

Didaktische Betrachtungen zur Unterrichtung von Software–Mustern im Hochschulbereich

Andreas Harrer und Markus Schneider

Gerhard–Mercator–Universität Duisburg, Technische Universität München

harrer@collide.info, schneima@in.tum.de

11. Oktober 2002

1. Workshop der GI-Fachgruppe Didaktik der Informatik

Gliederung:

- Einführung – Software–Muster
- Systematik für Muster – Stand der Forschung und Kritik
- Eine konzeptuell–didaktische Systematik für Muster
- Unterrichtung von Mustern in der Hochschule
- Zusammenfassung und Ausblick

Was ist ein Muster?

Muster sind abstrahierte Repräsentationen erfolgreich eingesetzten Problemlösungswissens

Was ist ein Muster?

Muster sind abstrahierte Repräsentationen erfolgreich eingesetzten Problemlösungswissens

- **abstrahiert:** Muster beziehen sich auf wiederkehrende Probleme

Was ist ein Muster?

Muster sind abstrahierte Repräsentationen erfolgreich eingesetzten Problemlösungswissens

- **abstrahiert:** Muster beziehen sich auf wiederkehrende Probleme
- **erfolgreich:** die Kenntnis von Mustern erspart es einem, das Rad immer wieder selbst erfinden zu müssen

Was ist ein Muster?

Muster sind abstrahierte Repräsentationen erfolgreich eingesetzten Problemlösungswissens

- **abstrahiert:** Muster beziehen sich auf wiederkehrende Probleme
- **erfolgreich:** die Kenntnis von Mustern erspart es einem, das Rad immer wieder selbst erfinden zu müssen
- **eingesetzt:** Muster in der Software-Technik sind damit „das Zunftbuch für Software-Entwickler“

Was ist ein Muster?

Muster sind abstrahierte Repräsentationen erfolgreich eingesetzten Problemlösungswissens

- **abstrahiert:** Muster beziehen sich auf wiederkehrende Probleme
- **erfolgreich:** die Kenntnis von Mustern erspart es einem, das Rad immer wieder selbst erfinden zu müssen
- **eingesetzt:** Muster in der Software-Technik sind damit „das Zunftbuch für Software-Entwickler“
- **Problemlösungswissen:** Muster geben einen Vorschlag für effiziente / elegante Lösungen der Problemstellung

Was ist ein Muster?

Muster sind abstrahierte Repräsentationen erfolgreich eingesetzten Problemlösungswissens

- **abstrahiert:** Muster beziehen sich auf wiederkehrende Probleme
- **erfolgreich:** die Kenntnis von Mustern erspart es einem, das Rad immer wieder selbst erfinden zu müssen
- **eingesetzt:** Muster in der Software-Technik sind damit „das Zunftbuch für Software-Entwickler“
- **Problemlösungswissen:** Muster geben einen Vorschlag für effiziente / elegante Lösungen der Problemstellung

Muster, die Probleme von Bestandteilen eines Software-Systems adressieren, werden **Software-Muster** genannt.

Beispiel eines Software-Musters: Adapter

Beispiel eines Software-Musters: Adapter

- **Name:** Adapter (auch Wrapper)

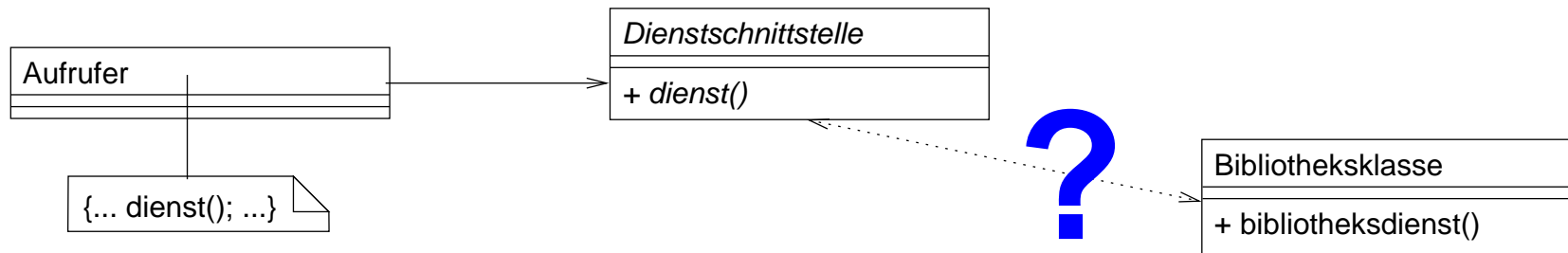
Beispiel eines Software-Musters: Adapter

