Software Design Patterns

讀書會#1

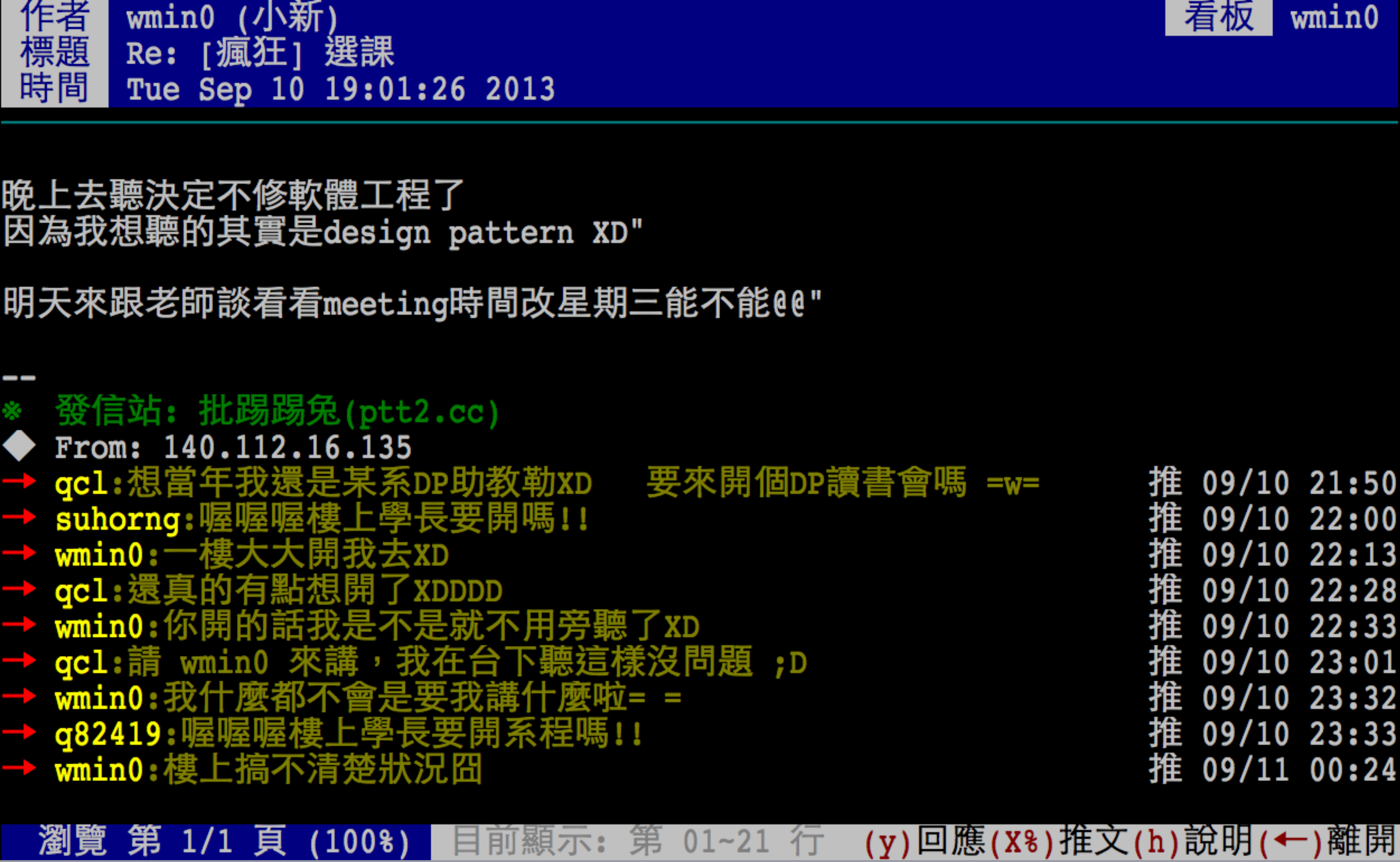
2013.09.25

qcl

- 臺大資訊
  - 自然語言處理實驗室
  - 碩士班二年級
- 土木所「設計模式與軟體開發」批改郎(100-2)
  - 所謂「批改郎」就是改作業給分數的啦

讀書會#1

- Introduction to 讀書會
- Design Patterns
- UML
  - Class diagram



```
作者  wmin0 (小新)                                                看板  wmin0
標題  Re: [瘋狂] 選課
時間  Tue Sep 10 19:01:26 2013

晚上去聽決定不修軟體工程了
因為我想聽的其實是design pattern XD"

明天來跟老師談看看meeting時間改星期三能不能@@"

--
※ 發信站: 批踢踢兔(ptt2.cc)
◆ From: 140.112.16.135
→ qcl:想當年我還是某系DP助教勒XD   要來開個DP讀書會嗎 =w=      推 09/10 21:50
→ suhorng:喔喔喔樓上學長要開嗎!!                                  推 09/10 22:00
→ wmin0:一樓大大開我去XD                                          推 09/10 22:13
→ qcl:還真的有點想開了XDDDD                                       推 09/10 22:28
→ wmin0:你開的話我是不是就不用旁聽了XD                            推 09/10 22:33
→ qcl:請 wmin0 來講，我在台下聽這樣沒問題 ;D                      推 09/10 23:01
→ wmin0:我什麼都不會是要我講什麼啦= =                             推 09/10 23:32
→ q82419:喔喔喔樓上學長要開系程嗎!!                               推 09/10 23:33
→ wmin0:樓上搞不清楚狀況囧                                        推 09/11 00:24
 瀏覽 第 1/1 頁 (100%)  目前顯示: 第 01~21 行  (y)回應(X%)推文(h)說明(←)離開
```

故事是這樣開始的

#1IBlm9PX (wmin0) [ptt2.cc] Re: [瘋狂] 選課

