CUSTIS®

Аспектно-ориентированное программирование. PostSharp

Докладчик:
Русецкий Георгий

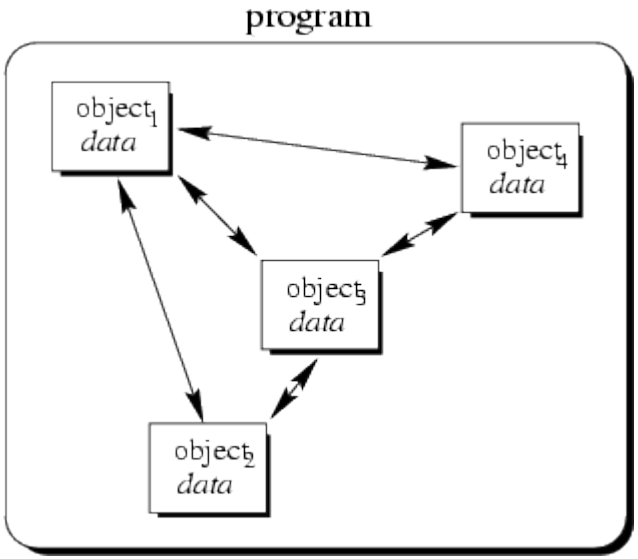
План

CUSTIS®

- Объектно-ориентированное программирование. Принцип единственной ответственности
- Общесистемные требования
  - Демонстрация
- Проблемы реализации общесистемной функциональности с использованием ООП
- Аспектно-ориентированное программирование
  - История
  - Реализации
- PostSharp
  - Основные понятия АОП в терминах PostSharp
  - Демонстрация
  - Основные подходы к использованию PostSharp
    - Простые аспекты
    - Композитные аспекты
  - Разработка композитных аспектов
    - Демонстрация
  - Преимущества и недостатки АОП
  - Вопросы

CUSTIS®

Объектно-ориентированное программирование



CUSTIS®

Принцип единственной ответственности

CUSTIS®

Принцип единственной ответственности

CUSTIS®

Результат применения принципа единственной ответственности:

- Прозрачный дизайн
- Легче модифицировать код
- Меньше вероятность возникновения ошибок

CUSTIS®

Примеры общесистемных требований:

- журналирование
- обработка исключений
- синхронизация доступа
- транзакционность

Проблемы?

CUSTIS®

Проблемы?

CUSTIS®

- трудно проследить предназначение модуля

Проблемы?

CUSTIS®

- трудно проследить предназначение модуля
- непригодность модуля для повторного использования

Проблемы?

CUSTIS®

- трудно проследить предназначение модуля
- непригодность модуля для повторного использования
- запутанный код => большая вероятность привнесения ошибок

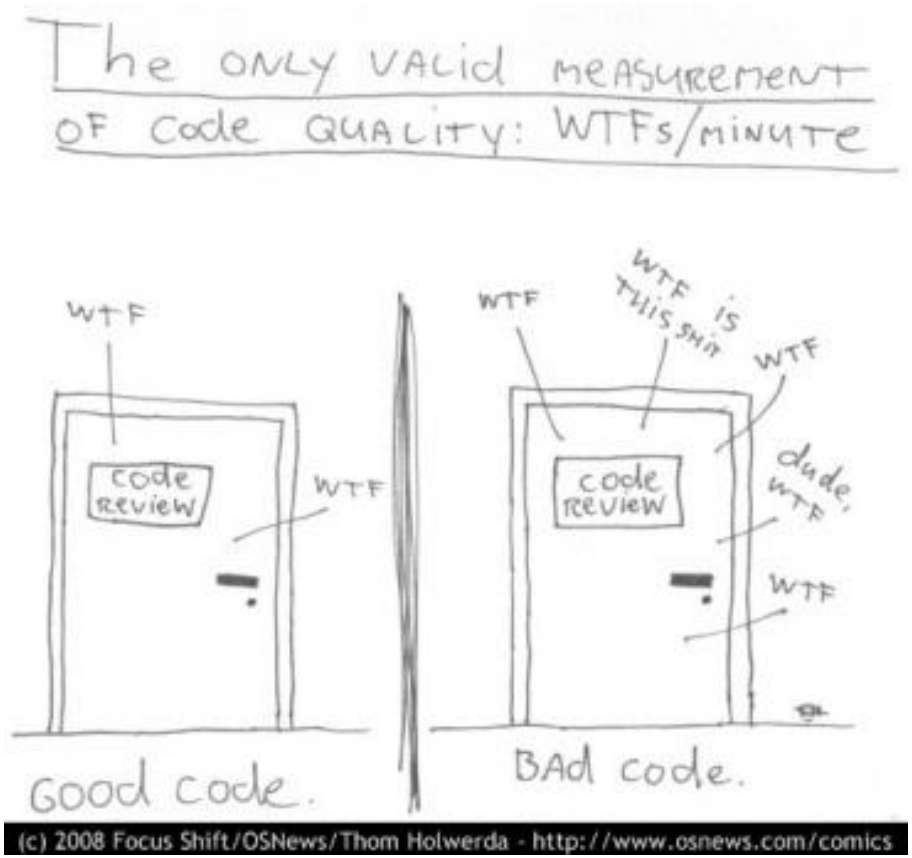
Проблемы?

