

# Software-Entwurfsmuster

*Prinzip von Entwurfsmustern und einige  
elementare Beispiele*

Malte Spiess

[malte@mathematik.uni-ulm.de](mailto:malte@mathematik.uni-ulm.de)

Seminar „Bildanalyse und Simulation mit Java“ im WS 2003/2004  
Universität Ulm

# Entwurfsmuster - was ist das?

Jedes Muster beschreibt ein in unserer Umwelt **beständig wiederkehrendes Problem** und erläutert den **Kern der Lösung** für dieses Problem, so dass Sie diese Lösung **beliebig oft anwenden** können, ohne sie jemals ein zweites Mal gleich auszuführen.

Christopher Alexander

# Vorteile/Motivation der Entwurfsmuster

## Entwurfsmuster leisten:

- höhere **Flexibilität**
  - bessere **Wiederverwendbarkeit**
  - leichter zu **erweitern**
  - **Vereinheitlichung** der Begriffe, einfacherer Sprachgebrauch
- ⇒ bessere Entwicklung, Wartung und Dokumentation

# Vorteile/Motivation der Entwurfsmuster

## Entwurfsmuster leisten:

- höhere **Flexibilität**
- bessere **Wiederverwendbarkeit**
- leichter zu **erweitern**
- **Vereinheitlichung** der Begriffe, einfacherer Sprachgebrauch

⇒ bessere Entwicklung, Wartung und Dokumentation

⇒ ähnliche Motivationen wie für objektorientiertes Programmieren allgemein

# Motivation Beispiel: FFT

## *Problematik*

FFT (Fast Fourier Transformation) wird oft benötigt  
⇒ Implementierung wird oft benutzt

### **mögliche Probleme:**

- bisherige Implementierung **zu langsam**
- bisherige Implementierung **zu ungenau**
- es gibt **neue und bessere Algorithmen**
- man möchte **mehrere Versionen gleichzeitig** nutzen  
(z. B. je nach Größe)

# Motivation Beispiel: FFT

## *Problematik*

FFT (Fast Fourier Transformation) wird oft benötigt  
⇒ Implementierung wird oft benutzt

### **mögliche Probleme:**

- bisherige Implementierung **zu langsam**
- bisherige Implementierung **zu ungenau**
- es gibt **neue und bessere Algorithmen**
- man möchte **mehrere Versionen gleichzeitig** nutzen  
(z. B. je nach Größe)

