

ASPEKTORIENTIERTE PROGRAMMIERUNG IN .NET

20. August 2009

Hilmar Bunjes

Agenda

2

- Motivation „Warum soll ich AOP lernen?“
- Was ist AOP?
- Tools für .NET
- Beispiele und Demos
- Zusammenfassung

Motivation

„Warum soll ich AOP lernen?“

Motivation

4

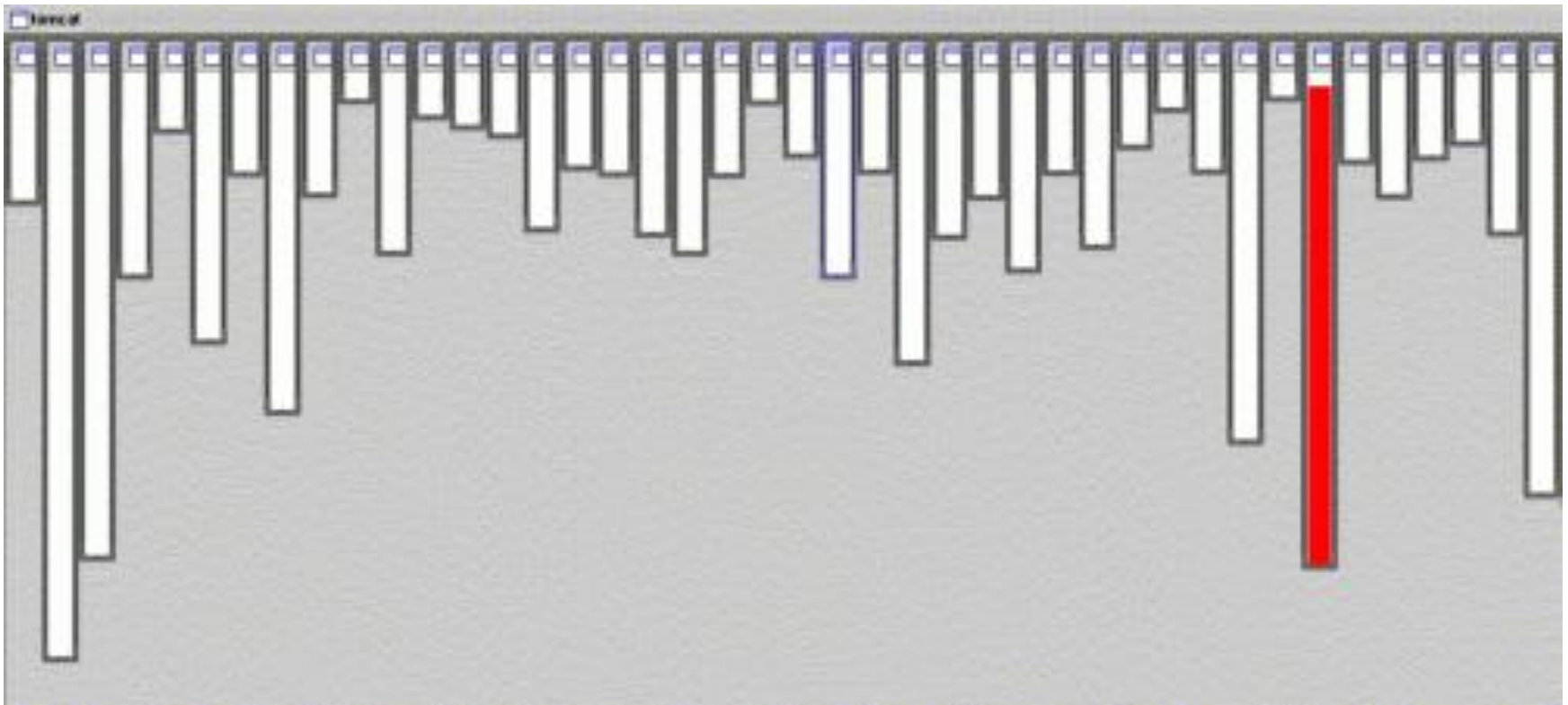
- Separations of Concern
 - Concern: Angelegenheit / Konzept
 - Fachlich: Geschäftslogik
 - Nichtfachlich: Logging, Sicherheit, ...

