Tarea 05: Ejercicios

Programación Avanzada

2023-1

Valeria Jimeno Villegas - vjimenov@ciencias.unam.mx

19 de septiembre de 2022

Ejercicio 5.1: Sobrecarga y Métodos Estáticos

Escribe una clase que sirva como clase de utilería que contenga los métodos necesarios para poder obtener el valor máximo de un arreglo de números. Dicho de otra forma, debemos poder hacer uso de estos métodos desde otra clase sin tener que crear una instancia de esta. Nota: Debe ser capaz de admitir cualquier tipo numérico: int, float, byte, etcétera.

```
public class MiClaseUtil{

// Se hace una sobrecarga del método Max(), el cual regresa el valor
// máximo de un arreglo de números. La sobrecarga se lleva a cabo por que
// se quiere poder admitir arreglos con todos los valores de tipo
// numéricos que existen en java.  Los métodos se define como public
// y static para que pueda ser llamado desde alguna otra clase sin tener
// que hacer una instancia de la clase MiClaseUtil.

        public static int Max(int [] array){
        int maximo = array[0];

        for (int i = 1; i < array.length; i++){
            if (array[i] > maximo)
                maximo = array[i];
        }
        return maximo;
        }

        public static short Max(short [] array){
```

```
short maximo = array[0];

for (int i = 1; i < array.length; i++){
    if (array[i] > maximo)
        maximo = array[i];
}
return maximo;
}

public static byte Max(byte [] array){
byte maximo = array[0];

for (int i = 1; i < array.length; i++){
    if (array[i] > maximo)
        maximo = array[i];
}
return maximo;
}

public static long Max(long [] array){
long maximo = array[0];

for (int i = 1; i < array.length; i++){
    if (array[i] > maximo)
        maximo = array[i];
}
return maximo;
}

public static float Max(float [] array){
float maximo = array[0];

for (int i = 1; i < array.length; i++){
    if (array[i] > maximo)
        maximo = array[i];
}
return maximo;
}

public static double Max(double [] array){
```

```
        double maximo = array[0];

        for (int i = 1; i < array.length; i++){
            if (array[i] > maximo)
                maximo = array[i];
        }
        return maximo;
        }
}
```

Ejercicio 5.2: Métodos y Variables Estáticas

- Crea las clases Inventario y Producto para que la clase Test funcione tal como se muestra y para que su salida sea igual a la que se observa.
- La clase Inventario debe otorgarle un id consecutivo a cada instancia de la clase Producto al momento que estas sean creadas, esto por medio de un método.
- La clase Producto no debe crear instancias de la clase Inventario.

```
public class Producto{

// La clase Producto sólo tendrá un método público( definido así
// para que se pudiera llamar desde la clase Test), el cual regresa
// una llamada al método estático, .ortorgarID() de la clase
// Inventario (ID).

        public Integer getId(){
                return Inventario.otorgarId();
        }
}
```

```
public class Inventario{

// Se crea una variable de tipo entero la cual se inicializa en
// 100 (dado el ejercicio) siendo static para poderla mandar
// llamar dentro del método estático otorgarID, el cual va
// aumentando una unidad la variable ID a medida que se van
//creando más objetos.
```

```
        public static Integer id = 100;
        public static Integer otorgarId(){
                return id++;
        }
}
```

Ejercicio 5.3: Pregunta

¿Qué modificador permite que múltiples variables de una misma instancia sean compartidas a través de todos los objetos de una clase? Explica brevemente un poco sobre este.

El modificador `static`, debido que al definir variables o métodos bajo este modificador estos se vuelvan independiente de los objetos de la clase, es decir que no es necesario instanciar a una clase para poder hacer uso de ellos.