# DESIGN AND IMPLEMENTATION
CHAPTER 7


- Presented By Shahriar Yazdipour
- November 2019
- Software & Systems Engineering Students Talk
- Prof. Dr.-Ing. Armin Zimmermann

# AGENDA

- DESIGN AND IMPLEMENTATION
- OBJECT-ORIENTED DESIGN USING UML
- DESIGN PATTERNS
- IMPLEMENTATION ISSUES
- OPEN SOURCE DEVELOPMENT

# DESIGN AND IMPLEMENTATION

**Software design & implementation is** the stage in the software engineering process that an executable software system is developed.

- **Software design** is a creative activity in which you identify software components and their relationships, based on a customer's requirements.
- **Implementation** is the process of realizing the design as a program.

# BUILD OR BUY

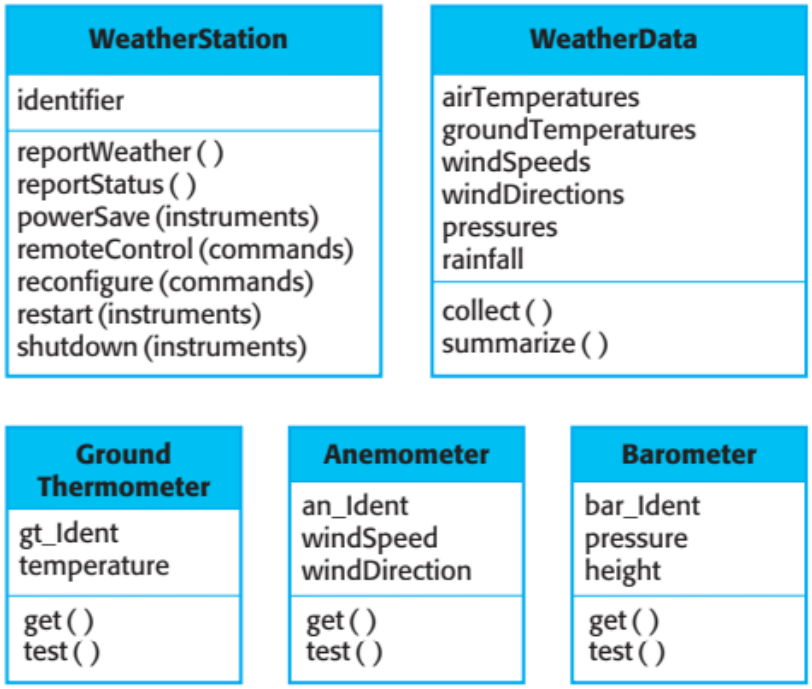
- In a wide range of domains, it is possible to buy off-the-shelf systems (COTS) that can be adapted and tailored to the users' requirements.
  - For Example: To implement a medical records system, we can buy a package that is already used in hospitals vs developing a system. One can be cheaper and faster to access but may not be scalable for our system.

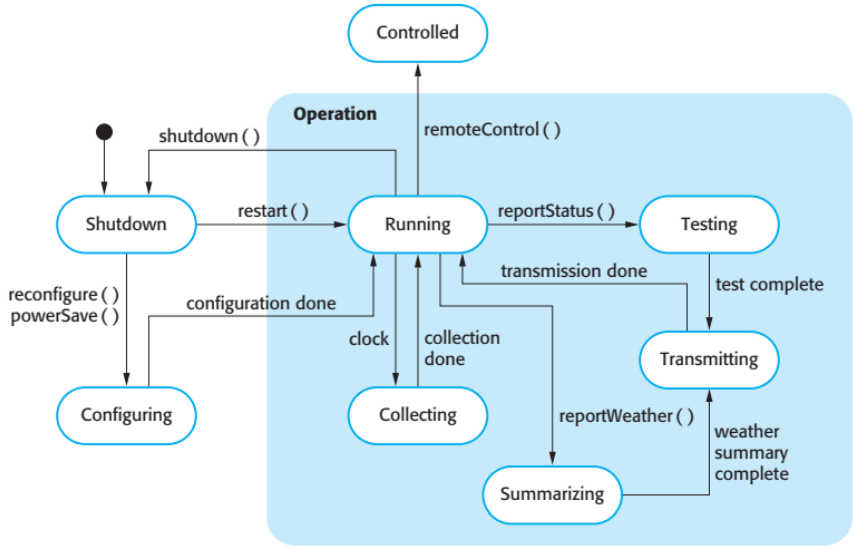
* COTS = Commercial off-the-shelf

- DESIGN AND IMPLEMENTATION
- **OBJECT-ORIENTED DESIGN**
- DESIGN PATTERNS
- IMPLEMENTATION ISSUES
- OPEN SOURCE DEVELOPMENT

# OBJECT-ORIENTED DESIGN PROCESS

- It involves developing a number of different system models.
- They require a lot of effort for development and maintenance which for **small systems** this may **not be cost-effective**. However, for **large systems** developed by different groups design models are an **important communication mechanism** for team members.




