



Contenidos:

- Definición
- Características
- Declaración
- Aplicación
- Ejemplos
- Actividades

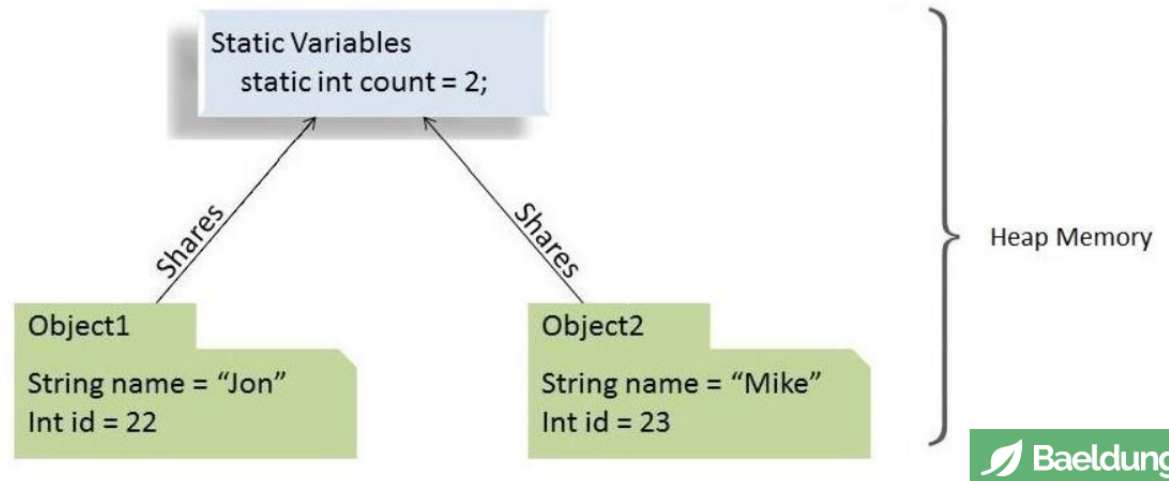


Un método o atributo declarado como estático puede ser accedido o invocado sin la necesidad de instanciar un objeto de su clase.

Es un **recurso compartido entre todos los objetos** creados a partir de una clase.

Pertenece a la clase y no depende de un objeto para usarse.

En la imagen se muestra como dos objetos distintos, que pertenecen a la misma clase, no duplican el valor de “count” en memoria, sino que la comparten.

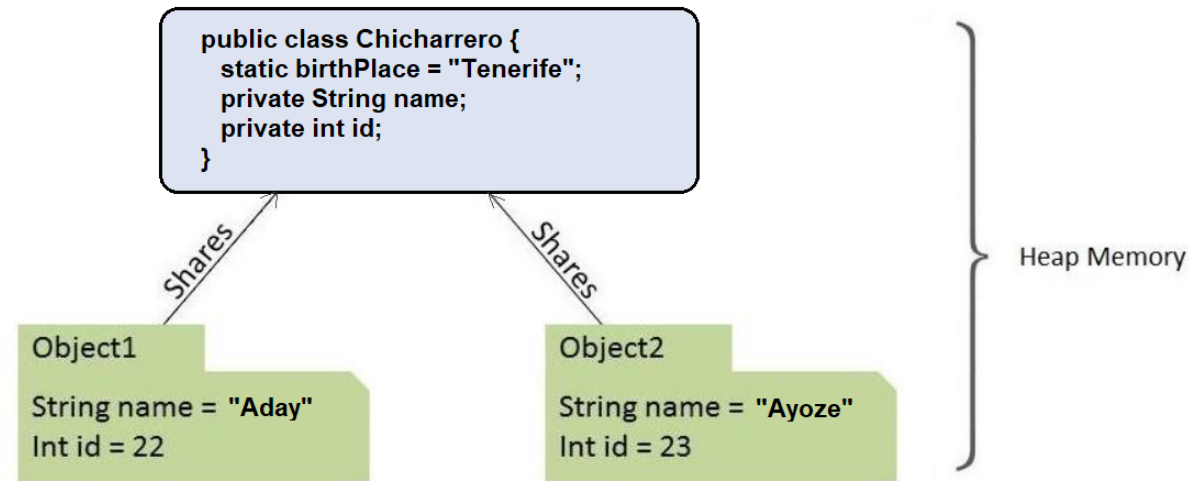


Al compartir un recurso entre distintas instancias de una clase se obtiene una **mejora de eficiencia** en el manejo de la memoria del programa.

