



# Java Records

JEP 395

Oli B. (Jul-2023)

**[Records] are classes that act as transparent for immutable data.**

**JEP 395**

# Java 11

```
public class Stapel {  
  
    private final long id;  
    private final BKZ branche;  
    private final StapelStatus status;  
  
    public Stapel(long id, BKZ branche, StapelStatus status) {  
        this.id = id;  
        this.branche = branche;  
        this.status = status;  
    }  
  
    public long getId() {  
        return id;  
    }  
  
    public BKZ getBranche() {  
        return branche;  
    }  
  
    public StapelStatus getStatus() {  
        return status;  
    }  
    ...  
}
```

# Java 11 (2)

```
...  
@Override  
public boolean equals(Object o) {  
    if (this == o) return true;  
    if (o == null || getClass() != o.getClass()) return false;  
    Stapel stapel = (Stapel) o;  
    return id == stapel.id && branche == stapel.branche && status == stapel.status;  
}
```

```
@Override  
public int hashCode() {  
    return Objects.hash(id);  
}
```

```
@Override  
public String toString() {  
    return "Stapel{" +  
        "id=" + id +  
        ", branche=" + branche +  
        ", status=" + status +  
        '}';  
}
```

```
}
```

# Java Klassisch

- Boilerplate-Code
- kann generiert werden
- vermindert Lesbarkeit
- wartungsunfreundlich

# @Lombok

```
@Value
public class Stapel {

    private final long id;
    private final BKZ branche;
    private final StapelStatus status;
}
```

# Java + Lombok

- cool, aber nur mit IDE-Unterstützung
- Builders
- starke Abhängigkeit
- it's a kind of magic
  - nicht immer problemlos
  - 278 Bugs (7.7.2023)



# Kotlin 1.1

```
data class Stapel(val id: Long, val branche: BKZ, val status: StapelStatus) {  
}
```



# Java 17

```
public record Stapel(long id, BKZ branche, StapelStatus status) {  
}
```

# Records

- wie Lombok, aber ohne Probleme
  - Getters
  - Equals/HashCode
  - toString()



# Einschränkungen

- final
- keine privaten Attribute



# Canonical Constructor

- zusätzliche Konstruktoren möglich
- Validierungen möglich
- kommen bei Deserialisierung zum Einsatz



```
public record Stapel(long id, BKZ branche, StapelStatus status) {  
  
    public Stapel {  
        if (id < 0) {  
            throw new IllegalArgumentException("not allowed: " + id);  
        }  
    }  
}
```

# Einsatzgebiete

- DTO
- Value Objects

```
record Stapel(  
    long id,  
    BKZ branche,  
    StapelStatus status) {  
  
}
```

[Records] are  
classes that act as  
transparent for  
immutable data.  
(JEP 395)



# Fazit

6.10. Workshop Testcontainers  
<https://www.jugs.org/veranstaltungen/testcontainers/>

Nie war es einfacher,  
Klassen für den  
Datenaustausch zu  
schreiben.

```
record Stapel(  
    long id,  
    BKZ branche,  
    StapelStatus status) {  
  
}
```

 [ob@jugs.org](mailto:ob@jugs.org)

 [@JitStorm](https://twitter.com/JitStorm)