CS 525 - ASD

Advanced Software Development

MS.CS Program

Department of Computer Science

Rene de Jong, MsC.

Maharishi University
OF MANAGEMENT

CS 525 - ASD

Advanced Software Development

Maharishi University
OF MANAGEMENT

Lesson 1

L1: ASD Introduction
L2: Strategy, Template method
L3: Observer pattern
L4: Composite pattern, iterator pattern
L5: Command pattern
L6: State pattern
L7: Chain Of Responsibility pattern

Midterm

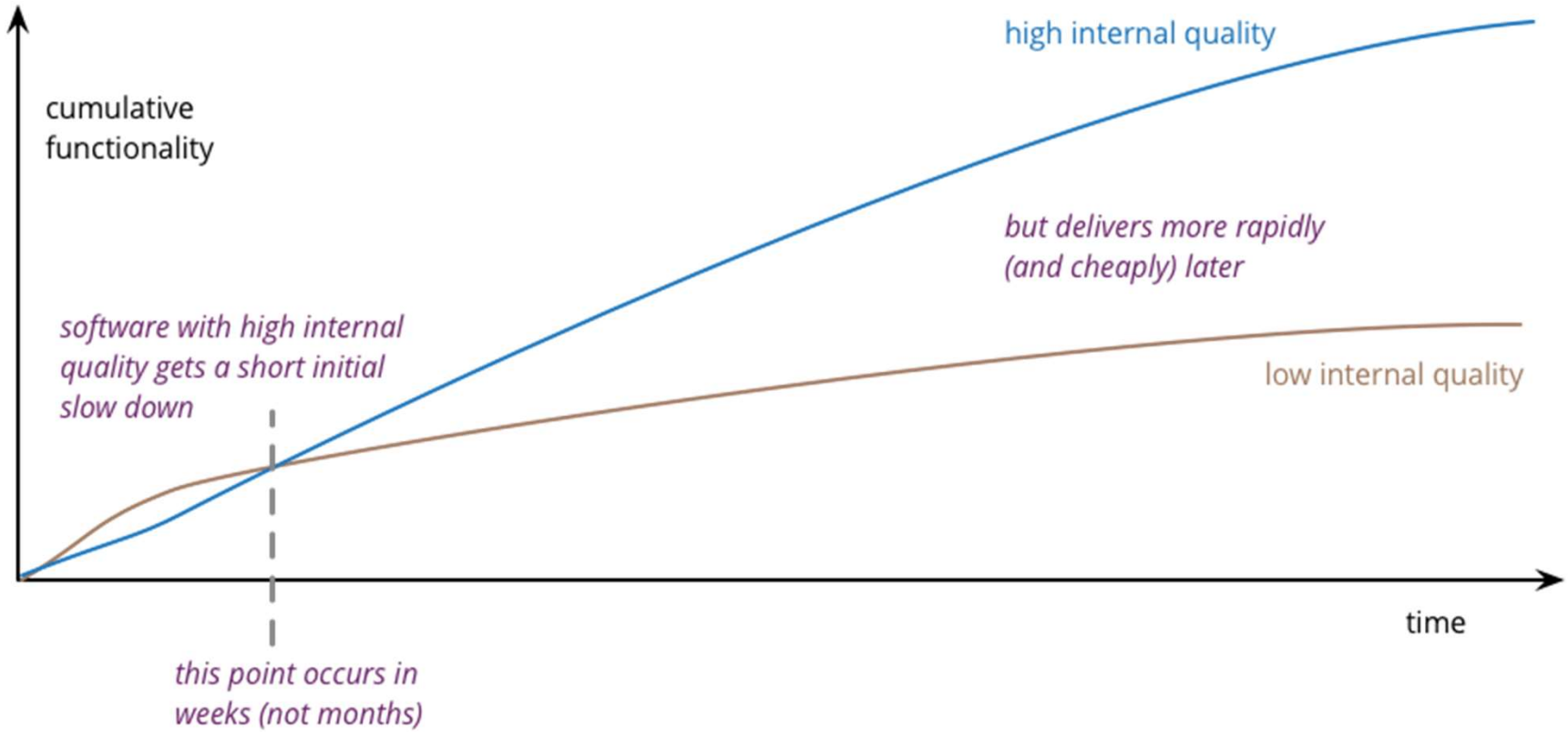
L8: Proxy, Adapter, Mediator
L9: Factory, Builder, Decorator, Singleton
L10: Framework design
L11: Framework implementation
L12: Framework example: Spring framework
L13: Framework example: Spring framework

