



Technische Universität Berlin



Fachgebiet Softwaretechnik

FlexTest

An Aspect-Oriented Framework for
Unit Testing

Dehla Sokenou and Matthias Vösgen

What is this talk about?

- A lot of test frameworks are around.
Reference: JUnit.
- They focus on organization, execution, reporting.

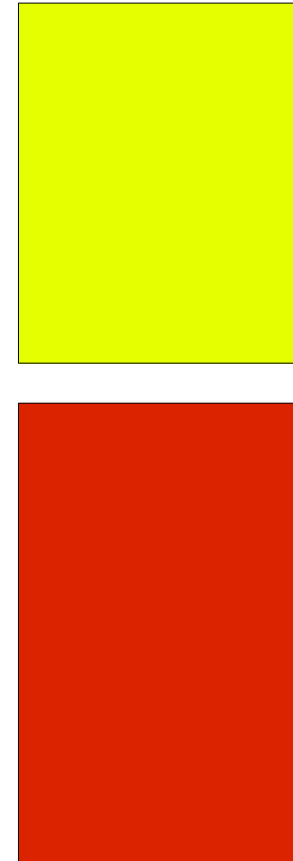
- FlexTest is also a test framework.
- It uses aspect-oriented programming (AOP).
- FlexTest applies AOP to OO-inherent test problems.
- It is implemented in AspectJ.

Outline

- AOP introduction
- Applications of AOP in the testing process
 - Encapsulation
 - Quantification over test cases
 - Hierarchy tests
 - Other
- Structure of FlexTest
- Conclusion

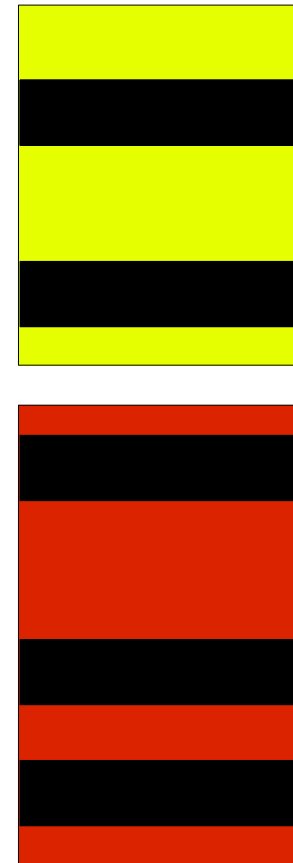
AOP introduction

- Concerns in the OO-world are separated by modules.