- **Name:** Adapter (auch Wrapper)
- **Kontext:** Bibliotheksklassen besitzen eine Schnittstelle, die vom Benutzer nicht veränderbar ist, weil der Quellcode im Allgemeinen nicht vorliegt.

Beispiel eines Software-Musters: Adapter

- **Name:** Adapter (auch Wrapper)
- **Kontext:** Bibliotheksklassen besitzen eine Schnittstelle, die vom Benutzer nicht veränderbar ist, weil der Quellcode im Allgemeinen nicht vorliegt.
- **Problem:** Die Eigenschaften einer Bibliotheksklasse sollen genutzt werden, unser Entwurf verwendet allerdings eine andere Schnittstelle, als die von der Bibliothek bereitgestellte. Die Aufrufe an die Schnittstelle passen nicht zu den von der Bibliotheksklasse bereitgestellten.

Problem des Adapter-Musters: Klassendiagramm



Lösung des Adapter-Musters

Definiere einen **Adapter**, der die Aufrufe, die in unserem System stattfinden, so umsetzt, dass die Eigenschaften der Bibliotheksklasse verwendet werden können.

Lösung des Adapter-Musters

Definiere einen **Adapter**, der die Aufrufe, die in unserem System stattfinden, so umsetzt, dass die Eigenschaften der Bibliotheksklasse verwendet werden können.

- Der Adapter muss die vom System verwendete Dienstschnittstelle implementieren und

Lösung des Adapter-Musters

Definiere einen **Adapter**, der die Aufrufe, die in unserem System stattfinden, so umsetzt, dass die Eigenschaften der Bibliotheksklasse verwendet werden können.

- Der Adapter muss die vom System verwendete Dienstschnittstelle implementieren und
- gleichzeitig die Eigenschaften der Bibliotheksklasse wiederverwenden:

Lösung des Adapter-Musters

Definiere einen **Adapter**, der die Aufrufe, die in unserem System stattfinden, so umsetzt, dass die Eigenschaften der Bibliotheksklasse verwendet werden können.

- Der Adapter muss die vom System verwendete Dienstschnittstelle implementieren und
- gleichzeitig die Eigenschaften der Bibliotheksklasse wiederverwenden:
 - Delegation an ein Objekt der Bibliotheksklasse: **Objektadapter**

Lösung des Adapter-Musters

Definiere einen **Adapter**, der die Aufrufe, die in unserem System stattfinden, so umsetzt, dass die Eigenschaften der Bibliotheksklasse verwendet werden können.