DP讀書會的目的

- 認識Software Design Patterns
  - 聽/說/讀/寫（？）
- Divide & Conquer
  - 每個人分配一點內容
  - 不必真的把書都翻過一次
  - 卻可以知道整本書的內容
- ~~自虐~~

DP讀書會的時間

- 每週一次
  - 原則上希望
- 逢期中、期末考週暫停
  - 好好溫書？
- 週三晚上
  - 根據調查, 這是最多人共同有空的時段
- 暗時七點
  - ~~因為 qcl 六點下班，七點開始才不會遲到~~

DP讀書會的地點

- 遊牧民族
- 游擊隊
- 資訊系館三樓研討室
  - ~~因為 qcl 的實驗室在三樓~~
- 319/340/3……..
  - 順時鐘

DP讀書會的書

- Design Patterns: Elements of Reusable Object-Oriented Software
  - GoF (四人幫) Design Patterns

- Head First Design Patterns
- Efective Java
- ...

進行方式

- qcl 參考「軟體設計模式」(100-1)、「設計模式與軟體開發」(98-2)以及眾神意見，每週分配2-3個patterns包成一組
- 請諸位大大認領一組回家培養感情
- 下一週起，每週請一位大大分享一組

- 每週將開一個issue於github上，請自行assign

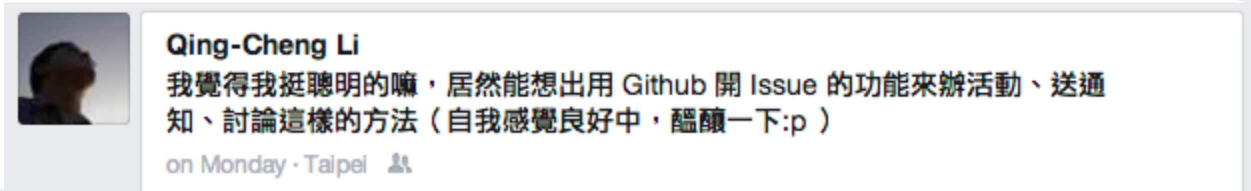
主題分組(暫定)

- UML/Adapter
- Composite/Decorator
- Bridge/Facade
- Flyweight/Proxy
- Builder
- Abstract Factory/Factory Method
- Singleton/Prorotype
- Template Method/ Strategy
- State/Interpreter
- Observer/Mediator
- Iterator/ Chain of Responsibility
- Command/Memento
- Visitor

cwahbong表示:根本亂分

DP讀書會的討論

- Github
  - https://github.com/qcl/dp-reading-club
- Watch this project!
  - email notification



Qing-Cheng Li

我覺得我挺聰明的嘛，居然能想出用 Github 開 Issue 的功能來辦活動、送通知、討論這樣的方法（自我感覺良好中，醞釀一下:p ）

on Monday · Taipei

Questions?

