



REFACTORING
· GURU ·

★ Premium Content

⌘ Refactoring

⌘ Design Patterns

★ Shop Now!

👤 SPRING SALE 🌸

🌐 English

✉ Contact us

👤 Log in

🏠 / [Design Patterns](#)

Classification of patterns

Design patterns differ by their complexity, level of detail and scale of applicability to the entire system being designed. I like the analogy to road construction: you can make an intersection safer by either installing some traffic lights or building an entire multi-level interchange with underground passages for pedestrians.

The most basic and low-level patterns are often called *idioms*. They usually apply only to a single programming language.

The most universal and high-level patterns are *architectural patterns*. Developers can implement these patterns in virtually any language. Unlike other patterns, they can be used to design the architecture of an entire application.

In addition, all patterns can be categorized by their *intent*, or purpose. This book covers three main groups of patterns:

- **Creational patterns** provide object creation mechanisms that increase flexibility and reuse of existing code.
- **Structural patterns** explain how to assemble objects and classes into larger structures, while keeping these structures flexible and efficient.
- **Behavioral patterns** take care of effective communication and the assignment of responsibilities between objects.

RETURN

READ NEXT

← Criticism of patterns

Catalog →

[Home](#) [Refactoring](#) [Design Patterns](#) [Premium Content](#) [Forum](#) [Contact us](#)



© 2014-2024 Refactoring.Guru. All rights reserved.
🖼 Illustrations by Dmitry Zhart

[Terms & Conditions](#) [Privacy Policy](#) [Content Usage Policy](#) [About us](#)

Ukrainian office:

📍 FOP Olga Skobeleva
📍 Abolmasova 7
Kyiv, Ukraine, 02002
✉ Email: support@refactoring.guru

Spanish office:

📍 Oleksandr Shvets
📍 Avda Pamplona 63, 4b
Pamplona, Spain, 31009
✉ Email: support@refactoring.guru