CYBAGE
Delivering Value. Scientifically.

Java 9 and 10 Features

Author & Presenter -Asfiya Khan
(Senior Technical Trainer)

Agenda

- The Java Platform Module System
- Jshell : The Interactive Java REPL
- Stream API Improvements
- Private interface methods
- Local variable type inference

The Java Platform Module System – Jigsaw Project

- One of the big changes in java 9 is the Module System i.e Jigsaw Project.
- Goals of Jigsaw Project
    1. The Modular JDK
    2. Modular Source Code
    3. Modular Run-Time Images
    4. "Allowing the user to create their modules to develop their applications"
    5. Ease of Testing and Maintainability.

Java 9 REPL

- Oracle Corp has introduced a new tool called "jshell". It stands for Java Shell and also known as REPL (Read Evaluate Print Loop)

- It is used to execute and test any Java Constructs like class, interface, enum, object, statements etc. very easily.

- Open command prompt and check java version to make sure you have java 9 or above, then only you can use jshell.

Java 9 REPL

```
G:\>jshell
| Welcome to JShell -- Version 9-ea
| For an introduction type: /help intro

jshell> int a = 10
a ==> 10

jshell> System.out.println("a value = " + a )
a value = 10
```

Stream API Improvements

- In Java SE 9, Oracle Corp has added four useful new methods to java.util.Stream interface.

- As Stream is an interface, all those new implemented methods are default methods. Two of them are very important: **dropWhile** and **takeWhile** methods.

- For example - **takeWhile()** takes a predicate as an argument and returns a Stream of subset of the given Stream values until that Predicate returns false for the first time.

Stream API Improvements

```
jshell> Stream.of(1,2,3,4,5,6,7,8,9,10).takeWhile(i -> i < 5 )
            .forEach(System.out::println);
1
2
3
4
```

Stream API Improvements

```
jshell> Stream<Integer> stream  =  Stream.of(1,2,3,4,5,6,7,8,9,10)
stream ==> java.util.stream.ReferencePipeline$Head@55d56113

jshell> stream.dropWhile(x -> x < 4).forEach(a -> System.out.println(a))
4
5
6
7
8
9
10
```

Private Interface Methods

- In Java 8, we can provide method implementation in Interfaces using default and static methods. However we cannot create private methods in Interfaces.

- From Java SE 9 on-wards, we can write private and private static methods too in an interface using ‘private’ keyword.

- To avoid redundant code and more re-usability.

Private Interface Methods

```
public interface Card{

  private Long createCardID(){
    // Method implementation goes here.
  }

  private static void displayCardDetails(){
    // Method implementation goes here.
  }

}
```

Local Variable Type Inference LVTI in Java 10

- **What is type inference?**

  Type Inference refers to the automatic detection of the datatype of a variable, done generally at the compiler time

- Local variable type inference is a feature in Java 10 that allows the developer to skip the type declaration associated with local variables

- The type is inferred by the JDK

- It will, then, be the job of the compiler to figure out the datatype of the variable.

Local Variable Type Inference LVTI in Java 10

```
// Java code for Normal local
// variable declaration

import java.util.ArrayList;
import java.util.List;

class A {
public static void main(String ap[]) {

List<Map> data = new ArrayList<>();

}

}
```

```
// Java code for local variable
// declaration using LVTI

import java.util.ArrayList;
import java.util.List;

class A {
public static void main(String ap[]) {

                var data = new ArrayList<>();
    }
}
```

CYBAGE
Delivering Value. Scientifically.

Thank You!

Any Questions ?