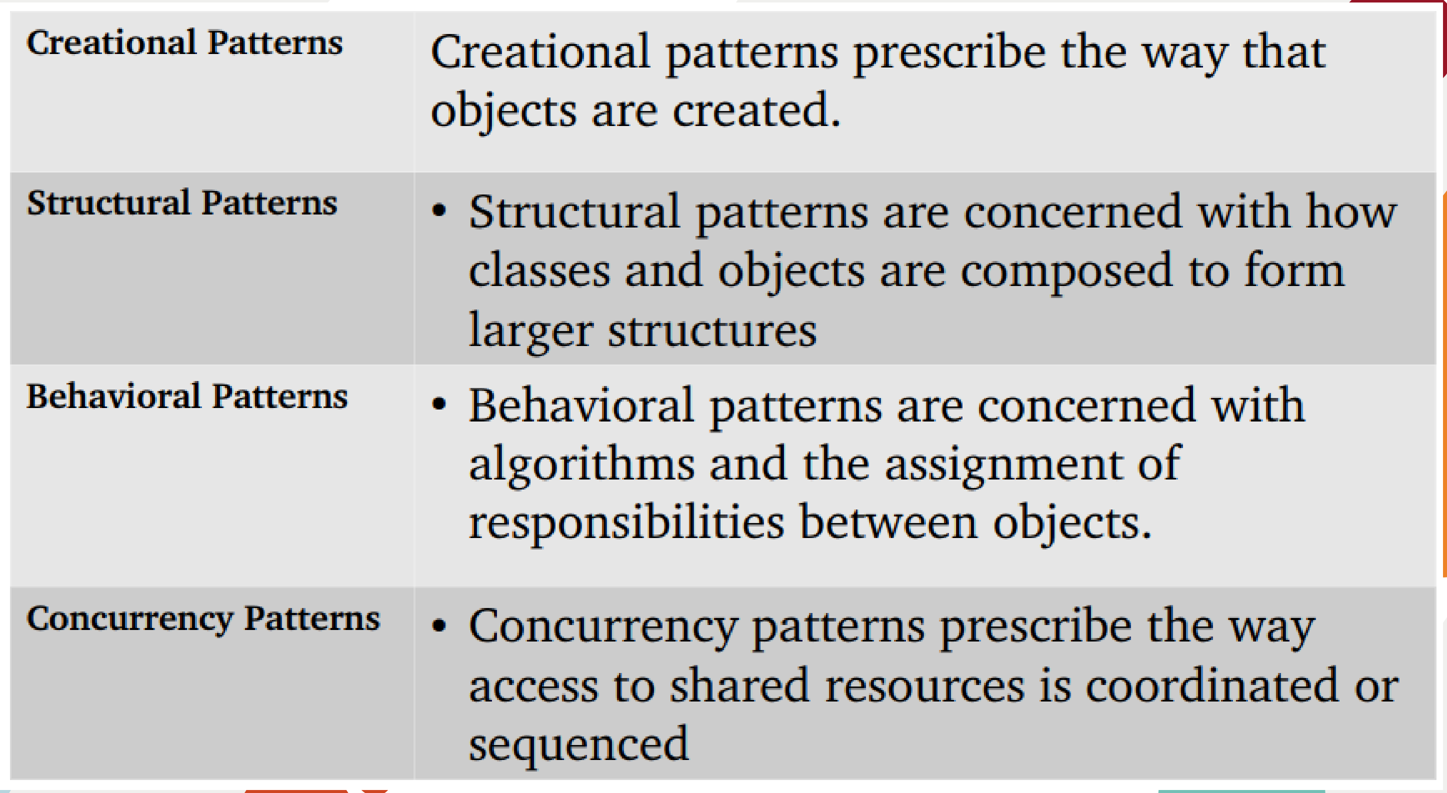
Pola Desain Perangkat Lunak

[Week] 9 – Structural Pattern, Proxy
Prepared by: Tifanny Nabarian

Design Patterns Category



Creational Patterns	Creational patterns prescribe the way that objects are created.
Structural Patterns	• Structural patterns are concerned with how classes and objects are composed to form larger structures
Behavioral Patterns	• Behavioral patterns are concerned with algorithms and the assignment of responsibilities between objects.
Concurrency Patterns	• Concurrency patterns prescribe the way access to shared resources is coordinated or sequenced

Design Patterns Scope

		Purpose		
		Creational	Structural	Behavioral
Scope	Class	• Factory method	• Adapter	• Interpreter • Template method
	Object	• Abstract factory • Builder • Prototype • Singleton	• Adapter • Bridge • Composite • Decorator • Fasad • Flyweight • Proxy	• Chain of responsibility • Command • Iterator • Mediator • Memento • Observer • State • Strategy • Visitor

Structural Pattern

Proxy

Proxy Concept

Definisi GoF

- **Provide** a **surrogate or placeholder for another object** to control access to it. A proxy is basically a **substitute** for an intended object. Access to the original object **is not always** possible due to many factors.

