



Google Collections

Orientation in Objects GmbH

Weinheimer Str. 68
68309 Mannheim

www.oio.de
info@oio.de

Version: 1.0

Gliederung

- Los geht's ...
- Praktische Helferlein
- Collections
- Wie geht's weiter?

Falk Sippach

Training

Beratung

Projekte



Java und XML

Besuchen Sie uns:
OIO @ JAX Expo

) Software Factory)

- Schlüsselfertige Realisierung von Java Software
- Individualsoftware
- Pilot- und Migrationsprojekte
- Sanierung von Software
- Software Wartung

) Object Rangers)

- Unterstützung laufender Java Projekte
- Perfect Match
- Rent-a-team
- Coaching on the project
- Inhouse Outsourcing

) Competence Center)

- Schulungen, Coaching, Weiterbildungsberatung, Train & Solve-Programme
- Methoden, Standards und Tools für die Entwicklung von offenen, unternehmensweiten Systemen

Gliederung

- Los geht's ...
- Praktische Helferlein
- Collections
- Wie geht's weiter?

Big Brother is watching you!

The slide features a central 3D orange octopus character with a grumpy expression, looking towards the right. Surrounding the octopus are several Google-related logos and icons:

- Google Deutschland
- Google Mail
- Google Calendar
- Picasa
- Google Earth
- Google Chrome
- Google Gears
- Google Guice
- ANDROID
- A red toolbox icon with a white 'G' on it.

A search bar is visible above the Google Mail logo, containing the text "Google-Suche" and "Auf gut Glück!".

At the bottom right, there is a yellow box with the text "Google Collections Library".

Die schönen Seiten ...

- Erweiterung des Java Collection Framework
- Neue Datenstrukturen und neue Implementierungen
- Praktische Helferlein
- Konsequente Ausnutzung von Java 5 Features



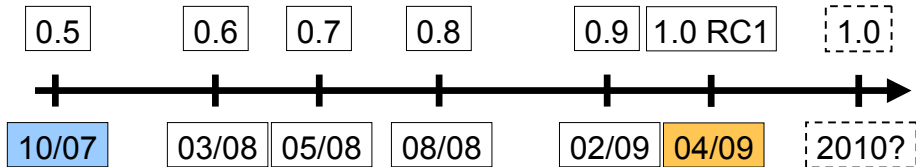
Die häßlichen Seiten ...

- keine bekannt ... (-:



Historie und Roadmap

- 20 % Projekt von Kevin Bourrillion und Jared Levy
- **Achtung:** "... the API is still subject to change."



Gliederung

- Los geht's ...
- **Praktische Helferlein**
- Collections
- Wie geht's weiter?

- Vorbedingungen
- equals, hashCode
- Transformationen, Filter, Randbedingungen
- ComparatorUtils, Iterables, Iterators
- Joiner

Vorbedingungen

vorher

```
if (arg.getDies() == null) {  
    throw new NullPointerException("arg dies")  
}  
if (!arg.isJenes()) {  
    throw new IllegalArgumentException(arg + " is not..")  
}
```

Google Collections

```
import static com.google...base.Preconditions.*;  
  
checkNotNull(arg.getDies(), "arg dies")  
checkArgument(arg.isJenes(), "%s is not..", arg)
```

```
@Override
public boolean equals(Object obj) {
    if (obj instanceof Person) {
        Person that = (Person) obj;
        return Objects.equal(this.name, that.name)
            && Objects.equal(this.age, that.age)
            && Objects.equal(this.address, that.address);
    }
    return false;
}

@Override
public int hashCode() {
    return Objects.hashCode(name, age, address);
}
```

```
Function<Float, Float> celToFahr = new Function<Float,Float>() {
    public Float apply(Float from) {
        return from * 9 / 5 + 32;
    }
};
```

```
List<Float> celsius = Lists.newArrayList(0.f, 20.f, 100.f);
```

```
Lists.transform(celsius, celToFahr); // => [32.0, 68.0, 212.0]
```

```
Sets.filter(Sets.newHashSet(null, 1, 2), Predicates.notNull());
// => [1, 2]
```

vorher

```
public void addToList(Element element) {  
    Preconditions.checkNotNull(element.getDies() > 0);  
    Preconditions.checkNotNull(element.getDas() != "");  
    list.add(element)  
}
```

Google Collections

```
list = Constraints.constrainedList(new  
    ArrayList<Element>(), new Constraint<Element>() {  
        public void checkElement(Element element) {  
            Preconditions.checkNotNull(..);  
            Preconditions.checkNotNull(..);  
        }  
    });
```

```
ComparatorUtils.min("bar", "foo", null); // => bar  
ComparatorUtils.max("bar", "foo", null); // => foo
```

```
List list1 = Arrays.asList(42);  
List list2 = Arrays.asList(1, 2, 3, 4);  
  
Iterables.getOnlyElement(list1); // => 42  
Iterators.getOnlyElement(emptySet.iterator(), 23); // => 23  
  
Iterables.get(list2, 1); // => 2  
Iterables.getLast(list2); // => 4
```


vorher

```
List<Integer> list = Arrays.asList(1, 2, null, 3, 4);
StringBuilder sb = new StringBuilder();
for (Integer number : list) {
    sb.append(number);
    sb.append("; ");
}
```

Google Collections

```
List<Integer> list = Arrays.asList(1, 2, null, 3, 4);
Joiner.on("; ").skipNulls().join(list); // => 1; 2; 3; 4
```

- Los geht's ...
- Praktische Helferlein
- **Collections**
- Wie geht's weiter?

- Statische Fabrikmethoden, Builder
- Multiset, Multimap
- BiMap
- ImmutableCollection/List/Set/Map/...

