Pattern Matching

Jesse Javier Cogollo Alvarez

Developer by passion

email: cogollo87@gmail.com

Functional programming group

September 9, 2015

Content

what is Pattern Matching?

Is the act of checking a given sequence of tokens for the presence of the constituens of some pattern.
https://en.wikipedia.org/wiki/Pattern_matching

characteristics

- **Useful**

Provide a powerful tool for declaring business logic in a concise and maintainable way.

characteristics

- **Basic pattern matching**

Allow you to make a programmatic choice between multiple conditions. cases can include types, wildcards, sequences,regular expressions and so forth.

characteristics

- **At its core**

Is a complex set of if/else expressions that lets you select from a number of alternatives.

characteristics

- **Guard**

It is useful to test particular conditions that cannot be tested in the pattern declaration itselft.

characteristics

- **Lists**

Pattern matching and Lists go hand in hand

Are inmutable. Is implemented as a linked list where the head of the list is called a cons cell. cons cell is represented by the ::case class

characteristics

- **case classes**

Case classes are classes that get to String, hashcode, and equal methods automatically. also can be constructed without using the new keyword. By default are read-only. and the case class is inmutable.

characteristics

- **Nested**

Scala's case classes give you a lot of flexibility for pattern matching, estracting values, nesting patterns, and so on. You can express a lot of logic in pattern declarations.

characteristics

- **Pattern matching as functions**

Are syntactic elements of the language when used with the match operator. You can also pass pattern matching as a parameter to other methods.

Pattern are functions and functions are instances, pattern are instances.

characteristics

- **Shape abstractions**

Visitor pattern: It is a pattern that allows you to add functionality to a class hierarchy after the hierarchy is already defined.

Summary

- Provide powerful declarative syntax for expressing complex logic.
- Provide a excellent and type-safe alternative to java's test/cast paradigm.
- Used with case class and extraction provide a powerful way to traverse object hierarchies.

DEMO

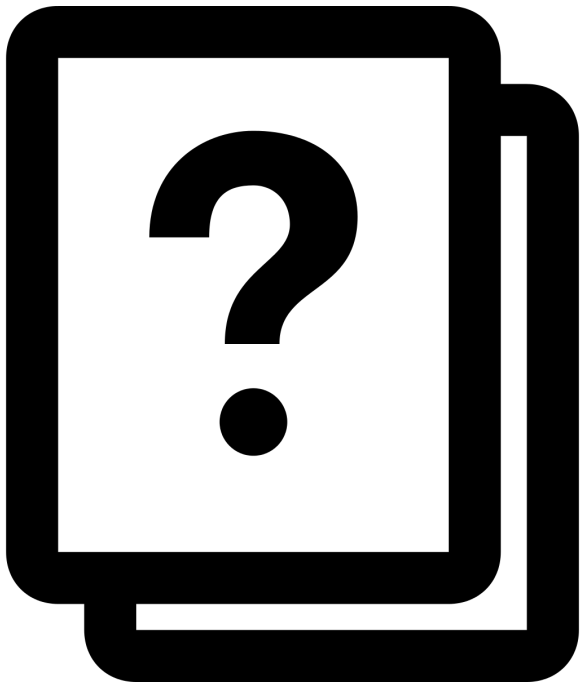
=)

Resource

Beginning Scala

http://www.amazon.com/Beginning-Scala-Vishal-Layka/dp/1484202333

Questions



Thanks !!! =)