Final

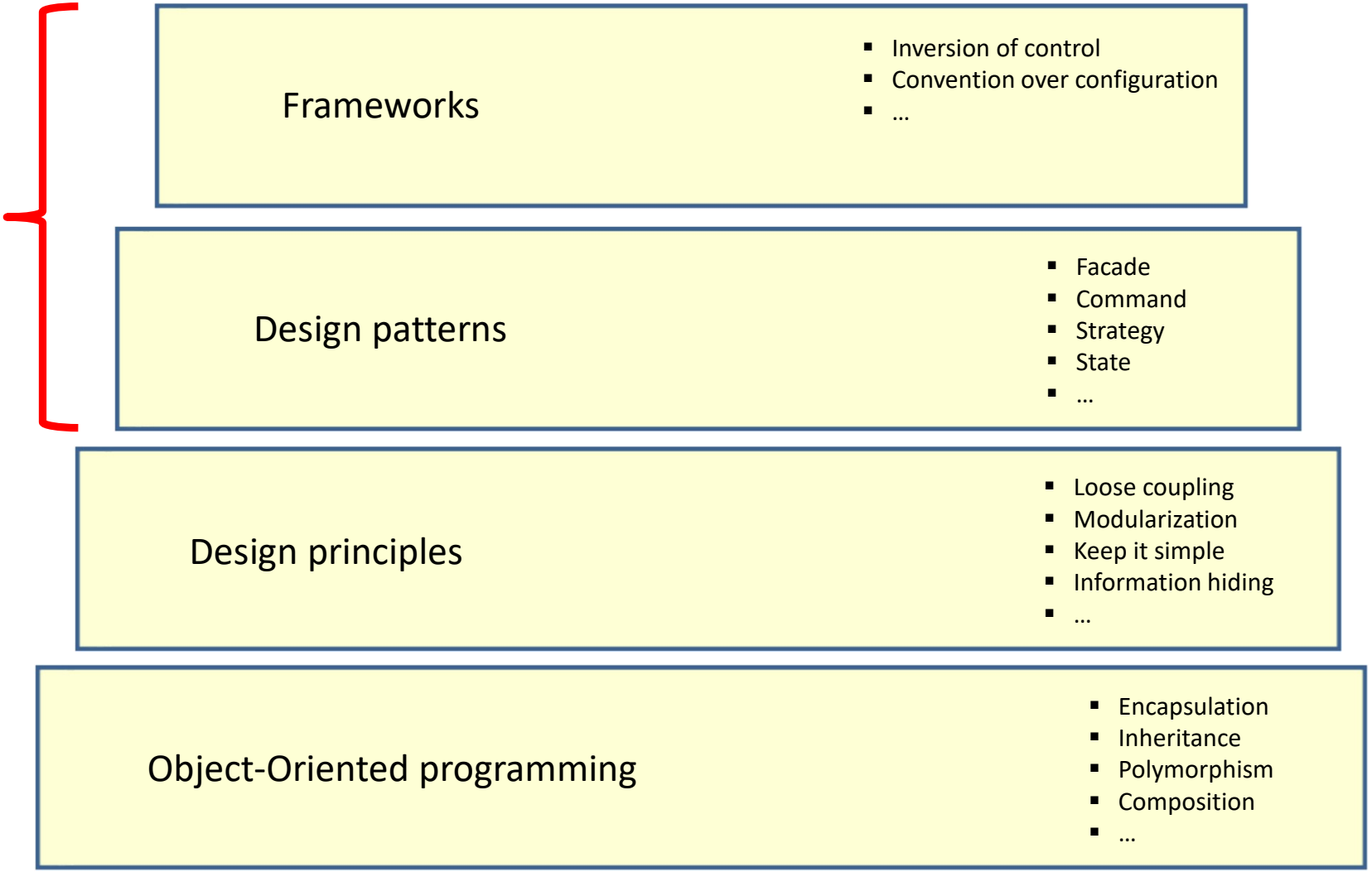
Advanced Software Development

- Principles and best practices of good software design
  - Design patterns
  - Frameworks
- Improve the quality of your design/code
  - Reduce technical debt

Why software quality matters?



ASD course



Lesson overview

L1: ASD Introduction
L2: Strategy, Template method
L3: Observer pattern
L4: Composite pattern, iterator pattern
L5: Command pattern
L6: State pattern
L7: Chain Of Responsibility pattern

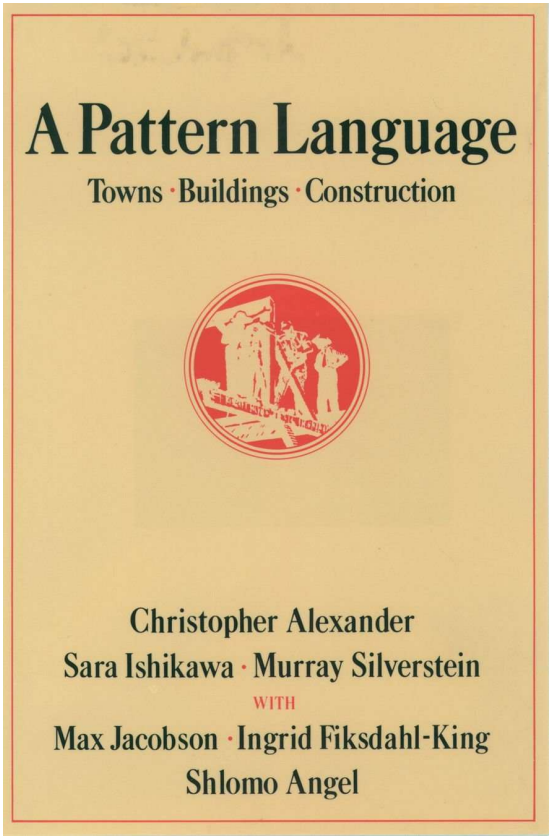
Midterm

L8: Proxy, Adapter, Mediator
L9: Factory, Builder, Decorator, Singleton
L10: Framework design
L11: Framework implementation
L12: Framework example: Spring framework
L13: Framework example: Spring framework