Design Patterns

What is it?

戰技/戰術/戰略

Object-Oriented Programming/Design

Design Issues

- UML emphasize design notations
  - Fine for specification, documentation
- But OOD is more than just drawing
  - Good draftsmen != good design
- Good OO designers rely on losts of experience
  - At least as important as syntax
- Most powerful reuse is design reuse
  - Match problem to design experience

這張是從臺大土木系陳俊杉教授的 DP投影片抄來的 :(

Design Issues

- Designing OO software is hard and desiging a reusable OO software is even harder
- A reusable and flexible design is difficult to get right at the first time
- Yet experienced OO designers do make good design!

這張是從臺大土木系陳俊杉教授的 DP投影片抄來的 :(

Recurring Design Structures

- OO system exhibit recurring structures that promote
  - abstraction
  - flexibility
  - modularity
  - elegance
- Therein lies valuable design knowledge
- Problem
  - Capture, communicate, apply this knowledge

這張是從臺大土木系陳俊杉教授的 DP投影片抄來的 :(

A Design Pattern

- Abstracts a recurring design structure
- Names and specifies the design structure explicity
- Distills design experience

這張是從臺大土木系陳俊杉教授的 DP投影片抄來的 :(

What is a Pattern?

- Comes from the architect Christopher Alexander
  - He studied ways to improve the porcess of designing buildings and urban areas
- "Each pattern is a three-part rule, which expresses a relation between a certain **context**, a **problem** and a **solution**."

- Pattern:

  A solution to a problem in a context

這張是從臺大土木系陳俊杉教授的 DP投影片抄來的 :(

Patterns

- Each pattern describes a **problem** which occurs over and over again in our environment, and then decribces the core of the **solution** to that problem, in such a way that you can **use this solution a million times over, without ever doing it the same way twice**.