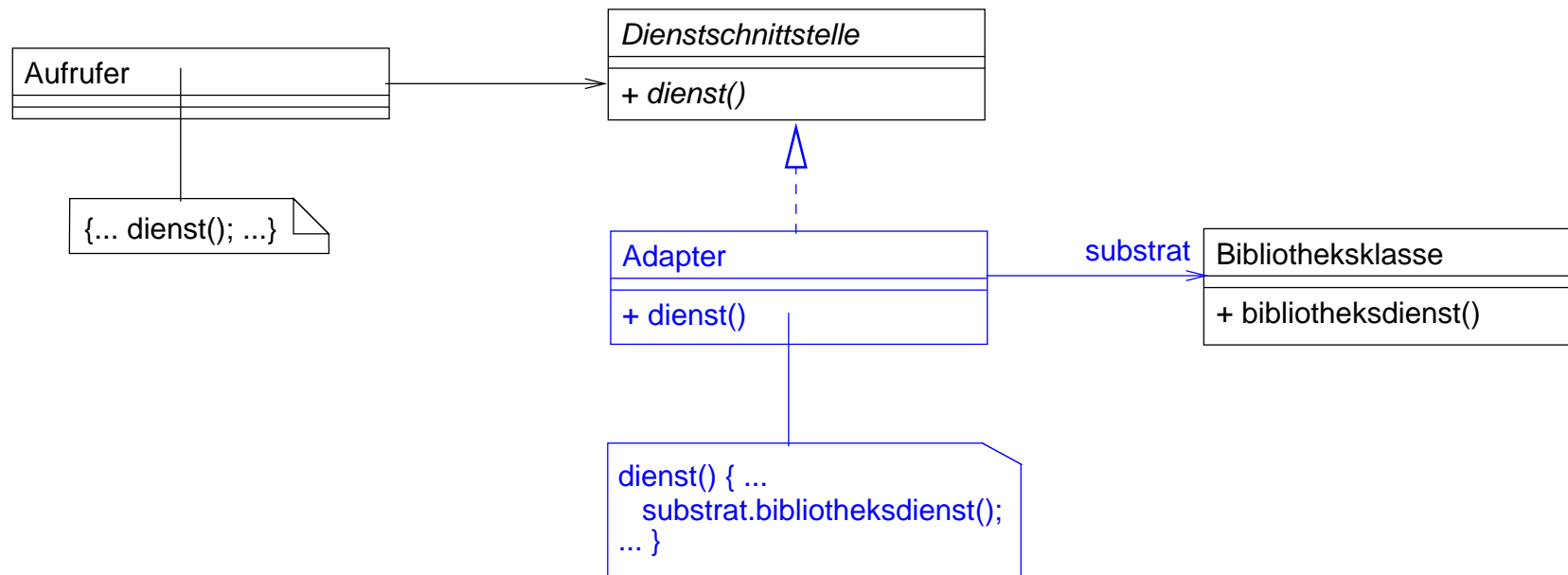
- Der Adapter muss die vom System verwendete Dienstschnittstelle implementieren und
- gleichzeitig die Eigenschaften der Bibliotheksklasse wiederverwenden:
 - Delegation an ein Objekt der Bibliotheksklasse: **Objektadapter**
 - Implementierungsvererbung von der Bibliotheksklasse: **Klassenadapter**

Lösung des Adapter-Musters

Definiere einen **Adapter**, der die Aufrufe, die in unserem System stattfinden, so umsetzt, dass die Eigenschaften der Bibliotheksklasse verwendet werden können.

- Der Adapter muss die vom System verwendete Dienstschnittstelle implementieren und
- gleichzeitig die Eigenschaften der Bibliotheksklasse wiederverwenden:
 - Delegation an ein Objekt der Bibliotheksklasse: **Objektadapter**
 - Implementierungsvererbung von der Bibliotheksklasse: **Klassenadapter**
- innerhalb der Implementierung des Schnittstellendienstes wird der Bibliotheksdienst verwendet.

Lösung des Adapter-Musters: Klassendiagramm für Objektadapter



Systematik für Muster

Systematik für Muster

- Schemata – Strukturierung eines einzelnen Musters

Systematik für Muster

- Schemata – Strukturierung eines einzelnen Musters
 - Name, Kontext, Problem, Lösung

Systematik für Muster

- Schemata – Strukturierung eines einzelnen Musters
 - Name, Kontext, Problem, Lösung
 - Beispiel, Varianten, Implementierungsvorschlag, Gegenanzeigen

Systematik für Muster

- Schemata – Strukturierung eines einzelnen Musters
 - Name, Kontext, Problem, Lösung
 - Beispiel, Varianten, Implementierungsvorschlag, Gegenanzeigen
- Klassifizierungen – Einteilung nach einer oder mehrerer Dimensionen

Systematik für Muster

- Schemata – Strukturierung eines einzelnen Musters
 - Name, Kontext, Problem, Lösung
 - Beispiel, Varianten, Implementierungsvorschlag, Gegenanzeigen
- Klassifizierungen – Einteilung nach einer oder mehrerer Dimensionen
 - Gruppierung von Mustern

Systematik für Muster

- Schemata – Strukturierung eines einzelnen Musters
 - Name, Kontext, Problem, Lösung
 - Beispiel, Varianten, Implementierungsvorschlag, Gegenanzeigen
- Klassifizierungen – Einteilung nach einer oder mehrerer Dimensionen
 - Gruppierung von Mustern
 - gezielte Suche möglich

