Java 8

Build #84

Robert Bachmann

JSUG Meeting #52

Outline

- Interface additions
- Lambda Syntax
- Library additions
    - java.util.function
    - java.time
    - java.util.Optional
    - java.util.streams

Do try this at home

- JDK8, http://jdk8.java.net/download.html
- Optional: IntelliJ IDEA 12.1

Interface additions

- Can add static methods to interfaces
- Can add default methods to interfaces
- Functional interfaces: An interface with one abstract method

Examples

```java
package java.util;

public interface List<E>
  extends Collection<E> {

  /* [...] */

  default void sort(Comparator<? super E> c) {
      Collections.sort(this, c);
  }
}
```

Examples

```
package java.util.function;

@FunctionalInterface
public interface Consumer<T> {
  void accept(T t);
}
```

Examples

```java
public interface Iterable<T> {
  Iterator<T> iterator();

  default void forEach(
    Consumer<? super T> action) {
        Objects.requireNonNull(action);
        for (T t : this) {
            action.accept(t);
        }
    }
}
```

Default method rules

- Specific interface over parent interface
- Object method over default method
- Can not provide toString, hashCode, equals

Lambda syntax

- Used in conjunction with functional interfaces
- Alternative to anonymous inner classes

Examples

```
List<String> list = Arrays.asList(”a”,”b”);

list.forEach(new Consumer<String>() { // Java 7
   public void accept(String s)
   { System.out.println(s); }
});

// Java 8
list.forEach( s -> {System.out.println(s);} );

list.forEach( s -> System.out.println(s) );

list.forEach( System.out::println );
```

Examples

```
List<Person> personList = ...;

// Java <= 7
Collections.sort(personList,
 new Comparator<Person>() {
  @Override
  public int compare(Person p1, Person p2) {
    return Long.compare(p1.getAge(), p2.getAge());
  }
});
```

Examples

```
List<Person> personList = ...;

// Java 8
personList.sort(
 (p1,p2) -> {
  return Long.compare(p1.getAge(), p2.getAge())
 });

// or:
personList.sort(
 (p1, p2) -> Long.compare(p1.getAge(), p2.getAge())
);

// alternative approach:
personList.sort(
 Comparators.comparing(Person::getAge));
```

java.util.function (1/2)

- Consumer (T $\rightarrow$ void)
- Supplier (void $\rightarrow$ T)
- Predicate (T $\rightarrow$ boolean)
- Function (T $\rightarrow$ R) and UnaryOperator (T $\rightarrow$ T)
- BiFunction, BiPredicate, BinaryOperator

java.util.function (2/2)

- IntConsumer (int $\to$ void)
- IntSupplier (void $\to$ int)
- IntPredicate (int $\to$ boolean)
- IntFunction (int $\to$ R)
- IntUnaryOperator (int $\to$ int)
- ToIntFunction (T $\to$ int)
- ObjIntConsumer, ToIntBiFunction
- Same for Double and Long

java.time

- Similar to JodaTime
- Instant & Clock
- LocalDateTime, LocalDate, LocalTime
- OffsetDateTime, OffsetTime
- ZonedDateTime

java.util.Optional

```
Optional<String> name = f();

// example 1
System.out
 .println(”Hello␣” + name.orElse(”user”));

// example 2
name
 .ifPresent( s -> System.out.println(”Hello␣” + s) );
```

java.util.stream

```java
// JDK example
int sumOfWeights =
  blocks.stream()
 .filter(b -> b.getColor() == RED)
 .mapToInt(b -> b.getWeight())
 .sum();

// Example with personList
personList // List<Person>
 .stream() // Stream<Person>
 .filter(p -> p.getAge() >= 18) // Stream<Person>
 .map( p -> p.getName() ) // Stream<String>
 .forEach(System.out::println) // void
;
```

Further reading

- http://www.techempower.com/blog/2013/03/26/everything-about-java-8/
- http://javadocs.techempower.com/jdk18/api/overview-summary.html
- src.zip in your JDK 8 installation

Questions?

Thanks

Twitter	@robertbachmann
Email	rb@ — .at