**ABER:** bisherige Implementierung ist direkt in das System eingebunden und nur schwer zu ersetzen

# Motivation Beispiel: FFT

## *Lösungskonzept*

### **Lösung:**

Entwurfsmuster, in diesem Fall: am besten **Fabrikmethode**

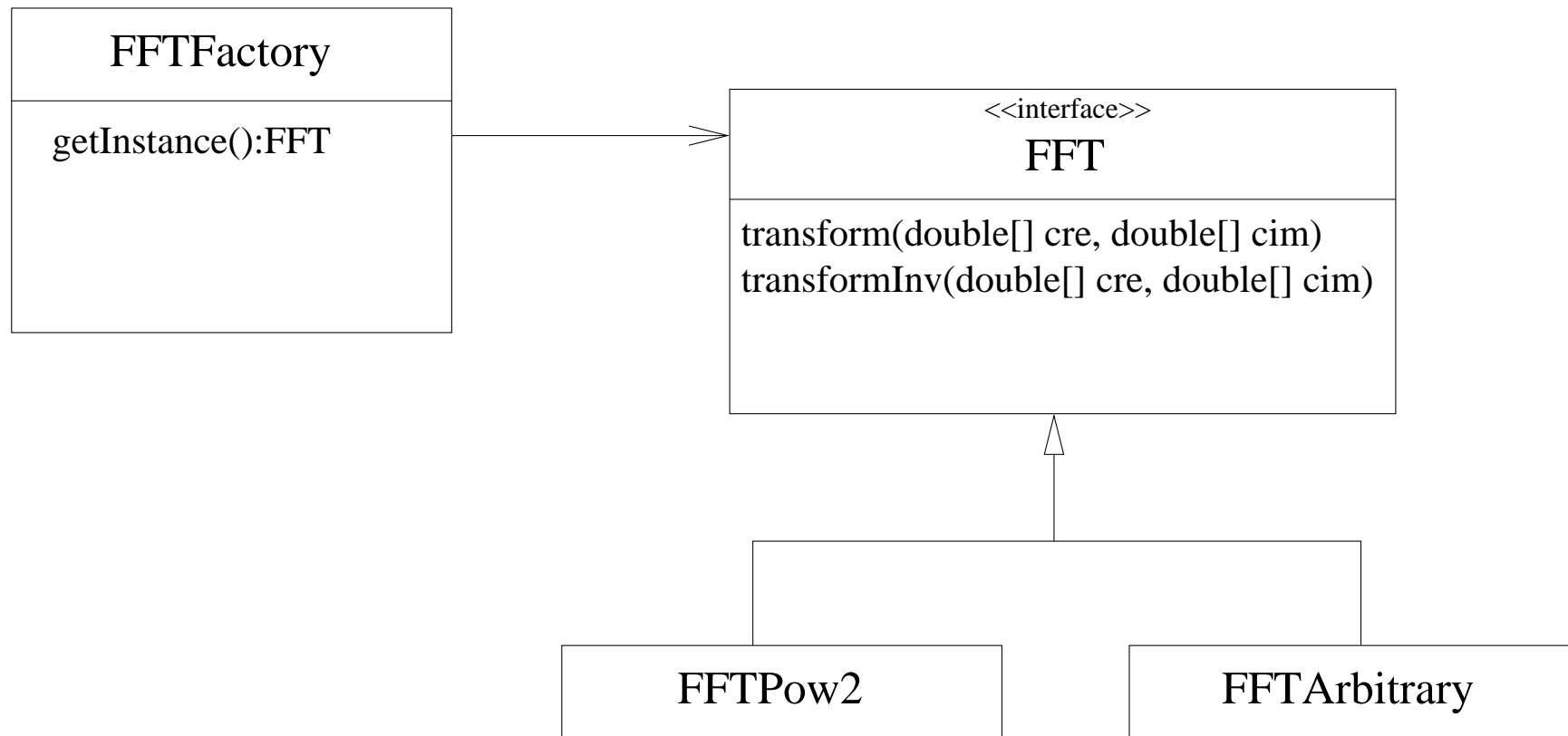
### **Dann:**

**Interface** definiert nur die **Schnittstellen** für das Objekt. Implementierung ist **nicht festgelegt**, Benutzung bleibt davon unabhängig und ist immer gleich.

# Motivation Beispiel: FFT

## *Geostoch-Implementierung - Schaubild*

So wird diese Idee zum Beispiel in der Geostoch-Bibliothek umgesetzt:





# Motivation Beispiel: FFT

## *Geostoch-Implementierung - Quellcode*

Hier der dazugehörige Quellcode:

```
public class FFTFactory {
    public static FFT getInstance(int m) {
        // compute n = 2^k minimal with n >= m
        ...
        if (m == 1)
            return new FFT() {
                public void transform(double[] cre, double[] cim) {
                    // nothing to do!
                }
                public void transformInv(double[] cre, double[] cim) {
                    // nothing to do!
                }
            };
        else if (n == m) // size is power of 2
            return FFTPow2.getInstance(m);
        else // arbitrary size > 1
            return FFTArbitrary.getInstance(m);
    }
    /** There should be no instance of this factory class. */
    private FFTFactory() {}
}
```

# Entwurfsmuster

## *Problemstellung und Lösungsansatz*

Erfahrene Programmierer haben gute Kenntnisse darüber, wie man Klassen sinnvoll anordnet u. s. w.

⇒ **Frage:**

Wie vermittelt man Neueinsteigern das Wissen?

# Entwurfsmuster

## *Problemstellung und Lösungsansatz*

Erfahrene Programmierer haben gute Kenntnisse darüber, wie man Klassen sinnvoll anordnet u. s. w.

⇒ **Frage:**

Wie vermittelt man Neueinsteigern das Wissen?

**Idee:**

**Muster** beschreiben gängige Vorgehensweisen von erfahrenen Software-Designern

⇒ Die Kenntnis dieser Muster ermöglicht **schnelles Erlernen komplexer Denkstrukturen.**

Also keine neuen Erkenntnisse im Sinne der bisherigen Methoden, sondern eine neue Herangehensweise an bekannte Probleme.

