Encouraging Functional Thinking via Cleverly Designed Collections

Vitthal Srinivasan

CO-FOUNDER, LOONYCORN

www.loonycorn.com

Overview

Understand different types of Scala containers

Use tuples as alternatives to lightweight objects

Work with core Scala collections

Learn to use options for sound error handling

Scala Containers

Tuples

Lists

Maps

Options

Arrays

Mutable Collections

Collections and Arrays

Immutable Collections

Neither collection length nor can its contents be changed

Mutable Collections

Collection length as well as contents can change

Arrays

Fixed length, however contents can be modified

Tuples

Tuples

Values enclosed in (parentheses)

Ordered containers

Elements can be of different types

Technically not collections

Implement traits Tuple1, Tuple2,..

Demo

Creating tuples

Accessing individual elements

Iterating over tuples

Passing tuples to functions

Lists

Lists

Immutable collections

Implemented as singly Linked Lists

Terminate in special value Nil

Nil is technically List[Nothing]

Demo

Creating lists

- With the cons operator
- With the List apply function
- From other lists

Demo

Simple operations on lists

Demo

Higher order methods

- foreach
- map
- filter
- partition
- sortBy

Maps

Maps

Immutable associative containers

Equivalent to sets of key-value pairs

Demo

Creating maps

Map lookup

Higher order methods

Conversion to/from lists

Options

Options

Used to indicate absence or presence of a value

Value absent? Option will take special value None

Value present? Option will be an instance of Some

Often used for error handling

Demo

Option collections

Error handling

Arrays and Mutable Collections

Mutable Collections and Arrays

Immutable Collections

Neither collection length nor its contents can be changed

Mutable Collections

Collection length as well contents can change

Arrays

Fixed length, however contents can be modified

Mutable Collections

Mutable and immutable versions of lists, maps, sets exist

Mutable lists are called buffers

By default, immutable versions used

Demo

Creating mutable collections

Converting immutable collections to mutable ones

Working with arrays

Arrays

Not mutable collections

Fixed number of elements

Elements can be modified, however

Demo

Working with arrays

Up Next - Higher Order Functions

Foreach

Map

Filter

Scan

Fold

Reduce

Summary

Classified containers and collections

Used tuples as substitutes for lightweight classes

Worked with core collections - lists and maps

Handled error-prone code with options