- Komplexität der Software sinkt nicht 😊

Motivation – Beispiel

5

- XML-Parsing im Apache Tomcat Server

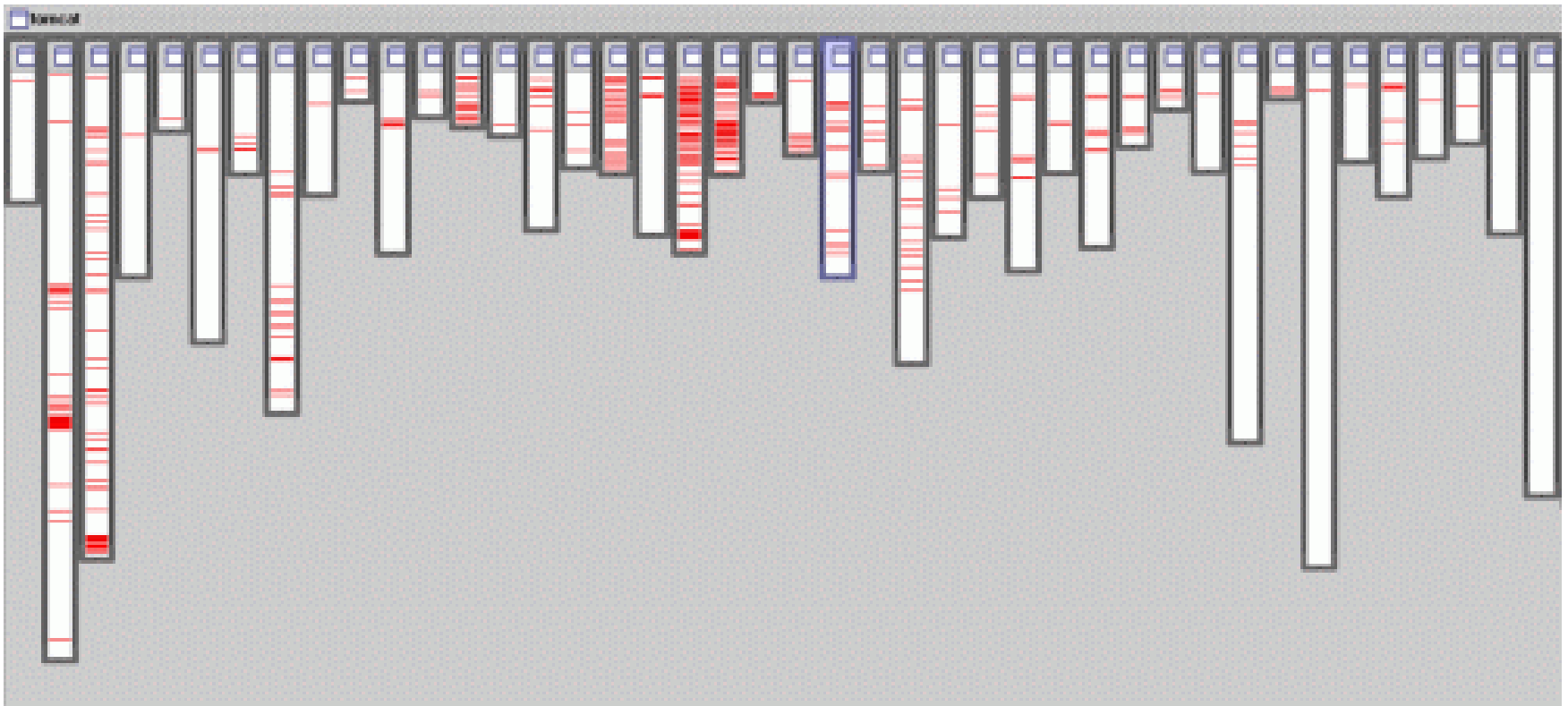


www.aspectj.org

Motivation – Beispiel

6

- Logging im Apache Tomcat Server



www.aspectj.org

Motivation – Beispiel

7

- Was passiert, wenn Log Framework geändert wird?
 - Namensänderungen -> Refactoring
 - Tiefere Änderungen -> Händische Arbeit

- Was passiert, wenn Autorisierung geändert wird?

- ...

Was ist AOP?

Was ist AOP?

9

- Programmierparadigma
- Getrennte Entwicklung logischer **Aspekte**
- Spätere Zusammenführung (Weaving/Weben)

Was ist AOP?

