



Functional Programming in Scala

Chapter 4

Handling errors without exceptions

Jordan Moldow

Oct. 9, 2014



Handling errors without exceptions



The good and bad aspects of exceptions

Possible alternatives to exceptions

The Option data type

- Usage patterns for Option - the Option functor

- Option composition and lifting - the monad laws

- Wrapping exception-oriented APIs

The Either data type

Exercises

Summary

Section 1



The good and bad aspects of exceptions

The good and bad aspects of exceptions

Possible alternatives to exceptions

The Option data type

- Usage patterns for Option - the Option functor

- Option composition and lifting - the monad laws

- Wrapping exception-oriented APIs

The Either data type

Exercises

Summary

Section 2



Possible alternatives to exceptions

The good and bad aspects of exceptions

Possible alternatives to exceptions

The Option data type

- Usage patterns for Option - the Option functor

- Option composition and lifting - the monad laws

- Wrapping exception-oriented APIs

The Either data type

Exercises

Summary

Section 3



The Option data type

The good and bad aspects of exceptions

Possible alternatives to exceptions

The Option data type

- Usage patterns for Option - the Option functor

- Option composition and lifting - the monad laws

- Wrapping exception-oriented APIs

The Either data type

Exercises

Summary

The Option data type



```
1 sealed trait Option[+A] {  
2   def map[B] (f: A => B): Option[B]  
3  
4   def getOrElse[B>:A] (default: => B): B  
5  
6   def flatMap[B] (f: A => Option[B]): Option[B]  
7  
8   def orElse[B>:A] (ob: => Option[B]): Option[B]  
9  
10  def filter(f: A => Boolean): Option[A]  
11 }  
12 case class Some[+A] (get: A) extends Option[A]  
13 case object None extends Option[Nothing]
```

Subsection 1



Usage patterns for Option - the Option functor

The good and bad aspects of exceptions

Possible alternatives to exceptions

The Option data type

Usage patterns for Option - the Option functor

Option composition and lifting - the monad laws

Wrapping exception-oriented APIs

The Either data type

Exercises

Summary

Subsection 2



Option composition and lifting - the monad laws

The good and bad aspects of exceptions

Possible alternatives to exceptions

The Option data type

Usage patterns for Option - the Option functor

Option composition and lifting - the monad laws

Wrapping exception-oriented APIs

The Either data type

Exercises

Summary

Subsection 3



Wrapping exception-oriented APIs

The good and bad aspects of exceptions

Possible alternatives to exceptions

The Option data type

Usage patterns for Option - the Option functor

Option composition and lifting - the monad laws

Wrapping exception-oriented APIs

The Either data type

Exercises

Summary

Section 4



The Either data type

The good and bad aspects of exceptions

Possible alternatives to exceptions

The Option data type

- Usage patterns for Option - the Option functor

- Option composition and lifting - the monad laws

- Wrapping exception-oriented APIs

The Either data type

Exercises

Summary

Section 5



Exercises

The good and bad aspects of exceptions

Possible alternatives to exceptions

The Option data type

- Usage patterns for Option - the Option functor

- Option composition and lifting - the monad laws

- Wrapping exception-oriented APIs

The Either data type

Exercises

Summary

Section 6



Summary

The good and bad aspects of exceptions

Possible alternatives to exceptions

The Option data type

- Usage patterns for Option - the Option functor

- Option composition and lifting - the monad laws

- Wrapping exception-oriented APIs

The Either data type

Exercises

Summary

Summary