CUSTIS®

- трудно проследить предназначение модуля
- непригодность модуля для повторного использования
- запутанный код => большая вероятность привнесения ошибок
- сложность сопровождения

CUSTIS®

Что делать? Как не писать такой код?



CUSTIS®

Можно применять поведенческие шаблоны (Visitor, Template Method)

CUSTIS®

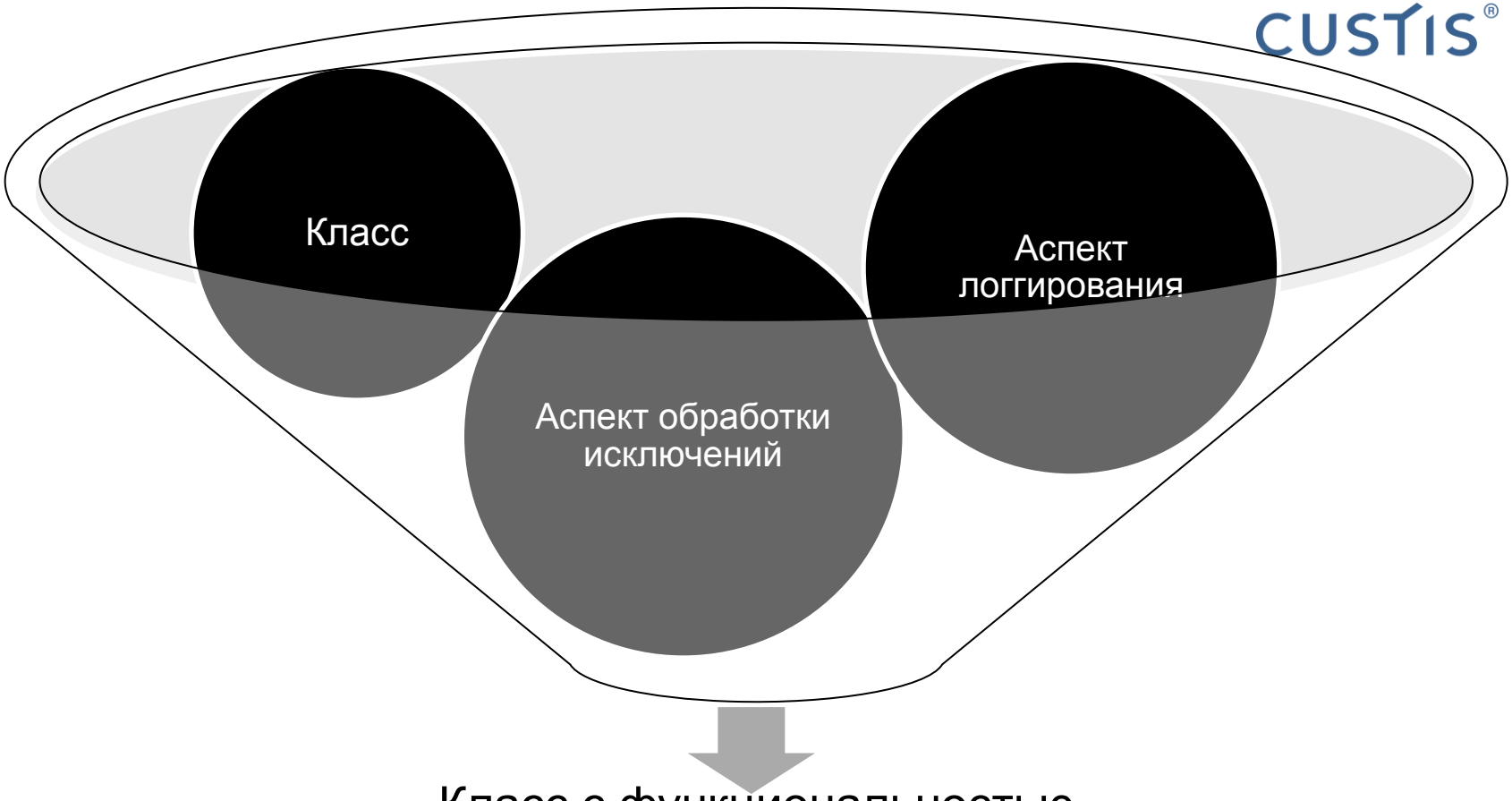
А можно использовать АОП.