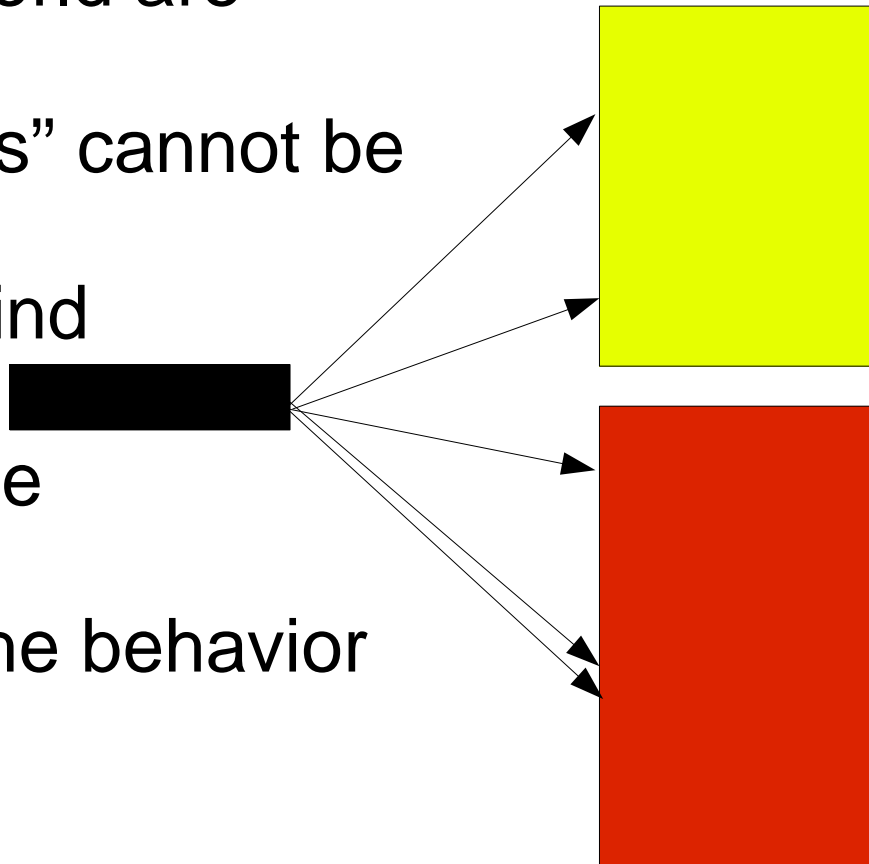
AOP introduction

- Concerns in the OO-world are separated by modules.
- “Crosscutting Concerns” cannot be isolated in this way.



AOP introduction

- Concerns in the OO-world are separated by modules.
- “Crosscutting Concerns” cannot be isolated in this way.
- AOP provides a new kind of module, the aspect: 
- The aspect contains the crosscutting behavior.
- It also defines where the behavior takes place.