# Entwurfsmuster: Detaillierte Beschreibung

## *Aufbau einer Musters*

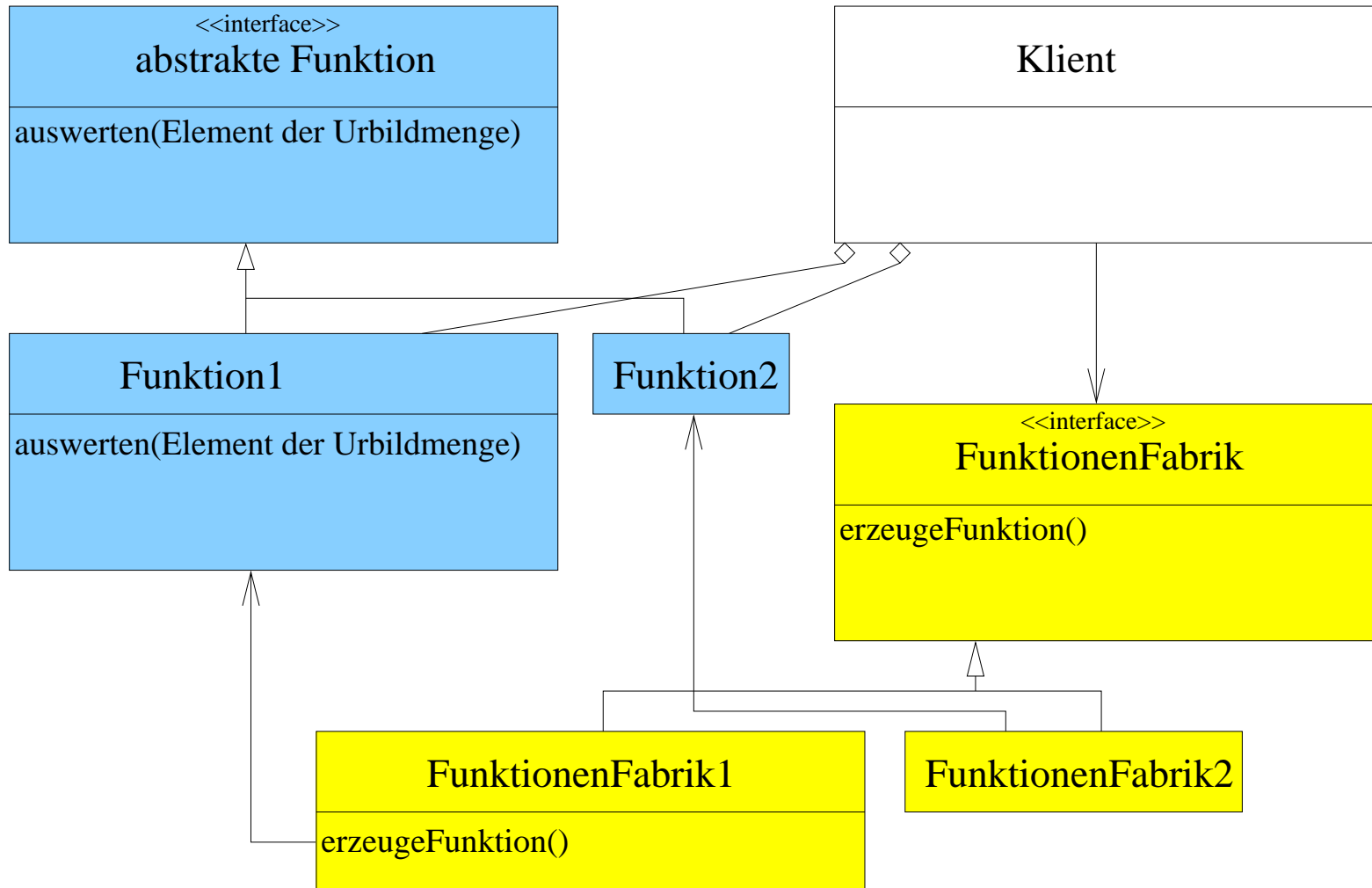
ein Muster besteht aus:

1. **Mustername**
2. **Problemabschnitt**  
(Problembeschreibung)
3. **Lösungsabschnitt**
4. **Konsequenzenabschnitt**

