

Scala

Gedanken zu einer Programmiersprache

Sebastian Eidecker

16. März 2016

*Wer als Werkzeug nur einen Hammer hat,
sieht in jedem Problem einen Nagel.*

— Paul Watzlawick

Inhalt

IT im Wandel

Inhalt

IT im Wandel

Herausforderungen

Inhalt

IT im Wandel

Herausforderungen

Manifeste

Inhalt

IT im Wandel

Herausforderungen

Manifeste

Scala

Inhalt

IT im Wandel

Herausforderungen

Manifeste

Scala

Management Summary

Inhalt

IT im Wandel

Herausforderungen

Manifeste

Scala

Management Summary

Ein wenig Code

Inhalt

IT im Wandel

Herausforderungen

Manifeste

Scala

Management Summary

Ein wenig Code

Spannendes

Software Engineering

Software Engineering

IT im Wandel

Herausforderungen

Forderungen an IT

Forderungen an IT

- Stabilität und Resilienz

Forderungen an IT

- Stabilität und Resilienz
- Wertbeitrag

Forderungen an IT

- Stabilität und Resilienz
- Wertbeitrag
- Businessstreiber

Forderungen an IT

- Stabilität und Resilienz
- Wertbeitrag
- Businessstreiber

— Matthias Magnor – CEO Surface und Contract Logistics

IT im Wandel

Manifeste

Manifeste

- Antwortbereit, Widerstandsfähig, Elastisch, Nachrichtenorientiert (2013)

Manifeste

- Gut gefertigt, Stets Mehrwert, Gemeinschaft aus Experten, Produktive Partnerschaften (2009)

Manifeste

- Individuen und Interaktionen, Funktionierende Software, Zusammenarbeit mit dem Kunden, Reagieren auf Veränderung (2001)

Manifeste

- Antwortbereit, Widerstandsfähig, Elastisch, Nachrichtenorientiert (2013)
- Gut gefertigt, Stets Mehrwert, Gemeinschaft aus Experten, Produktive Partnerschaften (2009)
- Individuen und Interaktionen, Funktionierende Software, Zusammenarbeit mit dem Kunden, Reagieren auf Veränderung (2001)

Wo stehen wir?

Wo stehen wir im Wettbewerb?

Scala

Management Summary

Scalable Language

Eigenschaften

Eigenschaften

- Objektorientiert

Eigenschaften

- Objektorientiert
- Funktional

Eigenschaften

- Objektorientiert
- Funktional
- Statisch typisiert

Eigenschaften

- Objektorientiert
- Funktional
- Statisch typisiert
- Gewohnte Syntax („Java ohne Semikolon“)

Eigenschaften

- Objektorientiert
- Funktional
- Statisch typisiert
- Gewohnte Syntax („Java ohne Semikolon“)
- Ausdrucksstark (APIs/DSLs schreiben)

Eigenschaften

- Objektorientiert
- Funktional
- Statisch typisiert
- Gewohnte Syntax („Java ohne Semikolon“)
- Ausdrucksstark (APIs/DSLs schreiben)
- Immutable Objects als Konzept

Versprechen

Versprechen

- Produktivitätssteigerung

Versprechen

- Produktivitätssteigerung
- Höhere Codequalität

Versprechen

- Produktivitätssteigerung
- Höhere Codequalität
- durch
- Weniger Code

Versprechen

- Produktivitätssteigerung
- Höhere Codequalität
- durch
- Weniger Code
- Höheres Abstraktionsniveau

Versprechen

- Produktivitätssteigerung
- Höhere Codequalität
- durch
- Weniger Code
- Höheres Abstraktionsniveau
- Mehr Spaß

