

Java 10th Anniversary



撰文 / 莫映

The Java Saga · Java 5.0

二〇〇四年 9 月 30 日，代号为 **Tiger**，研发历时近 3 年的 **J2SE 5.0** 发布正式版本，这是 **Java** 平台历来发布版本中改动面波及最大的一次。

应该说 **Tiger** 的发布，与 **JDK 1.4** 的成功不无关系。**2002** 年发布的 **1.4** 在原有 **1.3.x** 的基础上，对类库 **API** 的功能做了相当的扩展与完善，性能方面较之上一版本也有大幅提升，加上后续 **1.4.1** 和 **1.4.2** 两个版本的进一步巩固，**1.4.x** 已是一个相当成熟的 **Java** 平台。毋庸置疑，这为 **Tiger** 的研发提供了坚实的基础，也为迟来的发布赢得了时间。

Tiger 还受益于 **JCP** 的成功运作，它代表了 **Java** 社区的民意，包括泛型 (**JSR**)、元数据 (**JSR 175**)、新的语言特性 (**JSR 201**)、多线程库 (**JSR 166**) 在内的众多出现于 **Tiger** 中的新功能，都出自 **JCP**，源自广大 **Java** 开发者的积极反馈。



纵观 **Tiger**，“**Ease of development**”是其核心主题，这一点着重体现于语言特性上的改进，比如：对泛型和注解 (**annotation**) 的引入，还有“**enhanced for loop**”，自动封箱 (**auto-boxing**)，枚举类型，静态引入 (**static import**)，类似“**printf**”语法的格式化输出，可变参数等。如此多的新语言特性的引入，是历来 **Java** 版本发布中所罕见的。这在很大程度上，简化了开发人员日常的编程任务，以往一些琐碎的手工劳动都代之以轻松自然，而又安全可靠的自动化实现。其中的注解功能，以及随之而来的声明式编程，还对构筑与 **J2SE 5.0** 之上的 **J2EE 5.0** 产生了巨大影响。与此相关的应用包括 **JSR 181: Web Service** 元数据，以及 **JSR 220: EJB 3.0**，等等。有了基于元数据的声明式编程，**J2EE** 应用的开发将会比以往更为容易。

除了对语言做了较大改进以外，**Tiger** 在类库方面也有精进，大的改动包括多线程、**JDBC** 数据访问等。

尽管 **Tiger** 在语言特性上做了很大的动作，但作为 **Java** 技术的基础支撑，这些改动都是深思熟虑的结果。比如：像枚举类型，其实早在 **James Gosling** 研制 **Oak** 语言 (**Java** 的前身) 之时就已排入“**To do**”列表，只是当时看来这种特性的引入还欠成熟。又如泛型，核心平台小

组的人也早有耳闻，但是要将这一大特性平滑引入，且与既有语法保持内在一致性，这需要反复论证才行。

Tiger 发布至今也有大半年了，那么 **Sun** 又是如何规划 **J2SE** 的未来蓝图的呢？据悉，**J2SE** 的下两个版本分别是代号为 **Mustang** 的 **J2SE 6.0** 和代号为 **Dolphin** 的 **J2SE 7.0**，预计 **Mustang** 将于明年发布。在吸取了 **Tiger** 研发周期过长的教训之后，**Sun** 副总裁 **Graham Hamilton** 表示，**Mustang** 的发布周期将不会那么长。并且，**Sun** 还将“**Becoming more open**”作为 **Mustang** 的主题之一。未来 **JCP** 对 **Java** 技术的影响将会愈加深入，而整个研发过程也将愈加透明。**Mustang** 在正式发布前的内部版本也会陆续见诸于众，如此，广大 **Java** 开发者便可以更加及时的了解到 **Java** 发展的最新情况。在语言层面上的扩展依然会比较谨慎，比如像 **AOP** 这样的当下热门技术，依然不太可能会见诸其中。据 **Hamilton** 所言，一个有可能被引入的语法特性被称作“**friends**” **import** 机制，它将使由多个包组成的大型项目变得易于管理。

谨以本系列文章纪念 Java 诞生十周年