# abstrakte Fabrik / Fabrikmethode

## Beispiel: Funktionen

Anwendungsbeispiel: Funktionen (z. B. Variogramme)



# abstrakte Fabrik / Fabrikmethode

## *Eigenschaften*

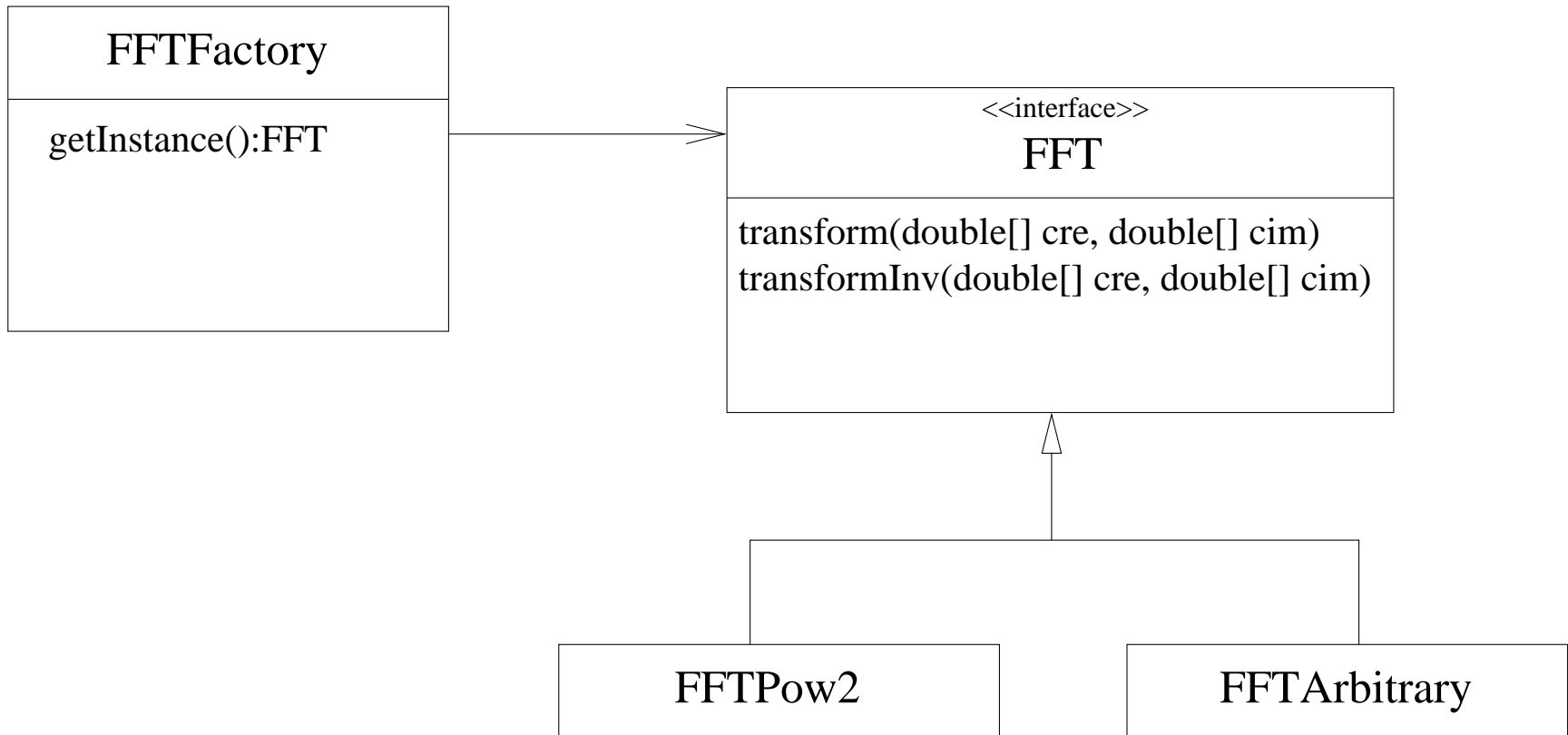
**allgemein:** Schnittstelle zum Erzeugen von Familien ähnlicher Objekte, ohne sie konkret zu benennen