CUSTIS®

Аспектно-ориентированное программирование (АОП) — это методика программирования в рамках классовой парадигмы, основанная на понятии аспекта — блока кода, инкапсулирующего сквозное поведение в составе классов и повторно используемых модулей (© http://ru.wikibooks.org)

CUSTIS®

CUSTIS®



Класс с функциональностью логгирования и обработки исключений

История АОП:

CUSTIS®

1974 – принцип разделения ответственности

1990е – исследования АОП

- Composition Filters
- Субъектно-ориентированное программирование
- Адаптивное программирование

1997 - Аспектно-ориентированное программирование (доклад на европейской конференции по ООП)

2001 – разработка АОП фреймворка AspectJ

Реализации АОП:

CUSTIS®

Для .NET:
PostSharp
Aspect.NET
LOOM.NET
Puzzle.NAspect
AspectDNG
Aspect#
Encase
Compose*
Seasar.NET
DotSpect (.SPECT)
Как часть функциональности The Spring.NET Framework

Для Object Pascal:
MeAOP
Infra

Для Java:
AspectJ
AspectWerkz (Now merged with AspectJ)
Byte Code Engineering Library
CaesarJ
Dynaop
JAC
Jakarta Hivemind
Javassist Home Page
JAsCo
JAML
JBoss AOP
Object Teams
PROSE
Reflex
The AspectBench Compiler for AspectJ (abc)
Как часть функциональности The Spring Framework
Seasar
The JMangler Project
InjectJ

Для JavaScript:
AOP плагин для библиотеки jQuery
Ajaxpect
AOP Fun with JavaScript (замещено проектом Ajaxpect)
Dojo — Dojo Toolkit
Aspectes

Для C/C++:
AspectC++
The XWeaver Project
FeatureC++

Для Lua:
AspectLua

Для Python:
Aspyct
Lightweight Python AOP
Logilab's aspect module
Python/Transwarp AOP Tutorial
PEAK
Pythius
PyPy

Для Ruby:
AspectR
Aquarium
Для ActionScript 2/3:
Fling
AIoC

Для PHP:
Seasar.PHP
AOP API for PHP
GAP: Generic Aspects for PHP
paspect

Для Perl:
The Aspect Module
Для Common Lisp:
AspectL

Для Cocoa:
AspectCocoa

Для XML:
AspectXML

Для Squeak Smalltalk
AspectS

Для ColdFusion:
ColdSpring

Для Flash ActionScript 2.0
as2lib

CUSTIS®

PostSharp

CUSTIS®

История PostSharp

Разработчик - Gael Fraiteur

- 2004 год - начало разработки
- 2006 год - первая альфа версия
- март 2007 - первая feature-complete версия
- осень 2007 - PostSharp 1.0 RC1. Начало разработки PostSharp 1.5
- осень 2008 - PostSharp 1.0 RTM
- начало 2009 - старт разработки PostSharp 2.0
- 2009 год - образование компании SharpCrafters

