

## Fundamentos de Java

### 7-3: Modificador estático e classes

### Atividades Práticas

#### Objetivos da Lição:

- Criar variáveis estáticas
- Usar variáveis estáticas
- Criar métodos estáticos
- Usar métodos estáticos
- Criar classes estáticas
- Usar classes estáticas

#### Vocabulário:

Identifique a palavra do vocabulário para cada definição a seguir.

	É um método disponível para uso sem primeiro criar uma instância da classe. Ela é declarada precedendo a definição com o modificador estático.
	Qualquer classe implementada como uma classe aninhada dentro de outra classe. Por definição, todas as classes internas são membros da classe container por composição.
	Qualquer variável do nível de classe declarada com o modificador estático. Isso significa que somente uma instância da variável de classe pode existir na JVM, independentemente do número de instâncias de classe.
	É uma palavra-chave que disponibiliza uma variável, um método ou uma classe interna sem criar primeiro uma instância de uma variável.
	É uma classe interna. As classes internas são definidas dentro de uma classe pai ou de container e são membros da classe de container por composição. Na verdade, as classes internas são a única forma de criar instâncias de classe por meio de composição.
	É uma classe interna disponível para uso sem primeiro criar uma instância da classe de container. Ela é declarada precedendo a definição com o modificador estático.
	Qualquer método Java definido com um modificador estático. É acessível fora da classe quando um especificador de acesso público, protegido ou padrão o precede. É privado e inacessível fora da classe quando um especificador privado o precede. Os métodos de classe são disponíveis sem primeiro criar uma instância da classe.
	É uma variável que pode estar disponível fora de uma classe sem primeiro criar uma instância de uma classe. Ela é declarada precedendo o nome da variável com o modificador estático.

## Tente/solucione:

1. Crie classes que serão usadas para comparar o uso de variáveis estáticas e de instância.
  - a. Crie um novo pacote denominado “veículos”.
  - b. Crie e implemente a seguinte classe concreta:
    - i. Crie uma classe denominada “Vehicle”.
    - ii. Crie suas variáveis estáticas da seguinte maneira:
      - a. Uma variável String estática pública denominada “MAKE” com um valor “Augur”.
      - b. Uma variável int estática pública denominada “numVehicles” com um valor inicial igual a 0.
    - iii. Crie suas variáveis de instância da seguinte maneira:
      - a. Uma variável String privada denominada “ChassisNo”.
      - b. Uma variável String privada denominada “model”.
    - iv. Crie um construtor que obtenha um único parâmetro formal do tipo String denominado “model”. O construtor deverá executar as seguintes tarefas:
      - a. Incrementar o valor da variável estática “numVehicles” em um algoritmo.
      - b. Definir o valor da variável de instância “chassisNo” igual à concatenação de “ch” com o valor reservado para “numVehicles”.
      - c. Definir a variável de instância “model” igual à variável do parâmetro “model” (dica: você precisará usá-la aqui).
      - d. Fazer com que o construtor exiba uma mensagem que informe “Vehicle manufactured”.
    - v. Implemente dois pares de métodos getter e setter que permitirão a você obter e definir os valores das duas instâncias de variável.
  - c. Crie e implemente a seguinte classe de driver:
    - i. Crie uma classe denominada “TestVehicle”.
    - ii. Crie um método principal estático que teste o seguinte:
      - a. Usando o valor da variável estática “MAKE”, crie a seguinte mensagem de saída:
        - “Manufacturer: Augur”
      - b. Usando o valor da variável estática “numVehicles”, crie a seguinte mensagem de saída:
        - “Number of vehicles manufactured: “
    - iii. Execute seu programa! Você verá que os valores identificados como estáticos são criados no runtime e podem, então, ser acessados.
    - iv. No método principal, use a variável “chassisNo” para criar a seguinte mensagem de saída:
      - “O número do chassi é *chassisNo*”.
    - v. Execute seu programa! Você verá que o programa não será executado. Isso acontecerá porque ainda não criamos uma instância (objeto) dessa classe Vehicle.
    - vi. No seu método principal, crie um objeto veículo denominado vehicle1 acima da instrução de saída do número do chassi.
    - vii. Atualize a instrução de saída do número do chassi para usar a notação de pontos, a fim de identificar o chassi de qual veículo desejamos ver (use vehicle1).
    - viii. Execute seu programa! Você verá que agora o programa mostra as variáveis estáticas e de instância.
    - ix. Atualize o código existente para mostrar também o modelo do carro.
    - x. Crie um segundo veículo (denominado “vehicle2”; use “Edict” como o parâmetro do construtor) e exiba

as variáveis de instância desse objeto.

