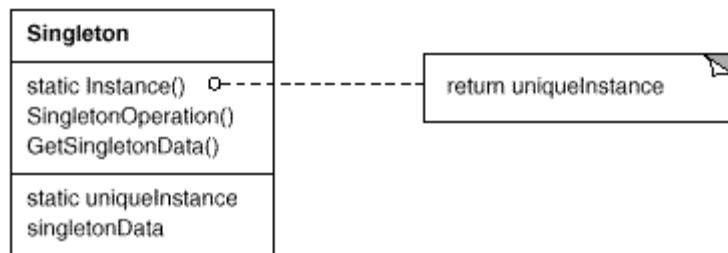


## PATRÓN DE DISEÑO: SINGLETON

**PROBLEMA:** Se necesita:

- que una clase tenga como máximo una **instancia única**
- y proporcionar **acceso global a dicha instancia**.

**SOLUCIÓN:**



- **Constructor privado** para que no se pueda acceder a él, y por tanto no se puedan crear nuevas instancias de la clase (`private Singleton()`).
- **Instancia única estática**, para que sea una instancia de clase (`private static Singleton uniqueInstance`)
- **Método estático** que proporciona acceso único a la única instancia de la clase (`static Instance()`)

Los métodos y variables estáticas son de la clase, no de las instancias.

### IMPLEMENTACIÓN EN JAVA:

```

public class Singleton {

    private static Singleton uniqueInstance;
    // puede tener otras variables de instancia (no estáticas)

    private Singleton() {}

    public static Singleton getInstance() {
        if (uniqueInstance == null) {
            uniqueInstance = new Singleton();
        }
        return uniqueInstance;
    }
    // puede tener otros métodos de instancia (no estáticos)

}
  
```

Para conseguir una instancia de la clase se utilizará el método **estático** `getInstance()`. No se puede llamar al constructor, al ser privado.

```
Singleton instancia = Singleton.getInstance();
```

`getInstance()` si todavía no se ha creado una instancia, llama al constructor para crearla, pero si ya existe una instancia, la devuelve (instanciación perezosa).

**PRÁCTICA A REALIZAR:**

1. **Descargar el proyecto** de eclipse PatronSingleton\_VAT desde ahora.
2. **Importar el proyecto descargado** en eclipse como un nuevo proyecto. (File->Import->General -> Existing Projects into Workspace).
3. **Ejecutar como aplicación JAVA** el fichero Main.java, comprobando la salida por consola. Dará un aviso de que hay errores de compilación en el proyecto (Ejecutarlo igualmente).  
**Comprobar en la consola** que hay tres instancias de la clase VATtoApply con distintos valores de VAT.
4. **Modificar la implementación de la clase VATtoApply** para que se comporte como global, obligando a que **como máximo exista una instancia única** de dicha clase (**APLICAR EL PATRÓN SINGLETON**).
5. **Hacer los cambios necesarios en la clase Main**, para que no haya errores sintácticos. (Sustituir las llamadas al constructor de la clase VATtoApply por llamadas a getInstance).
6. **Comprobar** que el test **VAT\_Test** del mismo proyecto se ejecuta satisfactoriamente.

```
1 package ule.iso1.singleton;
2
3
4
5 public class VATtoApply {
6
7     private double vatType;
8
9     /**
10      * MODIFICAR PARA CONSEGUIR QUE SEA SINGLETON
11      */
12
13     public VATtoApply(double vat) {
14         vatType=vat;
15     }
16
17
18
19
20     public double getVatType() {
21         return vatType;
22     }
23
24     public void setVatType(double vatType) {
25         this.vatType = vatType;
26     }
27
28 }
```