Текущая версия:

- PostSharp 2.0 - стабильная
- PostSharp 2.1 - CTP

CUSTIS®

Основные понятия АОП:

- Аспект (aspect)
- Совет (advice)
- Точка соединения (join point)
- Срез (pointcut)
- Внедрение (introduction)

CUSTIS®

Аспект

- модуль или класс, реализующий сквозную функциональность.

В PostSharp – класс, наследник System.Attribute.

CUSTIS®

Совет

- код, который может быть вызван из точки соединения.

В PostSharp – метод класса-аспекта, отмеченный специальным атрибутом.

CUSTIS®

Точка соединения

- точка в выполняемой программе, где следует применить совет.

В PostSharp – описывается атрибутом, который применяется при объявлении совета. Точками соединения могут быть: границы методов, обращения к свойствам, события, исключения и т.п.

CUSTIS®

Срез

- набор точек соединения. Срез определяет, подходит ли данная точка соединения к данному совету.

В PostSharp – описывается атрибутом, который определяет фильтр, к каким элементам кода применять совет. Пример совета: “границы всех публичных невиртуальных методов”.

CUSTIS®

Внедрение

- изменение структуры класса и/или изменение иерархии наследования для добавления функциональности аспекта в инородный код.

В PostSharp – выполняется на этапе посткомпиляции.

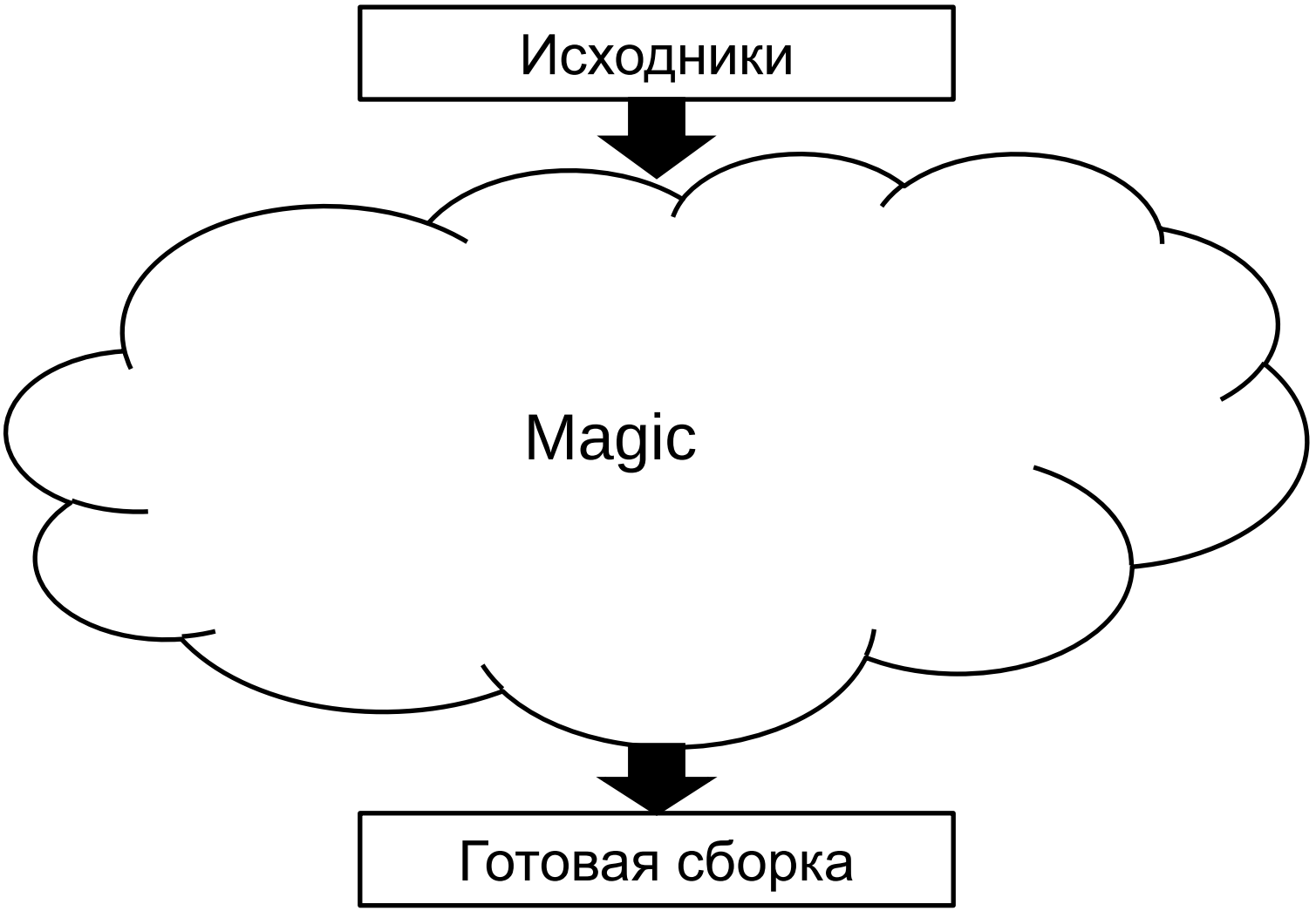
Реализация основных понятий АОП в PostSharp