Systematik für Muster

- Schemata – Strukturierung eines einzelnen Musters
 - Name, Kontext, Problem, Lösung
 - Beispiel, Varianten, Implementierungsvorschlag, Gegenanzeigen
- Klassifizierungen – Einteilung nach einer oder mehrerer Dimensionen
 - Gruppierung von Mustern
 - gezielte Suche möglich
 - Vergleichbarkeit der Eignung wird erhöht

Systematiken – die Hauptansätze

Klassifizierung nach Gamma et al. (1995)

Systematiken – die Hauptansätze

Klassifizierung nach Gamma et al. (1995)

- Bereich (*scope*): bezieht sich das Muster auf Beziehungen zwischen **Objekten** oder zwischen **Klassen**?

Systematiken – die Hauptansätze

Klassifizierung nach Gamma et al. (1995)

- Bereich (*scope*): bezieht sich das Muster auf Beziehungen zwischen **Objekten** oder zwischen **Klassen**?
- Zweck (*purpose*): ist das Muster mit **Erzeugung, Struktur** oder **Verhalten** befasst?

Systematiken – die Hauptansätze

Klassifizierung nach Gamma et al. (1995)

- Bereich (*scope*): bezieht sich das Muster auf Beziehungen zwischen **Objekten** oder zwischen **Klassen**?
- Zweck (*purpose*): ist das Muster mit **Erzeugung, Struktur** oder **Verhalten** befasst?

Eignung bezüglich Auswahl und Vermittlung von Mustern:

Systematiken – die Hauptansätze

Klassifizierung nach Gamma et al. (1995)

- Bereich (*scope*): bezieht sich das Muster auf Beziehungen zwischen **Objekten** oder zwischen **Klassen**?
- Zweck (*purpose*): ist das Muster mit **Erzeugung**, **Struktur** oder **Verhalten** befasst?

Eignung bezüglich Auswahl und Vermittlung von Mustern:

- lediglich grobe Unterscheidung

Systematiken – die Hauptansätze

Klassifizierung nach Gamma et al. (1995)

- Bereich (*scope*): bezieht sich das Muster auf Beziehungen zwischen **Objekten** oder zwischen **Klassen**?
- Zweck (*purpose*): ist das Muster mit **Erzeugung**, **Struktur** oder **Verhalten** befasst?

Eignung bezüglich Auswahl und Vermittlung von Mustern:

- lediglich grobe Unterscheidung
- Muster, die nur strukturelle, aber keine inhaltlichen Gemeinsamkeiten haben, gehören derselben Kategorie an

Systematiken – die Hauptansätze

Klassifizierung nach Buschmann et al. (1996)

Systematiken – die Hauptansätze

Klassifizierung nach Buschmann et al. (1996)

- Abstraktionsgrad: bezieht sich das Muster auf Gesamtsysteme, Subsysteme oder Implementierungsprobleme bestimmter Programmiersprachen?

Systematiken – die Hauptansätze

Klassifizierung nach Buschmann et al. (1996)

- Abstraktionsgrad: bezieht sich das Muster auf Gesamtsysteme, Subsysteme oder Implementierungsprobleme bestimmter Programmiersprachen?
- Problemkategorie: in welchem Kontext wird das Muster eingesetzt und welches Problemgebiet wird adressiert?

Systematiken – die Hauptansätze

Klassifizierung nach Buschmann et al. (1996)

- Abstraktionsgrad: bezieht sich das Muster auf Gesamtsysteme, Subsysteme oder Implementierungsprobleme bestimmter Programmiersprachen?
- Problemkategorie: in welchem Kontext wird das Muster eingesetzt und welches Problemgebiet wird adressiert?

Eignung bezüglich Auswahl und Vermittlung von Mustern:

Systematiken – die Hauptansätze

Klassifizierung nach Buschmann et al. (1996)

- Abstraktionsgrad: bezieht sich das Muster auf Gesamtsysteme, Subsysteme oder Implementierungsprobleme bestimmter Programmiersprachen?
- Problemkategorie: in welchem Kontext wird das Muster eingesetzt und welches Problemgebiet wird adressiert?