# OBJECT CLASS / STATE DIAGRAM

- DESIGN AND IMPLEMENTATION
- OBJECT-ORIENTED DESIGN
- **DESIGN PATTERNS**
- IMPLEMENTATION ISSUES
- OPEN SOURCE DEVELOPMENT

# DESIGN PATTERNS

- A way of reusing abstract knowledge about a common problem and its solution.
- Patterns are a great idea but you need experience of software design to use them effectively.
- You have to recognize situations where a pattern can be applied.

# DESIGN PATTERNS

Creational Patterns

- Object Pool / Resource Pool
- Prototype
- Factory Method (Abstract Factory Similar)
- Singleton
- Builders

Structural Patterns

- Adapter
- Decorator
- Bridge (Handle-Body Pattern)
- Composite
- Facade
- Flyweight
- Proxy

Behavioral Patterns

- Chain of Responsibility
- Command
- Interpreter
- Mediator
- Memento
- Template
- Visitor
- Null Object
- Iterator/Collection
- State
- Observer
- Strategy

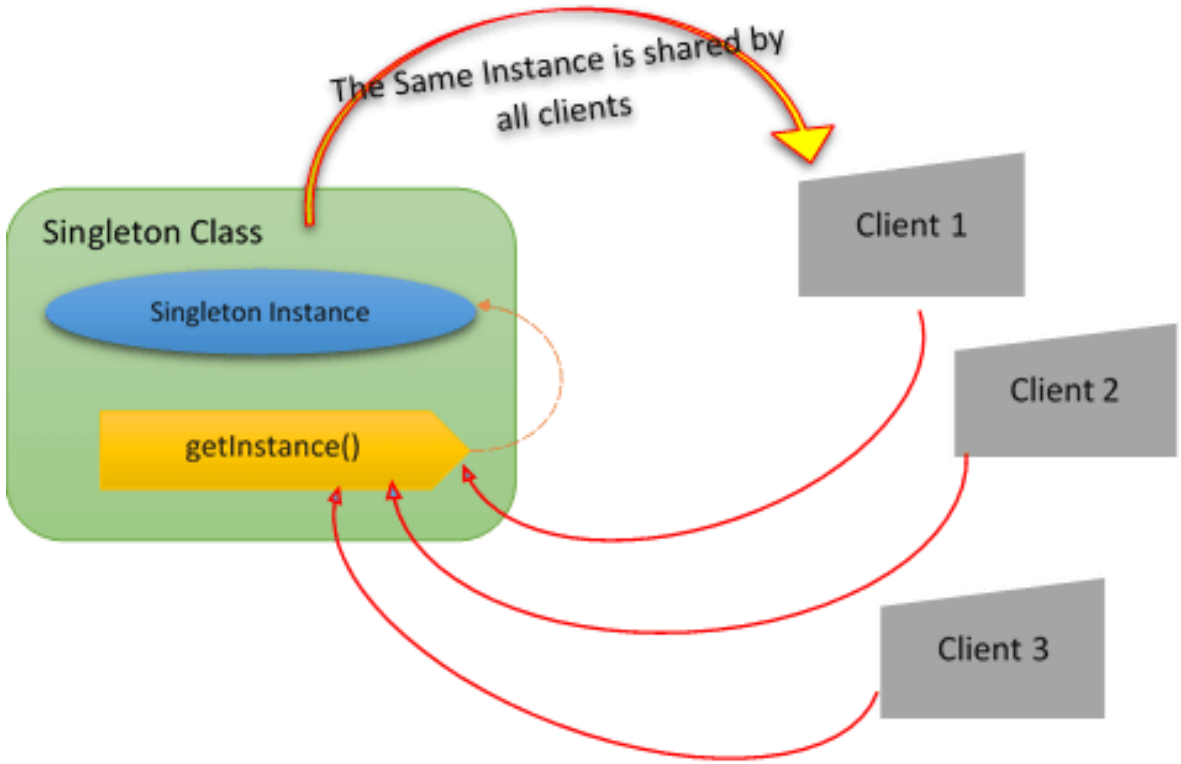


https://yazdipour.github.io/notes/#/principles/design-patterns

# SINGLETON

When? When the application needs an object frequently, and the object itself is very computationally expensive.