這張是從臺大土木系陳俊杉教授的 DP投影片抄來的 :(

Basic parts of a Pattern

1. Name
2. Problem
3. Solution
4. Consequences & trade-offs of application

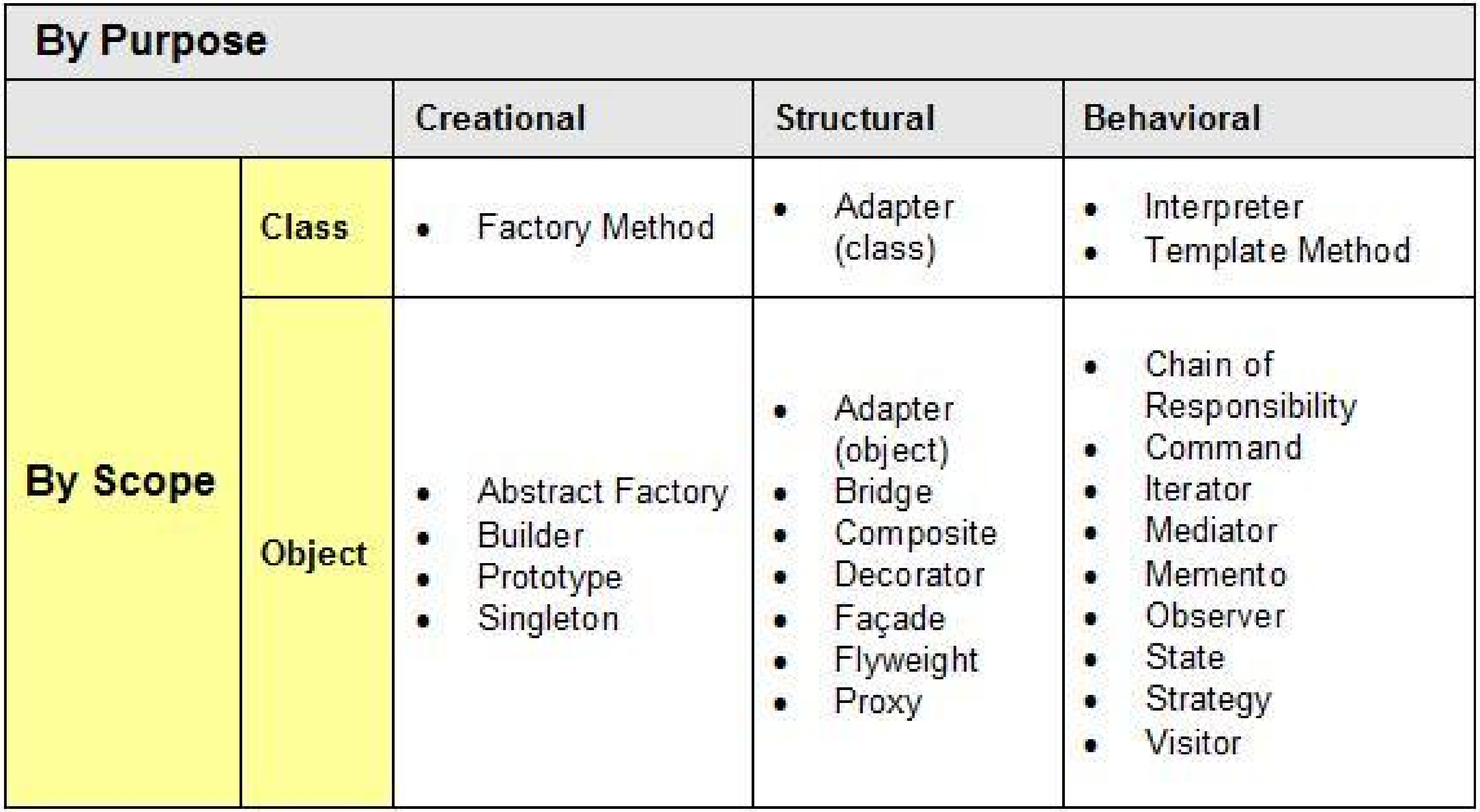
這張是從臺大土木系陳俊杉教授的 DP投影片抄來的 :(

Classification (GoF)

- Purpose - what a pattern does
  - Creational
  - Structure
  - Behavioral
- Scope - what the pattern applies to
  - Class
    - Focus on the relationships between classes & their subclasses
  - Object
    - Focus on the relationships between objects

這張是從臺大土木系陳俊杉教授的 DP投影片抄來的 :(

Classification (GoF)



By Purpose				
		Creational	Structural	Behavioral
By Scope	Class	• Factory Method	• Adapter (class)	• Interpreter • Template Method
	Object	• Abstract Factory • Builder • Prototype • Singleton	• Adapter (object) • Bridge • Composite • Decorator • Façade • Flyweight • Proxy	• Chain of Responsibility • Command • Iterator • Mediator • Memento • Observer • State • Strategy • Visitor

這張是從臺大土木系陳俊杉教授的 DP投影片抄來的 :(

DP Template (GoF)