Final

Pattern

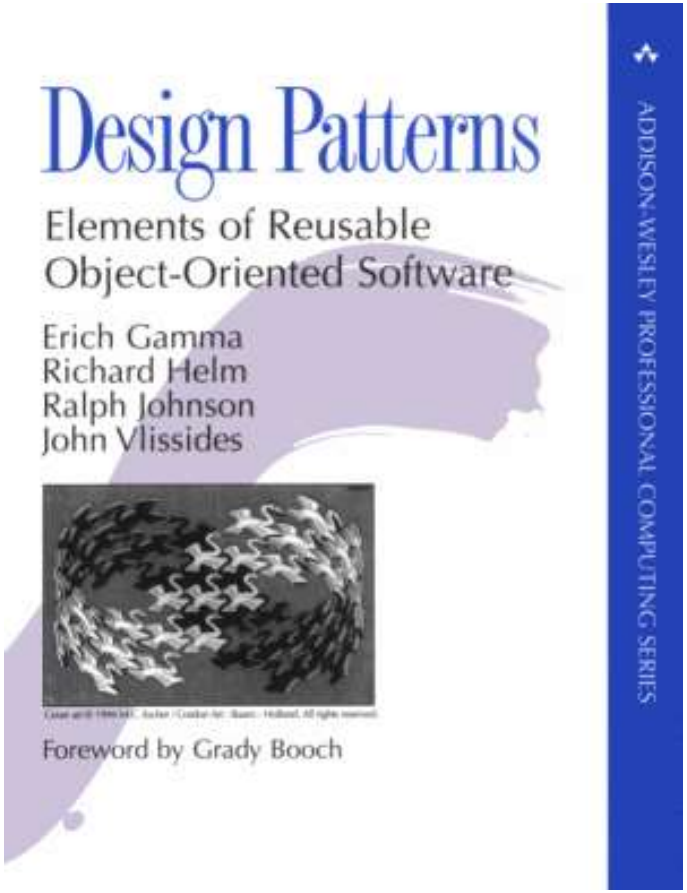
- Christopher Alexander



A pattern is a general, reusable solution to a commonly occurring problem within a given context

Design pattern

- A general repeatable solution to a commonly occurring problem in software design.
- Reuse of design (not code)
- GoF: Gang of Four



Patterns in software development

A pattern is a general, reusable solution to a commonly occurring problem within a given context

