

JDK9

- Java 9 ☐☐☐☐☐

Java 9 新特性

Java 9 发布于 2017 年 9 月 21 日，继 Java8 之后 3 年发布一次，Java 9 带来了许多新特性，Java 模块化是其中之一。

Java 模块化

Java 模块化是 Project Jigsaw 的一部分，Java 9 引入了模块化，DK 版本 94 开始，Java 引入了 jlink 工具。

模块化是 Java 9 的一个重要特性，它允许开发者将应用程序拆分成多个模块，每个模块可以独立运行，也可以与其他模块一起运行。Apache Maven 和 Gradle 都支持模块化。

Java 9 引入了 artifact 模块，module-info.class 文件，JAR 文件，Java 9 引入了 JMOD 文件。

- 模块可以 exports 公共或受保护的类，public 或 protected 类，public 或 protected 类。
- 模块可以 requires 公共或受保护的类，requires transitive 或 requires 公共或受保护的类。
- 模块可以 ServiceLocator 类，模块 provides with 公共或受保护的类。

例如，com.mycompany.sample 模块可以 exports 公共或受保护的类，com.mycompany.sample 模块可以 requires 公共或受保护的类，com.mycompany.common.DemoService 模块可以 provides with 公共或受保护的类。

1. 模块化

```
module com.mycompany.sample {  
    exports com.mycompany.sample;  
    requires com.mycompany.common;  
    provides com.mycompany.common.DemoService with  
        com.mycompany.sample.DemoServiceImpl;  
}
```

```
}
```

```
_ ????
```

```
##### Java ##### CLASSPATH ##### CL/
```

```
##### JVM ##### JV
```

Jshell

jshell Java 9 ##### jshell Java ##### NodeJS Python #####-##### Read-Evaluation-Print Loop #####

\$2 ##### jshell ##### Java ##### jshell #####

```
#####2##### add
```

2. jshell

```
jshell> int add(int x, int y) {  
    ...> return x + y;  
    ...> }  
| created method add(int,int)
```

```
_ ????
```

```
##### jshell#####
```

3. jshell

```
jshell> add(1, 2)  
$19 ==> 3
```

```
_ ????
```

Stream Optional

Java 9 List.of() Set.of() Map.of() Map.ofEntries()

4 .

```
List.of();
List.of("Hello", "World");
List.of(1, 2, 3);
Set.of();
Set.of("Hello", "World");
Set.of(1, 2, 3);
Map.of();
Map.of("Hello", 1, "World", 2);
```

???

Stream ofNullable dropWhile takeWhile iterate 5 1 4

5 . Stream dropWhile

```
@Test
public void testDropWhile() throws Exception {    final long count = Stream.of(1, 2, 3, 4,
5)
        .dropWhile(i -> i % 2 != 0)
        .count();
    assertEquals(4, count);
}
```

???

Collectors filtering flatMapping String flatMapping String Integer Integer

6 . Collectors flatMapping

```
@Test
public void testFlatMapping() throws Exception {
    final Set<Integer> result = Stream.of("a",
        "ab", "abc")
        .collect(Collectors.flatMapping(v -> v.chars().boxed(),
            Collectors.toSet()));
    assertEquals(3, result.size());
}
```

????

Optional ifPresentOrElse or streamOptional 3 2 flatMap 2

7 . Optional stream

```
@Test
public void testStream() throws Exception {
    final long count = Stream.of(
        Optional.of(1),
        Optional.empty(),
        Optional.of(2)
    ).flatMap(Optional::stream)
        .count();
    assertEquals(2, count);
}
```

????

API

Java 9 `ProcessHandle` `ProcessBuilder` `Process.toHandle()`
`ProcessHandle.Info` `ProcessHandle` `onExit()` `CompletableFuture`

8. `API`

```
final ProcessBuilder processBuilder = new ProcessBuilder("top")
    .inheritIO();
final ProcessHandle processHandle = processBuilder.start().toHandle();
processHandle.onExit().whenCompleteAsync((handle, throwable) -> {
    if (throwable == null) {
        System.out.println(handle.pid());
    } else {
        throwable.printStackTrace();
    }
});
```

`API`

Java 9 `JDK` `System.LoggerFinder` `JDK` `VM` `LoggerFinder`
`java.util.logging` `LoggerFinder` `getLogger()` `System.Logger` `System.Lo`
`JDK` `SUB4` `API`

9. `API`

```

public class Main {
    private static final System.Logger LOGGER = System.getLogger("Main");    public static void
main(final String[] args) {
    LOGGER.log(Level.INFO, "Run!");
}
}

```

_ ????

