

JDK10 新特性

- 局部变量类型推断
- 将JDK多存储库合并为单存储库
- 垃圾回收接口
- 并行Full GC 的G1
- 应用数据共享
- 线程局部管控
- 移除Native-Header Generation Tool (javah)
- Unicode 标签扩展
- 备用内存设备上分配堆内存
- 基于实验JAVA 的JIT 编译器
- Root 证书

一、局部变量类型推断

JDK10 可以使用`var`作为局部变量类型推断标识符，此符号仅适用于局部变量，增强for循环的索引，以及传统for循环的本地变量；它不能使用于方法形式参数，构造函数形式参数，方法返回类型，字段，catch形式参数或任何其他类型的变量声明。

标识符`var`不是关键字；相反，它是一个保留的类型名称。这意味着`var`用作变量，方法名或则包名称的代码不会受到影响；但`var`不能作为类或则接口的名字（但这样命名是比较罕见的，因为他违反了通常的命名约定，类和接口首字母应该大写）。

```
1 var str = "ABC"; //根据推断为 字符串类型
2 var l = 10L; //根据10L 推断long 类型
3 var flag = true; //根据 true推断 boolean 类型
4 var flag1 = 1; //这里会推断boolean类型。0表示false 非0表示true
5 var list = new ArrayList<String>(); // 推断 ArrayList<String>
6 var stream = list.stream(); // 推断 Stream<String>
7
```

二、将JDK多存储库合并为单存储库

为了简化和简化开发，将JDK多存储库合并到一个存储库中。多年来，JDK的完整代码已经被分解成多个存储库。在JDK9 中有八个仓库：root、corba、hotspot、jaxp、jaxws、jdk、langtools 和nashorn。在JDK10中被合并为一个存储库。

虽然这种多存储库模型具有一些有点，但它也有许多缺点，并且在支持各种可取的源代码管理操作方面做得很差。特别是，不可能在相互依赖的变更存储库之间执行原子提交。例如，如果一个bug 修复或RFE的代码现在同时跨越了jdk和hotspot 存储库，那么对于两个存储库来说，在托管这两个不同的存储库中，对两个存储库的更改是不可能实现的。跨多个存储库的变更是常见

三、垃圾回收接口

这不是让开发者用来控制垃圾回收的接口；而是一个在 JVM 源代码中的允许另外的垃圾回收器快速方便的集成的接口。

垃圾回收接口为HotSpot的GC代码提供更好的模块化；在不影响当前代码的基础情况下，将GC添加到HotSpot变的更简单；更容易从JDK构建中排除GC。实际添加或删除GC不是目标，这项工作将使HotSpot中GC算法的构建时间隔离取得进展，但它不是完全构建时间隔离的目标。

四、并行Full GC的G1

JDK10 通过并行Full GC，改善G1的延迟。G1垃圾收集器在JDK 9中是默认的。以前的默认值并行收集器中有一个并行的Full GC。为了尽量减少对使用GC用户的影响，G1的Full GC也应该并行。G1垃圾收集器的设计目的是避免Full收集，但是当集合不能足够快地回收内存时，就会出现完全GC。目前对G1的Full GC的实现使用了单线程标记-清除-压缩算法。JDK10 使用并行化标记-清除-压缩算法，并使用Young和Mixed收集器相同的线程数量。线程的数量可以由-XX:ParallelGCThreads选项来控制，但是这也会影响用Young和Mixed收集器的线程数量。

五、应用数据共享

为了提高启动和内存占用，扩展现有的类数据共享（CDS）特性，允许将应用程序类放置在共享档案中。

- 通过在不同的Java进程间共享公共类元数据来减少占用空间。
- 提升启动时间。
- CDS允许将来自JDK的运行时映像文件（\$JAVA_HOME/lib/modules）的归档类和应用程序类路径加载到内置平台和系统类加载器中。
- CDS允许将归档类加载到自定义类加载器中。

六、线程局部管控

在不执行全局VM安全点的情况下对线程执行回调的方法。让它停止单个线程而不是全部线程。

七、移除Native-Header Generation Tool (javah)

JDK10 从JDK中移除了javah 工具。该工具已被JDK8（JDK-7150368）中添加javac高级功能所取代。此功能提供了在编译java源代码时编写本机头文件的功能，从而无需使用单独的工具

