo de modificador de acesso. Permite acesso somente de dentro da mesma classe.
	Um construtor que não tem um parâmetro.
	Usado para especificar a acessibilidade de variáveis, métodos e classes.

Tente/solucione:

1. Crie um segmento de código que inicializa uma classe pública Fish. Deixe a classe conter uma string typeOfFish, e um inteiro friendliness. Não defina valores para essas variáveis ainda. Estas são variáveis de instâncias e serão definidas dentro dos construtores de classes.

2. Crie um construtor público (um método com o mesmo nome que a classe) dentro da classe Peixe. Este construtor não deve considerar argumentos. Dentro do construtor, defina typeOfFish como “Unknown” e friendliness como 3, que presumimos ser a simpatia genérica do peixe.

3. Crie outro construtor público dentro da classe Fish. Faça com que este construtor considere uma string t e um inteiro f. Deixe typeOfFish igual a t, e friendliness igual a f.

4. Explique por que é possível ter mais de um construtor com o mesmo nome e argumentos diferentes.

5. Crie um método dentro da classe Fish chamado getFriendliness que não considera argumentos e retorna o nível de simpatia do peixe.

6. Escreva um segmento de código que inicializa 2 novos peixes, conforme definido abaixo:

    a. Fish 1: Name – *Amber*, Type – *AngelFish,* Friendliness level – 5 (very friendly)

    b. Fish 2: Name – *James*, Type – *Guppy*, Friendliness level – 3 (neutral)

7. Crie um método nicestFish que considere dois peixes como parâmetros, compare o nível de simpatia dos dois peixes e retorne o peixe com o maior nível de simpatia. Teste esse método com o peixe definido no problema 6. (Escala de simpatia: 1 malvado, 2 antipático, 3 neutro, 4 simpático, 5 muito simpático)Dica: fishName.getFriendliness() fornece o número inteiro da simpatia de fishName. Você já criou getFriendliness no problema 5.

8. Modifique o método nicestFish para ser um método de argumento variável que considere um número variável de peixes e retorne o peixe mais simpático dentro os peixes incluídos. Dica: Dentro do método, crie um novo peixe chamado temp. Defina temp igual ao primeiro peixe passado no método. Use um loop positivo para passar por todos os peixes incluídos no método, e se você descobrir um peixe que é mais simpático do que temp, defina temp igual a esse peixe. Após a conclusão do loop, temp deverá ser o peixe mais simpático. Retornar temp.

9. Teste seu método nicestFish com o peixe descrito no problema 6. Qual peixe é retornado?

10. Determine o melhor modificador de acesso para cada uma das seguintes situações:

    a. Uma classe Employee grava o nome, endereço, salário e telefone.

    b. Um método de adição dentro de uma classe BasicMath que também é usado na aula de Álgebra.