Architecture patterns

- Client-server
- Layering
- Components
- Microservices
- Stream based
- …

Integration patterns

- Router
- Filter
- Point-to-point
- Transformer
- …

UI patterns

- Navigation
- Dealing with data
- Forms
- Menus
- …

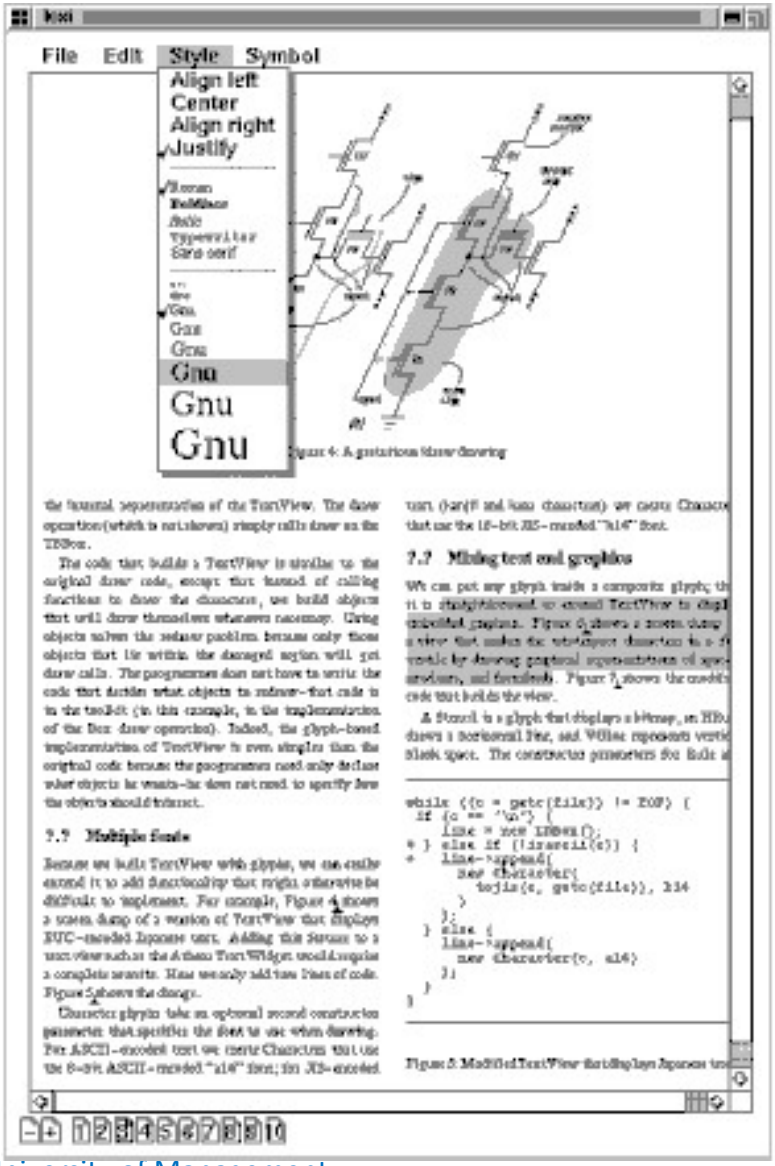
Data access patterns

- ORM
- Stored procedures
- SQL
- Lazy loading
