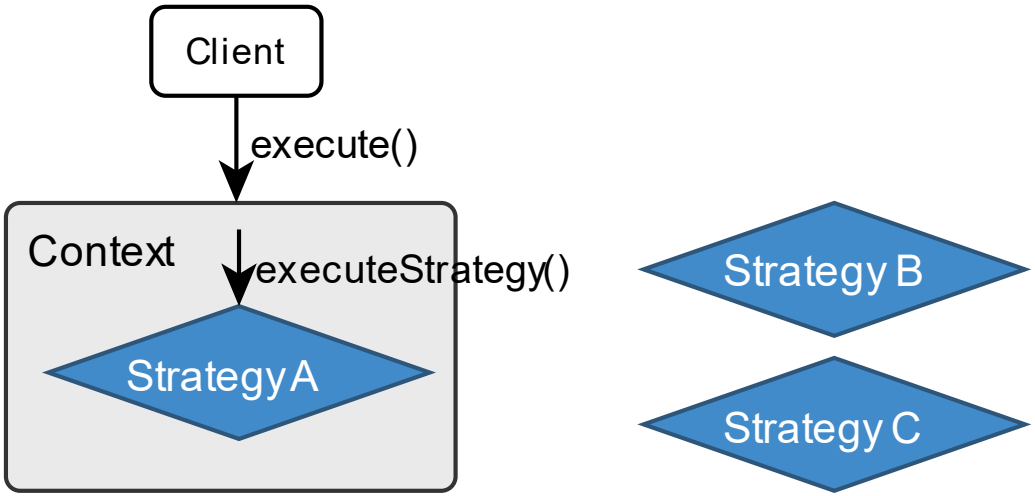
TBZ
Technische Berufsschule Zürich

Modul 326
Strategy Design Pattern

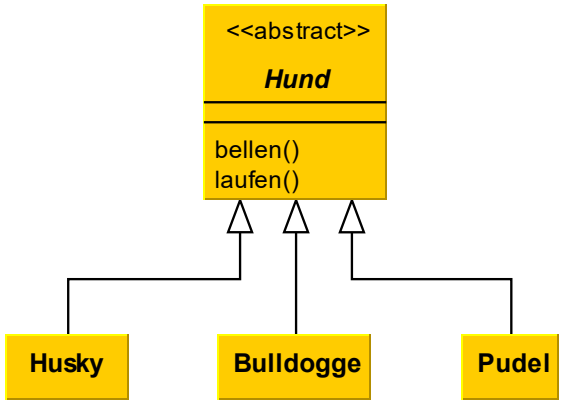
D. A. Waldvogel

12.03.2018

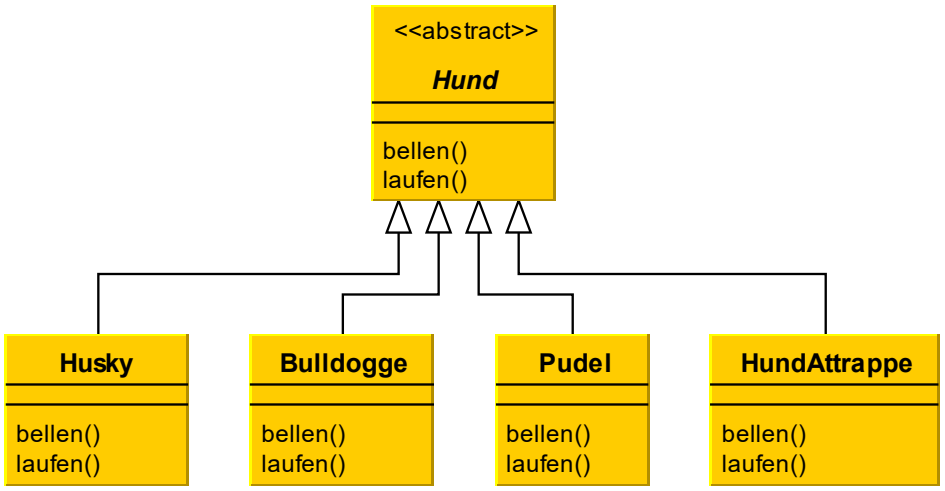
Flexibilität durch Strategien



Vererbung



Vererbung hat Nachteile



Nachteile diese Entwurfs

TBZ
Technische Berufsschule Zürich

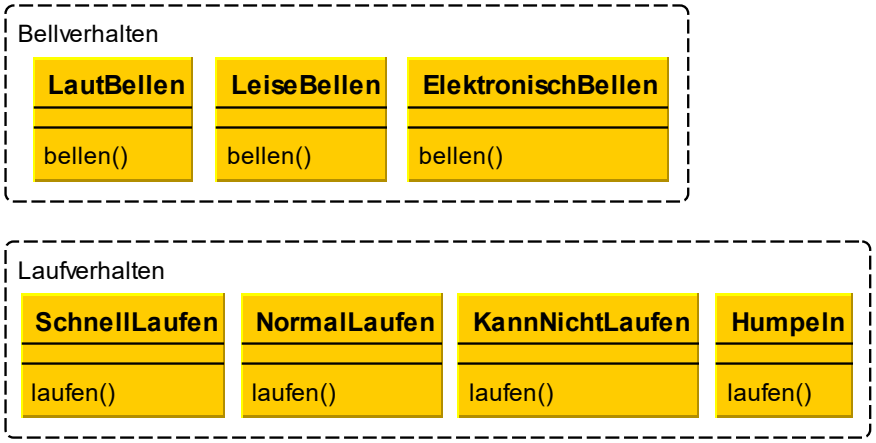
- bellen() und laufen() werden immer vererbt
- Code Redundanz
- Verhalten der Hunde zur Laufzeit nicht einfach zu ändern
- Wiederverwendbarkeit eingeschränkt möglich
- Keine allgemeine Aussagen über das Verhalten von Hunden möglich