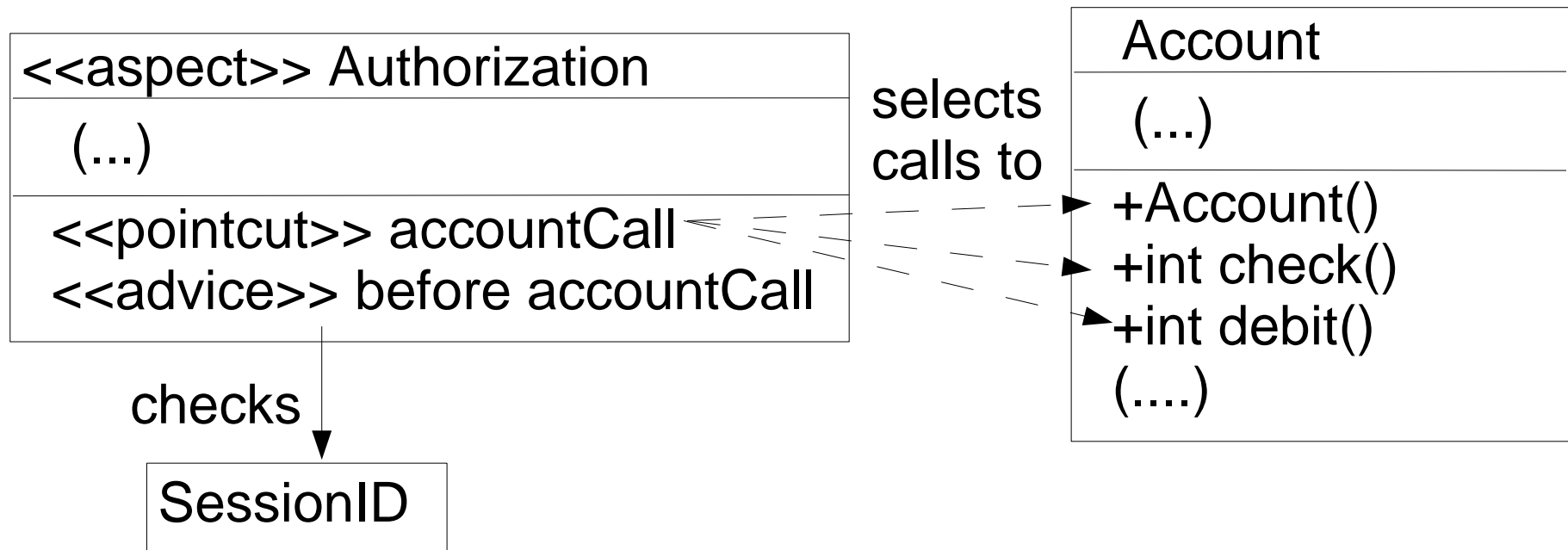


AOP terminology

- **Joinpoint**
A selectable point in the program (e.g. method call).
- **Pointcut**
A selection of a set of joinpoints.
- **Advice**
Specification of behavior for a pointcut.
- **Aspect**
Encapsulates pointcuts and advices.

AOP example

Specification: *“Before every method call of Account an authorization takes place”*



What to do in unit-testing

Abstraction level of the unit-test: Class-level.

Important tasks:

1. Write test cases.
2. Instrument the application so that it can be tested.
