

Mocking mit Mockito

Dr. Andreas Schroeder

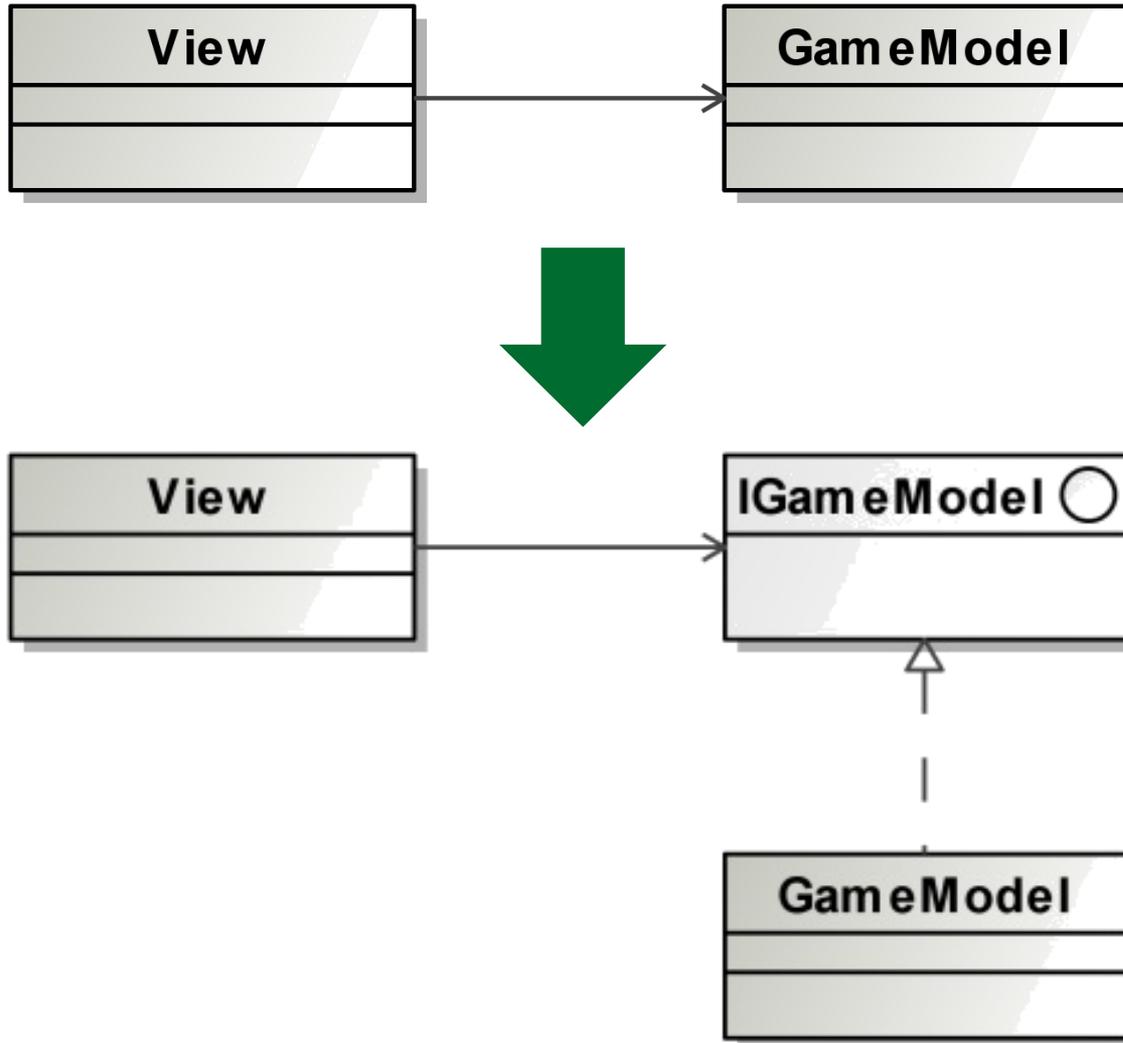


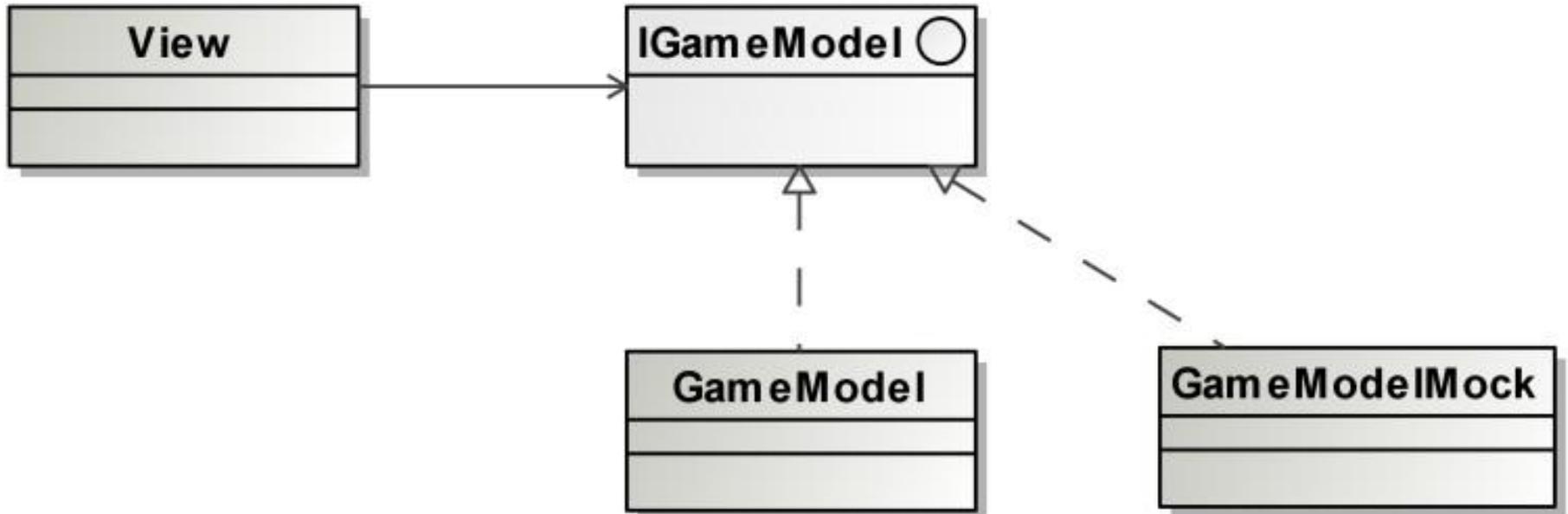
Dieses Video behandelt

- Das Problem: Unit-Testing mit Abhängigkeiten
- Dependency Inversion und Mocking
- Das Mockito-Framework



- Code-Units haben Abhängigkeiten, d.h. Referenzen auf andere Klassen:
 - Datenmodell
 - Benutzerschnittstelle
 - Datenbank-Schnittstelle
- Unit Tests sollen aber nur die Unit testen, nicht die Abhängigkeiten, denn das garantiert:
 - Bessere Fehlerlokalisierung
 - Unabhängigkeit der Testfälle







- Mock-Objekte
 - Ersetzen echte Objekte in Tests, um Abhängigkeiten der Unit zu erfüllen
 - Haben festgelegtes, test-spezifisches Verhalten
- Viele unterschiedliche Mocks mit unterschiedlichem Verhalten müssen generiert werden
- Mit Mocking-Frameworks kann viel Code gespart werden

Mockito lifecycle

- Create mock object

```
fWorld= mock(IWorld.class);
```

- Record behavior

```
when(fWorld.getWidth()).thenReturn(WIDTH);  
when(fWorld.getHeight()).thenReturn(HEIGHT);
```

- Use

```
fController= new GameOfLifeController(fWorld);
```

- Verify

```
verify(fWorld).changeCell(x, HEIGHT - 1);
```





In diesem Video wurde behandelt

- Das Problem: Unit-Testing mit Abhängigkeiten
- Dependency Inversion und Mocking
- Das Mockito-Framework