- Id generation
- …

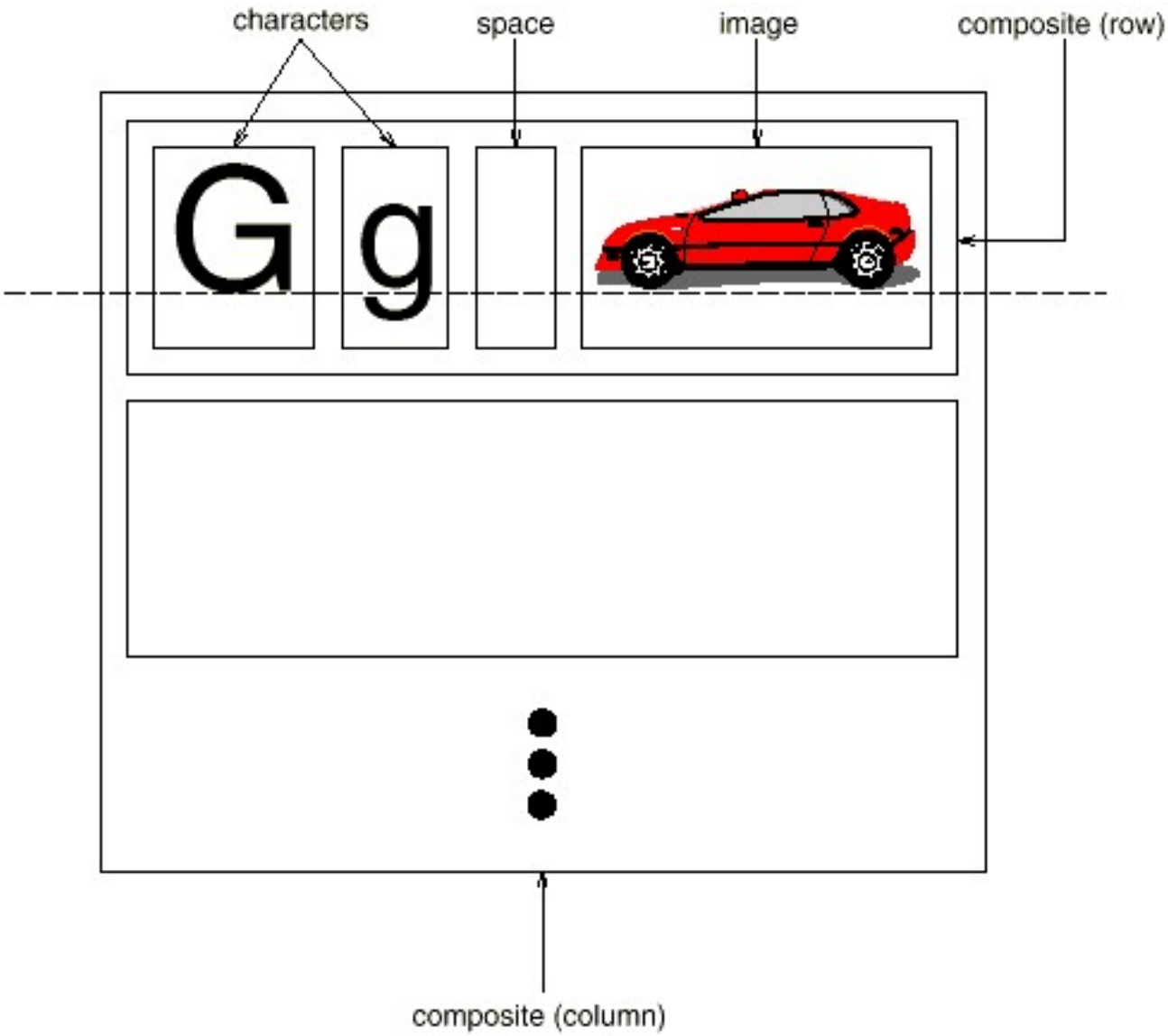
Design patterns

- Facade
- Command
- Strategy
- State
- …

Document editor

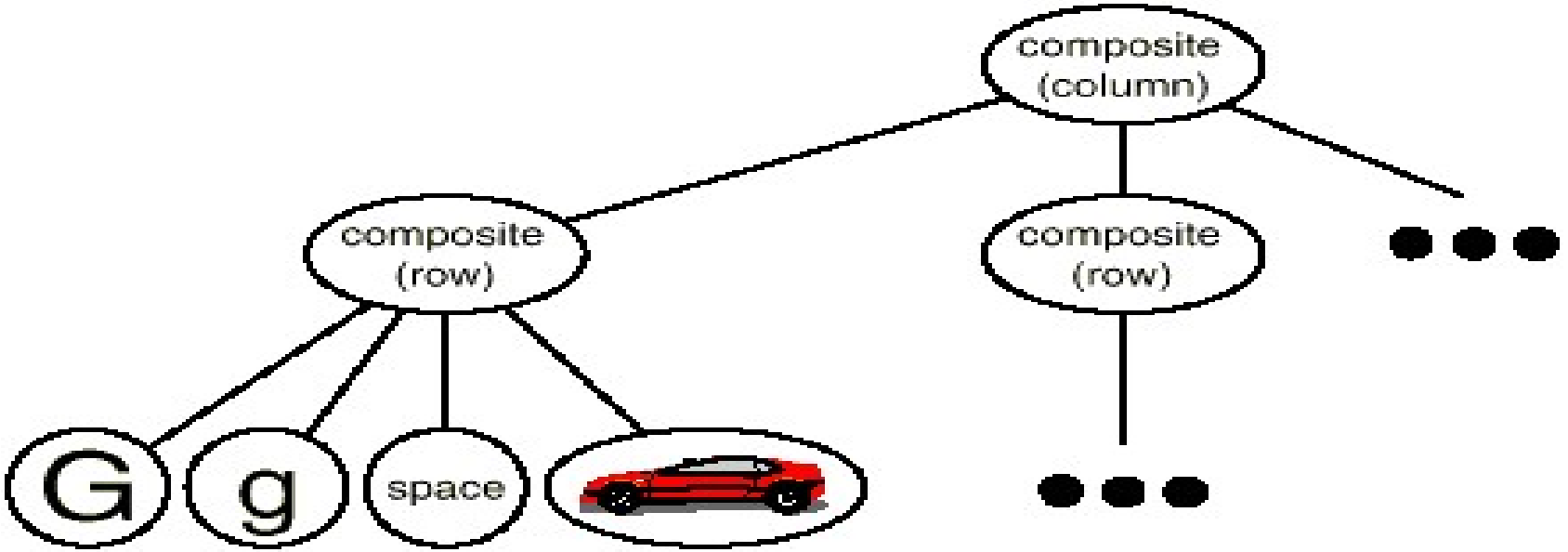


1. Document structure

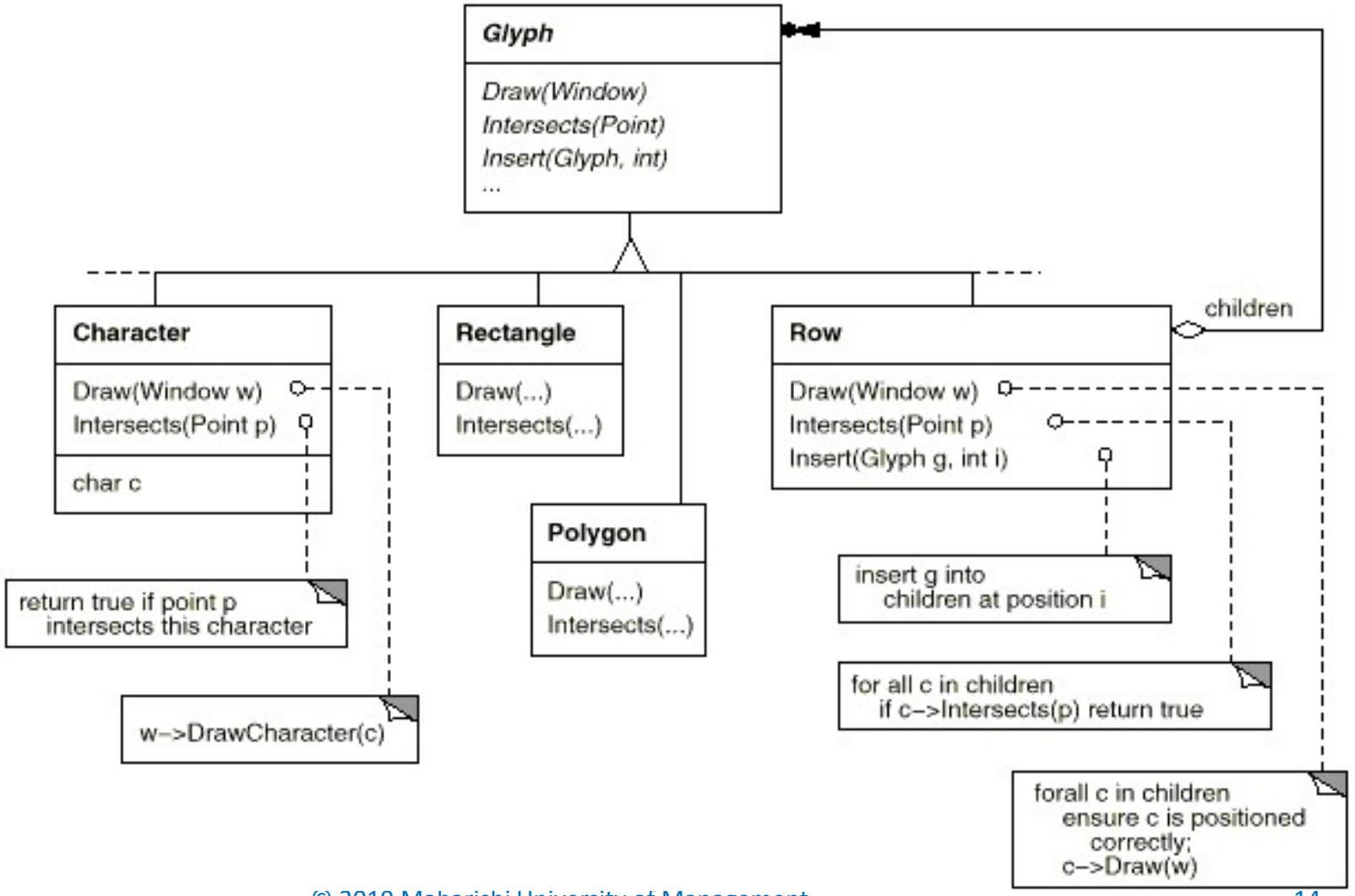


Tree structure

- We want to work with these tree elements in common way
  - Copy-paste
  - Drag & drop