3. Organize the tests in a modular way.
4. Use a framework that takes care of execution, evaluation.

FlexTest uses AOP to help with 2. - 4.

Testing encapsulated members

Problem:

- Classes encapsulate their members.
- But some tests might need these.

Common Solution:

- Make them public during testing.
- Use language features (friend in C++, Java reflection API).

VectorDouble3
- double x, y, z
(...) +void setX(double x) (...)

AOP used for testing encapsulated members

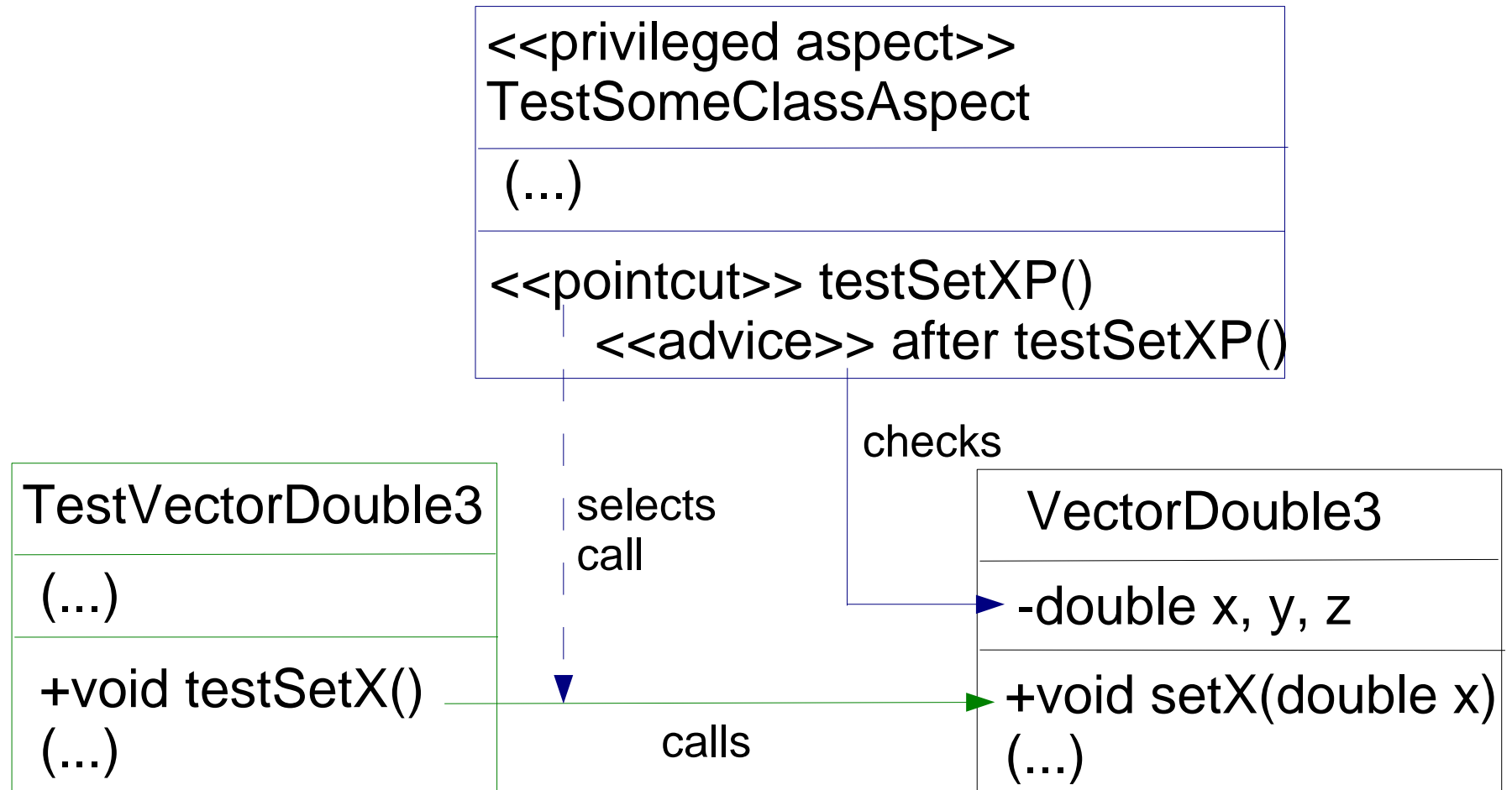
Idea:

- Encapsulation-breaking during testing can be seen as a structural crosscutting concern.
- The privileged aspect of AspectJ fits for this problem.

Solution:

- Test private members in a privileged aspect.

Example for testing encapsulated members



Quantifying over test cases

Problem:

- Some properties should hold over a whole context.
- They must be checked with every test case.

Common solution:

- Every test case checks for the context-property.
- The check must be added to every new test case.

AOP used for quantifying over test cases

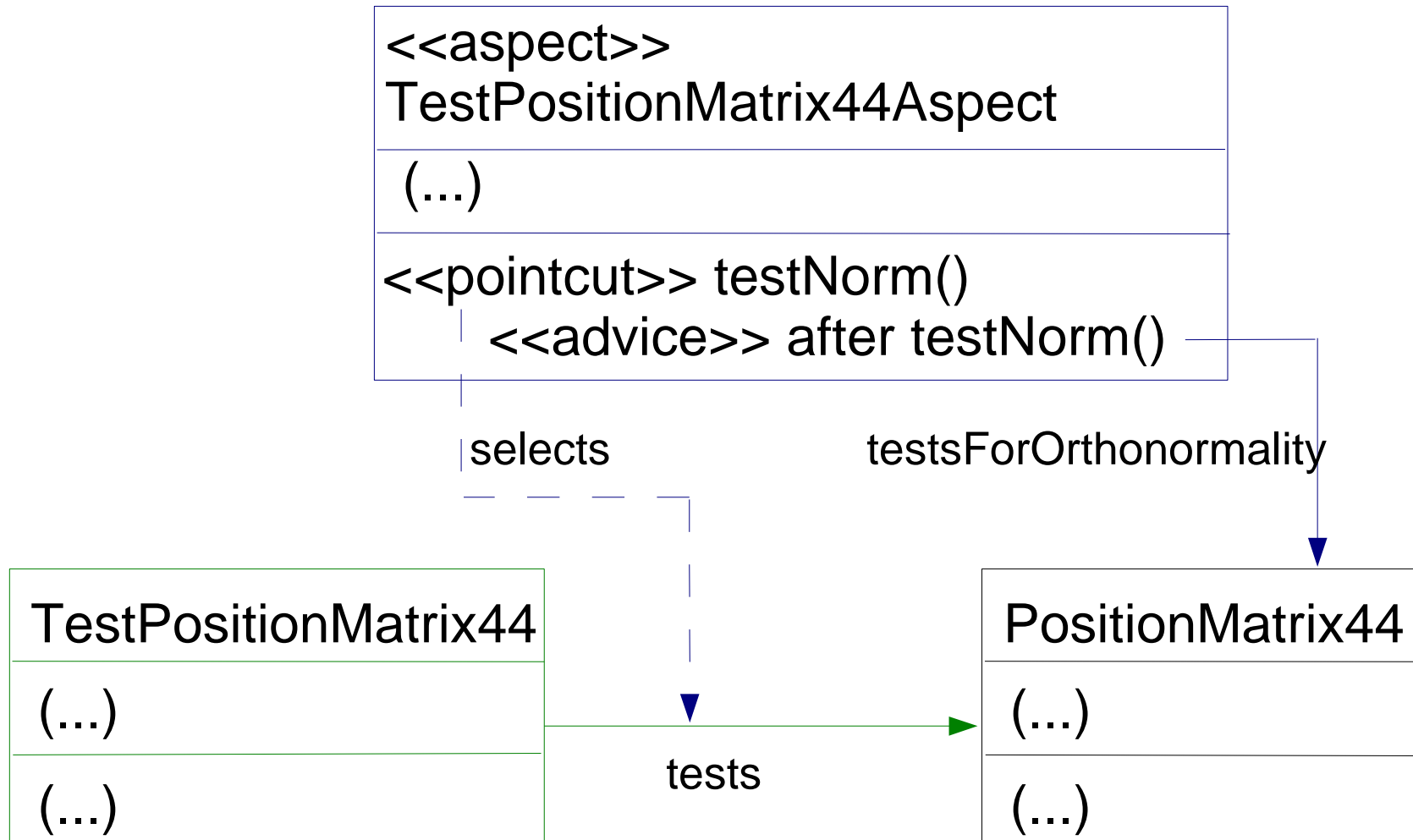
Idea:

- The test cases are implemented as method calls.
- So they can be selected by pointcuts.

Solution:

- A pointcut selects the test cases of the context.
- Its advice checks for the context-property.

Example for quantifying over test cases



Hierarchy testing

Problem:

- We can demand Liskov-conformity for a class hierarchy.
- This means (roughly): Every object of a superclass is replaceable by an object of one of its subclasses.
- How can we test this?

Common solution:

- Write a test suite for the superclass.
- Every new subclass-object must pass the suite.