Scala und die Java-Plattform

Scala und die Java-Plattform

- Java-Bytecode, läuft auf JVM

Scala und die Java-Plattform

- Java-Bytecode, läuft auf JVM
- Java-Bibliotheken nutzbar

Scala und die Java-Plattform

- Java-Bytecode, läuft auf JVM
- Java-Bibliotheken nutzbar
- Bekannte IDEs

Scala und die Java-Plattform

- Java-Bytecode, läuft auf JVM
- Java-Bibliotheken nutzbar
- Bekannte IDEs
- Ähnlicher Paketierungs- und Buildprozess (sbt)

Vorteile

Vorteile

- Modern

Vorteile

- Modern
- Verständlich funktional

Vorteile

- Modern
- Verständlich funktional
- Java-Ökosystem

Vorteile

- Modern
- Verständlich funktional
- Java-Ökosystem
- Macht Spaß

Vorteile

- Modern
- Verständlich funktional
- Java-Ökosystem
- Macht Spaß
- Statisch typisiert

Vorteile

- Modern
- Verständlich funktional
- Java-Ökosystem
- Macht Spaß
- Statisch typisiert

Nachteile

Vorteile

- Modern
- Verständlich funktional
- Java-Ökosystem
- Macht Spaß
- Statisch typisiert

Nachteile

- Komplex

Vorteile

- Modern
- Verständlich funktional
- Java-Ökosystem
- Macht Spaß
- Statisch typisiert

Nachteile

- Komplex
- Zukunftssicher?

Vorteile

- Modern
- Verständlich funktional
- Java-Ökosystem
- Macht Spaß
- Statisch typisiert

Nachteile

- Komplex
- Zukunftssicher?
- Zahl an
Entwicklungssklaven

Vorteile

- Modern
- Verständlich funktional
- Java-Ökosystem
- Macht Spaß
- Statisch typisiert

Nachteile

- Komplex
- Zukunftssicher?
- Zahl an
Entwicklungssklaven
- Binärkompatibilität nicht
in alle Ewigkeit

Scala

Ein wenig Code

Eine Java-Klasse

Eine Java-Klasse

```
1 public class Person {  
2     private final String firstName;  
3     private final String lastName;  
4     public Person(String firstName, String lastName) {  
5         this.firstName = firstName;  
6         this.lastName = lastName;  
7     }  
8     public String getFirstName() {  
9         return firstName;  
10    }  
11    public String getLastName() {  
12        return lastName;  
13    }
```

Eine Java-Klasse

```
1  @Override
2  public boolean equals(Object o) {
3      if (this == o) return true;
4      if (o == null || getClass() != o.getClass()) return false;
5      Person person = (Person) o;
6      if (firstName != null ?
7          !firstName.equals(person.firstName) :
8          person.firstName != null) return false;
9      if (lastName != null ?
10         !lastName.equals(person.lastName) :
11         person.lastName != null) return false;
12     return true;
13 }
```

Eine Java-Klasse

```
1  @Override
2  public int hashCode() {
3      int result = firstName != null ? firstName.hashCode() : 0;
4      result =
5          31 * result + (lastName != null ? lastName.hashCode() :
6              0);
7      return result;
8  }
```

Businesslogik?

Dasselbe in Scala

Dasselbe in Scala

```
1 case class Person(firstName:String, lastName:String)
```

Scala

Spannendes

Akka

Akka

- Scalable real-time transaction processing

Akka

- Scalable real-time transaction processing
- Will die aktuellen Probleme lösen

Akka

- Scalable real-time transaction processing
- Will die aktuellen Probleme lösen
-

ScalaTest

ScalaTest

-

ScalaTest

-

-

ScalaTest

-
-
-

Scala

Mehr für Nerds

Mehr für Nerds

Mehr für Nerds

- Sprecht mich an

Mehr für Nerds

- Sprecht mich an
- Hands on-Termin bei Interesse

Mehr für Nerds

- Sprecht mich an
- Hands on-Termin bei Interesse
- Heiko Seeberger: „Durchstarten mit Scala. Tutorial für Einsteiger (2. Aufl.)“

