# "SOLID Programming in C#"

Jesper Gulmann Henriksen





#### Presentation

- Name
- Company affiliation
- Area of expertise
- ▶ C# experience
- Expectations for the course



#### Prerequisites

- Working knowledge of newest C# and Visual Studio
  - Object-oriented development
    - Classes
    - Inheritance
    - Interfaces
    - Delegates
    - •
  - Some knowledge of Design Patterns might be helpful (but is not required)
- ▶ An interest in producing maintainable C# code ☺



- Module 0: Introduction
- Module 1: The SOLID Principles
  - Single Responsibility Principle
  - Open/Closed Principle
  - Liskov Substitution Principle
  - Interface Segregation Principle
  - Dependency Inversion Principle
- Module 2: SOLID in Practice
  - Workshop A.1
  - 2.1: Repository Pattern
  - Workshop A.2



- Module 2: SOLID in Practice (Cont'd)
  - (Optional) 2.2: Automatic Testing
  - Workshop A.3
- Module 3: Managing Dependencies
  - Workshop A.4
  - 3.1: Null Object Pattern
  - (Optional) Workshop A.5



- Module 04: Workshop B: Create User Service
  - 4.1: Strategy Pattern
  - 4.2: (Optional) Decorator / Proxy Pattern
  - 4.3: (Optional) Composite Pattern
- Module 05: Dependency Injection Containers
  - Containers
  - Workshop C.1: Adding Unity DI
  - Dependency Injection Patterns
  - Workshop C.2: Interception
  - Lifetime Management



Conclusion

▶ (For Later) Workshop D: If Time Permits...



#### Course Material

- Book
  - Mark Seemann and Steven van Deursen: "Dependency Injection Principles, Practices, and Patterns" Manning (2019), 1st Edition
- Slides
- Examples for every module
- Workshop and all workshop solutions
- Course evaluation after Day 2



#### Practical Information

- ▶ Each course day will be from 9.00 to 16.00
- Breaks
- ▶ Toilets
- Food and beverages
- Phones and devices
- Smoking
- Any questions...?