### **Vorteile:**

- Erzeugung von Objekten **ohne direkten Zugriff** (d.h. Kenntnis) der Klassen
- **einfacher Austausch** (nur eine Verwaltungsklasse)
- **einfacher Überblick und Kontrolle** über erzeugte Objekte

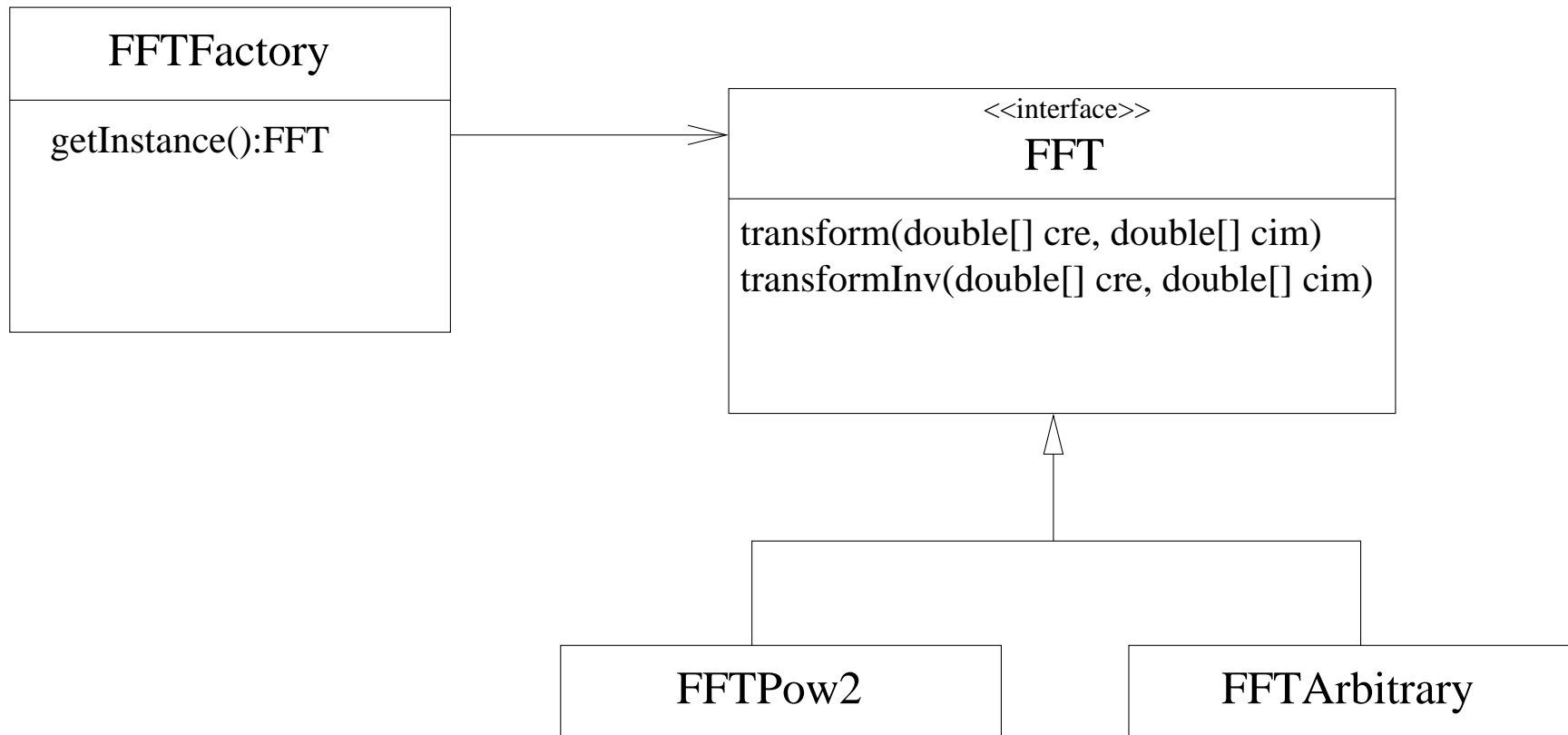
# Singleton

*Beispiel: FFT in der Geostoch, Eigenschaften*



# Singleton

*Beispiel: FFT in der Geostoch, Eigenschaften*



## Vorteile:

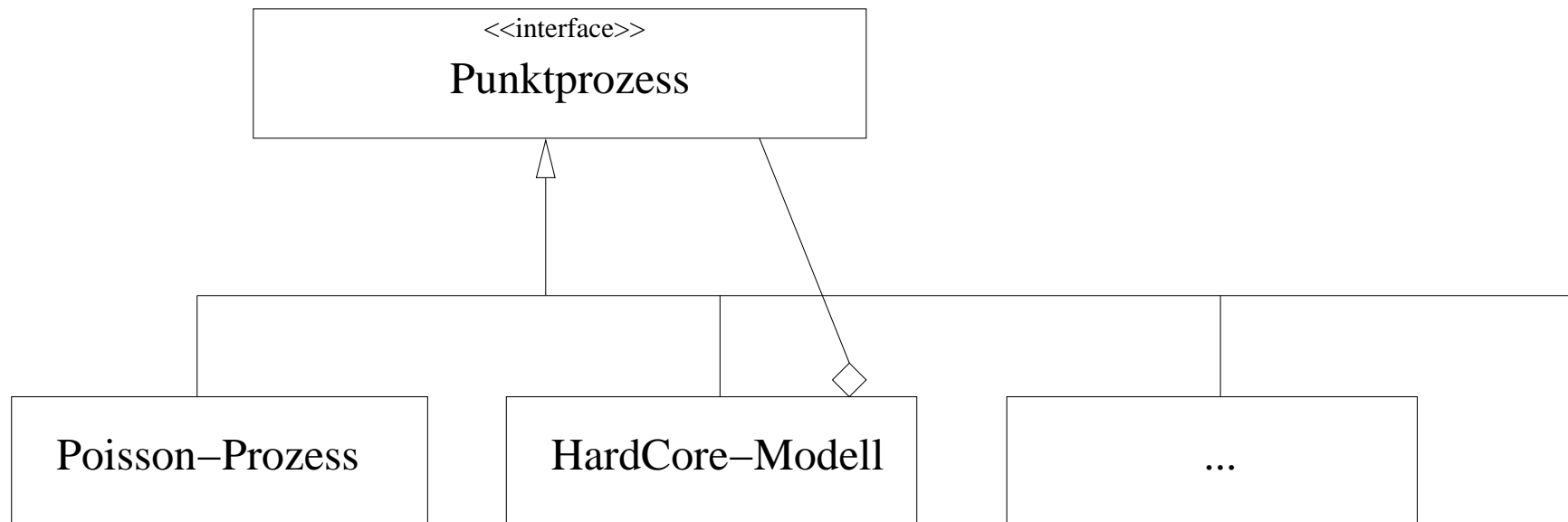
- **Spezialisierung / Veränderung leicht**



# Dekorierer

## *Beispiel: Punktprozesse / Hardcore-Modell*

ein schönes Beispiel zum Dekorierer liefert das Hardcore-Modell:



# Dekorierer

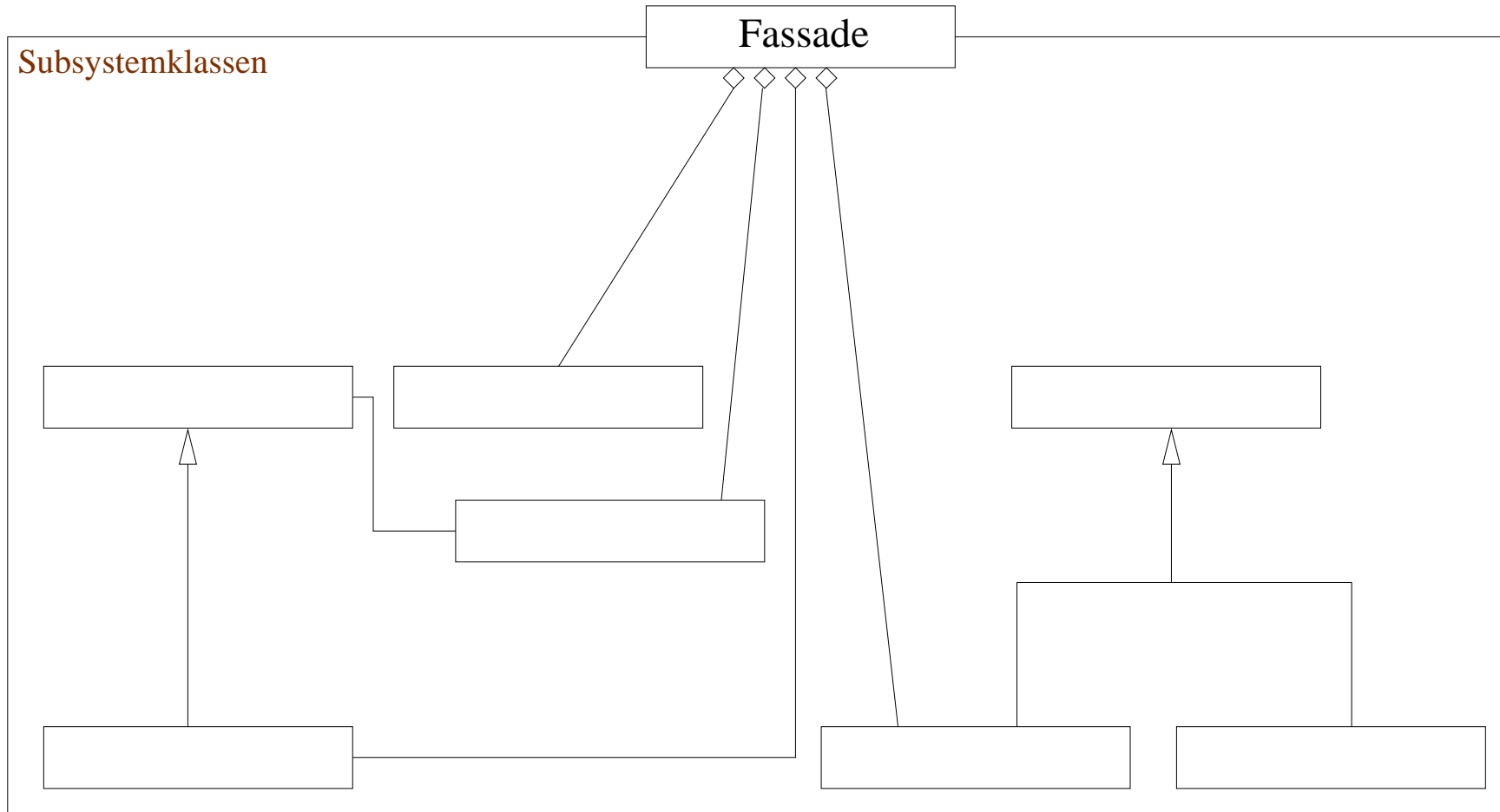
## *Eigenschaften*

### **Vorteile:**

- hohe **Flexibilität**
- Funktionalität nur **bei Bedarf**
- gute **kombinatorische** Möglichkeiten

