Implementing Closures with Java Lambda Expressions

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
- Know the benefits of applying Java lambda expressions
- Recognize how to implement lambda expressions

```
class ClosureExample {
  private int mRes;

  Thread makeThreadClosure
    (String s, int n) {
    return new Thread(() ->
      System.out.println
        (s + (mRes += n)));
  }
}
```

Know how to implement a (simple) variant of closures using Java lambda expressions

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

Implementing Closures with Java Lambda Expressions

Implementing Closures with Java Lambda Expressions

- Lambda expressions can implement (simple) variants of "closures"
  - A closure is a persistent scope that holds on to local variables even after the code execution has moved out of that block

See en.wikipedia.org/wiki/Closure_(computer_programming)

Implementing Closures with Java Lambda Expressions

- Lambda expressions can implement (simple) variants of “closures”

```
class ClosureExample {
  private int mRes;

  Thread makeThreadClosure(String s, int n) {
    return new Thread(()-> System.out.println(s + (mRes += n)));
  }


  ClosureExample() throw InterruptedException {
    Thread t = makeThreadClosure("result = ", 10);
    t.start(); t.join();
  }
}
```

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

Implementing Closures with Java Lambda Expressions

- Lambda expressions can implement (simple) variants of "closures"

```
class ClosureExample {
  private int mRes;

  Thread makeThreadClosure(String s, int n) {
    return new Thread(()-> System.out.println(s + (mRes += n)));
  }

  ClosureExample() throw InterruptedException {
    Thread t = makeThreadClosure("result = ", 10);
    t.start(); t.join();
  }
}
```

A closure in modern Java is an object that stores a method together w/ an environment with at least 1 "bound variable"

A bound variable is name that has a *value*, such as a number or a string

Implementing Closures with Java Lambda Expressions

- Lambda expressions can implement (simple) variants of "closures"

```
class ClosureExample {
  private int mRes;

  Thread makeThreadClosure(String s, int n) {
    return new Thread(()-> System.out.println(s + (mRes += n)));
  }


  ClosureExample() throw InterruptedException {
    Thread t = makeThreadClosure("result = ", 10);
    t.start(); t.join();
  }
}
```

This private field & the method params are "bound variables"

Implementing Closures with Java Lambda Expressions

- Lambda expressions can implement (simple) variants of "closures"

```
class ClosureExample {
  private int mRes;

  Thread makeThreadClosure(String s, int n) {
    return new Thread(()-> System.out.println(s + (mRes += n)));
  }
```

This lambda implements a closure that captures a private field & method params

```
  ClosureExample() throw InterruptedException {
    Thread t = makeThreadClosure("result = ", 10);
    t.start(); t.join();
  }
}
```

See bruceeckel.github.io/2015/10/17/are-java-8-lambdas-closures

Implementing Closures with Java Lambda Expressions

- Lambda expressions can implement (simple) variants of "closures"

```
class ClosureExample {
  private int mRes;

  Thread makeThreadClosure(String s, int n) {
    return new Thread(()-> System.out.println(s + (mRes += n)));
  }

  ClosureExample() throw InterruptedException {
    Thread t = makeThreadClosure("result = ", 10);
    t.start(); t.join();
  }
}
```

Values of private fields can be updated in a lambda, but not parameters or local variables (which are read-only)

See dzone.com/articles/java-8-lambas-limitations-closures

Implementing Closures with Java Lambda Expressions

- Lambda expressions can implement (simple) variants of "closures"

```
class ClosureExample {
  private int mRes;

  Thread makeThreadClosure(String s, int n) {
    return new Thread(()-> System.out.println(s + (mRes += n)));
  }
```

This factory method creates the closure

```
  ClosureExample() throw InterruptedException {
    Thread t = makeThreadClosure("result = ", 10);
    t.start(); t.join();
  }
}
```

See en.wikipedia.org/wiki/Factory_method_pattern

Implementing Closures with Java Lambda Expressions

- Lambda expressions can implement (simple) variants of "closures"