Name scope purpose

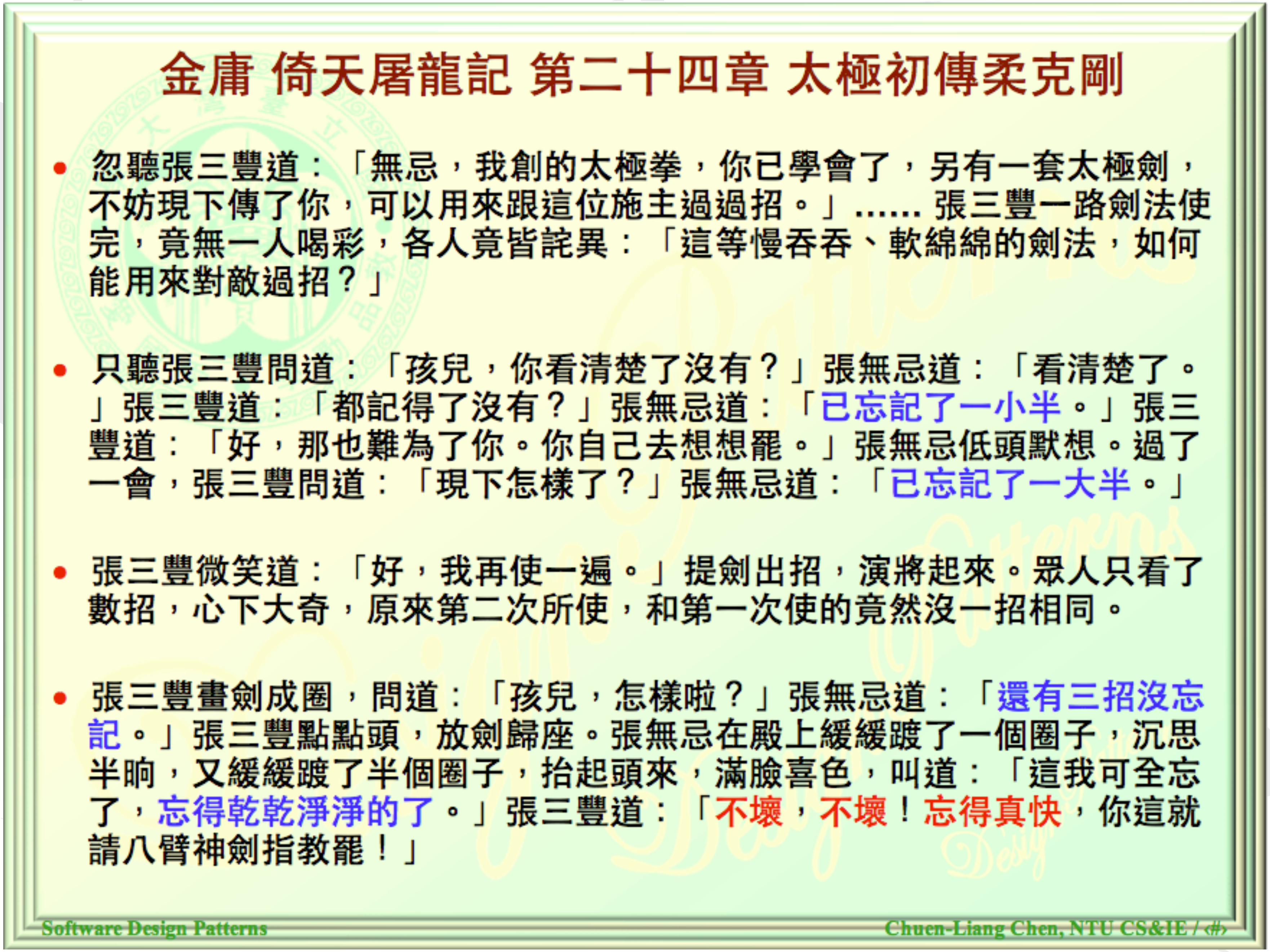
- Intent
- Also knows as
- Motivation
- Applicability
- Strcutre
- Participants

這張是從臺大土木系陳俊杉教授的 DP投影片抄來的 :(

DP Template (GoF)

- Collaborations
- Consequences
- Implementation
- Sample code
- Known uses
- Related patterns

這張是從臺大土木系陳俊杉教授的 DP投影片抄來的 :(



金庸 倚天屠龍記 第二十四章 太極初傳柔克剛

- 忽聽張三豐道：「無忌，我創的太極拳，你已學會了，另有一套太極劍，不妨現下傳了你，可以用來跟這位施主過過招。」…… 張三豐一路劍法使完，竟無一人喝彩，各人竟皆詫異：「這等慢吞吞、軟綿綿的劍法，如何能用來對敵過招？」
- 只聽張三豐問道：「孩兒，你看清楚了沒有？」張無忌道：「看清楚了。」張三豐道：「都記得了沒有？」張無忌道：「已忘記了一小半。」張三豐道：「好，那也難為了你。你自己去想想罷。」張無忌低頭默想。過了一會，張三豐問道：「現下怎樣了？」張無忌道：「已忘記了一大半。」
- 張三豐微笑道：「好，我再使一遍。」提劍出招，演將起來。眾人只看了數招，心下大奇，原來第二次所使，和第一次使的竟然沒一招相同。
- 張三豐畫劍成圈，問道：「孩兒，怎樣啦？」張無忌道：「還有三招沒忘記。」張三豐點點頭，放劍歸座。張無忌在殿上緩緩踱了一個圈子，沉思半晌，又緩緩踱了半個圈子，抬起頭來，滿臉喜色，叫道：「這我可全忘了，忘得乾乾淨淨的了。」張三豐道：「不壞，不壞！忘得真快，你這就請八臂神劍指教罷！」

