

[GoF] Singleton

Leonardo Bizzoni

June 23, 2024

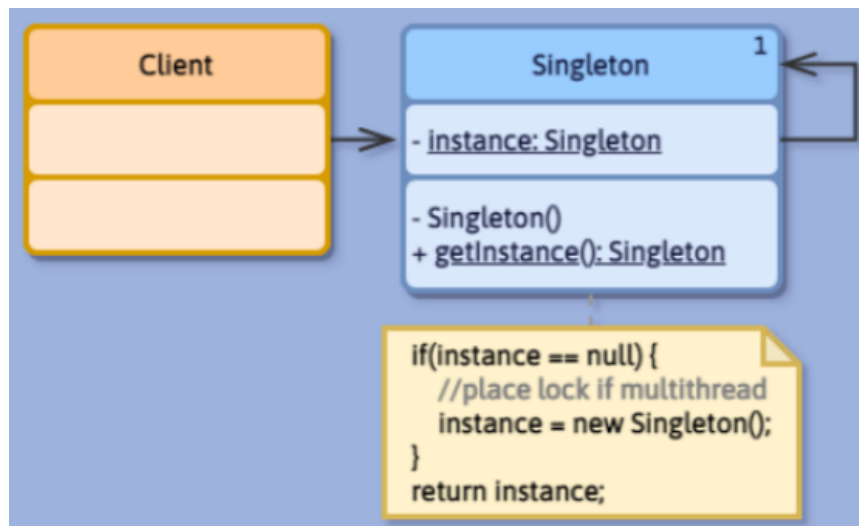
Questo pattern viene spesso utilizzato insieme al factory.

1 Problema

È consentita (o *richiesta*) esattamente una sola istanza di una classe, ovvero un **singleton**. Gli altri oggetti hanno bisogno di un punto di accesso globale e singolo a questo oggetto.

2 Soluzione

Definisci un metodo statico alla classe che restituisce l'oggetto singleton.



2.1 Esempio

```
use std::sync::Mutex;
static BURGER_FACTORY: Mutex<SimpleBurgerFactory> = Mutex::new(SimpleBurgerFactory::new());

enum BurgerType { Beef, Veggie }

trait Burger {
    fn prepare(&self);
}

struct Restaurant;
impl Restaurant {
    fn orderBurger(&self, request: BurgerType) -> Box<dyn Burger> {
        let factory = SimpleBurgerFactory::new();

        let burger = factory.createBurger(request);
        burger.prepare();
        burger
    }
}

struct SimpleBurgerFactory;
impl SimpleBurgerFactory {
    fn new() -> Self {
        Self { }
    }

    fn createBurger(&self, of_type: BurgerType) -> Box<dyn Burger> {
        match of_type {
            BurgerType::Beef => Box::new(BeefBurger),
            BurgerType::Veggie => Box::new(VeggieBurger),
        }
    }
}

struct BeefBurger;
impl Burger for BeefBurger {
    fn prepare(&self) {
        // Implementation
    }
}
```

```
    }  
}  
  
struct VeggieBurger;  
impl Burger for VeggieBurger {  
    fn prepare(&self) {  
        // Implementation  
    }  
}
```