10

□ Beispiel Bankkonto

fachlich	nichtfachlich
-Einzahlung -Auszahlung -Überweisung -...	-Logging -Performance -Sicherheit -...

Was ist AOP - Beispiel

11

```
public class Bankkonto {  
    public int Kontostand { get; private set; }  
    private static Logger _log = Logger.GetLogger(typeof(Bankkonto));  
    public void Einzahlen(int eurocent) {  
        CheckPermission();  
        _log.info("Einzahlung: alt: {0}, neu: {1}", Kontostand, Kontostand + eurocent);  
        Kontostand += eurocent;  
    }  
    public void Auszahlen(int eurocent) {  
        CheckPermission();  
        _log.info("Auszahlung: alt: {0}, neu: {1}", Kontostand, Kontostand + eurocent);  
        Kontostand -= eurocent;  
    }  
}
```

Was ist AOP - Beispiel

12

```
public class Bankkonto {  
    public int Kontostand { get; private set; }  
    private static Logger _log = Logger.GetLogger(typeof(Bankkonto));  
    public void Einzahlen(int eurocent) {  
        CheckPermission();  
        _log.info("Einzahlung: alt: {0}, neu: {1}", Kontostand, Kontostand + eurocent);  
        Kontostand += eurocent;  
    }  
    public void Auszahlen(int eurocent) {  
        CheckPermission();  
        _log.info("Auszahlung: alt: {0}, neu: {1}", Kontostand, Kontostand - eurocent);  
        Kontostand -= eurocent;  
    }  
}
```

Was ist AOP - Beispiel

13

```
public class Bankkonto {  
    public int Kontostand { get; private set; }  
    [LogCall]  
    [CheckPermission]  
    public void Einzahlen(int eurocent) {  
        Kontostand += eurocent;  
    }  
    [LogCall]  
    [CheckPermission]  
    public void Auszahlen(int eurocent) {  
        Kontostand -= eurocent;  
    }  
}
```

Was ist AOP - Begriffe

14

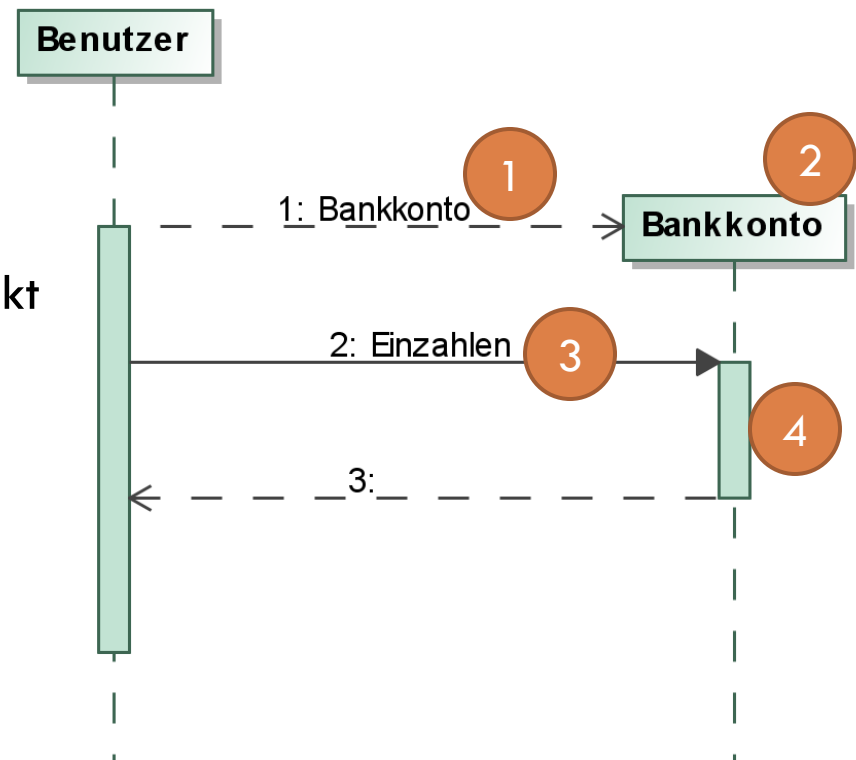
- Concern
 - Angelegenheit / Konzept (z.B. Logging)
- Joinpoint
 - Eingriffspunkt im Programm
- Pointcut
 - Selektion bestimmter Joinpoints
- Advice
 - Ratschlag/Code, der an Pointcut eingebaut wird
- Aspect
 - Gruppierung Pointcut/Advices, die zu Concern gehören