CUSTIS®

Термин АОП	Реализация в PostSharp
Аспект	Класс, наследник System.Attribute
Совет	Метод класса-аспекта, отмеченный специальным атрибутом
Точка соединения	Описывается атрибутом, который применяется при объявлении совета. Точками соединения могут быть: границы методов, обращения к свойствам, события, исключения
Срез	Описывается атрибутом, который определяет фильтр, к каким элементам кода применять совет.
Внедрение	Выполняется на этапе посткомпиляции

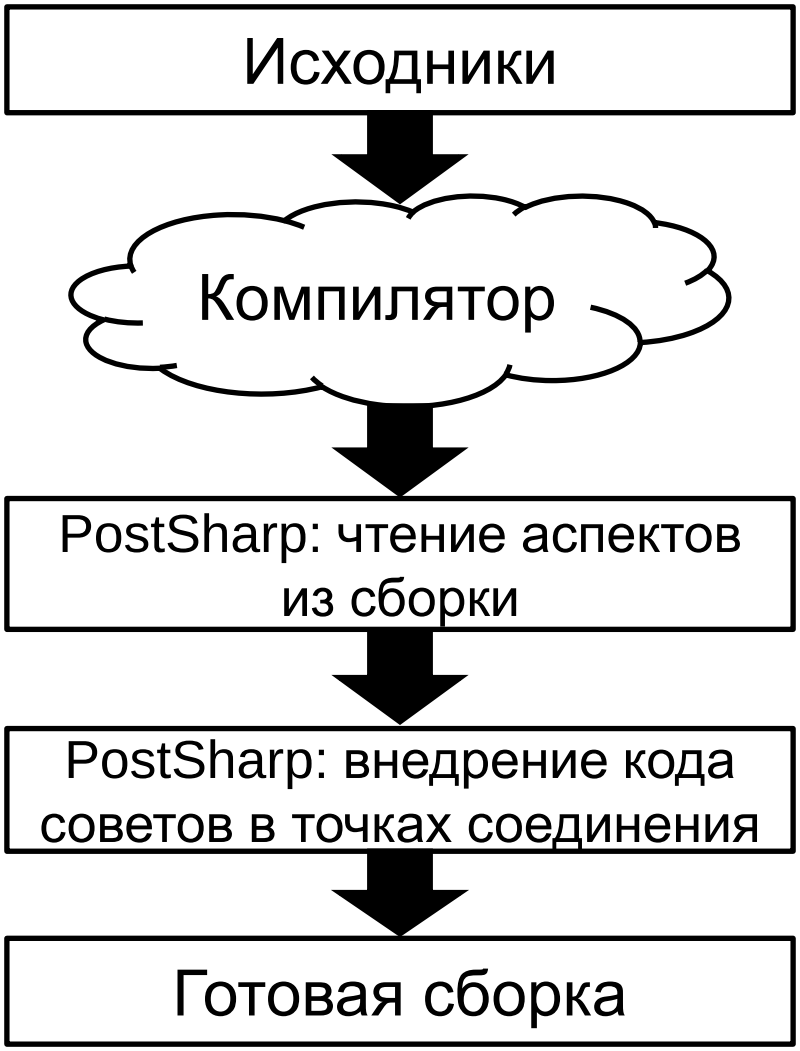
Как это работает?

