Design Patterns for the Object-oriented and functional mind

Kiko Fernandez-Reyes

Contents

Prologue

To my lovely wife, baby and Maya!

Introduction

This book covers simple design principles and advanced design patterns used in industry. All the design patterns and principles are written in 4 languages: Java, Python, Haskell and Clojure. These languages are already used in production in big companies and are well stablished. For instance, Boeing uses Clojure in their onboard diagnostic system, Facebook uses Haskell to prevent spam and phising attacks in their site, Pinterest uses Python for their backend operations and Java is widely used everywhere.

The mix of these languages cover two different paradigms (object-oriented and functional programming) and two different type systems (languages dynamically and statically typed), shown in Table 1.1.

Paradigm / Type	Static	Dynamic
Object-oriented	Java	Python
Functional	Haskell	Clojure

Tabla 1.1 Relation between language paradigm and type system

Why multiple languages?

You should choose the right language for the right job. Choosing a language is an important design decision as languages are tied to a paradigm and a type system, and these cannot be changed. For instance, Python has some functional features but, all in all, **Python is an object-oriented language**.

This book tries to shed some light in the design patterns of different languages, in relation to its paradigm and type system.

Scope of the book

This section covers the scope of the book. There may be some overlapping areas with the previous explanation…

Also, what is outside the scope of this book. Concepts such as:

- Classes,
- objects,
- functions,
- lambdas,
- high-order functions,
- etc

should be already known to the reader. However, it may be good to show a few definitions and simple examples to remind newcomers.

Organisation

Explain that the book covers the design patterns and needs to list:

- when to use it
- advantages

- disadvantages
- one or more ways of creating the design pattern and its drawbacks
- some UML?

Recap

This section covers basic stuff such as:

- Classes,
- objects,
- functions,
- lambdas,
- high-order functions,
- some UML
- etc

in Java, Python, Haskell and Clojure

GRASP principles

Explain GRASP design principles.

Design Patters: Creational

Covers creational design patterns

Design Patters: Structural

Covers structural design patterns

Design Patters: Behavioral

Covers behavioral design patterns