這張是從臺大資訊系陳俊良教授的 DP投影片抄來的 :(

先講故事
再看程式

一點心得 by qcl

UML

Unified Modeling Language

What is UML

- UML
  - Unified Modeling Language
- A graphical language for
  - Visualizing
  - Specifying
  - Constructing
  - Documenting

  a software system.

這張是從臺大土木系陳俊杉教授的 DP投影片抄來的 :(

Why use UML?

- Communication with
  - yourself
  - your parterns
  - others

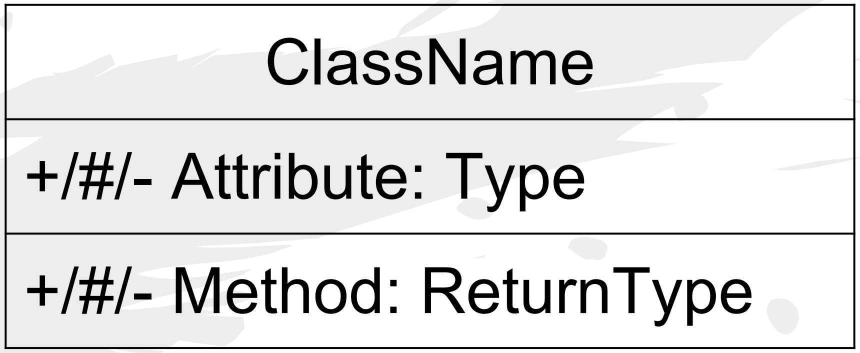
這張是從臺大土木系陳俊杉教授的 DP投影片抄來的 :(

Building blocks of UML

- Things
  - Modeling elements
- Relationships
  - Specify how two or more things are senantically related
- Diagrams
  - Views into UML models

