Smart ABAP

Design Patterns for Real-World Scenarios

George Drakos ABAPConf 2025
Vienna

Presentation Agenda

01 **Introduction**

02 **Object-Oriented ABAP Refresher**

03 **Harnessing Design Patterns in ABAP**

04 **Hands on:Applying Patterns in ABAP**

05 **Common Pitfalls and Anti-Patterns**

06 **Wrap-Up**

Brief personal introduction & session goals

About Me

- **SAP Technical Consultant @ Teya**
- **5+ years in SAP Development**
- **Speaker at ABAPConf 2024 – “Advanced New ABAP”**

Session Goals

- **Introduction to** Design Patterns and key principles
- Share **practical examples** you can **apply right away** in your projects/developments
- Recognize when design patterns add value—and when to avoid them

Building Better ABAP with Design Patterns

ABAP is evolving...

- From procedural-heavy legacy systems to modern architectures
- Complexity is increasing

Design patterns = proven solutions

- They offer **reusable, battle-tested** ways to solve recurring problems
- Help create **consistent**, **understandable**, and **testable** code
- Bridge the gap between **technical logic** and **business requirements**

In real projects, they help you...

- Separate concerns
- Encapsulate changes
- Enable teamwork
- Improve maintainability

Design patterns don’t add complexity. They organize it.

Yes but….where should I start?

"The secret of getting ahead is getting started." – Mark Twain

Basics First.Object Oriented Programming(OOP)

- Don't confuse OOP with class oriented programming
- Familiarize yourself with Object Oriented
- Start Small
- Prefer Interfaces for Flexibility
- Avoid Static Methods for Business Logic
- Use Access Modifiers Wisely

Adhere to Design Principles

- Clean code & SOLID principles
- Layered architecture considerations
- Single Level of Abstraction(SLA)
- Testing and mocking

Jump into Design Patterns

- Start with common patterns: Singleton, Factory, Strategy, Observer
- Focus on why a pattern is used, not just how
- Refactor existing code with patterns where it fits
- Don’t force patterns — let the problem guide you

Keep in mind

- Start from the most abstract level and move down to the concrete level.
- Your client program should only know about the interface but know nothing about the concrete classes

Hands on: Applying Patterns in ABAP

Adapter

MVC

Factory

Strategy

Common Pitfalls and Anti-Patterns

- Avoid adding unnecessary complexity
- Avoid forcing inheritance where composition would be more flexible and maintainable
- Skip design patterns if they introduce unnecessary complexity
- Don't apply design patterns where they aren’t needed
- Reuse code properly

Wrap up.Key Takeaways

- Design Patterns = Smarter ABAP
- OOP Done Right
- Avoid the Traps.
- Write code that scales and lasts

Call to Action

- Official SAP Help Portal for Design Patterns
- Design Patterns in ABAP Objects written by Kerem Koseoglu
- Pick one pattern from hands-on examples
- Apply it to your next project!
- Make patterns your design habit

The best developers don’t just write code—they craft solutions.

Thank you