AOP used for hierarchy testing

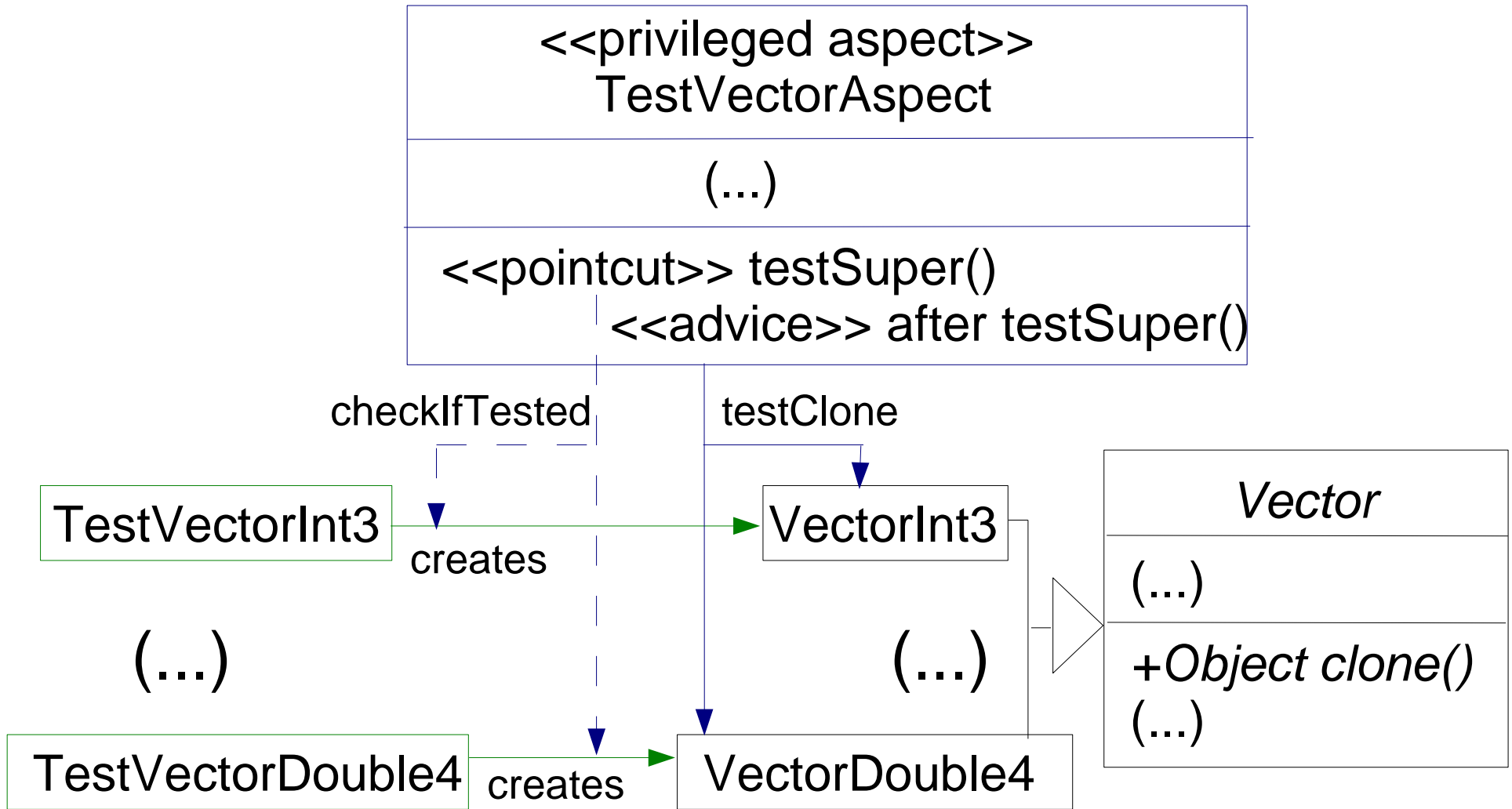
Idea:

- AOP can quantify over class hierarchies.
- Get relevant subclass-objects via AOP.

Solution:

- Use AOP to select the creation of subclass-objects.
- Run the superclass tests on any of those objects.
- Register the class as tested for this testsuite run.

