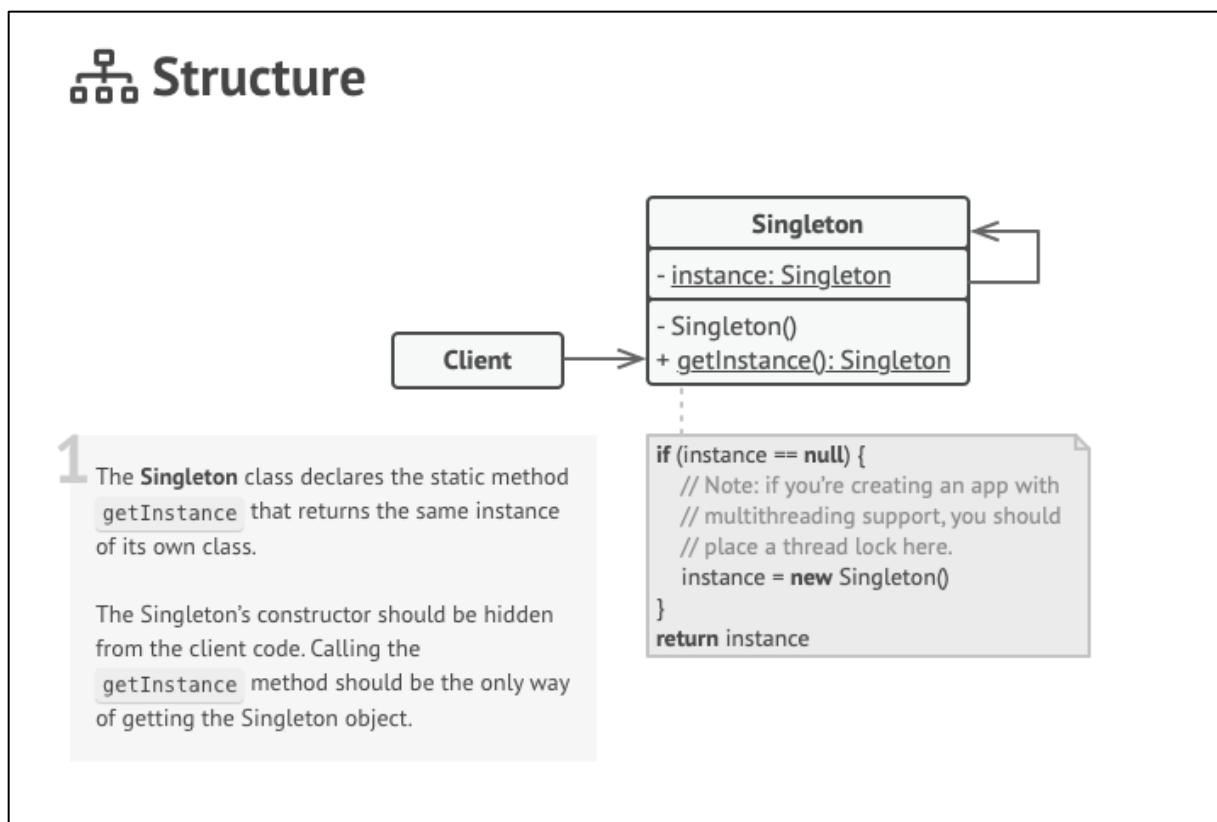


Das Singleton-Pattern

Bei manchen Klassen ist es wichtig, dass es genau ein Exemplar gibt. Obwohl es in einem System viele Drucker geben kann, sollte es nur einen Druckerspooler geben. Es sollte nur ein Dateisystem geben und nur eine Fensterverwaltung geben. Nur wie stellen wir sicher, dass es eine Klasse über genau ein Exemplar verfügt und das einfach auf dieses Exemplar zugegriffen werden kann? Eine globale Variable ermöglicht den Zugriff auf ein Objekt, verhindert aber nicht das Erzeugen mehrerer Exemplare.

Es ist besser, die Klasse selbst für die Verwaltung ihres einzigen Exemplars zuständig zu machen. Die Klasse kann durch Abfangen von Befehlen zur Erzeugung neuer Objekte sicherstellen, dass kein weiteres Exemplar erzeugt wird und sie kann die Zugriffsmöglichkeiten auf das Exemplar anbieten. Dies ist die Essenz des Singletonmusters.

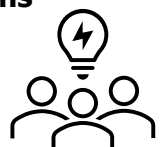
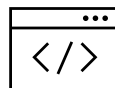


<https://refactoring.guru/design-patterns/1>



Arbeitsauftrag

1. **Informieren** Sie sich über das **Singleton-Pattern**. Nutzen Sie dafür gerne die **Hilfestellung**, welche Sie bei den vorherigen Entwurfsmustern bekommen haben oder eigene seriöse Quellen.
2. Setzen Sie das **Singleton-Pattern** mittels **SQL-Datenbank-Verbindung** in der **Programmiersprache Ihres Projektes** um. Nehmen Sie Ihre **Lösung mit ins QA-Gespräch** mit der Lehrkraft.



Datenbank-Beispiel zum Testen**SQL-Datei**<https://t1p.de/67ymu>**Hilfestellung bei Umsetzung****Git-Link für Beispiel in JAVA**<https://t1p.de/5pi03>**Refactoring Guru**<https://t1p.de/s7w4m>