

Google Guava

Miroslav Sikora

Wincor Nixdorf



Common Concepts

- Immutable Objects
defensive programming
- Builder pattern
step by step creation of immutables
class names in plural: Map -> Maps
- Fluent Interface
more readable code
- Static import

Core packages

- **base**
the "corest of the core", Joiner, Splitter,...
- **primitives**
working with int, long, double, byte, **unsigned** . . .
- **collect**
collection types, implementations, algorithms
- **io**
streams, buffers, files, . . .
- **net**
URLs, domain names, IP addrs,
- **cache**
cache loader, builder, stats,...
- **eventbus**
publish-subscribe-style communication

Collections misc.

- **Diamond operator in Java 6**

```
Map<String, Map<Long, List<String>>> map =  
Maps.newHashMap();
```

- **Multisets**
 - allows duplicate elements
 - also called bag
 - provide histograms
 - **Multimaps**
 - each unique key has a list of values
 - ListMultimap
 - SetMultimap – doesn't store duplicate key-value pairs
 - **BiMap**
 - map with unique values
 - can be inverted so
bimap.inverse().inverse() == bimap
- Table - index on more than one key at a time

ClassToInstanceMap -

Collections - Iterables & Iterators

- each method in Iterables has a corresponding Iterator-based method in the Iterators class
- Iterators – iterator manipulation
- all of the iterators produced in Iterators are *lazy*
- Most useful methods: concat, cycle, filter, **forArray**, limit, **skip**, peekingIterator, size

Collections – Predicates & Functions

- Predicate interface – filter collection
 - Built-in predicates: `isNull()`, `notNull()`, `alwaysTrue()`, `and()`, `or()`

```
Iterables.filter(listOfNames, not(in(blackList)));
```

- Function interface – transform collection
 - `Functions.forPredicate(Predicate)` – transform predicate to function.
 - `Functions.forMap(Map)` – performs a map lookup

Immutable & Forwarding Collections

- They don't like null
- They have builders
- Memory efficiency
- Order is preserved

```
ImmutableList<String> of =  
    ImmutableList.of("a", "b", "c", "d");
```

Forwarding Collections

- collection which forwards all its method calls to another collection
- Decorator pattern

Orderings

- Combine several Comparators
- Provides min, max, leastOf, isOrdered,...
- ```
List<Person> sortedCopy = Ordering
 .from(byLastName)
 .compound(byFirstName)
 .nullsFirst()
 .reverse()
 .sortedCopy(persons);
```



# Joiner & Splitter

- From the *Top 3 Most Bizarrely Missing From The JDK Class Libraries* list: joining pieces of text with a separator.

```
int[] numbers = { 1, 2, 3, 4, 5 };
String numbersAsString = Joiner.on(";")
 .skipNulls()
 .useForNulls("NODATA")
 .join(Ints.asList(numbers))
```

- Break strings into substrings

```
Iterable<String> split = Splitter.on(",")
 .omitEmptyStrings()
 .trimResults()
 .split(names);
```

# Object Helpers

- `Objects.equal(Object, Object)` - null safe equals.
- `Objects.hashCode(Object...)` - easy way to get a hash code based on multiple fields of your class.
- `Objects.toStringHelper(Object)`  

```
// Returns "ClassName{x=1, y=foo}"
Objects.toStringHelper(this)
 .add("x", 1)
 .add("y", "foo")
 .toString();
```
- `Objects.firstNonNull(Object, Object)` - Returns the first of two given parameters that is not null

# Preconditions

- methods to verify correct arguments and state

```
if (count <= 0) {
 throw new IllegalArgumentException("must be
 positive: " + count);
}
```

- With `import static Preconditions.*`

```
checkArgument(count > 0, "must be positive: %s",count);
```

- the sense of the expression is inverted
- declare what you expect to be *true*
- easy assignment after not null check

```
this.field = checkNotNull(parameter);
```

# Pros & Cons



- Nice Joiner
- **Static Constructors**
- Shorter and readable code
- Functional style
- Generics
- Immutability
- Factory methods
- Builder pattern



- Debugging of fluent interfaces
- Size ~ 1,5MB
- Tutorials

# Useful Links

- <http://code.google.com/p/guava-libraries/w/list>
- <http://codemunchies.com/tag/guava/>
- <http://scaramoche.blogspot.com/search/label/guava>
- <http://www.javalobby.org/articles/google-collections/>