Änderungsstabilität

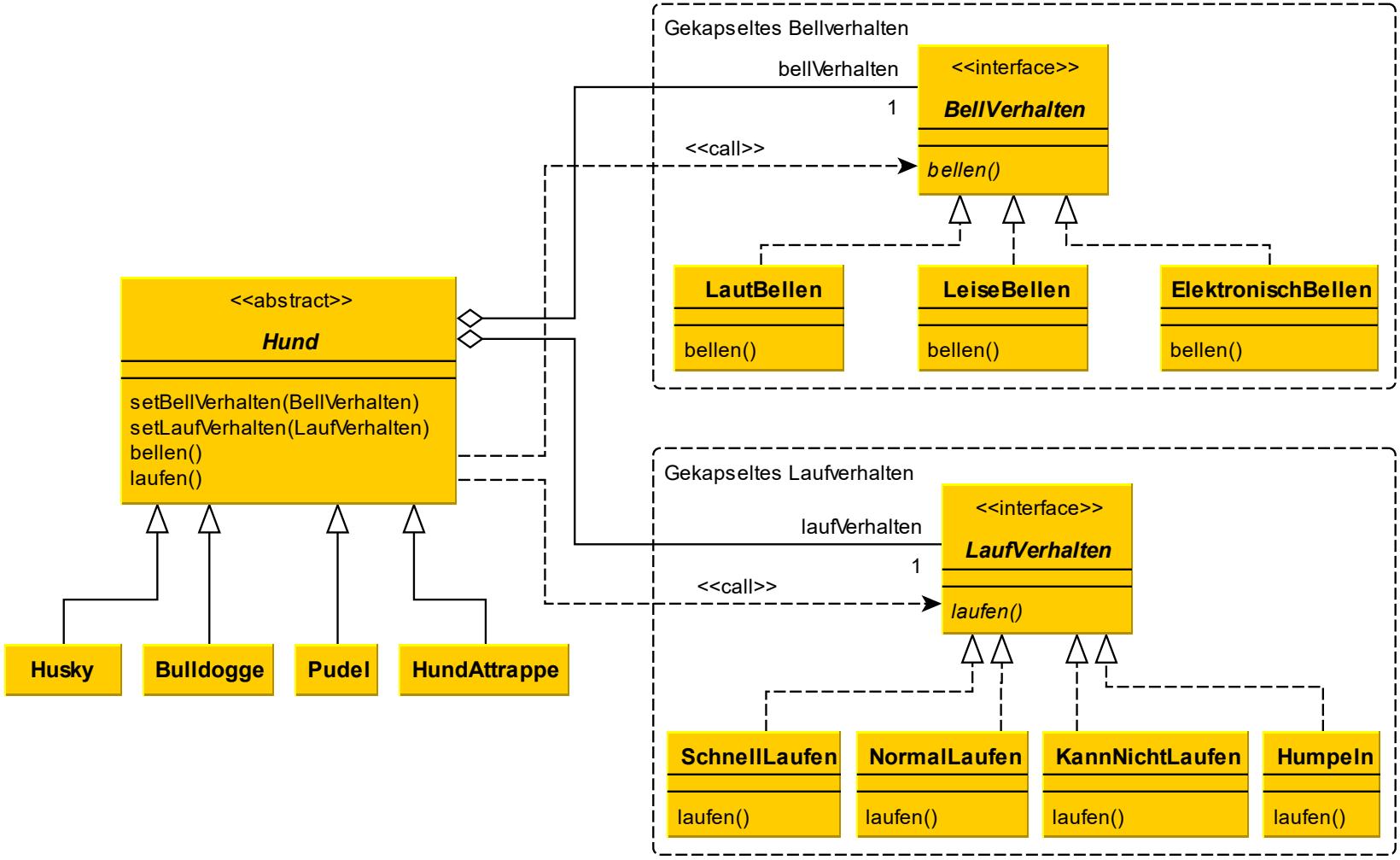
TBZ Technische Berufsschule Zürich

Entwurfsprinzip:

Identifiziere jene Aspekte, die sich ändern und trenne sie von jenen, die konstant bleiben



Lösungsansatz mit Delegation



Delegation

Technische Berufsschule Zürich

Delegation von Verhalten an das Verhaltensobjekt:

```
public void bellen(){
        bellVerhalten.bellen();
}
```

```
public class Husky extends Hund {
    //Defaultverhalten: LeiseBellen
    private BellVerhalten bellVerhalten = new LeiseBellen();

    public void bellen(){
        bellVerhalten.bellen();
    }

    public void setBellVerhalten(BellVerhalten bellVerhalten){
        this.bellVerhalten = bellVerhalten;
    }
}
```

Abstrakte Hunde Klasse

Technische Berufsschule Zürich

```
abstract class Hund {

    //Instanzvariablen vom Typ des Interfaces. Defaultverhalten
    BellVerhalten bellVerhalten = new LautBellen();
    LaufVerhalten laufVerhalten = new SchnellLaufen();


    public void setBellVerhalten(BellVerhalten bellVerhalten) {
        this.bellVerhalten = bellVerhalten;
    }


    public void setLaufVerhalten(LaufVerhalten laufVerhalten) {
        this.laufVerhalten = laufVerhalten;