八、Unicode标签扩展

JDK10 改善 java.util.Locale 类和相关的 API 以实现额外 BCP 47 语言标签的 Unicode 扩展。尤其以下扩展支持：

- cu：货币类型
- fw：一周的第一天
- rg：区域覆盖
- tz：时区

为支持以上扩展，JDK10对以下API进行更改：

- java.text.DateFormat::get*Instance：将根据扩展ca、rg或tz返回实例。

- `java.text.DateFormatSymbols::getInstance`: 将根据扩展rg返回实例。
- `java.text.DecimalFormatSymbols::getInstance`: 将根据扩展rg返回实例。
- `java.text.NumberFormat::getInstance`: 将根据nu或rg返回实例。
- `java.time.format.DateTimeFormatter::localizedBy`: 将返回DateTimeFormatter 根据ca, rg或rz的实例。
- `java.time.format.DateTimeFormatterBuilder::getLocalizedDateTimePattern`: 将根据rg返回String。
- `java.time.format.DecimalStyle::of`: 将返回DecimalStyle根据nu或rg的实例。
- `java.time.temporal.WeekFields::of`: 将返回WeekFields根据fw或rg的实例。
- `java.util.Calendar::{getFirstDayOfWeek,getMinimalDaysInWeek}`: 将根据fw或rg返回值。
- `java.util.Currency::getInstance`: 将返回Currency根据cu或rg返回实例。
- `java.util.Locale::getDisplayName`: 将返回一个包含这些U扩展名的显示名称的字符串。
- `java.util.spi.LocaleNameProvider`: 将为这些U扩展的键和类型提供新的SPI。

九、备用内存设备上分配堆内存

启用HotSpot VM以在用户指定的备用内存设备上分配Java对象堆。随着廉价的NV-DIMM内存的可用性，未来的系统可能配备了异构的内存架构。这种技术的一个例子是英特尔的3D XPoint。这样的体系结构，除了DRAM之外，还会有一种或多种类型的非DRAM内存，具有不同的特征。具有与DRAM具有相同语义的可选内存设备，包括原子操作的语义，因此可以在不改变现有应用程序代码的情况下使用DRAM代替DRAM。所有其他的内存结构，如代码堆、metaspace、线程堆栈等等，都将继续驻留在DRAM中。

参考以下使用案例：

- 在多JVM部署中，某些JVM（如守护进程，服务等）的优先级低于其他JVM。与DRAM相比，NV-DIMM可能具有更高的访问延迟。低优先级进程可以为堆使用NV-DIMM内存，允许高优先级进程使用更多DRAM。
- 诸如大数据和内存数据库等应用程序对内存的需求不断增加。这种应用可以将NV-DIMM用于堆，因为与DRAM相比，NV-DIMM可能具有更大的容量，成本更低。

十、基于实验JAVA的JIT编译器

启用基于Java的JIT编译器Graal，将其用作Linux / x64平台上的实验性JIT编译器。Graal是一个基于Java的JIT编译器，它是JDK 9中引入的Ahead-of-Time (AOT) 编译器的基础。使它成为实验性JIT编译器是Project Metropolis的一项举措，它是下一步是研究JDK的基于Java的JIT的可行性。使Graal可用作实验JIT编译器，从Linux / x64平台开始。Graal将使用JDK 9中引入的JVM编译器接口 (JVMCI) 。Graal已经在JDK中，因此将它作为实验JIT将主要用于测试和调试工作。要启用Graal作为JIT编译器，请在java命令行上使用以下选项：

```
-XX:+UnlockExperimentalVMOptions -XX:+UseJVMCICompiler
```

十一、ROOT证书

在JDK中提供一组默认的root 认证权威（CA）证书。在Oracle的Java SE根CA程序中开源root证书，以使OpenJDK构建对开发人员更有吸引力，并减少这些构建和Oracle JDK构建之间的差异。cacerts密钥存储库是JDK的一部分，它的目的是包含一组root证书，这些root证书可以用来在各种安全协议中使用的证书链中建立信任。然而，JDK源代码中的cacerts密钥库目前是空的。因此，诸如TLS之类的关键安全组件在OpenJDK构建中不会默认工作。为了解决这个问题，用户必须配置和填充cacerts密钥库，并使用一组root证书来记录，例如， [JDK 9 release notes](#)。