

Mustergültige Softwaredesigns:
Flexible Applikationen dank
Entwurfsmuster und **MEF**

Manfred Steyer

CAMPUS **02** **BASTA! ON TOUR**
ARCHITECTURE & GOOD PRACTICES

Aktuelles Buch

.Net 4.0 Update
ISBN 978-3866454439
<http://tinyurl.com/net4update>

Microsoft
.net 4.0
Update

Visual Basic 2010
Visual C# 2010
ASP.NET 4.0
ASP.NET 4.0
LINQ
Communication
Workflow

Microsoft Press

Kontakt

- **[www]** www.SoftwareArchitekt.at
- **[mail]** Manfred.Steyer@SoftwareArchitekt.at

Inhalt

- Design Patterns anhand eines Fallbeispiels
- Managed Extensibility Framework (MEF)

Documentmanagement-System

- Plattformunabhängiges Dokument-Format
- Textdateien mit Formatierungsanweisungen
- Ähnlichkeiten zu bekannten Formaten
sind rein zufällig!

Beispiel

%LSP Überschrift 1 Ü b e r s c h r i f t 1

%UC Überschrift 2 ÜBERSCHRIFT 2

%STD Das ist ein Das ist ein TOLLER Text
%UC toller %STD Text

DEMO

DOCUMENT MANAGER - VERSION 1

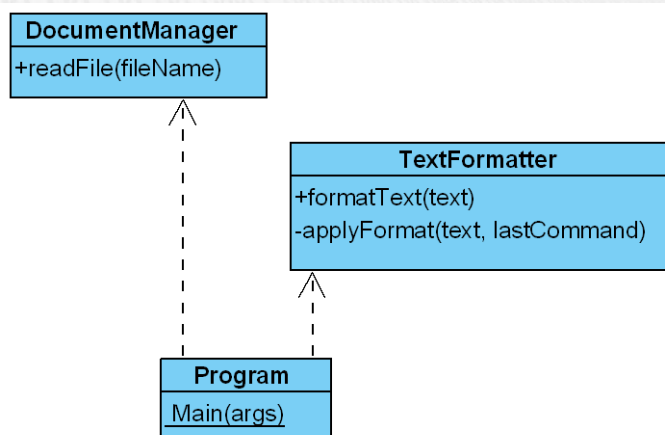
Gedanken zu Version 1

- Gesamte Funktionalität in einer Klasse
- Unübersichtlich
- Schwierig zu warten
- Schwierig zu testen
- Schwierig wieder verwendbar

Design-Prinzip

- Jede Klasse soll ein einziges, klar umrissenes Aufgabengebiet haben.

Refactoring



DEMO

DOCUMENT MANAGER - VERSION 2

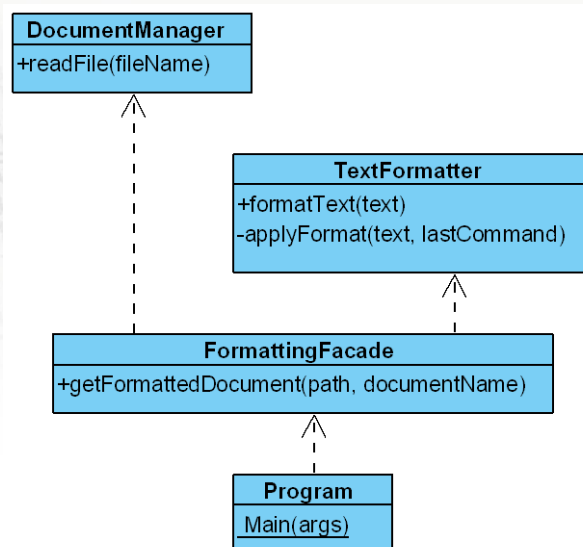
Gedanken zu Version 2

- Einzelne Klasse können einfach erweitert bzw. wiederverwendet werden
- Entwickler muss wissen, wie diese Klasse zusammenspielen

Design-Pattern: Facade

- Fassaden koordiniert verschiedene Objekte um ein Standardverhalten zu erzielen
- Objekte können auch separat verwendet werden

Design



DEMO

DOCUMENT MANAGER - VERSION 3

Auslagern von Kommandos

- Die Applikation soll in Zukunft einfach um weitere Kommandos erweitert werden kann