    }

    public void bellen(){
        //Delegation des Verhaltens an Verhaltensobjekt
        bellVerhalten.bellen();
    }

    public void laufen(){
        //Delegation des Verhaltens an Verhaltensobjekt
        laufVerhalten.laufen();
    }
}
```

Beispiel client

TBZ
Technische Berufsschule Zürich

```java
public class Client {
    public static void main(String[] args) {
        Husky husky = new Husky();
        husky.bellen(); //ganz leises bellen...
        husky.laufen(); //Schnelles laufen
        husky.setLaufVerhalten(new Humpeln());
        husky.laufen(); //Humpeln
        //...
    }
}
```

Flexibilität und Dynamik

TBZ
Technische Berufsschule Zürich

- Alternative zur Unterklassenbildung
- Hund ist von seinem Verhalten entkoppelt
- Keine Code Redundanz
- Wiederverwendbarkeit möglich
- Verhalten kann zur Laufzeit geändert werden
- Es lassen sich Aussagen über das allgemeine Verhalten von Hunden sagen

Aufträge

- Bauen Sie diese Hundebeispiele mit eigenen Klassen nach (sie dürfen auch andere Tiere nehmen, wenn Sie Hunde nicht mögen)
  - Version Vererbung
  - Version Delegation
- Erweitern Sie Ihr Portfolio mit einem konkreten Beispiel