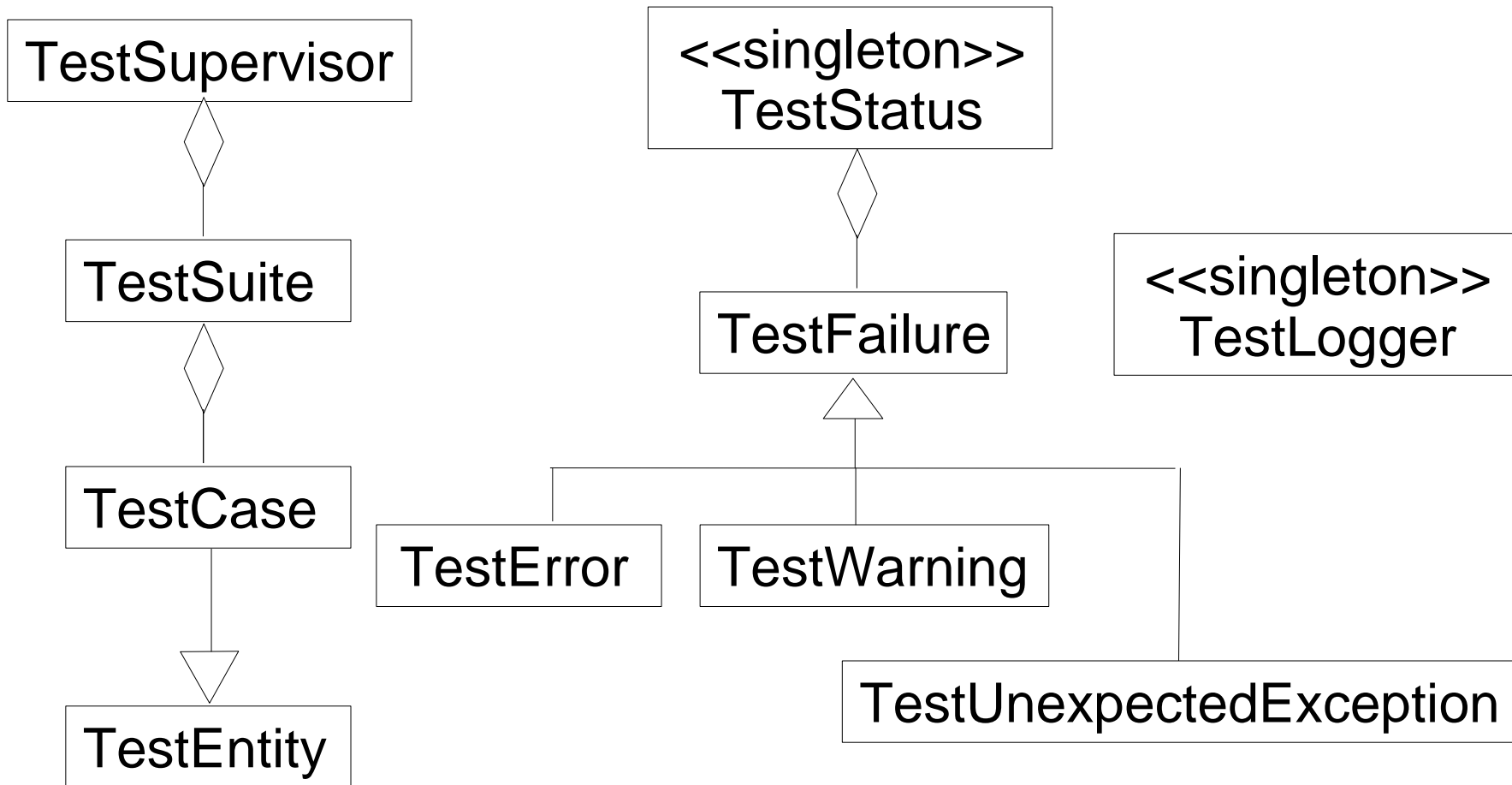
Example for hierarchy testing



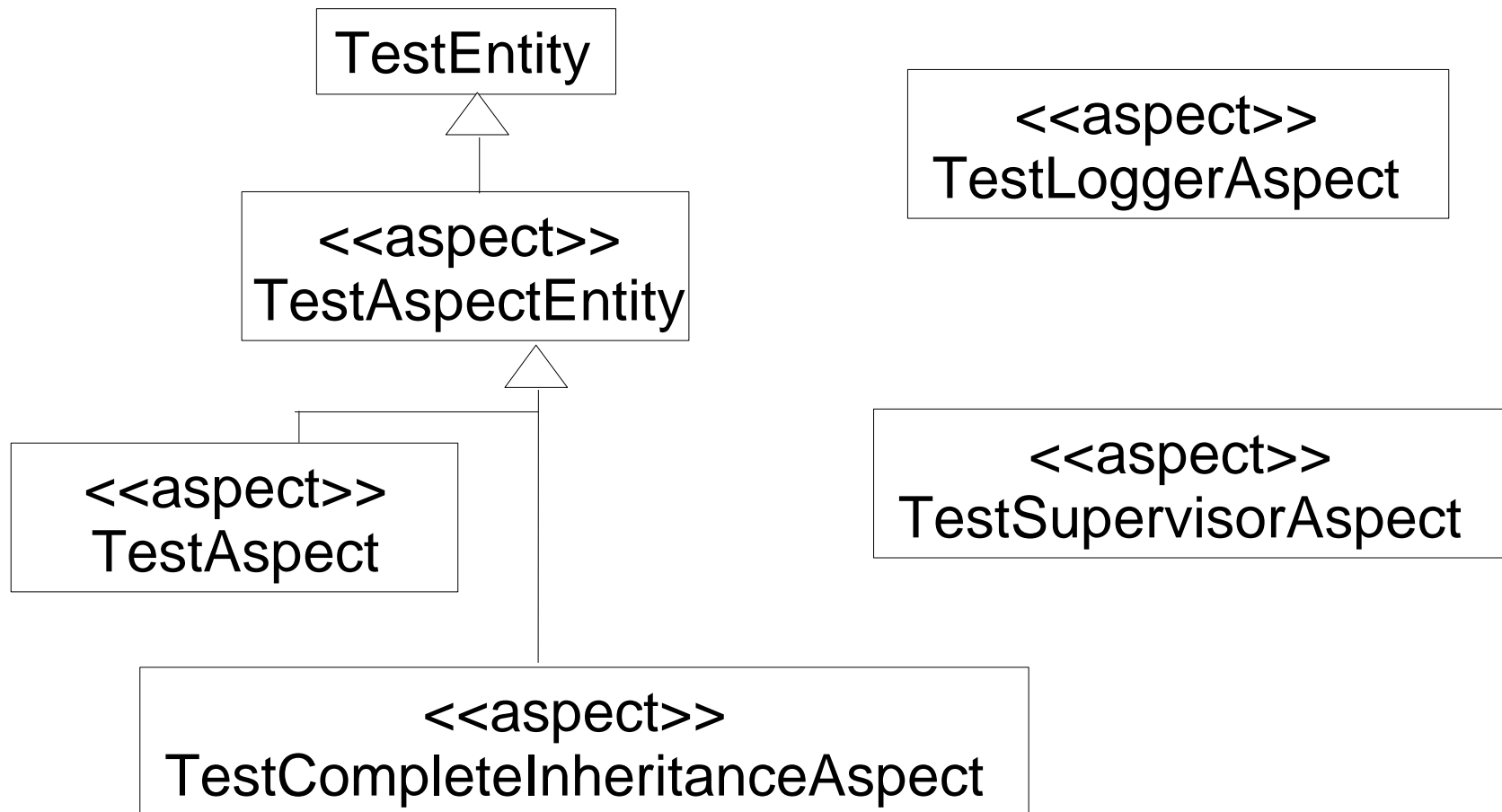
Other applications of AOP

- Logging
- Failure localization for report generation
- Mock Objects

FlexTest structure, classes



FlexTest structure, aspects



Pros and Cons of FlexTest

- + No additional instrumentation
- + Tests hierarchies
- + Checks properties for more than one test case at once
- + Separates concerns
- AOP skills needed
- Compiles slower
- “Hidden” tests
- Generally more complex

Conclusion

- FlexTest is a working test framework.
- It is enriched with AOP.
- Features include
 - modularized encapsulation breaking.
 - testing for properties over multiple test cases.
 - testing for Liskov-conformity (hierarchy testing).
 - logging implemented in a modular way
 - report generation implemented without using exceptions
- Worth approach is only measurable by case studies.

Thanks for listening!!!



You can download FlexTest from

<http://swt.cs.tu-berlin.de/~mvoesgen/Stuff/flextest.zip>