Software Design Techniques and Mechanisms

Course : Andrei Poștaru, senior lecturer

Seminar : Vasile Drumea, university assistant

Chișinău 2021

Contacts

1. **Mobile:** +37369886119
2. **Email:** vasile.drumea@faf.utm.md
3. **Github:** https://github.com/DrVasile
4. **Facebook**

Prerequisites

1. OOP Principles
2. OO Modeling (e.g. UML)
3. One OOP programming language

SOLID Principles

1. Single Responsibility Principle
2. Open-Closed Principle
3. Liskov Substitution Principle
4. Interface Segregation Principle
5. Dependency Inversion Principle

Design Patterns

A *Design Pattern* is a generic, repeatable solution

to a commonly occurring problem in software design. [2]

Creational Patterns

1. **Singleton**
2. **Prototype**
3. **Builder**
4. **Factory method**
5. **Abstract factory**

Structural Patterns

1. Adapter
2. Bridge
3. Composite
4. Decorator
5. Facade

Behavioral Patterns

1. Chain of Responsibility
2. Command
3. Iterator
4. Observer
5. Strategy

References

1. The “Gang of four”, 1994, *Design Patterns: Elements of Reusable Object-Oriented Software*
2. https://sourcemaking.com/design_patterns
3. https://github.com/DrVasile/SDTM-Examples
4. https://github.com/DrVasile/SDTM-Labs

Thanks for your attention!
Questions?