

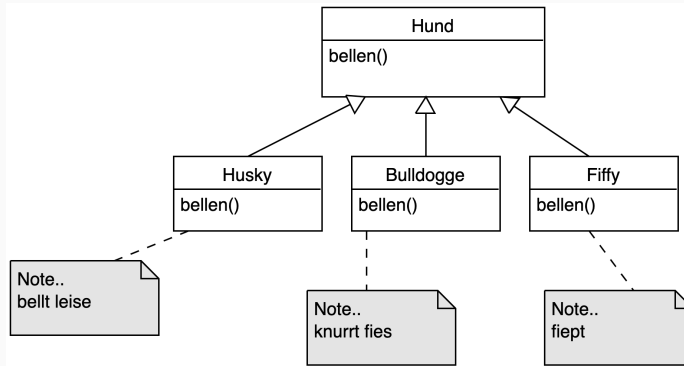
# Strategy-Pattern

---

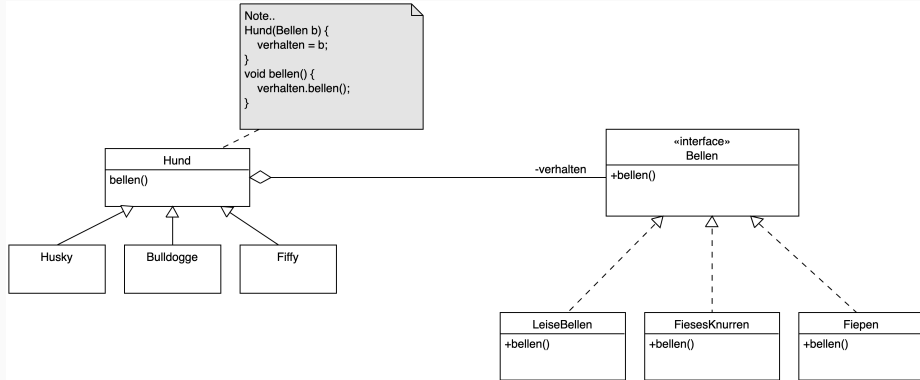
Carsten Gips (FH Bielefeld)

Unless otherwise noted, this work is licensed under CC BY-SA 4.0.

# Wie kann man das Verhalten einer Klasse dynamisch ändern?



# Lösung: Delegation der Aufgabe an geeignetes Objekt



Entwurfsmuster: **Strategy Pattern**

# Hands-On: Strategie-Muster

Implementieren Sie das Strategie-Muster für eine Übersetzungsfunktion:

- Eine Klasse liefert eine Nachricht (`String`) mit `getMessage()` zurück.
- Diese Nachricht ist in der Klasse in Englisch implementiert.
- Ein passendes Übersetzerobjekt soll die Nachricht beim Aufruf der Methode `getMessage()` in die Ziel-Sprache übersetzen.

1. Wie muss das Pattern angepasst werden?
2. Wie sieht die Implementierung aus?

Strategy-Pattern: Verhaltensänderung durch Delegation an passendes Objekt

- Interface oder abstrakte Klasse als Schnittstelle
- Konkrete Klassen implementieren Schnittstelle => konkrete Strategien
- Zur Laufzeit Instanz dieser Klassen übergeben (Aggregation) ...
- ... und nutzen (Delegation)

# LICENSE



Unless otherwise noted, this work is licensed under CC BY-SA 4.0.