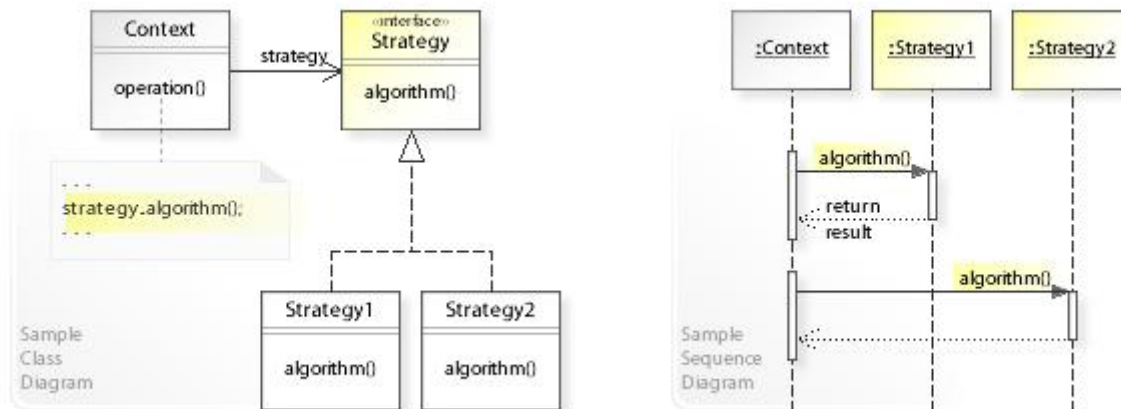


# Seminar 2

1. In einer Kiste befinden sich Sägen, Hämmer, Nägel und Scheren. Implementiere eine Klasse Filter mit 2 Methoden:
  - A. eine Methode, die alle Entitäten ausgibt, deren Gewicht größer als ein Kilogramm ist.
  - B. eine Methode, die alle Entitäten ausgibt, die schneiden können.

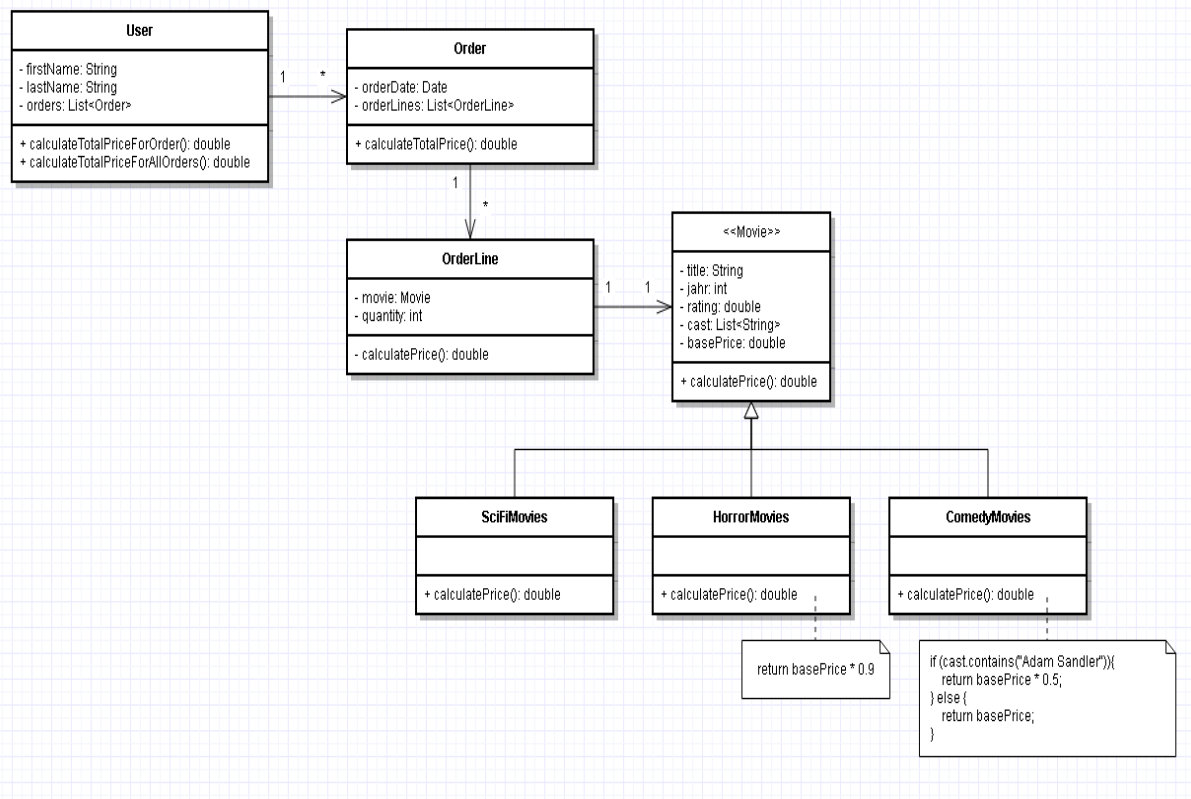
## 2. Strategy Pattern (Beispiel)



Die Klasse **Strategie** definiert nur eine Schnittstelle für alle unterstützten Algorithmen. Die Implementierung der eigentlichen Algorithmen findet sich erst in den Erweiterungen wieder. Der **Kontext** hält eine Variable der Schnittstelle Strategie, die mit einer Referenz auf das gewünschte Strategieobjekt belegt ist. Auf diese Weise wird der konkrete Algorithmus über die Schnittstelle eingebunden und kann bei Bedarf selbst zur Laufzeit noch dynamisch gegen eine andere Implementierung ausgetauscht werden.

3. Ein Benutzer möchte eine kleine Anwendung haben, wo er mehrere Filme bestellen kann. Eine Bestellung kann mehrere Filme beinhalten, bzw. ein Film kann in einer Bestellung mehrmals erscheinen.

- Jeder Film hat einen Titel, ein Erscheinungsjahr, eine Auswertung, eine Liste mit Schauspielern und einen Preis.
- Es gibt drei Arten von Filmen: **Sci-Fi-Filme**, **Horror-Filme** und **Komödien**.
- Der Preis eines Films hängt von seiner Art ab.
  - o Sci-Fi-Filme werden mit dem normalen Preis kalkuliert.
  - o Horror-Filme haben 10% Rabatt.
  - o Komödien, in denen Adam Sandler spielt, haben 50% Rabatt.
- Der Benutzer hat die Möglichkeit den endgültigen Preis einer Bestellung bzw. den Preis von allen bisherigen Bestellungen zu kalkulieren.



- A. Implementiere mit der Anwendung von Strategy die folgenden Klassen: **Movie**, **SciFiMovies**, **HorrorMovies**, **ComedyMovie**. **Movie** wird als eine abstrakte Klasse implementiert.
- B. Implementiere die Klasse **OrderLine** mit der Methode **calculatePrice()**.