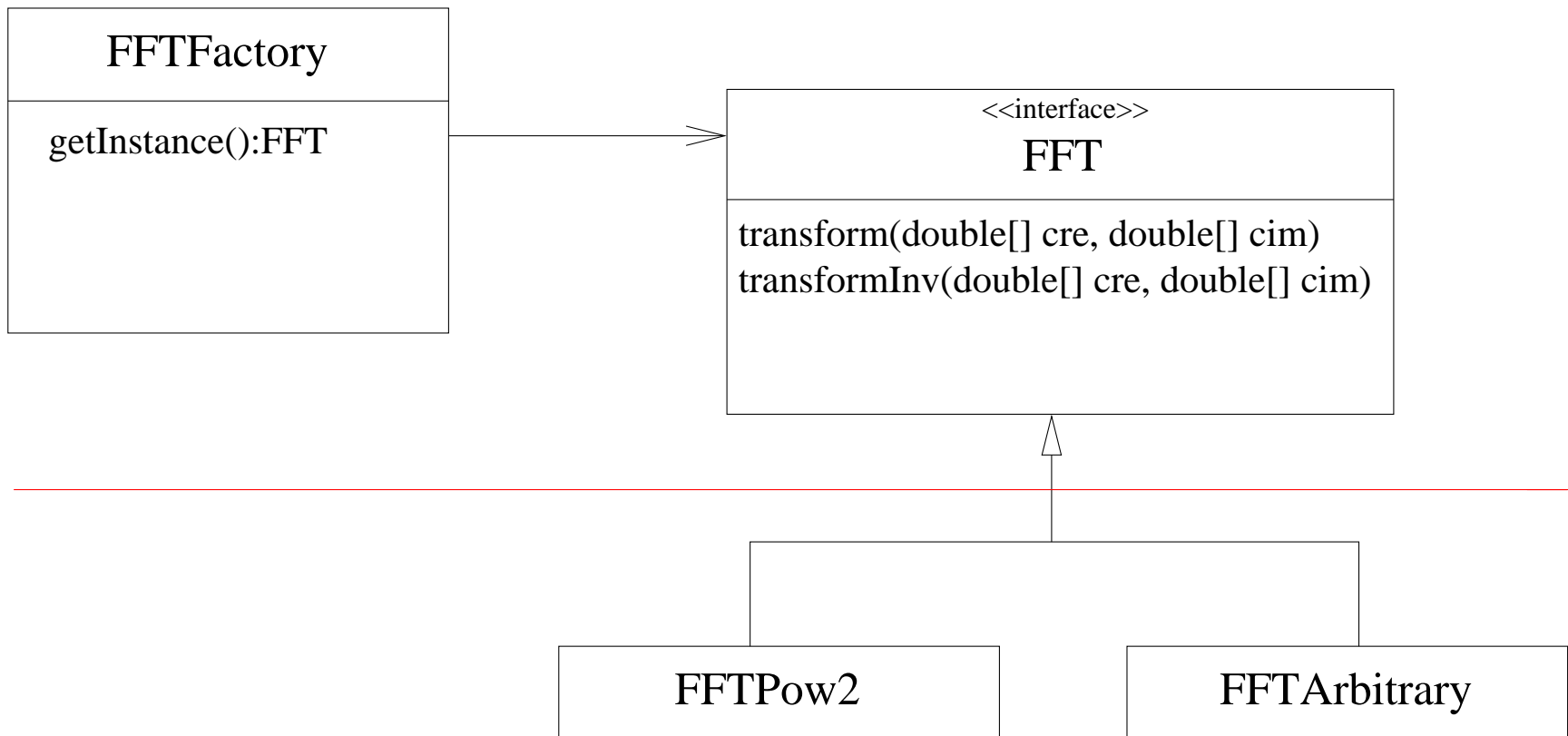
# Fassade

*allgemeines Diagramm*



# Fassade

## Beispiel: FFT in der Geostoch



# Fassade

## *Eigenschaften*

### **Vorteile:**

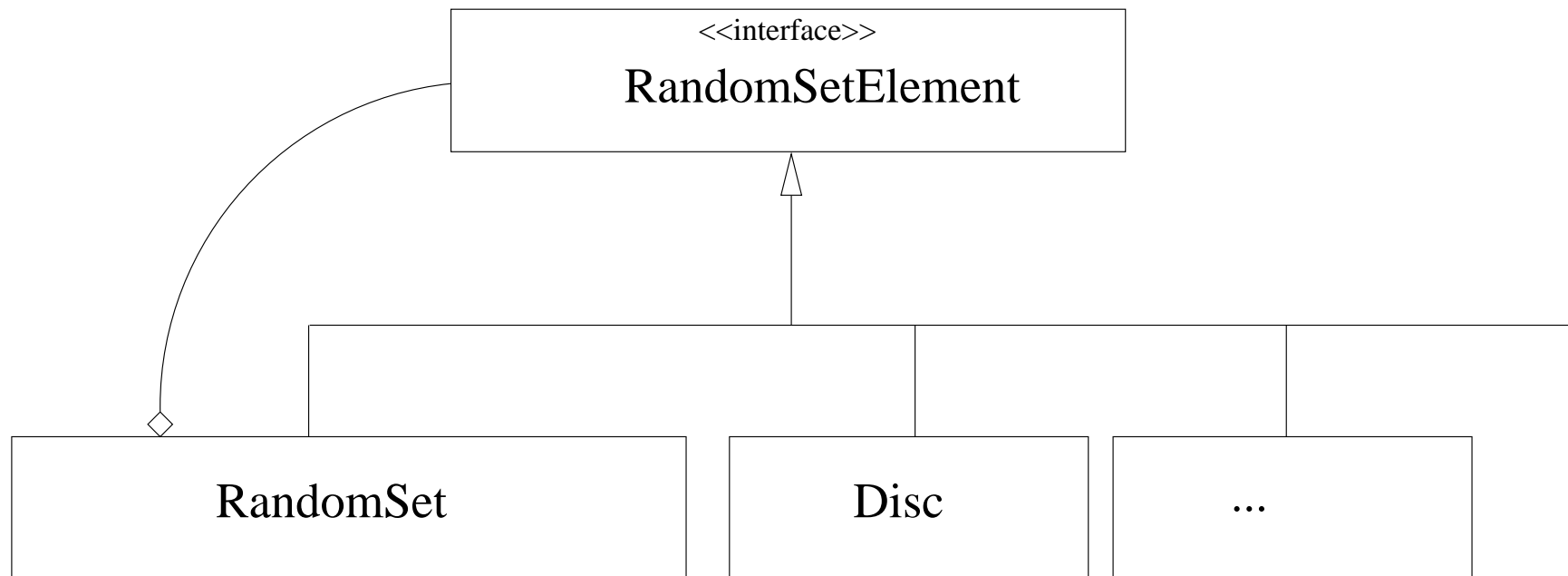
- Subsysteme **leichter zu benutzen**
- Subsysteme **leichter auszutauschen**
- einfacheres **Anordnen** der Subsysteme (welches kommuniziert mit welchem)
- weitere Vorteile bei abstrakter Fassade (**alternative** Subsysteme)

