



TEMPLE

UNIVERSITY®

Thanh Nguyen • Adjunct Instructor • Temple University CIS

<https://refactoring.guru/design-patterns/singleton>

22 common steps

Implementations of Singleton

- Make the **default constructor private** - prevent other objects using the **new** operator with the Singleton class.
- Create a **static creation method** that acts as a **constructor** - This method calls the private constructor to create an object and saves it in a static field. All following calls to this method **return the cached object**.

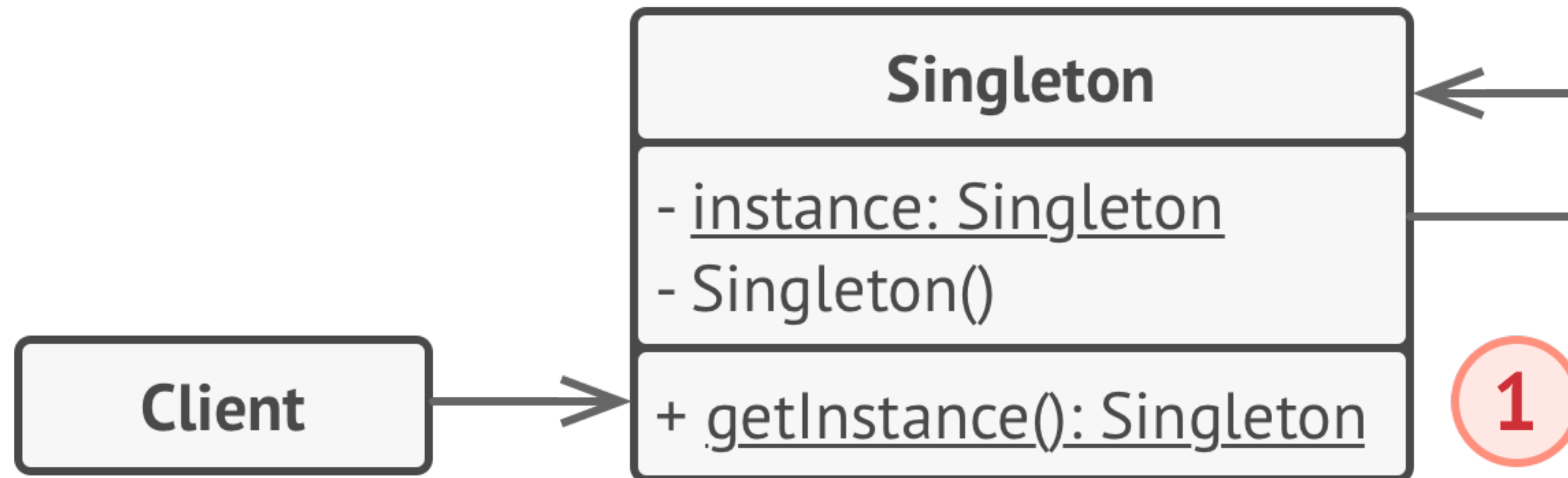
If your code has access to the Singleton class, then it's able to call the Singleton's static method. So whenever that method is called, the same object is always returned.

Implementations of Singleton

2 common steps

- Make the **default constructor private** - prevent other objects using the **new** operator with the Singleton class.
- Create a **static creation method** that acts as a **constructor** - This method calls the private constructor to create an object and saves it in a static field. All following calls to this method **return the cached object**.

If your code has access to the Singleton class, then it's able to call the Singleton's static method. So whenever that method is called, the same object is always returned.



```
if (instance == null) {  
    // Note: if you're creating an app with  
    // multithreading support, you should  
    // place a thread lock here.  
    instance = new Singleton()  
}  
return instance
```