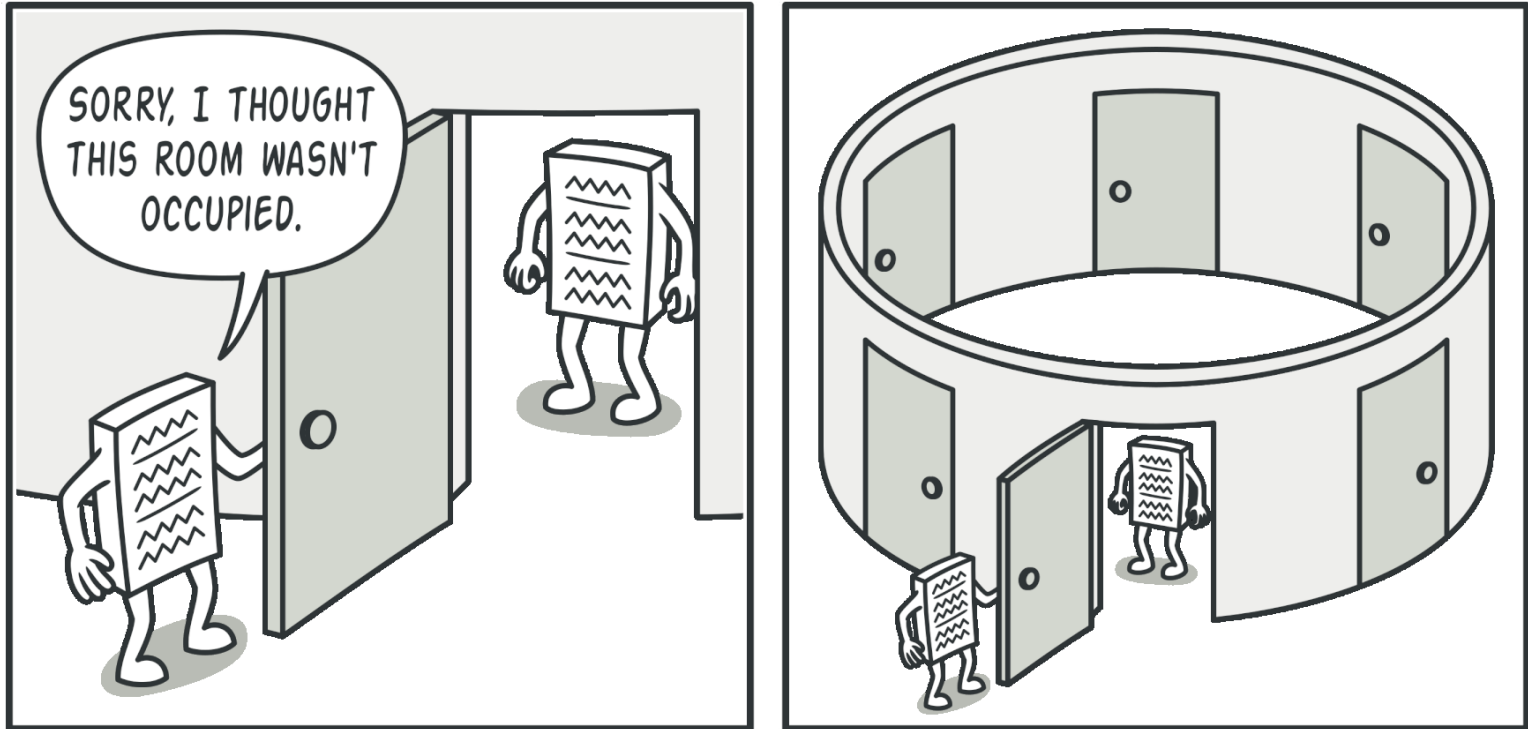


Singleton

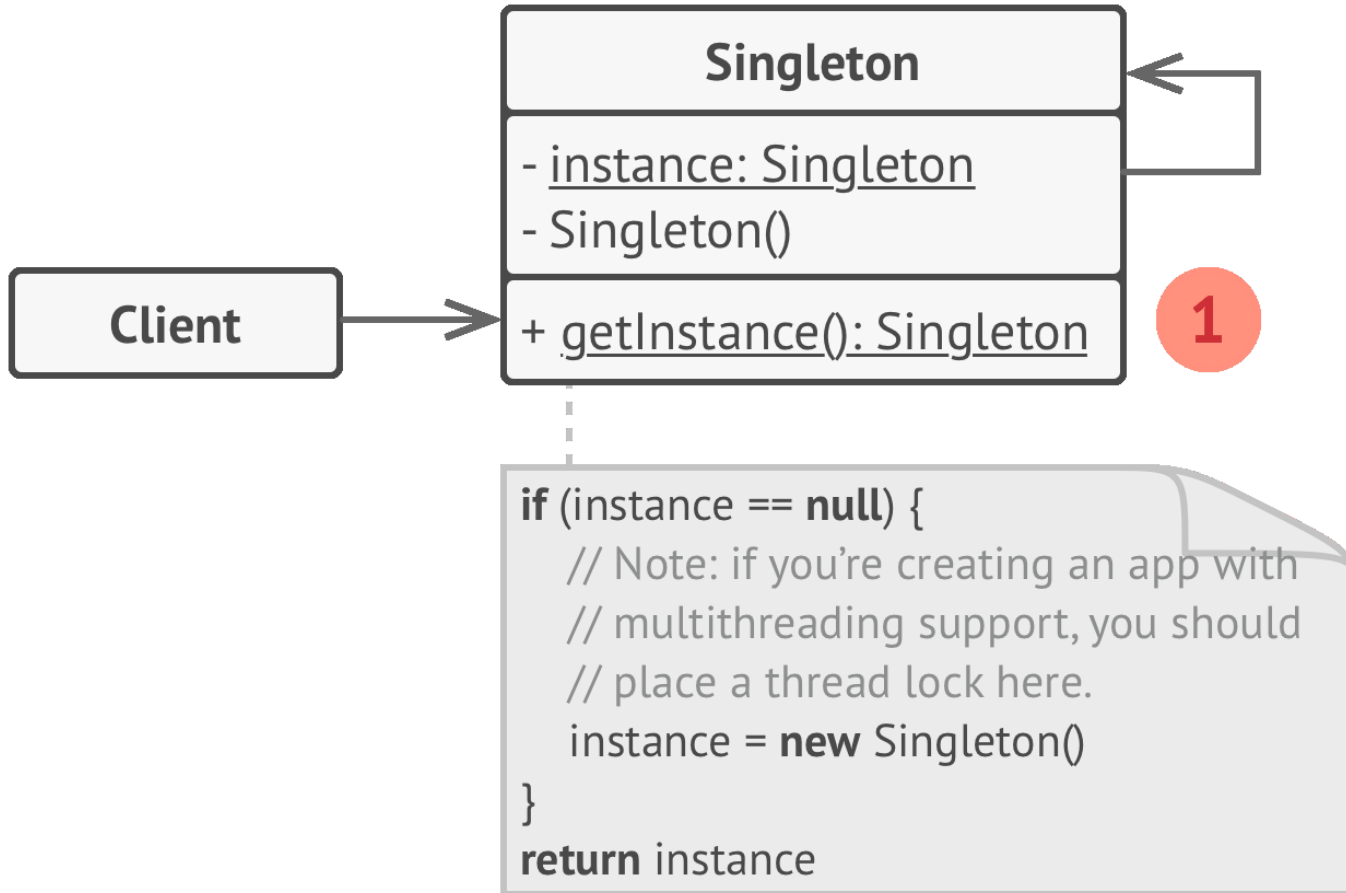
Singleton is a creational design pattern that lets you ensure that a class has only one instance, while providing a global access point to this instance.

Problem



Clients may not even realize that they're working with the same object all the time.

Solution



Application

- Use the Singleton pattern when a class in your program should have just a single instance available to all clients; for example, a single database object shared by different parts of the program.
- Use the Singleton pattern when you need stricter control over global variables.

Pros & Cons

- You can be sure that a class has only a single instance.
- You gain a global access point to that instance.
- The singleton object is initialized only when it's requested for the first time.
- Violates the *Single Responsibility Principle*. The pattern solves two problems at the time.
- The Singleton pattern can mask bad design, for instance, when
 - the components of the program know too much about each other.
 - The pattern requires special treatment in a multithreaded environment so that multiple threads won't create a singleton object several times.