Java 10 New Features Overview

Guide   Java   New Java Features   About 1457 words   About 5 minutes

Java 10 was released on March 20, 2018. Its most well-known feature is probably `var` the introduction of the keyword (local variable type inference). Other new features include garbage collector improvements, GC improvements, performance improvements, thread management, and so on.

Overview (selected part) :

- JEP 286: Local Variable Type Inference
- JEP 304: Garbage Collector Interface
- JEP 307: G1 Parallel Full GC
- JEP 310: Application Class Data Sharing (Extended CDS Capabilities)
- JEP 317: Experimental Java-based JIT Compiler

Local variable type inference (var)

Since so many Java developers wanted local variable inference to be introduced in Java, it came in Java 10, which is what everyone wanted!

Java 10 provides `var` keyword to declare local variables.

```java
var id = 0;
var codefx = new URL("https://mp.weixin.qq.com/");
var list = new ArrayList<>();
var list = List.of(1, 2, 3);
var map = new HashMap<String, String>();
var p = Paths.of("src/test/java/Java9FeaturesTest.java");
var numbers = List.of("a", "b", "c");
for (var n : list)
    System.out.print(n+ " ");
```

The var keyword can only be used with local variables in constructors and for loops.

```
var count=null; //❌编译不通过，不能声明为 null
var r = () -> Math.random();//❌编译不通过,不能声明为 Lambda表达式
var array = {1,2,3};//❌编译不通过,不能声明数组
```

var doesn't change the fact that Java is a statically typed language; the compiler is responsible for inferring types.

In addition, Scala and Kotlin already have `val` keywords (`final var` combined keywords).

Related reading: "Java 10 new features: local variable type inference" .

Garbage Collector Interface

In earlier JDK architectures, the components that made up the garbage collector (GC) implementation were scattered across the codebase. Java 10 separates the source code for different garbage collectors by introducing a set of pure garbage collector interfaces.

G1 Parallel Full GC

G1 has been the default garbage collector since Java 9. It is designed as a low-latency garbage collector to avoid Full GC. However, Java 9's G1 Full GC still uses a single thread to perform the mark-and-sweep algorithm, which may cause Full GC to be triggered when the garbage collection period cannot reclaim memory.

To minimize the impact of application pauses caused by Full GC, starting from Java 10, G1's Full GC is changed to a parallel mark-sweep algorithm, and the same number of parallel worker threads as young generation collection and mixed collection are used, thereby reducing the occurrence of Full GC and bringing better performance improvements and greater throughput.

Collection Enhancement

`List` , `Set` , `Map` provides a static method `copyOf()` that returns an immutable copy of the input parameter collection.

```java
static <E> List<E> copyOf(Collection<? extends E> coll) {
    return ImmutableCollections.listCopy(coll);
}
```

The collection created using `copyOf()` is an immutable collection. You cannot add, delete, replace, or sort it. Otherwise, `java.lang.UnsupportedOperationException` an exception will be reported. IDEA will also give a corresponding prompt.

```
List<Integer>
integerList.a
```

Immutable object is modified
Wrap 'integerList' with 'ArrayList'
More actions...

In addition, `java.util.stream.Collectors` new static methods are added to collect elements in a stream into an immutable collection.

```java
var list = new ArrayList<>();
list.stream().collect(Collectors.toUnmodifiableList());
list.stream().collect(Collectors.toUnmodifiableSet());
```

Optional Enhancement

`Optional` A new parameterless `orElseThrow()` method has been added as a simplified version of the parameterized `orElseThrow(Supplier<? extends X> exceptionSupplier)` method. By default, a NoSuchElementException is thrown when there is no value.

```java
Optional<String> optional = Optional.empty();
String result = optional.orElseThrow();
```

Application Class Data Sharing (Extended CDS Functionality)

Class Data Sharing (CDS) was introduced in Java 5. This mechanism allows a group of classes to be pre-processed into a shared archive file, which can be memory-mapped at runtime to reduce Java program startup time. When multiple Java Virtual Machines (JVMs) share the same archive file, it also reduces dynamic memory usage and reduces resource usage when multiple JVMs run on the same physical or virtual machine. CDS was a commercial feature of the Oracle JDK at the time.

Java 10 further expands upon the existing CDS functionality to allow application clas be placed in a shared archive. Building upon the existing bootstrap class, the CDS feat adds support for application classes in CDS (Application Class-Data Sharing, AppCDS), significantly expanding the applicability of CDS. The principle is to record the class loading process at startup and write it to a text file. Upon subsequent startup, this startup text is directly read and loaded. Assuming the application environment remains unchanged, this should result in faster startup times.

Experimental Java-based JIT compiler

Graal is a JIT compiler written in the Java language and is the basis for the experimental Ahead-of-Time (AOT) compiler introduced in JDK 9.

Oracle's HotSpot VM ships with two JIT compilers implemented in C++: C1 and C2. In Java 10 (Linux/x64, macOS/x64), HotSpot still uses C2 by default, but `-XX:+UnlockExperimentalVMOptions -XX:+UseJVMCICompiler` you can replace C2 with Graal by adding a parameter to the java command.

Related reading: A simple introduction to Java 10's experimental JIT compiler Graal - Zheng Yudi

other

- **Thread-local control** : Java 10 introduces the concept of JVM safepoints for thread control, which allows thread callbacks to be executed by the thread itself or by a JVM thread without running a global JVM safepoint, while keeping the thread in a blocked state. This makes it possible to stop a single thread, rather than just enabling or stopping all threads.
- **Heap allocation on alternative storage devices** : Java 10 will enable the JVM to allocate heap memory on alternative memory devices using heaps suitable for different types of storage mechanisms.
- ...

refer to

- Java 10 Features and Enhancements: https://howtodoinjava.com/java10/java10-features/
- Guide to Java10: https://www.baeldung.com/java-10-overview
- 4 Class Data Sharing: https://docs.oracle.com/javase/10/vm/class-data-sharing.htm#JSJVM-GUID-7EAA3411-8CF0-4D19-BD05-DF5E1780AA91

JavaGuide官方公众号
（微信搜索JavaGuide）

1、公众号后台回复“PDF”获取原创PDF面试手册

2、公众号后台回复“学习路线”获取Java学习路线最新版

3、公众号后台回复“开源”获取优质Java开源项目合集

4、公众号后台回复“八股文”获取Java面试真题+面经

Recently Updated2025/7/3 15:04
Contributors: guide , Guide , Mr.Hope , Clear , xoc