Singleton

Creational Design Pattern

**What is a Singleton:**

1. A singleton class has only one instance, accessible globally through a single point.
2. Main problem this pattern solves is to ensure that only a single instance of this class exists.
3. Any state you add in your singleton becomes part of global state of your application

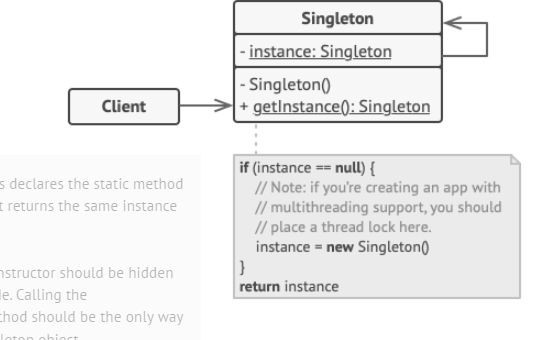
**Implementing a Singleton:**

1. Class constructors must be not accessible globally
2. Subclassing/inheritance must not be allowed
3. Keep track of instance (in the class itself)
4. Giving access to the singleton instance; Public static method

**Types:**

1. Eager Singleton - Early initialization; create singleton as soon as class is loaded.
2. Lazy Singleton - Lazy initialization; Singleton is created when it is first required.

**Structure:**



**Implementing:**

1. Create a class with an empty constructor
2. Create a class parameter that is of the same type as the class itself and assign it.
3. Create method getInstance that returns this parameter.