Real World Example

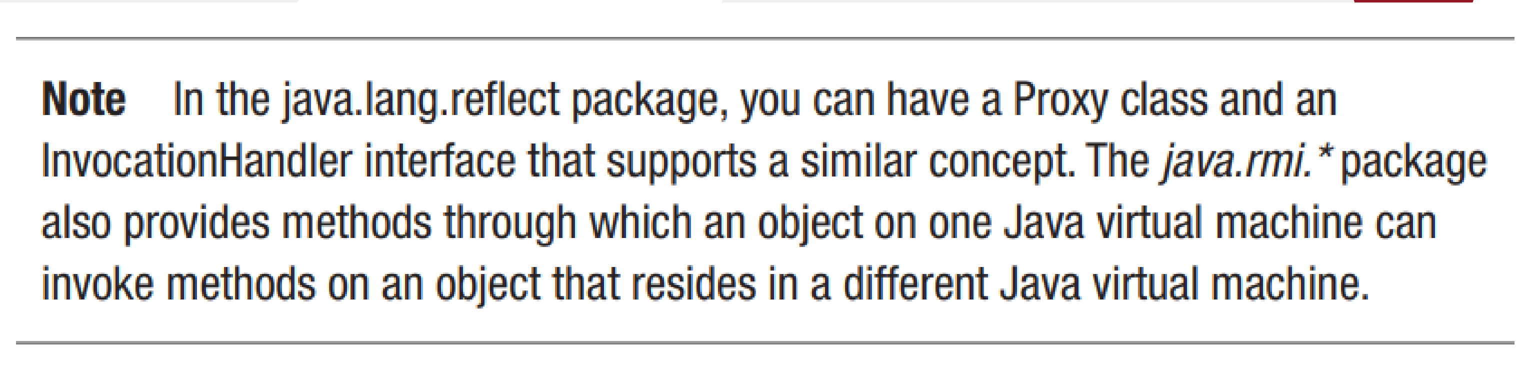
- In a classroom, when a student is **absent**, his best friend may try to mimic his voice during roll call to try to get attendance for his friend

Proxy Concept

Computer World Example

- In the programming world, to create multiple instances of a complex object (heavy object) is **costly**
- So, whenever you are in need, you can create **multiple proxy objects** that point to the original object.
- This mechanism can also **help save your system/application memory**.
- An **ATM can implement this pattern** to hold proxy objects for bank information that may exist on a remote server.

Proxy Example



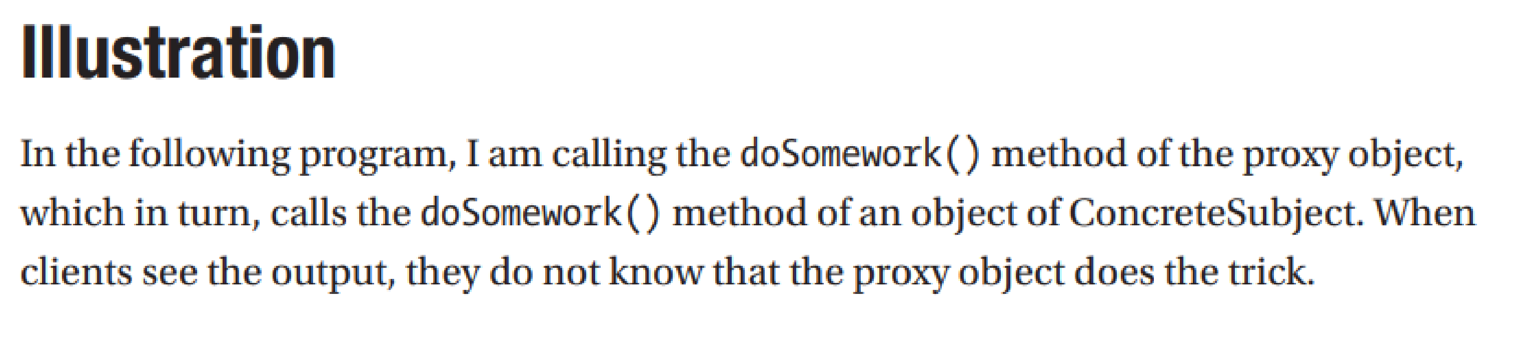
Note In the java.lang.reflect package, you can have a Proxy class and an InvocationHandler interface that supports a similar concept. The *java.rmi.** package also provides methods through which an object on one Java virtual machine can invoke methods on an object that resides in a different Java virtual machine.

Tipe Proxy

- ***Remote proxies***. Hide the actual object that stays in a different address space.
- ***Virtual proxies***. Perform optimization techniques, such as the creation of a heavy object on a demand basis.
- ***Protection proxies***. Deal with different access rights.
- ***Smart reference***. Performs additional housekeeping work when an object is accessed by a client.

