Applying Java Structured Concurrency:

Case Study ex5

Douglas C. Schmidt <u>d.schmidt@vanderbilt.edu</u> www.dre.vanderbilt.edu/~schmidt



Professor of Computer Science

Institute for Software Integrated Systems

Vanderbilt University Nashville, Tennessee, USA



Learning Objectives in this Part of the Lesson

- Understand Java's structured concurrency model
- Recognize classes used to program Java's structure concurrency model
- Evaluate the design & performance of various Java concurrency models
- Learn how StructuredTaskScope is implemented

```
y models isPrime(number) == 0
cope is ? number
custom return scope.result();
}
```

numbers

try (var scope = new

ShutdownOnNonNullSuccess

.forEach(number -> scope

<Integer>()) {

.fork(() ->

- Know how to implement a custom StructuredTaskScope
 - Case study ex5 shows how to create/apply a custom StructuredTaskScope & compares it to applying ShutdownOnSuccess with exceptions

Applying Reactive Java Concurrency to Case Study ex5

Applying Reactive Java Concurrency to Case Study ex5

ex5 – ShutdownOnNonNullSuccess.java [ex5.main]					
e	x5 $ angle$ src $ angle$ main $ angle$ java $ angle$ utils $ angle$ $igsquare$ ShutdownOnNonNullS	Success		🏨 → 🔨 🗐 ex5 → 🕨 🎍 🕼 🖓 → 🔳 Git: 🖌 🗸 🧷 🕤 🔍 🕤	
ect	■ Project ▼ ④ Ξ 茶 ϕ −	C Shutdo	ownOnNon	NullSuccess.java 🗴 🞯 ex5.java 🗴 🧿 PrimeUtils.java 🗵	: ())
Proj	ex5 ~/Dropbox/Documents/opp/Pearson/LiveLe	7	-/**	<u>۸</u> ۱۸۰	Data
	> 🖿 .gradle	8	*	A {@link StructuredTaskScope} that captures the result of the first	base
ці.	> idea	9	*	subtask to complete successfully (i.e., without returning a {Acade	
Com	> bin	10		Sublast to compare Geode pulling (inc. subtack completes	G
-0-	> build	10		Notify of retorns (<u>notify</u> in the solution completes)	radle
s	> 🖿 gradle	11	*	successfully. Unce captured, it invokes the {[d <u>code</u> shutdown()}	
luest	✓ Im src	12	*	method to interrupt unfinished threads and wakeup the owner.	,
I Req	🗠 🖿 main	13	*		lotific
Pul.	✓ ■ java	14	*	The policy implemented by this class is intended for cases where	catio
141		15	*	the result of any subtask will do ("invoke any") and where the	ns
		16	*	results of other unfinished subtask are no longer needed	
	© Options	17		/	
	C PrimeUtils	17			
	ShutdownOnNonNullSuccess	18	put	CLASS ShutdownUnNonNullSuccess	
	¢ ex5	19		extends StructuredTaskScope <t> {</t>	
	 classpath sition and 	20		/**	
		21		* Stores the first computation to match or null if there are no	
	✓ project <i>№</i> build.gradle	22		* matches.	
	gradlew	23		*/	
ks	🗧 gradlew.bat	20		"/ nniveto volotilo I mPocult:	
cmar	R settings.gradle	24		private votatite / mikesott,	
Bool	External Libraries	25			
	Scratches and Consoles	26		/**	
e		27		* Creates an unnamed structured task scope that creates virtual	
uctu		28		* threads.	
Str.		29		*/	
		7.0		nuhlia ShutdownOnNonNullSuccess() / suman(name: null Thread of/intual() factory()). 1	
	🖞 Git 🗮 TODO 😶 Problems 🛂 Terminal 🜑 Services 🚱 Profiler 📚 Dependencies				
				✓ 18:26 LF UTF-8 4 spaces P master	r 🔒 🖌

See github.com/douglascraigschmidt/LiveLessons/tree/master/Loom/ex5

End of Applying Java Structured Concurrency: Case Study ex5