Reactive Streams

Java RxJava Reactor non-blocking backpress

Flow Flow.Publisher Flow.Subscriber Flow.Subscription Flow.Processor 4 java 9 Subr
Flow.Publisher RxJava 2 Reactor Flow

Java java.lang.invoke.VarHandle
VarHandle VarHandle byte[] Byte

HandleTarget count

10.

```

public class HandleTarget {
    public int count = 1;
}

public class VarHandleTest {
    private HandleTarget handleTarget = new HandleTarget();
    private VarHandle varHandle;
    @Before
    public void setUp() throws Exception {

```

```

        this.handleTarget = new HandleTarget();
        this.varHandle = MethodHandles
            .lookup()
            .findVarHandle(HandleTarget.class, "count", int.class);
    }

    @Test
    public void testGet() throws Exception {
        assertEquals(1,
            this.varHandle.get(this.handleTarget));
        assertEquals(1,
            this.varHandle.getVolatile(this.handleTarget));
        assertEquals(1,
            this.varHandle.getOpaque(this.handleTarget));
        assertEquals(1,
            this.varHandle.getAcquire(this.handleTarget));
    }
}

```

_

Method Handle

java.lang.invoke.MethodHandles

- arrayConstructor
- arrayLength
- varHandleInvoker varHandleExactInvoker VarHandle
- zero
- empty MethodType
- loop countedLoop iteratedLoop whileLoop doWhileLoop for while do-while
- tryFinally try-finally

iteratedLoop S tring

11.


```

public class IteratedLoopTest {
    static int body(final int sum, final String value) {
        return sum + value.length();
    }
    @Test
    public void testIteratedLoop() throws Throwable {
        final MethodHandle iterator =
MethodHandles.constant(
            Iterator.class,
            List.of("a", "bc", "def").iterator());
        final MethodHandle init =
MethodHandles.zero(int.class);
        final MethodHandle body = MethodHandles
            .lookup()
            .findStatic(
                IteratedLoopTest.class,
                "body",
                MethodType.methodType(
                    int.class,
                    int.class,
                    String.class));
        final MethodHandle iteratedLoop =
MethodHandles
            .iteratedLoop(iterator, init, body);
        assertEquals(6,
iteratedLoop.invoke());
    }
}

```

_ []

[] CompletableFuture [] completeAsync [] CompletableFuture [] orTimeout []
[] TimeoutException [] CompletableFuture [] completeOnTimeout [] o rTimeout [] Com
Thread.onSpinWai t []

Nashorn

Nashorn 与 Java 8 兼容 JavaScript 与 Java 9 兼容 Nashorn 兼容 ECMAScript 6 兼容
Syntax Tree 与 AST 兼容 ECMAScript 兼容

I/O 与

与 java.io.InputStream 与 InputStream 与

- readAllBytes() 与 InputStream 与
- readNBytes() 与 InputStream 与
- transferTo() 与 InputStream 与 OutputStream 与

与 12 与

与 12. InputStream 与

```
public class TestInputStream {
    private InputStream inputStream;
    private static final String CONTENT = "Hello World";
    @Before
    public void setUp() throws Exception {
        this.inputStream =
TestInputStream.class.getResourceAsStream("/input.txt");
    }
    @Test
    public void testReadAllBytes() throws Exception {        final String content = new
String(this.inputStream.readAllBytes());
        assertEquals(CONTENT, content);
    }
    @Test
    public void testReadNBytes() throws Exception {        final byte[] data = new
byte[5];
        this.inputStream.readNBytes(data, 0, 5);        assertEquals("Hello", new
String(data));
    }
}
```

```

@Test
public void testTransferTo() throws Exception {
    final ByteArrayOutputStream
    outputStream = new ByteArrayOutputStream();
    this.inputStream.transferTo(outputStream);
    assertEquals(CONTENT,
    outputStream.toString());
}
}

```

????

ObjectInputFilter [] ObjectInputStream [] ObjectInputStream [] setObject

java 9 4 SHA-3 SHA3-224 SHA3-256 SHA3-384 SHA3-512 java.security.SecureRa
SHA-3

13. SHA-3

```

import org.apache.commons.codec.binary.Hex;
public class SHA3 {
    public static void main(final String[] args) throws
    NoSuchAlgorithmException {
        final MessageDigest instance = MessageDigest.getInstance("SHA3-
        224");
        final byte[] digest = instance.digest("").getBytes();
        System.out.println(Hex.encodeHexString(digest));
    }
}

```

????

java.awt.Desktop addAppEventListener

JVM

Java 9 虚拟机 (JVM) 日志 (Xlog) JVM 日志 (Xlog) Java 9

Java 9 新特性 try-with-resources 有效地 final java.lang.StackWalker 3.0 ResourceBundle ISO-8859-1 UTF-8 native2ascii @Deprecated

14 buildMessage SayHi sayHi

14. 接口

```
public interface SayHi {  
    private String buildMessage() {  
        return "Hello";  
    }  
    void sayHi(final String message);  
    default void sayHi() {  
        sayHi(buildMessage());  
    }  
}
```

???

Java 9 新特性 Java 9 新特性 Java 9 新特性 Java 9 新特性

<https://developer.ibm.com/zh/articles/the-new-features-of-java-9/>