Gutes Design:
Design-Prinzipien, Design Pattern

- Prinzipien für ein gutes Design
- Design Pattern / Entwurfsmuster
  - Sinnhaftigkeit
  - Kataloge
  - Beispiele

DHBW Duale Hochschule Baden-Württemberg Mannheim

Vorgehen beim Entwurf: es gibt leider *keine* Kochrezepte!

- Ein Entwurfsvorgehen, das immer passt, ***gibt es nicht***!
- Ob ein Entwurf(sergebnis) gut ist, zeigt sich bei seinem Einsatz
- Häufig verwendet man einen *objektorientierten Entwurf*

Sonstige Hilfestellungen

- Bestimmte *Prinzipien* beachten
  -> nächster Abschnitt
- Bei „typischen“ Problemen
  bekannte / etablierte (Lösungs-)*Muster* einsetzen
  -> übernächster Abschnitt

Objektorientiertes Design (OOD)

- Prinzip

  Jedes Objekt / jedes Ding (jede "Art" von Objekt) in der Realität wird durch ein Objekt (eine Klasse von Objekten) im Modell repräsentiert

- Ein Objekt...
  - besitzt einen *Zustand*:
    Der Zustand wird durch die Werte der Objektattribute dargestellt
  - hat eine Menge von *Operationen*:
    Operationen sind Dienste (Methoden), die dieses Objekts anderen Objekten anbietet

- Die Erzeugung eines Objekts...
  - erfolgt gemäß der Definition einer *Klasse*

Acht Prinzipien für ein gutes Design 1/2

- Abstraktion
  - Weglassen unwichtiger Eigenschaften -> größere Allgemeingültigkeit
- Modularisierung und ...
  - Zerlegen in beherrschbare Einheiten
- ... Kapselung
  - Verbergen der Details einer Einheit

Acht Prinzipien für ein gutes Design 2/2

- **S**ingle Responsibility Principle
  - Jedes Objekt hat (nur) eine einzelne Verantwortung (die aber ganz)
- **O**ffen-geschlossen-Prinzip
  - Schnittstelle vollständig, erweiterbar ohne Änderungen
- **L**iskov‘sches Substitutionsprinzip
  - Klassen müssen ihre Oberklassenklassen (ohne Einschränkung) ersetzen können
- **I**nterface Segregation Principle
  - Aufteilen großer Schnittstellen in mehrere kleine, spezialisierte Schnittstellen
- **D**elegation
  - Verantwortlichkeit für eine bestimmte Aufgabe an eine andere Klasse „übergeben“ ohne das LSP zu verletzen

DHBW Duale Hochschule Baden-Württemberg Mannheim

F R A G E N

Ursprung von „Mustern“ in der Informatik

- Ursprung:
  Muster bei Gebäudeentwürfen von
  C. Alexander (*A Pattern Language, 1977*)
- Muster „gibt es“:
  - Sie werden *nicht erfunden*,
  - sondern *erkannt* (und dokumentiert)
- Von Erich Gamma (gang of four) auf die Informatik übertragen
- Im Großen:
  Architektur-Stile, Architecture Pattern
- Im Kleinen:
  Entwurfsmuster, Design Pattern

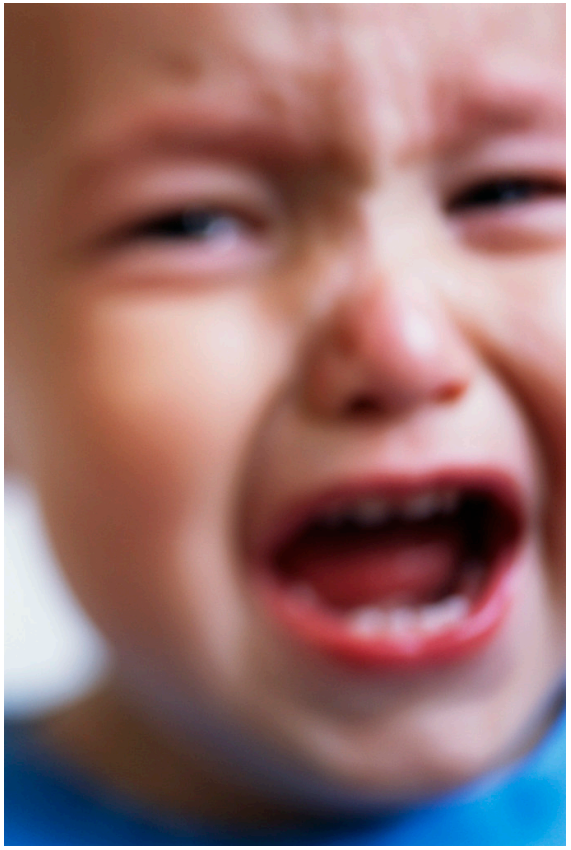
Eigenschaften von Mustern

- Problembeschreibung + Kern der Problemlösung + Konsequenzen (pos. + neg.)
- Keine genaue Spezifikation (möglich)
- Die Lösung kann in verschiedenen Umgebungen wieder verwendet werden
- Es gibt auch Entwurfsmuster-Kataloge vieler anderer Autoren und Muster aus anderen Gebieten (nicht nur Entwurf)
- Ermöglichen *Wiederverwendung* auf Entwurfsebene

Vorteile von Mustern

- Entwurfsmuster können…
  - die Umsetzung nicht-funktionaler Anforderungen (z.B. Änderbarkeit, Zuverlässigkeit) unterstützen
  - ein Vokabular für den Entwurf schaffen
    - erleichtern Dokumentation
    - erleichtern Kommunikation
    - erleichtern Verständnis, ermöglichen schnelleres Einarbeiten

Enttäuschungen bzgl. Mustern



- Entwurfsmuster können ***nicht***…
  - Sind kein Allheilmittel
  - Sind zwar einfach zu verstehen, aber man braucht Erfahrung, um die Entwurfsmuster *sinnvoll* einzusetzen