Eignung bezüglich Auswahl und Vermittlung von Mustern:

- Abstraktionsgrad ist als Dimension gut geeignet für Grobeinteilung

Systematiken – die Hauptansätze

Klassifizierung nach Buschmann et al. (1996)

- Abstraktionsgrad: bezieht sich das Muster auf Gesamtsysteme, Subsysteme oder Implementierungsprobleme bestimmter Programmiersprachen?
- Problemkategorie: in welchem Kontext wird das Muster eingesetzt und welches Problemgebiet wird adressiert?

Eignung bezüglich Auswahl und Vermittlung von Mustern:

- Abstraktionsgrad ist als Dimension gut geeignet für Grobeinteilung
- Problemkategorie vermischt Systemklassen und Konzepte, dadurch verliert die Zuordnung an Exaktheit

Vorschlag einer konzeptuell-didaktischen Systematik

Ziel ist eine klare Trennung der Klassifizierungsdimensionen für die

Vorschlag einer konzeptuell-didaktischen Systematik

Ziel ist eine klare Trennung der Klassifizierungsdimensionen für die

- geeignete Auswahl von Mustern in der Unterrichtsplanung

Vorschlag einer konzeptuell-didaktischen Systematik

Ziel ist eine klare Trennung der Klassifizierungsdimensionen für die

- geeignete Auswahl von Mustern in der Unterrichtsplanung
 - Schwierigkeit
 - Umfang

Vorschlag einer konzeptuell-didaktischen Systematik

Ziel ist eine klare Trennung der Klassifizierungsdimensionen für die

- geeignete Auswahl von Mustern in der Unterrichtsplanung
 - Schwierigkeit
 - Umfang
- klare Darstellung der Muster und ihrer informatischen Essenz in der Unterrichtsdurchführung

Vorschlag einer konzeptuell-didaktischen Systematik

Ziel ist eine klare Trennung der Klassifizierungsdimensionen für die

- geeignete Auswahl von Mustern in der Unterrichtsplanung
 - Schwierigkeit
 - Umfang

- klare Darstellung der Muster und ihrer informatischen Essenz in der Unterrichtsdurchführung
 - Einsetzbarkeit
 - Konzept

Prinzipien objektorientierten Software-Entwurfs

Aufgrund des starken Bezugs von Mustern zu objektorientierten Ansätzen leiten wir zunächst die Grundkonzepte aus dem Prozess objektorientierter Modellierung und Programmierung ab

Prinzipien objektorientierten Software-Entwurfs

Aufgrund des starken Bezugs von Mustern zu objektorientierten Ansätzen leiten wir zunächst die Grundkonzepte aus dem Prozess objektorientierter Modellierung und Programmierung ab

- Festlegung einer Architektur – Komponenten und deren Interaktion

Prinzipien objektorientierten Software–Entwurfs

Aufgrund des starken Bezugs von Mustern zu objektorientierten Ansätzen leiten wir zunächst die Grundkonzepte aus dem Prozess objektorientierter Modellierung und Programmierung ab

- Festlegung einer Architektur – Komponenten und deren Interaktion
- Verfeinerung des Modells – detaillierter Entwurf von Objekten