Let's Practice!

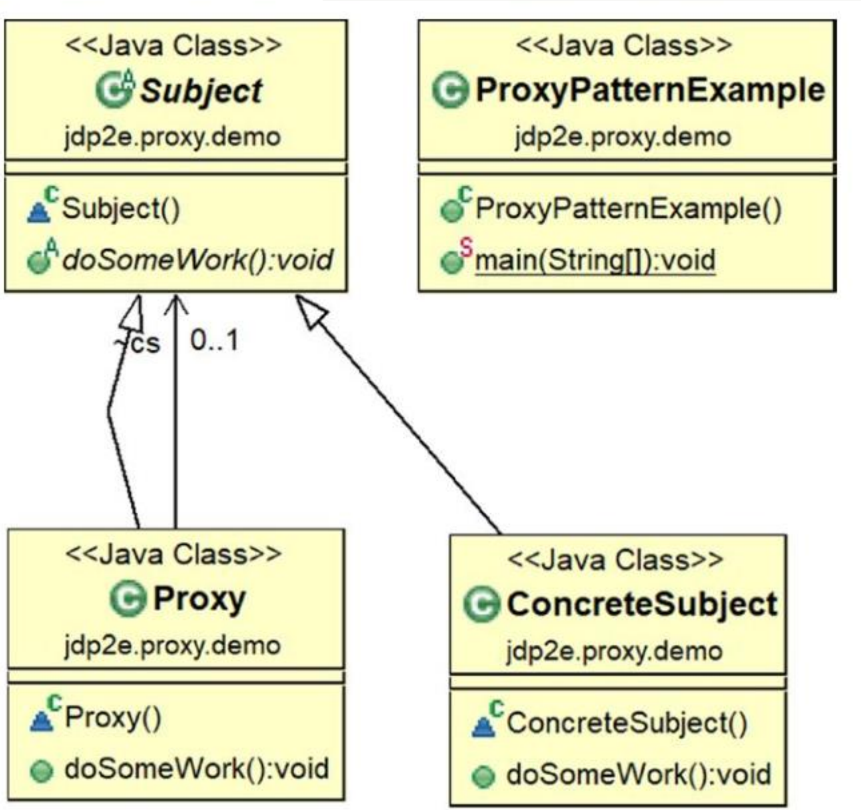
Proxy Example - Illustration



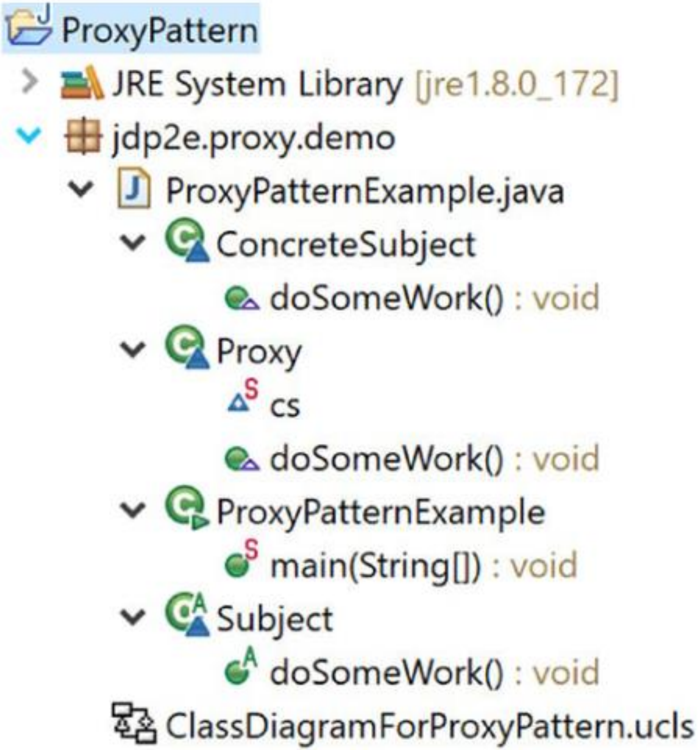
Illustration

In the following program, I am calling the `doSomework()` method of the proxy object, which in turn, calls the `doSomework()` method of an object of ConcreteSubject. When clients see the output, they do not know that the proxy object does the trick.

Class Diagram



Proxy Example (Package Explorer)



```
ProxyPattern
  > JRE System Library [jre1.8.0_172]
  v jdp2e.proxy.demo
      v ProxyPatternExample.java
          v ConcreteSubject
                doSomeWork() : void
          v Proxy
                cs
                doSomeWork() : void
          v ProxyPatternExample
                main(String[]) : void
          v Subject
                doSomeWork() : void
      ClassDiagramForProxyPattern.ucls
```

Silahkan Kerjakan Tugas Praktikkum..

CHARACTER BUILDING CAMPUS

STT - NF

NURUL FIKRI