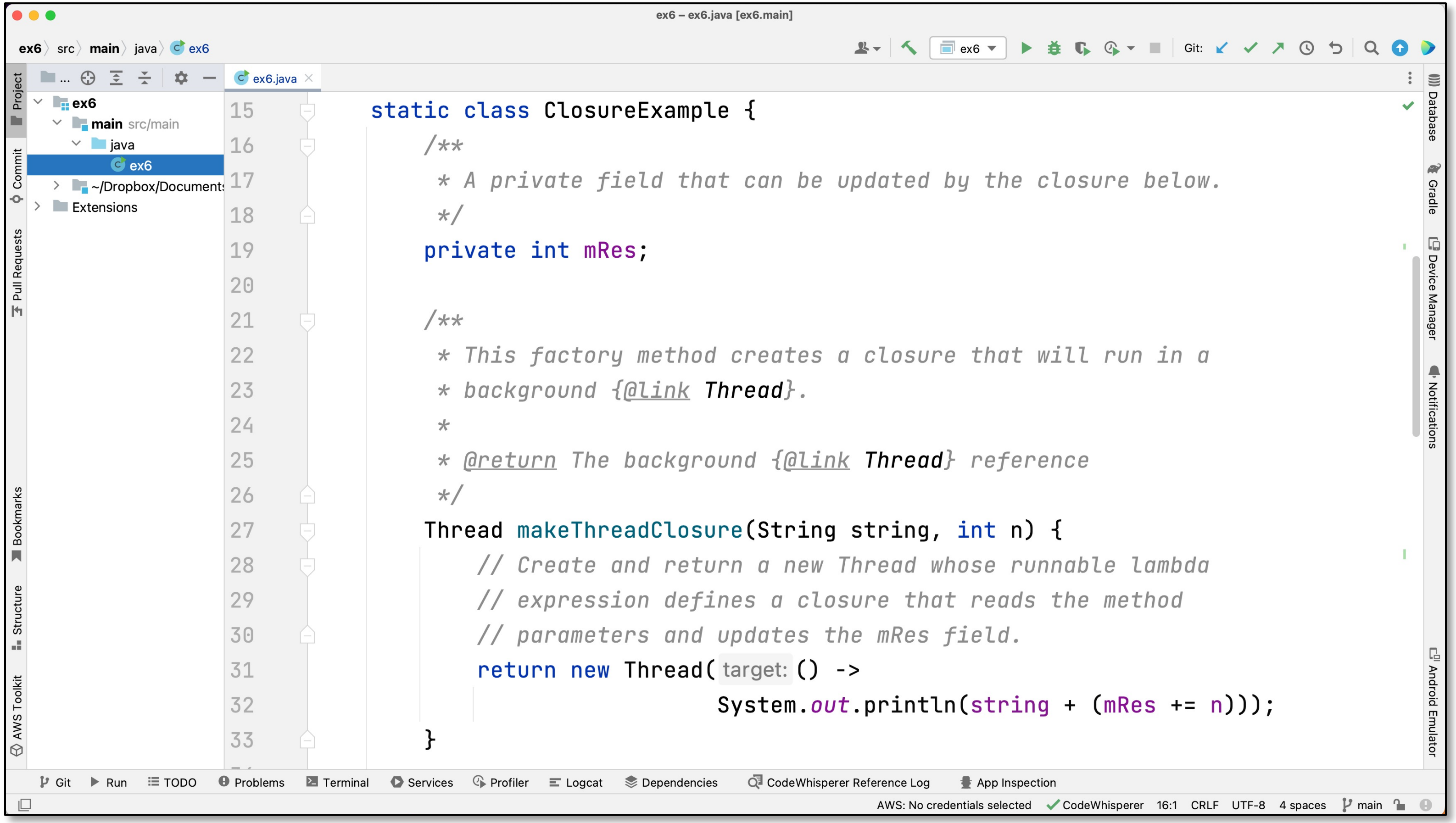
```
class ClosureExample {
  private int mRes;

  Thread makeThreadClosure(String s, int n) {
    return new Thread(()-> System.out.println(s + (mRes += n)));
  }



  ClosureExample() throw InterruptedException {
    Thread t = makeThreadClosure("result = ", 10);
    t.start();
    ...
}
```

This closure then runs in a background thread

See yakovfain.com/2014/07/21/closures-in-java-with-lambdas

Applying Java Lambda Expressions to Implement Closures in Case Study ex6

Applying Java Lambda Expressions in Case Study ex6



```java
    static class ClosureExample {
        /**
         * A private field that can be updated by the closure below.
         */
        private int mRes;

        /**
         * This factory method creates a closure that will run in a
         * background {@link Thread}.
         *
         * @return The background {@link Thread} reference
         */
        Thread makeThreadClosure(String string, int n) {
            // Create and return a new Thread whose runnable lambda
            // expression defines a closure that reads the method
            // parameters and updates the mRes field.
            return new Thread( target: () ->
                                  System.out.println(string + (mRes += n)));
        }
```

ex6 – ex6.java [ex6.main]
ex6 src main java ex6
ex6 main src/main java ex6 ~/Dropbox/Documents Extensions
ex6.java
Git Run TODO Problems Terminal Services Profiler Logcat Dependencies CodeWhisperer Reference Log App Inspection
AWS: No credentials selected CodeWhisperer 16:1 CRLF UTF-8 4 spaces main

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

End of Implementing Closures with Java Lambda Expressions