Prinzipien objektorientierten Software–Entwurfs

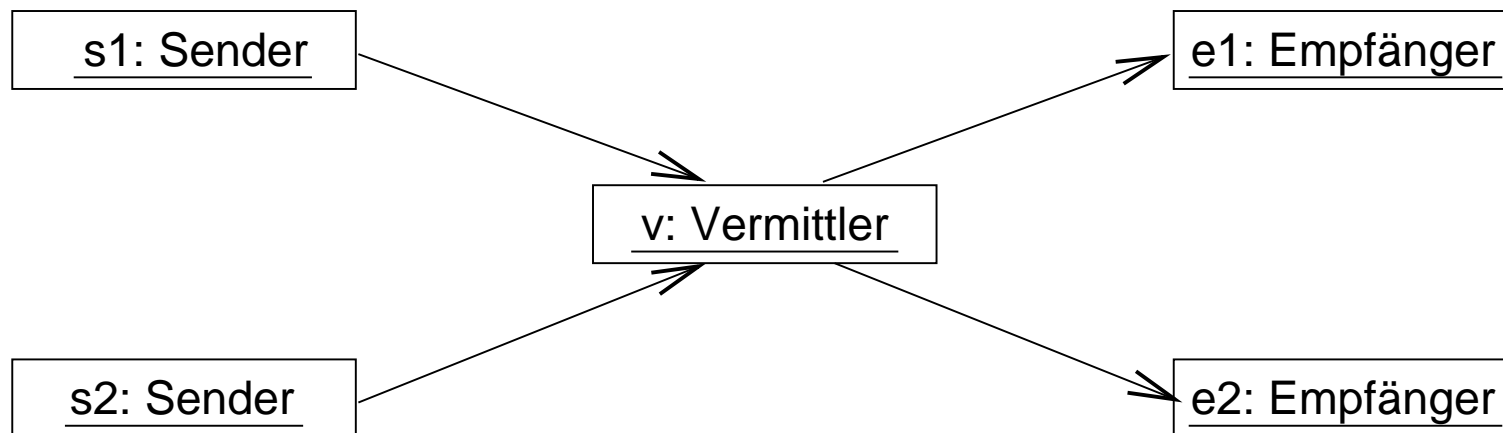
Aufgrund des starken Bezugs von Mustern zu objektorientierten Ansätzen leiten wir zunächst die Grundkonzepte aus dem Prozess objektorientierter Modellierung und Programmierung ab

- Festlegung einer Architektur – Komponenten und deren Interaktion
- Verfeinerung des Modells – detaillierter Entwurf von Objekten
- Lösung implementierungsnaher Probleme der Verwaltung von Objekten

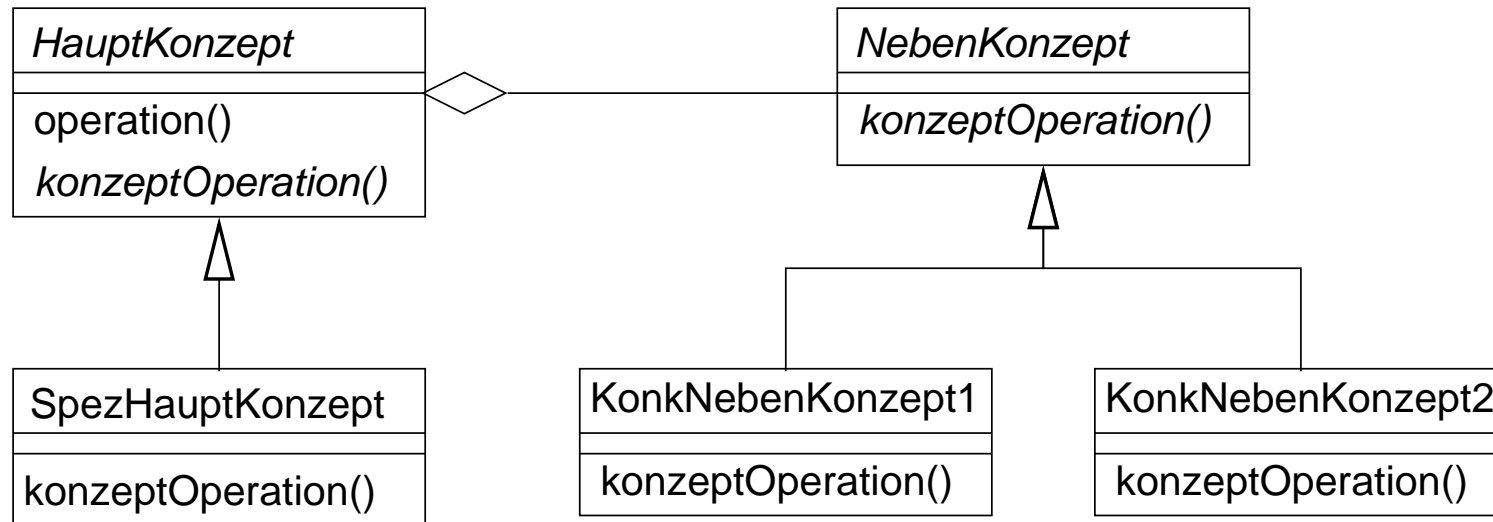
Isolierte Informatische Konzepte:

- Interaktion und deren Regelung
- Dekomposition
- Rekursive Dekomposition
- Wiederverwendung
- Management

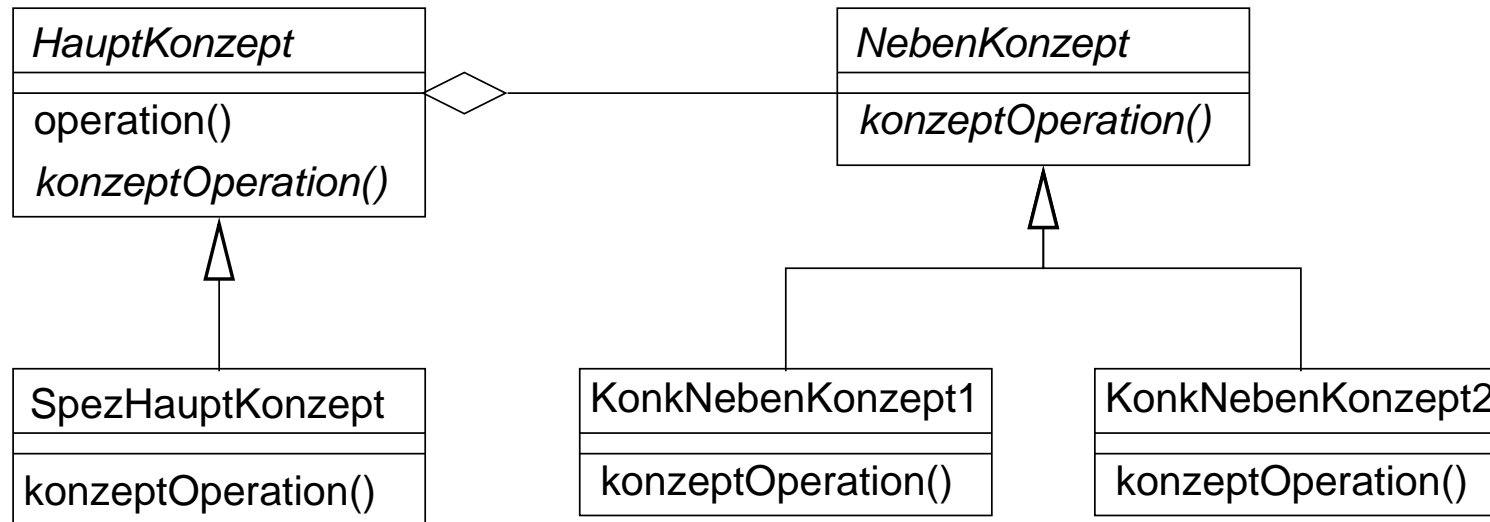
Grundidee der Muster zur Interaktionsregelung



Grundidee der Dekompositions-Muster



Grundidee der Dekompositions-Muster



rekursive Dekomposition durch Identifizierung von Haupt- und Nebenkonzept

Klassifizierungsdimension Komplexität:

Klassifizierungsdimension Komplexität:

- Schwierigkeit des Verständnisses

Klassifizierungsdimension Komplexität:

- Schwierigkeit des Verständnisses
- Umfang der Diskussionspunkte

Klassifizierungsdimension Komplexität:

- Schwierigkeit des Verständnisses
- Umfang der Diskussionspunkte
- Variationsmöglichkeiten

Klassifizierungsdimension Art des Musters:

- Software-Muster

Klassifizierungsdimension Art des Musters:

- Software-Muster
 - Architekturmuster

Klassifizierungsdimension Art des Musters:

- Software–Muster
 - Architekturmuster
 - Entwurfsmuster

Klassifizierungsdimension Art des Musters:

- Software-Muster
 - Architekturmuster
 - Entwurfsmuster
 - Idiome

Klassifizierungsdimension Art des Musters:

- Software–Muster
 - Architekturmuster
 - Entwurfsmuster
 - Idiome
- Prozess–Muster

Klassifizierungsdimension Einsatzgebiet:

Klassifizierungsdimension Einsatzgebiet:

- generell

Klassifizierungsdimension Einsatzgebiet:

- generell
- speziell
 - verteilte Systeme
 - benutzer–interaktive Systeme
 - intelligente Lehr–/Lernsysteme
 - ...

