

# Software Engineering in der Praxis

## Praktische Übungen

# Strukturelles Testen

Matthias Meitner    Marc Spisländer

Lehrstuhl für Software Engineering  
Friedrich-Alexander-Universität Erlangen-Nürnberg

- 1 Inhalt
- 2 Nachlese
  - Funktionales Testen mit jUnit
- 3 Strukturelles Testen
  - White-Box Testverfahren
  - Überdeckungskriterien
- 4 Werkzeuge
  - djUnit
  - CodeCover
  - Clover



# Funktionales Testen

- **Was macht die Software**
- Betrachtet nur die Schnittstellen
- Stichworte: Äquivalenzklassen, Grenzwertanalyse, Error Guessing



# Funktionales Testen

- *Was macht die Software*
- *Betrachtet nur die Schnittstellen*
- *Stichworte: Äquivalenzklassen, Grenzwertanalyse, Error Guessing*



# Funktionales Testen

- Was macht die Software
- Betrachtet nur die Schnittstellen
- Stichworte: Äquivalenzklassen, Grenzwertanalyse, Error Guessing



# Strukturelles Testen

- Betrachte auch, *wie* die Software arbeitet
- Ziel: möglichst hohe strukturelle Überdeckung
  - Anweisungsüberdeckung
  - Verzweigungsüberdeckung
  - (Einfache) Bedingungsüberdeckung
  - Pfadüberdeckung
- Alternativbezeichnung: *White-Box-Testen*

# Strukturelles Testen

- Betrachte auch, *wie* die Software arbeitet
- Ziel: möglichst hohe strukturelle Überdeckung
  - Anweisungsüberdeckung
  - Verzweigungsüberdeckung
  - (Einfache) Bedingungsüberdeckung
  - Pfadüberdeckung
- Alternativbezeichnung: *White-Box-Testen*



# Strukturelles Testen

- Betrachte auch, *wie* die Software arbeitet
- Ziel: möglichst hohe strukturelle Überdeckung
  - Anweisungsüberdeckung
  - Verzweigungsüberdeckung
  - (Einfache) Bedingungsüberdeckung
  - Pfadüberdeckung
- Alternativbezeichnung: *White-Box-Testen*

# Einfache Überdeckungskriterien

## Anweisungsüberdeckung

**Gesucht:** Testfallmenge so, dass möglichst alle Anweisungen ausgeführt werden.

## Verzweigungsüberdeckung

**Gesucht:** Testfallmenge so, dass möglichst alle Verzweigungen verfolgt werden.

# Einfache Überdeckungskriterien

## Anweisungsüberdeckung

**Gesucht:** Testfallmenge so, dass möglichst alle Anweisungen ausgeführt werden.

## Verzweigungsüberdeckung

**Gesucht:** Testfallmenge so, dass möglichst alle Verzweigungen verfolgt werden.

# Komplexere Überdeckungskriterien

## Bedingungsüberdeckung

**Gesucht:** Testfallmenge so, dass möglichst

- alle atomaren Bedingungen jeden Wahrheitswert annehmen (einfach),
- alle Kombinationen von Wahrheitswerten getestet werden (mehrfach),
- Kombinationen, bei denen eine falsch ausgewertete atomare Bedingung das Ergebnis ändern würde (minimal mehrfach).

# Komplexere Überdeckungskriterien

## Bedingungsüberdeckung

**Gesucht:** Testfallmenge so, dass möglichst

- alle atomaren Bedingungen jeden Wahrheitswert annehmen (einfach),
- alle Kombinationen von Wahrheitswerten getestet werden (mehrfach),
- Kombinationen, bei denen eine falsch ausgewertete atomare Bedingung das Ergebnis ändern würde (minimal mehrfach).

# Komplexere Überdeckungskriterien

## Bedingungsüberdeckung

**Gesucht:** Testfallmenge so, dass möglichst

- alle atomaren Bedingungen jeden Wahrheitswert annehmen (einfach),
- alle Kombinationen von Wahrheitswerten getestet werden (mehrfach),
- Kombinationen, bei denen eine falsch ausgewertete atomare Bedingung das Ergebnis ändern würde (minimal mehrfach).

# Komplexere Überdeckungskriterien

## Bedingungsüberdeckung

**Gesucht:** Testfallmenge so, dass möglichst

- alle atomaren Bedingungen jeden Wahrheitswert annehmen (einfach),
- alle Kombinationen von Wahrheitswerten getestet werden (mehrfach),
- Kombinationen, bei denen eine falsch ausgewertete atomare Bedingung das Ergebnis ändern würde (minimal mehrfach).

# Komplexere Überdeckungskriterien

## Pfadüberdeckung

**Gesucht:** Testfallmenge so, dass alle möglichen Pfade beschriftet werden.

- mit n-facher Schleifenüberdeckung
- lineare Basen
- ...



# Komplexere Überdeckungskriterien

## Pfadüberdeckung

**Gesucht:** Testfallmenge so, dass alle möglichen Pfade beschriftet werden.

- mit n-facher Schleifenüberdeckung
- lineare Basen
- ...

# Komplexere Überdeckungskriterien

## Pfadüberdeckung

**Gesucht:** Testfallmenge so, dass alle möglichen Pfade beschriftet werden.

- mit n-facher Schleifenüberdeckung
- lineare Basen
- ...

# djUnit

- basiert auf Cobertura und JUnit
- integriert in Eclipse
- misst Anweisungs- und Verzweigungsüberdeckung
- kinderleichte Bedienung
- frei verfügbar (Open Source)

# djUnit

- basiert auf Cobertura und JUnit
- integriert in Eclipse
- misst Anweisungs- und Verzweigungsüberdeckung
- kinderleichte Bedienung
- frei verfügbar (Open Source)

# djUnit

- basiert auf Cobertura und JUnit
- integriert in Eclipse
- misst Anweisungs- und Verzweigungsüberdeckung
- kinderleichte Bedienung
- frei verfügbar (Open Source)

# djUnit

- basiert auf Cobertura und JUnit
- integriert in Eclipse
- misst Anweisungs- und Verzweigungsüberdeckung
- kinderleichte Bedienung
- frei verfügbar (Open Source)

# djUnit

- basiert auf Cobertura und JUnit
- integriert in Eclipse
- misst Anweisungs- und Verzweigungsüberdeckung
- kinderleichte Bedienung
- frei verfügbar (Open Source)

# CodeCover

- vergleichbar mit djUnit
- unterstützt auch Bedingungs- und Schleifenüberdeckung
- ebenfalls frei verfügbar (Open Source) von der Uni Stuttgart



# CodeCover

- vergleichbar mit djUnit
- unterstützt auch Bedingungs- und Schleifenüberdeckung
- ebenfalls frei verfügbar (Open Source) von der Uni Stuttgart

# CodeCover

- vergleichbar mit djUnit
- unterstützt auch Bedingungs- und Schleifenüberdeckung
- ebenfalls frei verfügbar (Open Source) von der Uni Stuttgart

# Clover

- vergleichbar mit djUnit
- zusätzliche Komfortfunktionen: Historie, Coverage-Cloud
- kommerziell

# Clover

- vergleichbar mit djUnit
- zusätzliche Komfortfunktionen: Historie, Coverage-Cloud
- kommerziell

# Clover

- vergleichbar mit djUnit
- zusätzliche Komfortfunktionen: Historie, Coverage-Cloud
- kommerziell