Modern Java - A Guide to Java 8

wizardforcel

Published with GitBook

Table of Contents

Modern Java - A Guide to Java 8

Author: winterbe

From: java8-tutorial

Modern Java - A Guide to Java 8

> “Java is still not dead—and people are starting to figure that out.”

Welcome to my introduction to Java 8. This tutorial guides you step by step through all new language features. Backed by short and simple code samples you'll learn how to use default interface methods, lambda expressions, method references and repeatable annotations. At the end of the article you'll be familiar with the most recent API changes like streams, functional interfaces, map extensions and the new Date API. **No walls of text, just a bunch of commented code snippets. Enjoy!**

This article was originally posted on my blog. You should follow me on Twitter.

Table of Contents

Default Methods for Interfaces

Java 8 enables us to add non-abstract method implementations to interfaces by utilizing the `default` keyword. This feature is also known as virtual extension methods.

Here is our first example:

```
interface Formula {
    double calculate(int a);

    default double sqrt(int a) {
        return Math.sqrt(a);
    }
}
```

Besides the abstract method `calculate` the interface `Formula` also defines the default method `sqrt`. Concrete classes only have to implement the abstract method `calculate`. The default method `sqrt` can be used out of the box.

```
Formula formula = new Formula() {
    @Override
    public double calculate(int a) {
        return sqrt(a * 100);
    }
};

formula.calculate(100);     // 100.0
formula.sqrt(16);           // 4.0
```

The formula is implemented as an anonymous object. The code is quite verbose: 6 lines of code for such a simple calculation of `sqrt(a * 100)`. As we'll see in the next section, there's a much nicer way of implementing single method objects in Java 8.

Lambda expressions