Einsatzbeispiel der Systematik

Klassifizierung des Adapters

- **Art des Musters:**
- **Einsatzgebiet:**
- **Informatisches Konzept:**
- **Komplexität:**

Einsatzbeispiel der Systematik

Klassifizierung des Adapters

- **Art des Musters:** Softwaremuster – Entwurfsmuster
- **Einsatzgebiet:**
- **Informatisches Konzept:**
- **Komplexität:**

Einsatzbeispiel der Systematik

Klassifizierung des Adapters

- **Art des Musters:** Softwaremuster – Entwurfsmuster
- **Einsatzgebiet:** generell
- **Informatisches Konzept:**
- **Komplexität:**

Einsatzbeispiel der Systematik

Klassifizierung des Adapters

- **Art des Musters:** Softwaremuster – Entwurfsmuster
- **Einsatzgebiet:** generell
- **Informatisches Konzept:** Interaktionsregelung, Wiederverwendung
- **Komplexität:**

Einsatzbeispiel der Systematik

Klassifizierung des Adapters

- **Art des Musters:** Softwaremuster – Entwurfsmuster
- **Einsatzgebiet:** generell
- **Informatisches Konzept:** Interaktionsregelung, Wiederverwendung
- **Komplexität:** einfach, wenig Diskussionspunkte, variantenreich

Einsatzbeispiel der Systematik

Klassifizierung des Adapters

- **Art des Musters:** Softwaremuster – Entwurfsmuster
- **Einsatzgebiet:** generell
- **Informatisches Konzept:** Interaktionsregelung, Wiederverwendung
- **Komplexität:** einfach, wenig Diskussionspunkte, variantenreich

Im Aufsatz findet sich die Klassifizierung aller Muster in Gamma et al., um die Verwendbarkeit der Systematik aufzuzeigen.

Unterrichtung von Mustern in der Hochschule:

1. Unterrichtsplanung – Auswahl von Mustern

Unterrichtung von Mustern in der Hochschule:

1. Unterrichtsplanung – Auswahl von Mustern

- Entscheidend ist Anspruchsniveau und Grundwissen

Unterrichtung von Mustern in der Hochschule:

1. Unterrichtsplanung – Auswahl von Mustern

- Entscheidend ist Anspruchsniveau und Grundwissen
- Erfahrungen in Grundstudium und Spezialvorlesung im Hauptstudium

Unterrichtung von Mustern in der Hochschule:

1. Unterrichtsplanung – Auswahl von Mustern
 - Entscheidend ist Anspruchsniveau und Grundwissen
 - Erfahrungen in Grundstudium und Spezialvorlesung im Hauptstudium
2. Unterrichtsdurchführung – Vermittlung von Mustern

Unterrichtung von Mustern in der Hochschule:

1. Unterrichtsplanung – Auswahl von Mustern

- Entscheidend ist Anspruchsniveau und Grundwissen
- Erfahrungen in Grundstudium und Spezialvorlesung im Hauptstudium

2. Unterrichtsdurchführung – Vermittlung von Mustern

- Muster eignen sich gut zur Kombination verschiedener didaktischer Vorgehensweisen

Unterrichtung von Mustern in der Hochschule:

1. Unterrichtsplanung – Auswahl von Mustern

- Entscheidend ist Anspruchsniveau und Grundwissen
- Erfahrungen in Grundstudium und Spezialvorlesung im Hauptstudium

2. Unterrichtsdurchführung – Vermittlung von Mustern

- Muster eignen sich gut zur Kombination verschiedener didaktischer Vorgehensweisen
- In der Spezialvorlesung wurde induktives, deduktives und spiralförmiges Vorgehen miteinander kombiniert

Zusammenfassung und Ausblick:

- Kenntnis von Mustern erhöht Modellierungs- und Programmierfertigkeiten, erleichtert Software-Entwicklung
- Zur Unterrichtung von Mustern, ist eine geeignete Systematik wünschenswert
- Entwicklung einer konzeptuell–didaktischen Systematik
- Praxiserfahrungen aus dem Hochschulunterricht
- Muster als Unterrichtsgegenstand in Schulen?