Table of Contents

Preface		. vii
1.	Introduction	. 1
	Why Did They Need to Change Java Again?	1
	What Is Functional Programming?	2
	Example Domain	3
2.	Lambda Expressions.	5
	Your First Lambda Expression	5
	How to Spot a Lambda in a Haystack	6
	Using Values	8
	Functional Interfaces	9
	Type Inference	11
	Key Points	13
	Exercises	14
3.	Streams	17
	From External Iteration to Internal Iteration	17
	What's Actually Going On	20
	Common Stream Operations	21
	collect(toList())	22
	map	22
	filter	24
	flatMap	25
	max and min	26
	A Common Pattern Appears	27
	reduce	28
	Putting Operations Together	30
	Refactoring Legacy Code	31

	Multiple Stream Calls	34
	Higher-Order Functions	36
	Good Use of Lambda Expressions	36
	Key Points	37
	Exercises	37
	Advanced Exercises	39
4.	Libraries	41
	Using Lambda Expressions in Code	41
	Primitives	42
	Overload Resolution	45
	@FunctionalInterface	47
	Binary Interface Compatibility	47
	Default Methods	48
	Default Methods and Subclassing	49
	Multiple Inheritance	52
	The Three Rules	53
	Tradeoffs	54
	Static Methods on Interfaces	54
	Optional	55
	Key Points	56
	Exercises	57
	Open Exercises	58
F	Advanced Collections and Collectors	50
5.	Auvalieu Collections and Collectors	50
	Element Ordering	39
	Enter the Collector	60
	Inte Other Collections	62
	To Values	62
	10 values Partitioning the Data	63
	Crowning the Data	64
	Strings	65
	Composing Collectors	66
	Refactoring and Custom Collectors	6/
	Reduction as a Collector	69
	Collection Niceties	76
	Key Dointe	77
	Evercises	78
	LACICISCS	78
6	Data Parallelism	81
	Parallelism Versus Concurrency	81
	·	01

۸.

	Why Is Parallelism Important?	83
	Parallel Stream Operations	83
	Simulations	85
	Caveats	88
	Performance	89
	Parallel Array Operations	92
	Key Points	94
	Exercises	94
7.	Testing, Debugging, and Refactoring	. 97
	Lambda Refactoring Candidates	97
	In, Out, In, Out, Shake It All About	98
	The Lonely Override	98
	Behavioral Write Everything Twice	99
	Unit Testing Lambda Expressions	102
	Using Lambda Expressions in Test Doubles	105
	Lazy Evaluation Versus Debugging	106
	Logging and Printing	106
	The Solution: peek	107
	Midstream Breakpoints	107
	Key Points	108
8.	Design and Architectural Principles.	109
8.	Design and Architectural Principles. Lambda-Enabled Design Patterns	109 110
8.	Design and Architectural Principles. Lambda-Enabled Design Patterns Command Pattern	109 110 110
8.	Design and Architectural Principles. Lambda-Enabled Design Patterns Command Pattern Strategy Pattern	109 110 110 114
8.	Design and Architectural Principles. Lambda-Enabled Design Patterns Command Pattern Strategy Pattern Observer Pattern	109 110 110 114 117
8.	Design and Architectural Principles. Lambda-Enabled Design Patterns Command Pattern Strategy Pattern Observer Pattern Template Method Pattern	109 110 110 114 117 119
8.	Design and Architectural Principles. Lambda-Enabled Design Patterns Command Pattern Strategy Pattern Observer Pattern Template Method Pattern Lambda-Enabled Domain-Specific Languages	109 110 110 114 117 119 123
8.	Design and Architectural Principles. Lambda-Enabled Design Patterns Command Pattern Strategy Pattern Observer Pattern Template Method Pattern Lambda-Enabled Domain-Specific Languages A DSL in Java	109 110 110 114 117 119 123 124
8.	Design and Architectural Principles. Lambda-Enabled Design Patterns Command Pattern Strategy Pattern Observer Pattern Template Method Pattern Lambda-Enabled Domain-Specific Languages A DSL in Java How We Got There	109 110 110 114 117 119 123 124 125
8.	Design and Architectural Principles. Lambda-Enabled Design Patterns Command Pattern Strategy Pattern Observer Pattern Template Method Pattern Lambda-Enabled Domain-Specific Languages A DSL in Java How We Got There Evaluation	109 110 110 114 117 119 123 124 125 127
8.	Design and Architectural Principles. Lambda-Enabled Design Patterns Command Pattern Strategy Pattern Observer Pattern Template Method Pattern Lambda-Enabled Domain-Specific Languages A DSL in Java How We Got There Evaluation Lambda-Enabled SOLID Principles	109 110 114 117 119 123 124 125 127 127
8.	Design and Architectural Principles. Lambda-Enabled Design Patterns Command Pattern Strategy Pattern Observer Pattern Template Method Pattern Lambda-Enabled Domain-Specific Languages A DSL in Java How We Got There Evaluation Lambda-Enabled SOLID Principles The Single Responsibility Principle	109 110 114 117 119 123 124 125 127 127 127
8.	Design and Architectural Principles. Lambda-Enabled Design Patterns Command Pattern Strategy Pattern Observer Pattern Template Method Pattern Lambda-Enabled Domain-Specific Languages A DSL in Java How We Got There Evaluation Lambda-Enabled SOLID Principles The Single Responsibility Principle The Open/Closed Principle	109 110 114 117 119 123 124 125 127 127 127 128 130
8.	Design and Architectural Principles. Lambda-Enabled Design Patterns Command Pattern Strategy Pattern Observer Pattern Template Method Pattern Lambda-Enabled Domain-Specific Languages A DSL in Java How We Got There Evaluation Lambda-Enabled SOLID Principles The Single Responsibility Principle The Open/Closed Principle	109 110 110 114 117 119 123 124 125 127 127 127 127 128 130 134
8.	Design and Architectural Principles. Lambda-Enabled Design Patterns Command Pattern Strategy Pattern Observer Pattern Template Method Pattern Lambda-Enabled Domain-Specific Languages A DSL in Java How We Got There Evaluation Lambda-Enabled SOLID Principles The Single Responsibility Principle The Open/Closed Principle The Dependency Inversion Principle Further Reading	109 110 110 114 117 119 123 124 125 127 127 128 130 134 137
8.	Design and Architectural Principles. Lambda-Enabled Design Patterns Command Pattern Strategy Pattern Observer Pattern Template Method Pattern Lambda-Enabled Domain-Specific Languages A DSL in Java How We Got There Evaluation Lambda-Enabled SOLID Principles The Single Responsibility Principle The Open/Closed Principle The Dependency Inversion Principle Further Reading Key Points	109 110 110 114 117 119 123 124 125 127 127 128 130 134 137 137
8.	Design and Architectural Principles.Lambda-Enabled Design PatternsCommand PatternStrategy PatternObserver PatternTemplate Method PatternLambda-Enabled Domain-Specific LanguagesA DSL in JavaHow We Got ThereEvaluationLambda-Enabled SOLID PrinciplesThe Single Responsibility PrincipleThe Open/Closed PrincipleThe Dependency Inversion PrincipleFurther ReadingKey Points	109 110 110 114 117 119 123 124 125 127 127 127 128 130 134 137 139
8.	Design and Architectural Principles.	109 110 114 117 119 123 124 125 127 127 128 130 134 137 137 139 139

Index		
10.	Moving Forward	159
	Exercises	156
	Key Points	155
	When and Where	155
	Reactive Programming	153
	Completable Futures	149
	Futures	147
	The Pyramid of Doom	145
	Message Passing Architectures	144