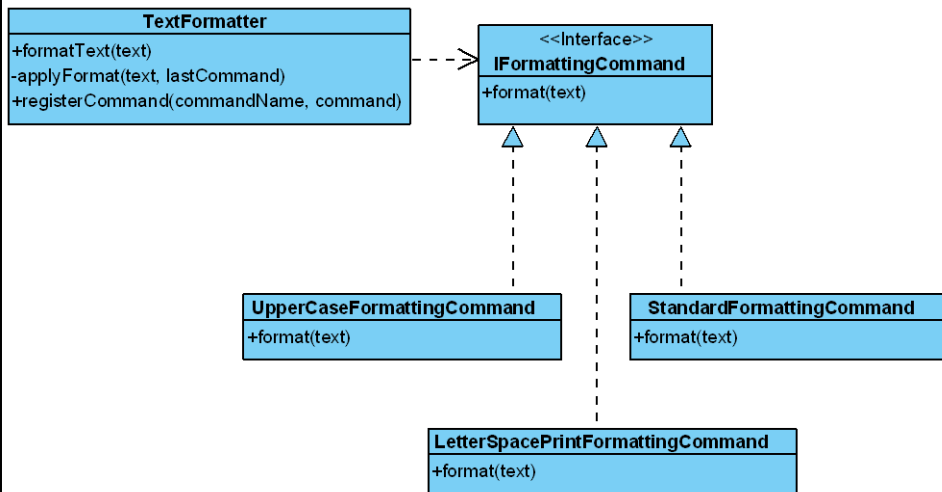
Designprinzip

- Programmteile, welche sich häufig ändern sollen von anderen Programmteilen isoliert werden.

Verbesserungsvorschlag

- Kommandos auslagern

Design



Gedanken

- Implementierung der Kommandos ist entkoppelt
- Um neue Kommandos hinzuzufügen muss jedoch noch immer Code abgeändert werden!

Wunsch

- Namen von neuen Kommando-Klassen sollen lediglich in der Konfiguration hinterlegt werden

Design-Pattern: Factory

- Eigentlich: SimpleFactory
- Nimmt „Friendly-Name“ des Kommandos entgegen
- Schlägt in Konfiguration nach
- Instanziert dynamisch Kommando

DEMO

DOCUMENT MANAGER - VERSION 4

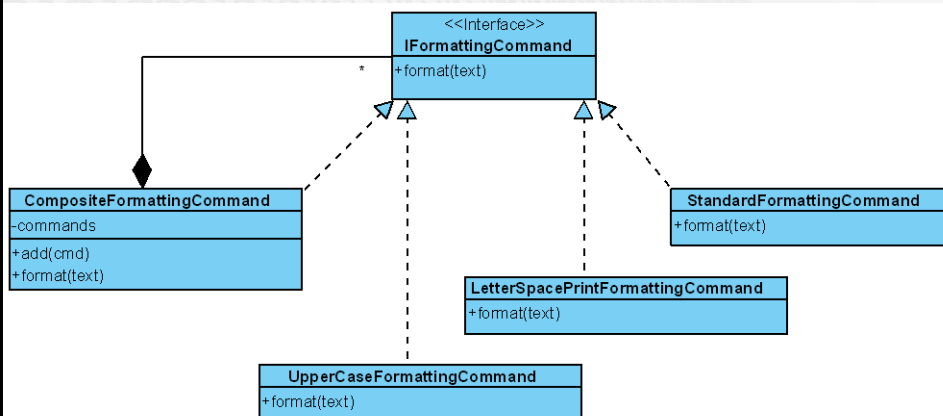
Erweiterungswunsch

- Künftig sollen auch zusammengesetzte Kommandos unterstützt werden!
- %UC,LSP Test → T E S T

Design-Pattern: Composite

- Ein Composite ermöglicht es, ein „Ganzes“ wie seine „Teile“ zu behandeln
- Implementiert das selbe Interface, wie die Teile (hier: IFormattingCommand)
- Nimmt seine Teile auf und delegiert an diese weiter.

Design



DEMO

DOCUMENT MANAGER - VERSION 5

Erweiterungswunsch

- Plugins sollen selbständig (ohne Konfigurationsdatei) gesucht werden.

DEMO

DOCUMENT MANAGER - VERSION 6

Literatur

- Gamma et. al., **Entwurfsmuster. Elemente wiederverwendbarer objektorientierter Software**

ISBN 978-3827321992

Literatur

- Freeman et. al,
Entwurfsmuster von Kopf bis Fuß
ISBN 978-3897214217

Kontakt

- manfred.steyer@softwarearchitekt.at
- www.softwarearchitekt.at

.Net 4.0 Update
ISBN 978-3866454439
<http://tinyurl.com/net4update>