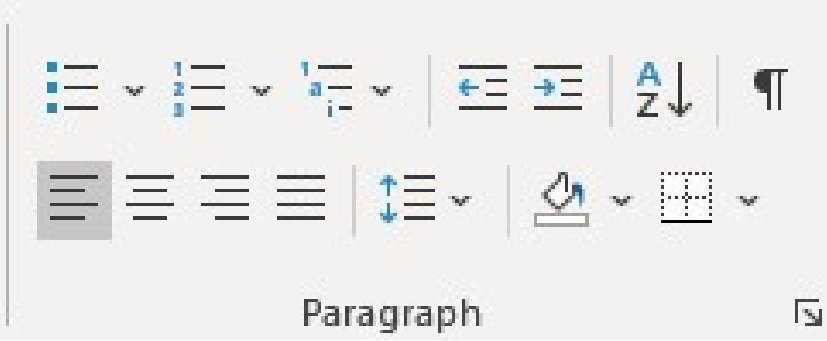


Composite pattern

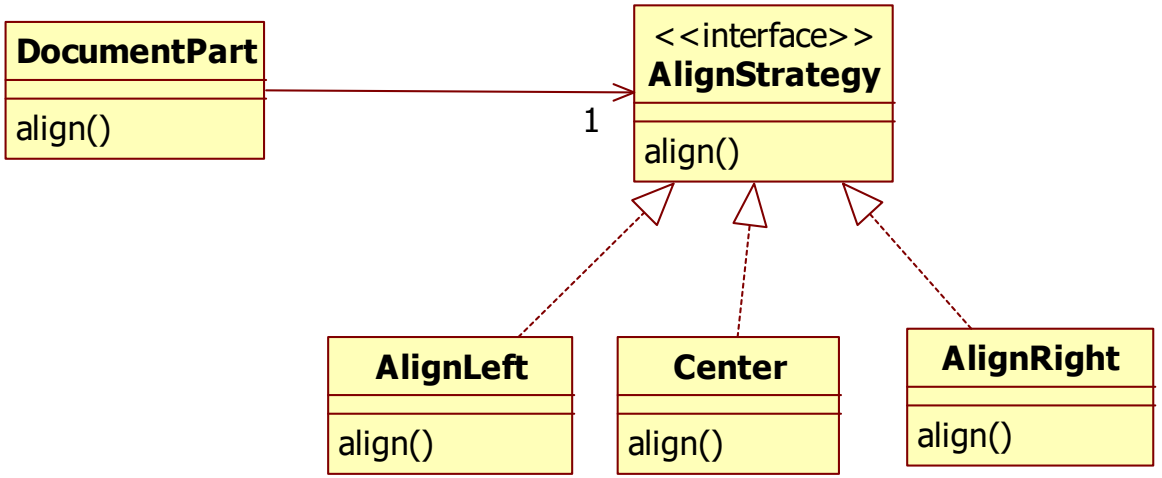


2. Formatting

- How can we format the structure of the document?
- We need to allow different ways to format.