# Kompositum

*Beispiel: komplexes Bild in der Geostoch*

## Prinzip:

**komplexe** Klasse, die für **primitive Objekte und Container** steht



# Kompositum

## *Eigenschaften*

### **Vorteile:**

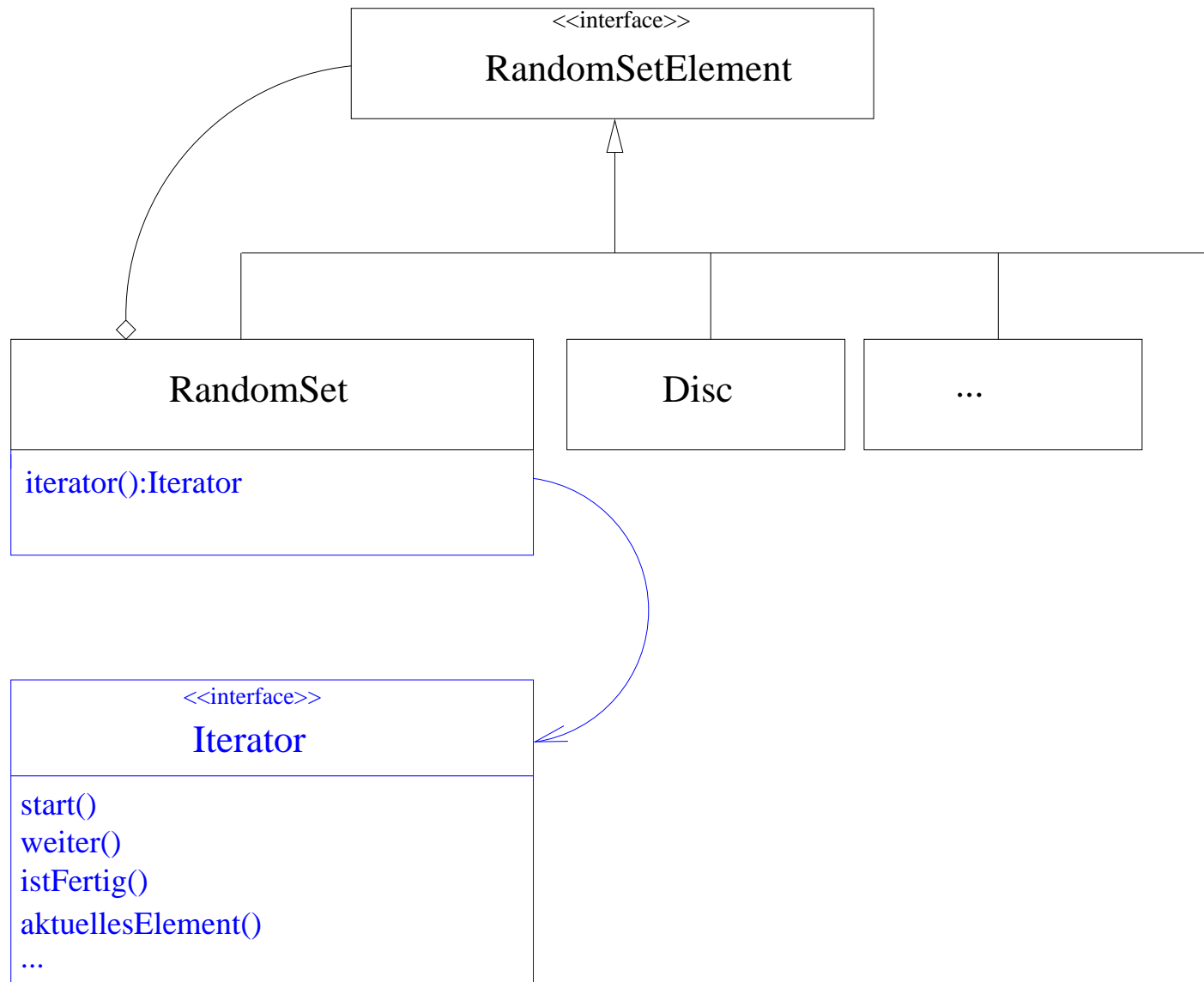
- einfache Struktur
- leicht erweiterbar

### **Nachteile:**

- Klassen evtl. zu allgemein (wenige Beschränkungen für Objekte möglich)

# Iterator

*Beispiel: komplexes Bild in der Geostoch*





# Iterator

## *Eigenschaften*

### Vorteile:

- stellt **Schnittstelle zu Containern** bereit, ohne dass man ihre genaue **Implementierung** kennen muss
- **unabhängig** von Containerart

# Zusammenfassung

## *Vorteile von Entwurfsmustern*

Entwurfsmuster haben sich in den letzten Jahren sehr schnell verbreitet. Was ist der Grund für ihre Beliebtheit?

- **Standardisierung** von Namen für gängige Software-Designs und damit auch der Dokumentation
- **Erleichterung für Neueinsteiger**
- Vorteile bei **Reorganisation** von Software
- leichte **Wiederverwendung von Ideen** zum Software-Design

# Zusammenfassung

## *Schlusswort*

Entwurfsmuster sind

Beschreibungen **zusammenarbeitender** Objekte und Klassen, die **maßgeschneidert** sind, um ein **allgemeines Entwurfsproblem** in einem bestimmten **Kontext** zu lösen.

Erich Gamma

basierend auf

## Literatur

[1] Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides *Entwurfsmuster*