

A BUNCH OF NICE SCALA FEATURES

CASES CLASSES,  
PATTERN MATCHING &  
IMMUTABILITY

mail@timsteffens.de ~ <https://github.com/tmstff>



# WHAT TO EXPECT FROM THIS TALK?

- My experience:
  - After years of Java: some things are really annoying
  - Scala offers more suitable solutions
- My intention:
  - start questioning whether Java is always the appropriate tool
  - get you interested in Scala
    - => show you some nice Scala Features
    - => few slides, much demo code



# PLEASE INTERRUPT ME IN CASE...

- ... you have questions
- ... I'm going too fast - or too slow
- ... I'm talking nonsense ;-)
- I like things to be interactive!



# SHORT SURVEY

- # Java Developers
- # Java8 incl. Lambdas
- # Scala Coders
- # Scala Experts
- # never written Scala or Java
- # Like to use Scala at work



# JAVA BEANS VS. SCALA CASE CLASSES

## JAVA

```
public class User {

    private String name;
    private List<Order> orders;

    public User(String name, List<Order> orders) {
        this.name = name;
        this.orders = orders;
    }

    public User(String name) {
        this(name, new ArrayList<>());
    }

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

    public List<Order> getOrders() {
        return orders;
    }

    public void setOrders(List<Order> orders) {
        this.orders = orders;
    }

    @Override
    public boolean equals(Object o) {
        if (this == o) return true;
        if (o == null || getClass() != o.getClass()) return false;

        User user = (User) o;

        if (name != null ? !name.equals(user.name) : user.name != null) return false;
        return orders != null ? orders.equals(user.orders) : user.orders == null;
    }

    @Override
    public int hashCode() {
        int result = name != null ? name.hashCode() : 0;
        result = 31 * result + (orders != null ? orders.hashCode() : 0);
        return result;
    }

    @Override
    public String toString() {
        return "User{" +
            "name='" + name + '\'' +
            ", orders=" + orders +
            '}';
    }
}
```

```
public class Order {

    private int id;
    private List<Product> products;

    public Order(int id, List<Product> products) {
        this.id = id;
        this.products = products;
    }

    public Order(int id) {
        this(id, new ArrayList<>());
    }

    public int getId() {
        return id;
    }

    public void setId(int id) {
        this.id = id;
    }

    public List<Product> getProducts() {
        return products;
    }

    public void setProducts(List<Product> products) {
        this.products = products;
    }

    @Override
    public boolean equals(Object o) {
        if (this == o) return true;
        if (o == null || getClass() != o.getClass()) return false;

        Order order = (Order) o;

        if (id != order.id) return false;
        return products.equals(order.products);
    }

    @Override
    public int hashCode() {
        int result = id;
        result = 31 * result + products.hashCode();
        return result;
    }

    @Override
    public String toString() {
        return "Order{" +
            "id=" + id +
            ", products=" + products +
            '}';
    }
}
```

```
public class Product {

    private int id;
    private String category;

    public Product(int id, String category) {
        this.id = id;
        this.category = category;
    }

    public int getId() {
        return id;
    }

    public void setId(int id) {
        this.id = id;
    }

    public String getCategory() {
        return category;
    }

    public void setCategory(String category) {
        this.category = category;
    }

    @Override
    public boolean equals(Object o) {
        if (this == o) return true;
        if (o == null || getClass() != o.getClass()) return false;

        Product product = (Product) o;

        if (id != product.id) return false;
        return category != null ? category.equals(product.category) : product.category == null;
    }

    @Override
    public int hashCode() {
        int result = id;
        result = 31 * result + (category != null ? category.hashCode() : 0);
        return result;
    }

    @Override
    public String toString() {
        return "Product{" +
            "id=" + id +
            ", category='" + category + '\'' +
            '}';
    }
}
```



# JAVA BEANS VS. SCALA CASE CLASSES

## SCALA

```
case class User(name: String, orders: List[Order] = Nil)

case class Order(id: Int, products: List[Product] = Nil)
private List<Order> orders;

case class Product(id: Int, category: String)
public User(String name, List<Order> orders) {
    this.name = name;
    this.orders = orders;
}

public User(String name) {
    this(name, new ArrayList<>());
}

public String getName() {
    return name;
}

public void setName(String name) {
    this.name = name;
}

public List<Order> getOrders() {
    return orders;
}

public void setOrders(List<Order> orders) {
    this.orders = orders;
}

@Override
public boolean equals(Object o) {
    if (this == o) return true;
    if (o == null || getClass() != o.getClass()) return false;

    User user = (User) o;

    if (name != null ? !name.equals(user.name) : user.name != null) return false;
    return orders != null ? orders.equals(user.orders) : user.orders == null;
}

@Override
public int hashCode() {
    int result = name != null ? name.hashCode() : 0;
    result = 31 * result + (orders != null ? orders.hashCode() : 0);
    return result;
}

@Override
public String toString() {
    return "User{" +
        "name='" + name + '\'' +
        ", orders=" + orders +
        '\'';
}
}
```

```
public class Order {

    private int id;
    private List<Product> products;

    public Order(int id, List<Product> products) {
        this.id = id;
        this.products = products;
    }

    public Order(int id) {
        this(id, new ArrayList<>());
    }

    public int getId() {
        return id;
    }

    public void setId(int id) {
        this.id = id;
    }

    public List<Product> getProducts() {
        return products;
    }

    public void setProducts(List<Product> products) {
        this.products = products;
    }

    @Override
    public boolean equals(Object o) {
        if (this == o) return true;
        if (o == null || getClass() != o.getClass()) return false;

        Order order = (Order) o;

        if (id != order.id) return false;
        return products.equals(order.products);
    }

    @Override
    public int hashCode() {
        int result = id;
        result = 31 * result + products.hashCode();
        return result;
    }

    @Override
    public String toString() {
        return "Order{" +
            "id=" + id +
            ", products=" + products +
            '\'';
    }
}
```

```
public class Product {

    private int id;
    private String category;

    public Product(int id, String category) {
        this.id = id;
        this.category = category;
    }

    public int getId() {
        return id;
    }

    public void setId(int id) {
        this.id = id;
    }

    public String getCategory() {
        return category;
    }

    public void setCategory(String category) {
        this.category = category;
    }

    @Override
    public boolean equals(Object o) {
        if (this == o) return true;
        if (o == null || getClass() != o.getClass()) return false;

        Product product = (Product) o;

        if (id != product.id) return false;
        return category != null ? category.equals(product.category) : product.category == null;
    }

    @Override
    public int hashCode() {
        int result = id;
        result = 31 * result + (category != null ? category.hashCode() : 0);
        return result;
    }

    @Override
    public String toString() {
        return "Product{" +
            "id=" + id +
            ", category='" + category + '\'' +
            '\'';
    }
}
```



# JAVA BEANS VS. SCALA CASE CLASSES

## SCALA

**case class** User (name: String, orders: List[Order] = Nil)

**case class** Order (id: Int, products: List[Product] = Nil)

**case class** Product (id: Int, category: String)



DEMO



# SHORT SURVEY (END)

- # Like to use Scala at work



# PLEASE LEAVE FEEDBACK

- Thank you!