

「一入 Java 深似海」系列课程 - 第七期

第三节：Java APT

小马哥 @mercyblitz

「一入 Java 深似海」系列课程

- 讲师信息

小马哥，Java 劝退师，Apache 和 Spring Cloud 等知名开源架构成员。

- 微博：@mercyblitz

- GitHub： <https://github.com/mercyblitz>

- 课程详情： <https://segmentfault.com/n/1330000017785588>

- 课件资源： <https://github.com/mercyblitz/segmentfault-lessons/>



小马哥 VIP 交流群

扫一扫二维码，加入群聊。

主要议题

- Java APT 基础
- Language Model API
- Annotation Processing Tool API

Java APT 基础

Java APT 基础

- 什么是 Java APT

The command-line utility apt, annotation processing tool, finds and executes annotation processors based on the annotations present in **the set of specified source** files being examined. The annotation processors use a set of reflective APIs and supporting infrastructure to perform their processing of program annotations (JSR 175). The apt reflective APIs provide a **build-time, source-based, read-only view of program structure**. These reflective APIs are designed to cleanly model the Java™ programming language's type system after the addition of generics (JSR 14). First, apt runs annotation processors that can produce new source code and other files. Next, apt can cause compilation of both original and generated source files, thus easing the development cycle.

Java APT 基础

- 为什么要用 Java APT

Many of the intended use cases for annotations involve having annotations in a base file hold information that is used to **generate new derived files** (source files, class files, deployment descriptors, etc.) that are logically consistent with the base file and its annotations. In other words, instead of manually maintaining consistency among the entire set of files, only the base file would need to be maintained since the derived files are generated. The apt tool is designed for creating the derived files.

Java APT 基础

- 为什么要用 Java APT

Compared to using a doclet to generate the derived files based on annotations, apt

- has a cleaner model of the declarations and current type structure of programs
- uses a more contemporary API design, such as returning generic collections instead of arrays and providing visitors to operate on declarations and types
- supports recursive processing of newly generated files and can automatically cause compilation of original and generated source files

Java APT 基础

- APT 外部依赖
 - Java 编译环境: javac 工具或者 Java Compiler API
 - Java Language Model API
 - Java Annotation Processing Tool API

Language Model API

Annotation Processing Tool API

下期预告

Java 装配

谢谢观看