- xi. Crie um método toString() na classe Vehicle que exibirá o fabricante, o modelo e o número do chassi do veículo na tela usando outra linha de cada instrução de saída.
- xii. Exiba o conteúdo do veículo no método principal usando o método toString().
- xiii. Adicione um método de saída final que exibirá o número total de carros fabricados.

**A saída do programa deverá ser semelhante a esta:**

Manufacturer: Augur

Number of vehicles manufactured: 0

Vehicle manufactured

The vehicle is manufactured by: Augur

The model type is Vision

The chassis number is ch1

Vehicle manufactured

The vehicle is manufactured by: Augur

The model type is Edict

The chassis number is ch2

Number of vehicles manufactured: 2

- 2. Para chamar a atenção para o fato de as variáveis estáticas serem armazenadas em um único espaço de memória que é acessado por cada instância de uma classe, vamos modificar o código criado na pergunta 1.
  - a. No método principal, acima da linha que exibe o número total de carros fabricados, adicione a linha de código a seguir:
    - vehicle2.setMake("Seer");
  - Estamos usando vehicle2 para modificar o valor, mas poderíamos usar qualquer nome de instância.
  - b. Adicione duas instruções de saída abaixo dessa linha que usarão o método toString para exibir o valor de vehicle1 e vehicle2.

**A saída do programa deverá ser semelhante a esta:**

..

..

..

The vehicle is manufactured by: Seer

The model type is Vision

The chassis number is ch1

The vehicle is manufactured by: Seer

The model type is Edict

The chassis number is ch2

Number of vehicles manufactured: 2

3. Usando a solução criada na pergunta 2, você criará uma classe estática aninhada que conterá os detalhes do motor usado dentro no veículo.
  - a. Abra a classe do veículo.
  - b. No método do construtor, crie uma classe pública denominada Engine.
  - c. Crie duas variáveis finais estáticas privadas denominadas MAKE e CAPACITY. A variável Make conterá o texto e a variável Capacity armazenará números inteiros.
  - d. Os valores que deverão ser atribuídos às variáveis são "Predicter" e "1600"
  - e. Não crie um construtor para essa classe.
  - f. Crie dois métodos getter estáticos públicos das duas variáveis criadas.
  - g. Atualize o método toString na classe Vehicle para exibir o fabricante e o modelo do carro. Lembre-se de que Engine é uma classe estática.

**A saída do programa de cada veículo agora deverá ser semelhante a esta:**

The vehicle is manufactured by: Seer

The model type is Edict

The chassis number is ch2

The engine make is Predicter

The engine capacity is 1600cc

4. Você adaptará nosso código para usar uma classe estática interna capaz de retornar informações da instância de sua classe de contêiner.
  - a. Altere sua declaração de Engine para corresponder ao seguinte:
    - `public static class Engine extends Vehicle {`
  - b. Crie um construtor que aceite um parâmetro String denominado model.
  - c. O construtor deverá ter uma única instrução que envie a variável de modelo para o superconstrutor.
  - d. Vá até o método principal e crie um objeto Engine denominado vehicle3 acima da linha de código que exibe o número total de carros fabricados. Para fazer isso, você precisará seguir estas diretrizes:
    - `Outerclass.InnerClass object name = new Outerclass.InnerClass (parameter);`
    - `Vehicle.Engine vehicle3 = new Vehicle.Engine("Fortune");`
  - e. Isso dá acesso tanto aos métodos quanto aos campos da classe interna, bem como a quaisquer métodos e campos em sua classe de delimitador. Crie uma instrução de saída que usará os métodos getter apropriados para exibir o seguinte:
    - `"Vehicle number ch3 is a Fortune model and has an engine capacity of 1600cc"`
  - f. Para ver a diferença entre os diferentes tipos de veículos, tente criar a instrução anterior usando vehicle1 ou vehicle2 (isso não funcionará porque eles não têm acesso aos trabalhos internos da classe estática Engine).