Statische Fabrikmethoden

vorher

```
Map<KeyLongTypeName, ValueLongTypeName>  
myVeryImportMap = new HashMap<KeyLongTypeName,  
ValueLongTypeName> ();
```

Google Collections

```
Map<KeyLongTypeName, ValueLongTypeName>  
myVeryImportMap = Maps.newHashMap ();
```

```
Lists.newArrayList("Montag", "Dienstag", "Mittwoch");
Sets.newHashSet("Bube", "Dame", "König");

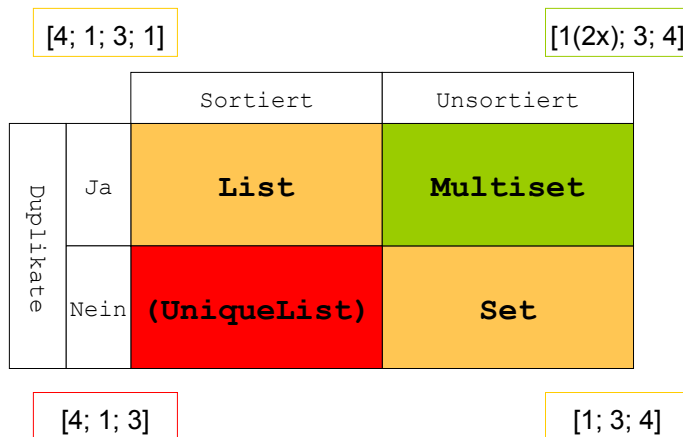
LinkedList<Double> linkedList = Lists.newLinkedList();
Collections.addAll(linkedList, 1.0, 2.0, 3.0);
```

```
EnumMap<SomeEnum, String> enumMap =
    Maps.newEnumMap(SomeEnum.class);

Map<SomeEnum, String> hashMap =
    Maps.newHashMap(enumMap);

ClassToInstanceMap<String> cToIMap =
    Maps.newClassToInstanceMap();
```

- add(4); add(1); add(3); add(1);



```
import static java.util.Arrays.asList;

Multiset<String> fahrzeuge = HashMultiset.create();

fahrzeuge.addAll(asList("Auto", "Boot", "Roller"));
fahrzeuge.addAll(asList("Auto", "Roller", "Boot"));
fahrzeuge.addAll(asList("Auto", "Fahrrad"));

fahrzeuge.count("Gibts nicht"); // => 0
fahrzeuge.count("Auto"); // => 3
fahrzeuge.size(); // => 8

fahrzeuge.remove("Auto");
fahrzeuge.add("Auto", 3);
fahrzeuge.count("Auto"); // => 5
```

vorher

```
Map<Person, List<Address>> map = ...
List<Address> addresses = map.get(p1);
if (addresses == null) {
    addresses = new ArrayList();
    map.put(p1, addresses);
}
addresses.add(a3);
```

Google Collections

```
MultiMap<Person, Address> multimap =
    ArrayListMultimap.create();
multimap.put(p1, a3);
```

```
BiMap<String,Integer> biMap =  
    HashBiMap.create();  
  
biMap.put("eins", 1);  
biMap.put("zwei", 2);  
biMap.put("drei", 3);  
  
assertEquals(biMap, biMap.inverse().inverse());
```

```
ImmutableSet<Integer> LUCKY_NUMBERS =  
    ImmutableSet.of(4, 8, 15, 16, 23, 42);
```

```
ImmutableMap<String, Integer> ENGLISH_TO_INT =  
    new ImmutableMap.Builder<String, Integer>()  
        .put("four", 4)  
        .put("eight", 8)  
        .put("fifteen", 15)  
        .put("sixteen", 16)  
        .put("twenty-three", 23)  
        .put("forty-two", 42)  
        .build();
```

- Los geht's ...
- Praktische Helferlein
- Collections
- **Wie geht's weiter?**

- sehr stabil
- nahtlose Erweiterung des Java Collection Framework
 - Komfortfunktionen, Fluent API
- gute API (Joshua Bloch approved)
- gute Dokumentation (JavaDoc)
- geringe Größe: 500 KByte
- Performance (intern erprobt bei Google)

Warum vielleicht (noch) nicht verwenden?

- bis 1.0 Änderung der API möglich
- manches könnte in Java 7 überflüssig werden
 - z. B. Type-Inference Vorschlag für Generics

Josh Bloch als Vorredner eines Google Collection Vortrags

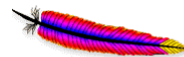
1. "Minimize Mutability"
2. "When in doubt, leave it out"
 - kleine, klare Schnittstellen schaffen
3. "Know and use the libraries"

"I [J. Bloch] believe that Google collections does all of these things."

http://www.youtube.com/watch?v=ZeO_J2OcHYM

Google vs. Apache Commons Collections

- Commons nicht Java 5 fähig
- Commons brechen Verträge
- Weiterentwicklung?



Apache Commons



Fragen ?

Orientation in Objects GmbH

Weinheimer Str. 68
68309 Mannheim

www.oio.de
info@oio.de



Vielen Dank für Ihre Aufmerksamkeit !

Orientation in Objects GmbH

Weinheimer Str. 68
68309 Mannheim

www.oio.de
info@oio.de

Version: 1.0

Mehr von OIO zum Thema...

- Schulung: Einführung in GWT (Google Web Toolkit)
 - <http://www.oio.de/google-web-toolkit-schulung.htm>
- Artikel: JavaScript-Code mit Google Web Toolkit erzeugen
 - <http://www.oio.de/public/opensource/gwt/google-web-toolkit.pdf>