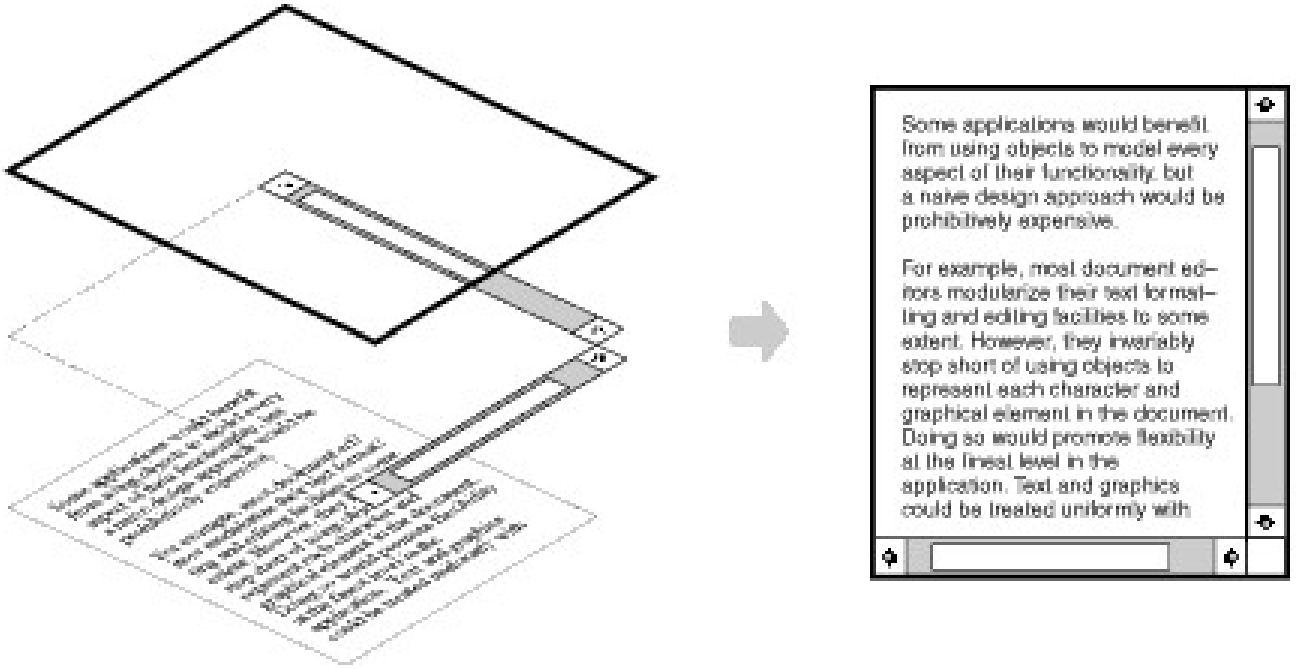


Strategy pattern

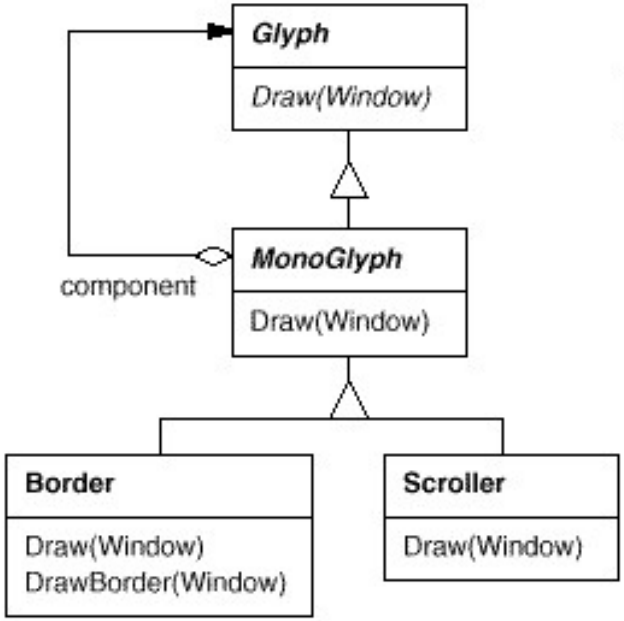


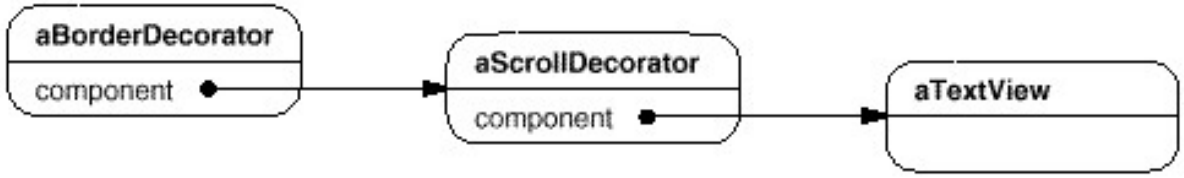
3. Embellishing the user interface

- Add a Border around the text editing area
- Add scroll bars



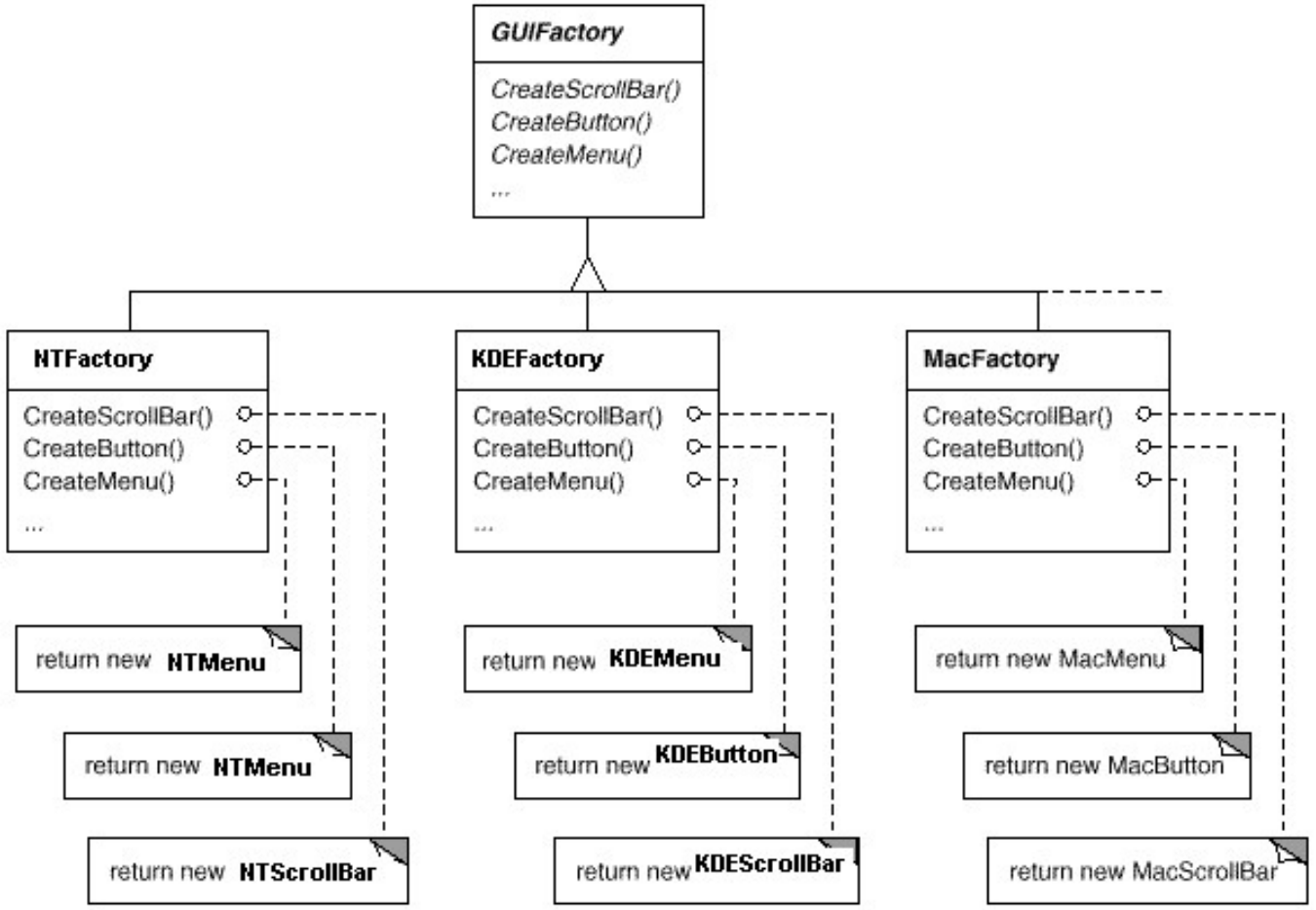
Decorator pattern





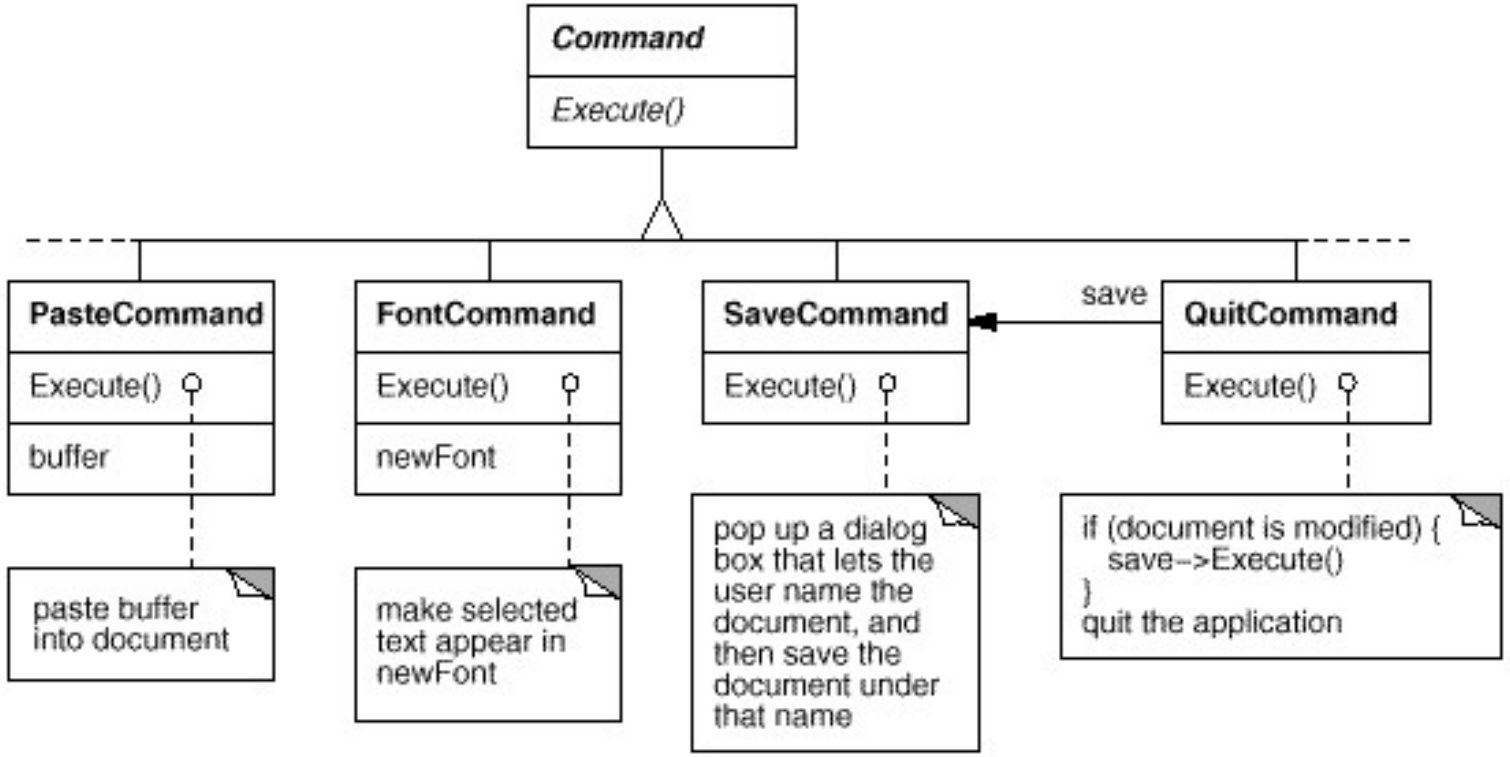
4. Supporting multiple look and feels

- Abstract factory pattern



5. User operations

- Should be independent of the UI
- We also want undo/redo
- Command pattern



Categories of patterns

Creational	Structural	Behavioral
▪ Factory method ▪ Abstract factory ▪ Builder ▪ Singleton	▪ Composite ▪ Decorator ▪ Adapter ▪ Façade ▪ Proxy	▪ Command ▪ Iterator ▪ Mediator ▪ Chain of responsibility ▪ Observer ▪ State ▪ Strategy ▪ Template method

Half baked

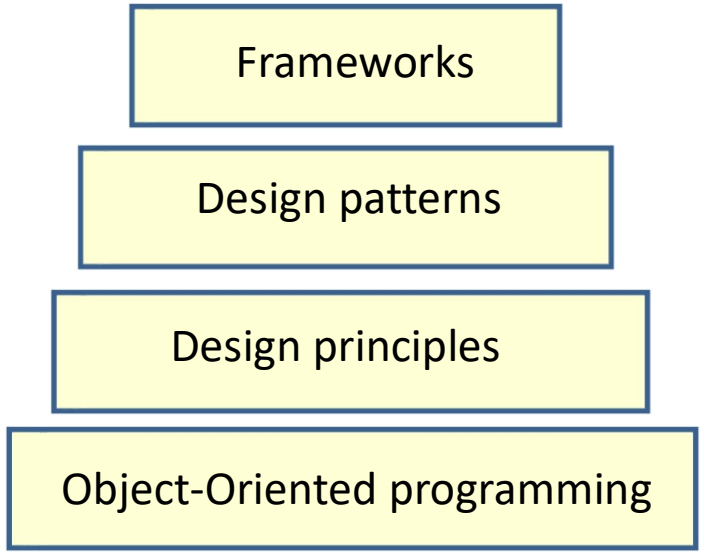
- A design pattern isn't a finished design that can be transformed directly into code.
- Design patterns are half baked
- You have to tailor them for your situation

How to become a good designer?

- By designing software
  - Class diagrams
  - Sequence diagrams
  - Code

Design principles

- Keep it simple
- Keep it flexible
- Loose coupling
- Separation of concern
- Information hiding
- Principle of modularity
- DRY: Don’t repeat yourself
- Encapsulate what varies
- Solid
  - **S**ingle Responsibility Principle (SRP)
  - **O**pen-Closed Principle (OCP)
  - **L**iskov Substitution Principle (LSP)
  - **I**nterface Segregation Principle (ISP)
  - **D**ependency Inversion Principle (DIP)



Main point

- A design pattern is a reusable solution for a generic design problem within a context
- The unified field is the field of all possibilities which contains the intelligence to solve all problems in the most optimal way.