CUSTIS®



Как это работает?

CUSTIS®



Подходы к использованию PostSharp

CUSTIS®

1. Использование базовых аспектов (OnMethodBoundaryAspect и т.п.). Хорошо подходит для решения простых задач.
2. Создание и использование композитных аспектов. Позволяет реализовывать сложные шаблоны поведения

Простые аспекты PostSharp

CUSTIS®

Аспект	Что делает	Где можно использовать
OnMethodBoundaryAspect	Позволяет отслеживать события начала и окончания вызова метода + исключения и успешное завершение вызова метода	Журналирование, транзакции
OnExceptionAspect	Добавляет обработчик исключений в метод, к которому применён аспект	Везде, где нужна обработка исключений
MethodInterceptionAspect	Заменяет вызов метода вызовами OnInvoke у аспекта	Диспетчеризация вызовов в другой поток
LocationInterceptionAspect	Заменяет вызов метода вызовами OnInvoke у аспекта	Реализация шаблона Наблюдатель (например, для добавления поддержки интерфейса INotifyPropertyChanged)
EventInterceptionAspect	Заменяет обращения к add и remove на методы аспекта. Также заменяет вызовы обработчиков события на вызов метода OnInvokeHandler у аспекта	Асинхронные события

Композитные аспекты PostSharp

CUSTIS®

- AssemblyLevelAspect
- TypeLevelAspect
- InstanceLevelAspect
- MethodLevelAspect
- LocationLevelAspect
- EventLevelAspect

Разработка композитных аспектов

CUSTIS®

1. Создать пустой аспект, унаследованный от базового аспекта (например InstanceLevelAspect)
2. Объявить реализуемый аспектом совет (метод, отмеченный атрибутом)
3. Отметить набор точек соединения, в которых следует применять совет (атрибутами)

Преимущества использования АОП:

CUSTIS®

- Улучшение декомпозиции системы на отдельные модули
- Упрощение сопровождения программы и внесения в нее изменений
- Возможность повторного использования кода, реализующего сквозную функциональность

Недостатки использования АОП:

CUSTIS®

- Нет структурированной информации, как разрабатывать программы с использованием аспектов
- В некоторых случаях использование АОП ухудшает понимание функционирования программной системы

Мифы об АОП:

CUSTIS®

1. АОП хорошо только для ведения журналов
2. АОП не решает никаких новых проблем
3. Шаблоны проектирования заменяют АОП
4. Аспекты скрывают схему работы программы
5. Аспекты затрудняют отладку
6. Нельзя выполнить модульный тест аспектов
7. АОП слишком сложен

Полезные ссылки:

CUSTIS®

1. http://www.sharpcrafters.com/postsharp/documentation
2. http://www.cs.ubc.ca/~gregor/
3. http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.37.4987&rep=rep1&type=pdf
4. http://en.wikipedia.org/wiki/Aspect-oriented_programming
5. http://ru.wikipedia.org/wiki/Аспектно-ориентированное_программирование

CUSTIS®

Вопросы?

CUSTIS®

Спасибо за внимание!