Was ist AOP - Begriffe

15

□ Joinpoint

- Aufrufen einer Methode
- Ausführen einer Methode
- Zugriff auf Variable
- Werfen einer Exception
- Initialisierung einer Klasse/Objekt



Was ist AOP - Begriffe

16

□ Advice

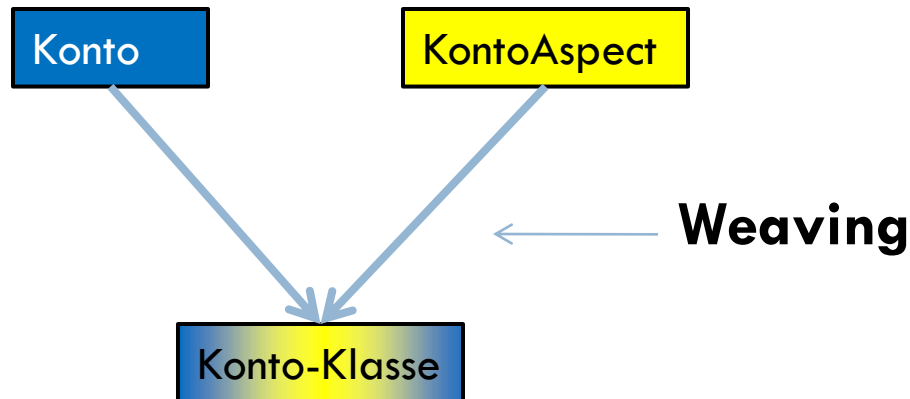
- Code, der am Joinpoint eingebaut wird

```
public override void OnEntry(MethodExecutionEventArgs eventArgs)
{
    Trace.WriteLine("Entry method " + eventArgs.Method);
}
```

Was ist AOP - Begriffe

17

- Weaving
 - ▣ Zusammenführung fachliche Klasse und Aspekt



Tools für .NET

Tools für .NET

19

- PostSharp
 - Version 1.5 vom 28. Juli 2009
 - Aktueller Webcast: AOP in Silverlight
(<http://www.microsoft.com/germany/msdn/webcasts/library.aspx?id=1032422768>)
- Spring.NET AOP
 - Auch ohne Spring Framework nutzbar
- dotspect
 - Seit 2006 kein commit
- Aspect#
 - Letzte Änderung 2005

Tools für .NET – PostSharp

20

- .NET Assemblies nach Compilerlauf ändern
- PostSharp Laos: AOP-Plugin
- Ausführung nach C#/VB Compiler
- Leichte Verlängerung der Build-Zeit (5-100%)
- Leichtgewichtig zur Laufzeit
- Wenig Reflection-Kenntnisse notwendig

Tools für .NET – PostSharp

21

- **Einstiegspunkte**
 - **Methoden:** Vorher, Nachher, Exception, Erfolg
 - Auftreten einer **Exception**
 - **Felder:** Lesen, Schreiben
 - **Methodenaufruf** (ohne Zugriff auf Methode)
 - **Interface/Implementierung** zu Klasse hinzu
 - Implementierung **abstract/extern Methode**
 - **CustomAttributes** hinzufügen

Beispiele und Demos

-> Visual Studio

Zusammenfassung

Zusammenfassung

24

- Fachlichen/Nichtfachlichen Code trennen
- Einfache Änderung Implementierungsdetails
- Benötigt Eingewöhnung
- Kann Performance kosten, aber beschleunigt Entwicklung

Vielen Dank für die
Aufmerksamkeit!

Kontakt:
Hilmar Bunjes
hilmar@bluecall.de

Ressourcen

26

- PostSharp:
<http://www.postsharp.org>
- AOP Konferenz:
<http://aosd.net/>
- Spring AOP Einstieg:
<http://www.developer.com/lang/article.php/3795031>
- Spring AOP Guide:
<http://www.springframework.net/doc-1.1-P3/reference/html/aop-quickstart.html>