

Applying Closures & Java Lambda Expressions in Case Study ex6

Douglas C. Schmidt

d.schmidt@vanderbilt.edu

www.dre.vanderbilt.edu/~schmidt

Professor of Computer Science

Institute for Software
Integrated Systems

Vanderbilt University
Nashville, Tennessee, USA



Learning Objectives in this Lesson

- Understand how lambda expressions provide a foundational functional programming feature in Modern Java
- Recognize how to apply lambda expressions
- Know the benefits of applying Java lambda expressions
- Be aware of how to apply Java lambda expressions to implement closures

```
var cp1 = new CheckPrimality(...)  
        .start();  
  
var cp2 = new CheckPrimality(...)  
        .start();  
  
var pr1 = cp1.getResult();  
var pr2 = cp2.getResult();  
  
if (pr1.isPrime() && pr2.isPrime()) {  
    var keyPair = RSAKeyUtils  
        .generateKeyPair  
        (pr1.primeCandidate(),  
         pr2.primeCandidate());
```

Apply closures, Java Thread objects, & lambda expressions to create public & private RSA keys

Applying Java Lambda Expressions to Implement Closures in Case Study ex6

Applying Java Lambda Expressions in Case Study ex6

The screenshot shows a Java code editor within an IDE. The project structure on the left shows a file named 'ex6.java' under 'src/main/java/ex6'. The code itself uses Java 8's Stream API and lambda expressions to perform concurrent primality checks on large numbers.

```
// Create two closures that concurrently check the primality of
// large numbers in separate threads.
var checkPrimality1: CheckPrimality =
    new CheckPrimality( n: generateProbablePrime( bitLength: 1024) )
        .start();
var checkPrimality2: CheckPrimality =
    new CheckPrimality( n: generateProbablePrime( bitLength: 1024) )
        .start();

// Get the results of both primality checks, blocking until
// the results are available. The results are returned in the
// order the threads in the closures were started.
var primeResult1: PrimeResult = checkPrimality1.getResult();
var primeResult2: PrimeResult = checkPrimality2.getResult();

// Print the results.
System.out.println("The value "
    + primeResult1.primeCandidate()
    + "\nis "
    + (primeResult1.isPrime() ? "" : "not ")
```

At the bottom of the screen, there are several tabs and status indicators:

- Git, Run, TODO, Problems, Terminal, Services, Profiler, Logcat, Dependencies, CodeWhisperer Reference Log, App Inspection, Diffblue Cover
- AWS Toolkit
- * daemon started successfully (24 minutes ago)
- AWS: No credentials selected, CodeWhisperer 19:15, CRLF, UTF-8, 4 spaces
- main

See github.com/douglascraigschmidt/ModernJava/tree/main/FP/ex6

End of Applying Closures & Java Lambda Expressions in Case Study ex6