When not a good idea? When we using it in multi-thread scenario.




```
class Singleton {
    private static Singleton instance = new Singleton();

    private Singleton() { }

    public static Singleton getInstance() {
        return instance;
    }
}
```



javagists.com/singleton-design-pattern

- DESIGN AND IMPLEMENTATION
- OBJECT-ORIENTED DESIGN
- DESIGN PATTERNS
- **IMPLEMENTATION ISSUES**
- OPEN SOURCE DEVELOPMENT

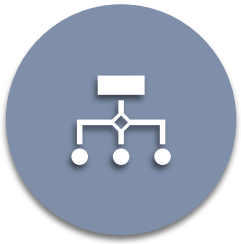
# IMPLEMENTATION ISSUES

- **Reuse**
  - Make as much use as possible of existing code.
- **Configuration management**
  - During the development process, you have to keep track of the many different versions of each software component in a configuration management system.
- **Host-target development**
  - Production software does not usually execute on the same computer as the software development environment

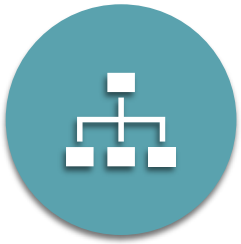
# REUSE LEVELS

ABSTRACTION LEVEL

OBJECT LEVEL



COMPONENT LEVEL



SYSTEM LEVEL

- DESIGN AND IMPLEMENTATION
- OBJECT-ORIENTED DESIGN
- DESIGN PATTERNS
- IMPLEMENTATION ISSUES
- **OPEN SOURCE DEVELOPMENT**

# OPEN SOURCE

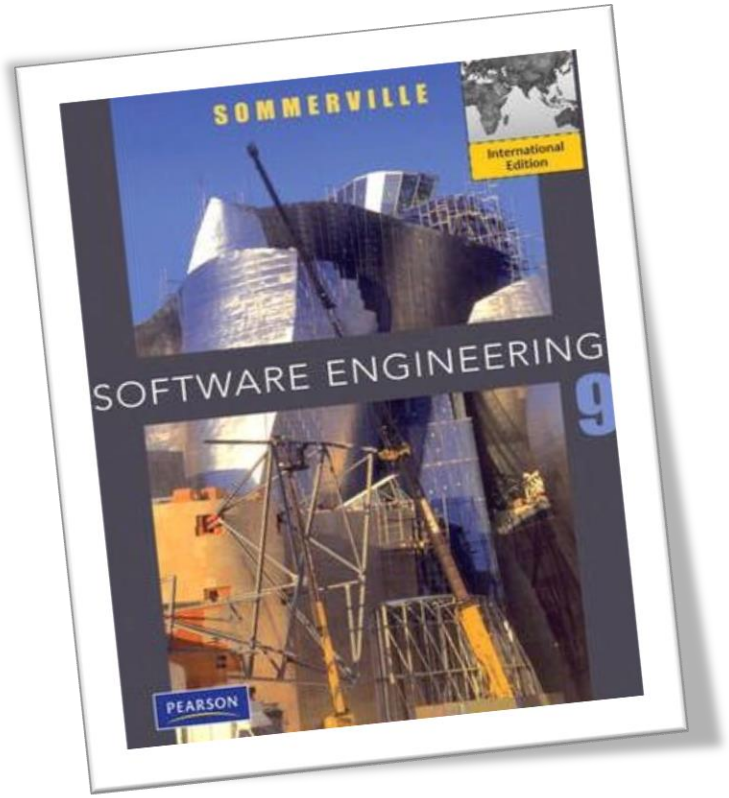
Is an approach that we publish our source code and volunteers are invited to participate in the development process.

BUSINESS

- Open source projects business model is not reliant on selling a software product but on selling support for that product.
- It is believed that involving the open source community will allow software to be developed more cheaply, more quickly and will create a community of users for the software.

LICENSE: GPL, LGPL, BSD, …

# REFERENCE




SOFTWARE ENGINEERING
SOMMERVILLE

# THANK YOU FOR YOUR ATTENTION.