Conceptos fundamentales de Java
7-3: Modificador static y clases anidadas
Actividades de práctica

Objetivos de la lección:

- Crear variables estáticas
- Usar variables estáticas
- Crear métodos estáticos
- Usar métodos estáticos
- Crear clases estáticas
- Utilizar clases estáticas

Vocabulario:

Identifique el término correspondiente a cada una de las siguientes definiciones.

	Es un método que se puede utilizar sin crear primero una instancia de la clase. Se declara al anteponer el modificador static a su definición.
	Es cualquier clase implantada como una clase anidada dentro de otra clase. Por definición, todas las clases internas son miembros de la clase de contenedor por composición.
	Cualquier variable de nivel de clase Java que se declara con el modificador static. Esto significa que solo puede existir una instancia de la variable de clase en la JVM, independientemente del número de instancias de clase.
	Es una palabra clave que permite que una variable, un método o una clase interna esté disponible sin crear primero una instancia de una variable.
	Es una clase interna. Las clases internas están definidas dentro de una clase principal o de contenedor y son miembros de la clase de contenedor por composición. De hecho, las clases internas son la única forma en que puede crear instancias de clase mediante composición.
	Es una clase interna que está disponible para su uso sin crear primero una instancia de la clase de contenedor. Se declara al anteponer el modificador static a su definición.
	Cualquier método de Java definido con un modificador static. Es accesible fuera de la clase cuando se le antepone un especificador de acceso public, protected o default. Es privado e inaccesible fuera de la clase cuando se le antepone un especificador private. Los métodos de clase están disponibles sin crear primero una instancia de la clase.
	Es una variable que puede estar disponible fuera de una clase sin crear primero una instancia de una clase. Se declara al anteponer el modificador static al nombre de variable.

Inténtelo/resuélvalo:

1. Cree clases que se utilicen para comparar el uso de variables static y de instancia.
    a. Cree un paquete nuevo denominado "vehicles".
    b. Cree e implante la siguiente clase concreta:
        i. Cree una clase denominada "Vehicle".
        ii. Cree las variables static de la siguiente forma:
            a. Una variable public static String denominada "MAKE" con un valor "Augur".
            b. Una variable public static int denominada "numVehicles" con un valor inicial 0.
        iii. Cree sus variables de instancia de la siguiente forma:
            a. Una variable private String denominada "ChassisNo".
            b. Una variable private String denominada "model".
        iv. Cree un constructor que tome un único parámetro formal de tipo String llamado "model". El constructor debe ejecutar las siguientes tareas:
            a. Aumente el valor de la variable static "numVehicles" en uno.
            b. Defina el valor de la variable de instancia "chassisNo" igual a la concatenación de "ch" con el valor que tiene "numVehicles".
            c. Defina la variable de instancia "model" igual al valor del parámetro "model" (indicación: tendrá que utilizar esto aquí).
            d. Haga que el constructor muestre un mensaje que indique "Vehicle manufactured".
        v. Implante dos pares de métodos getter y setter que le permitan obtener y definir los valores de las dos variables de instancia.
    c. Cree e implante la siguiente clase de controlador:
        i. Cree una clase denominada "TestVehicle".
        ii. Cree un método main estático que pruebe lo siguiente:
            a. El uso del valor de la variable estática "MAKE" crea el siguiente mensaje de salida:
                - "Manufacturer: Augur"
            b. El uso del valor de la variable estática "numVehicles" crea el siguiente mensaje de salida:
                - "Number of vehicles manufactured: "
        iii. Ejecute el programa. Verá que los valores identificados como estáticos se crean en tiempo de ejecución y, por lo tanto, se puede acceder a ellos.
        iv. En el método main use la variable "chassisNo" para crear el siguiente mensaje de salida:
            - "The chassis number is *chassisNo*".
        v. Ejecute el programa. Verá que el programa no se ejecutará. Esto es porque todavía no hemos creado una instancia (objeto) de la clase Vehicle.
        vi. En el método main, cree un objeto vehicle denominado vehicle1 por encima de la sentencia de salida para el número de chasis.
        vii. Actualice la sentencia de salida de número de chasis para utilizar la notación de puntos para identificar el chasis del vehículo que queremos ver (utilice vehicle1).
        viii. Ejecute el programa. Verá que el programa ahora muestra las variables estáticas y de instancia.
        ix. Actualice el código existente para mostrar también el modelo del coche.

x. Cree un segundo vehículo (denominado "vehicle2", utilice "Edict" como parámetro para el constructor) y visualice las variables de instancia de ese objeto.

xi. Cree un método toString() en la clase Vehicle que mostrará la marca, modelo y número de chasis del vehículo en la pantalla utilizando una línea diferente para cada sentencia de salida.

xii. Visualice el contenido del vehículo en el método main mediante el método toString().

xiii. Agregue un método de salida final que muestre el número total de coches fabricados.

La salida del programa se debe parecer a la siguiente:

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 resaltar el hecho de que las variables estáticas se almacena en un único espacio de memoria al que accede cada instancia de una clase, vamos a corregir el código creado en la pregunta 1.

   a. En el método main, por encima de la línea que muestra el número total de coches fabricados, agregue la siguiente línea de código:

      - vehicle2.setMake("Seer");

      Estamos utilizando vehicle2 para corregir el valor, pero podríamos utilizar cualquier nombre de instancia.

   b. Agregue dos sentencias de salida bajo esta línea que utilicen el método toString para mostrar el valor de vehicle1 y vehicle2.

La salida del programa se debe parecer a la siguiente:

..

..

..

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. Con la solución creada en la pregunta 2, va a crear una clase estática anidada que contendrá los detalles del motor utilizado en el vehículo.
    a. Abra la clase vehicle.
    b. Bajo el método constructor cree una clase public static denominada Engine.
    c. Cree dos variables private static final denominadas MAKE y CAPACITY. Make contendrá texto, mientras que capacity almacenará números enteros.
    d. Los valores que debe asignar a las variables son "Predicter" y "1600"
    e. No cree un constructor para esta clase.
    f. Cree dos public static getters para ambas variables creadas.
    g. Actualice el método toString de la clase Vehicle para mostrar la marca y modelo del motor. Recuerde que Engine es una clase estática.

La salida del programa para cada vehículo se debe parecer ahora a la siguiente:

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. Adaptará el código para utilizar una clase estática interior que pueda devolver información de la instancia de su clase de contenedor.
    a. Cambie la declaración Engine para que coincida con lo siguiente:
        - public static class Engine extends Vehicle {
    b. Cree un constructor que acepte un parámetro String denominado model.
    c. El constructor debe tener una única instrucción que envíe la variable model al superconstructor.
    d. Vaya a su método main y cree un objeto Engine denominado vehicle3 por encima de la línea de código que muestra el número total de coches fabricados. Para ello, tendrá que seguir estas directrices:
        - Outerclass.InnerClass object name = new Outerclass.InnerClass (parameter);
        - Vehicle.Engine vehicle3 = new Vehicle.Engine("Fortune");
    e. Esto le da acceso a los métodos y los campos de la clase interna, así como a los métodos y los campos en su clase delimitadora. Cree una sentencia de salida que utilice los métodos getter adecuados para mostrar lo siguiente:
        - "Vehicle number ch3 is a Fortune model and has an engine capacity of 1600cc"
    f. Para ver la diferencia entre los diferentes tipos de vehículos, intente crear la sentencia anterior con vehicle1 o vehicle2 (no funciona ya que no tienen acceso a los trabajos internos de la clase estática Engine).