Los campos de una clase que se declaran como estáticos **se resuelven en tiempo de compilación**. Es decir, se inicializan en el momento en que se arranca el programa y se mantienen **en un espacio de memoria** dentro de la clase.

Puntos importantes:

- Se puede acceder a ellos desde cualquier parte del código.
- Pueden usarse sin instanciar objetos de la clase que los contiene.
- No requieren el uso de “this”.
- No pueden ser sobrescritos, ni abstractos.





Para indicar que un **atributo** es estático, debemos utilizar como modificador la **palabra** reservada **static** en la declaración del mismo.

```
src > Chicharrero.java > Chicharrero
1  public class Chicharrero {
2
3      static String birthPlace = "Tenerife";
4
5      private String name;
6      private int id;
7
8      public Chicharrero(String name, int idNumber) {
9          this.name = name;
10         this.id = idNumber;
11     }
```




Un atributo declarado como estático **también** puede ser **constante**.

```
src > SalesianosStudent.java > SalesianosStudent > setSchoolName(String)
1  public class SalesianosStudent {
2
3      final static String SCHOOL_GROUP = "Salesianos";
4
5      private String schoolName;
6      private String username;
7
8      public SalesianosStudent(String schoolName, String studentName) {
9          this.schoolName = schoolName;
10         this.username = studentName;
11     }
```



Para indicar que un **método** es estático, debemos utilizar como modificador la palabra reservada **static** en la declaración del mismo.

src >  Chicharrero.java > ...

```
28
29     public static void singAlong() {
30         System.out.println("Chicharreeero, chicharreeero,");
31         System.out.println("Chicharreeero de coorazoooon");
32     }
33
```



Si queremos **acceder** a un atributo o método estático **desde una clase ejecutable**, basta con escribir el nombre de la clase seguido con el miembro en cuestión:

```
public static void main(String[] args) throws Exception {  
  
    System.out.println(Chicharrero.birthPlace);  
    System.out.println(SalesianosStudent.SCHOOL_GROUP);  
  
    Chicharrero.singAlong();  
  
}
```

Tenerife
Salesianos

Chicharreeero, chicharreeero,
Chicharreeero de coorazoooon



Hemos usado muchas referencias estáticas de **clases propias de Java**:

- Los métodos `random()`, `ceil()`, `floor()` o la constante `PI` de la clase `Math`.
- El método `parseInt()` de la clase `Integer`.
- Método `valueOf()` de `String`.

```
public static void main(String[] args) throws Exception {  
  
    String numberStringified = String.valueOf(14);  
  
    int numberParsed = Integer.parseInt(numberStringified);  
  
    double result = Math.PI + numberParsed;  
  
    System.out.println(Math.ceil(result));  
}
```




- Crea una clase SalesianosSchool que contenga los siguientes atributos no estáticos:
id (int), *name* (String)
- Añade un atributo estático y constante llamado “GROUP” con valor “Salesianos”.
- Añade un atributo estático “totalSchools” con valor inicial a cero.
- Crea un constructor para esta clase que inicialice los atributos no estáticos y en el que se añada una unidad al atributo “totalSchools”.
- Ve a la clase ejecutable y accede a “totalSchools” y “GROUP” para mostrarlos por pantalla.
- A continuación, instancia tres objetos de la clase SalesianosSchool con un id cualquiera para los siguientes centros: “La Cuesta”, “La Orotava” y “Hogar Escuela de Santa Cruz”.
- Vuelve a mostrar por pantalla el valor del atributo “totalSchools” y comprueba que ahora su valor es tres.



1) Crea una clase llamada NumberComparator que contenga los siguientes métodos estáticos:

- `int getBigger(int numberOne, int numberTwo)`
- `boolean areEquals(int numberOne, int numberTwo)`
- `int getMinimum(int[] numbersArray)`
- `boolean isNegative(int number)`

2) Ve a la clase ejecutable y comprueba que esos métodos funcionan accediendo a ellos de manera estática para mostrar por pantalla los siguientes textos:

- “Entre 14 y 47, el mayor es 47”.
- “Los números 5 y 6 son DISTINTOS, pero 7 y 7 son IGUALES”.
- “En el array [12, 8, 45, -13, 67, 21, 190] el número más pequeño es -13”.
- “El número 6 es POSITIVO, pero -10 es NEGATIVO”.

¿Conoces los QuBits?

