

Design Pattern & Clean Code

Zero Impact Injection

Thomas Ley | @CleanCodeCoach

Goals

- Refactor Brownfield Project
- Single Responsibility Principle
- Interface Segregation Principle
- Dependency Injection
- Zero Impact Injection
- Testable Code.

Steps

- Extract Code (Method → Class → Project)
- Create Single Dependency (`new()`)
- Inject Dependency
- Null-Object Pattern.

Extract code [SRP]

- Extract method
 - Move to class
 - Introduce field
- 💡 Namespace indicates dependencies
- 💡 Create a documentation of the class without "and"

vs Demo: Barcode Code

[Zero Impact Injection]

- Add interface to class
- Inject as interface into class
- Use `??` for default implementation

vs Demo: Barcode class

[Null-Object Pattern]

- Reduces null-checks
- "Identity Element" for an interface
- Empty implementation
- Returns `default`
- IFooBar --> EmptyFooBar
- Implemented along with Interface

vs Demo: Empty logger class