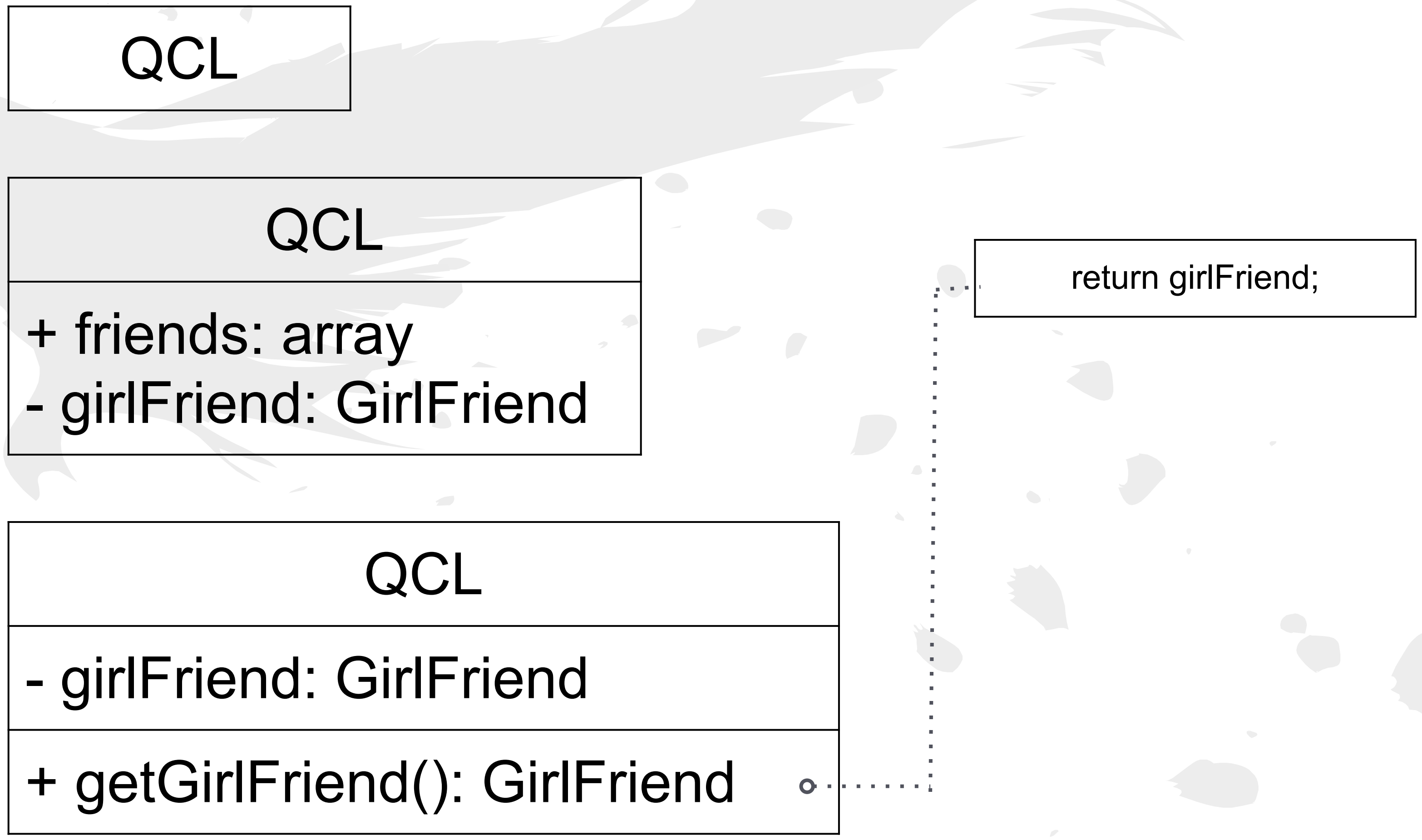
這張是從臺大土木系陳俊杉教授的 DP投影片抄來的 :(

Class Diagram



- +: Public
- #: Protected
- -: Private

Class Diagram

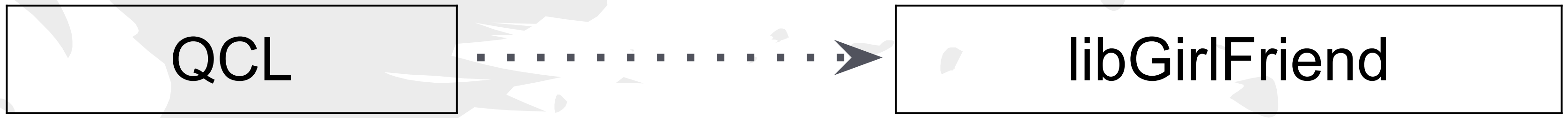


Relationships

- Dependency
- Aggregation
- Composition
- Generalization

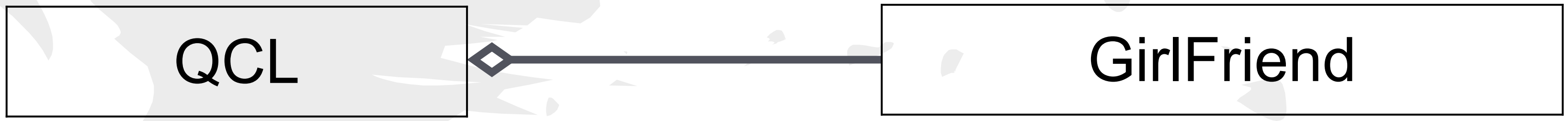
Dependency

- 依賴



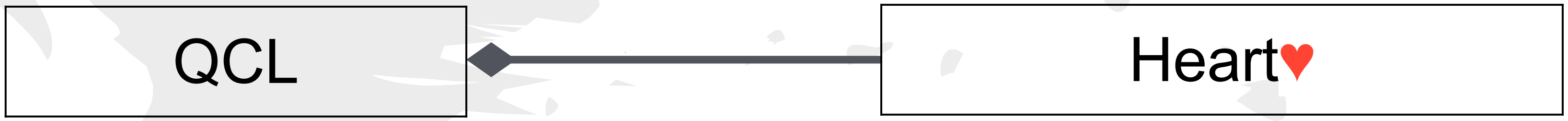
Aggregation

- has-a
- >=0



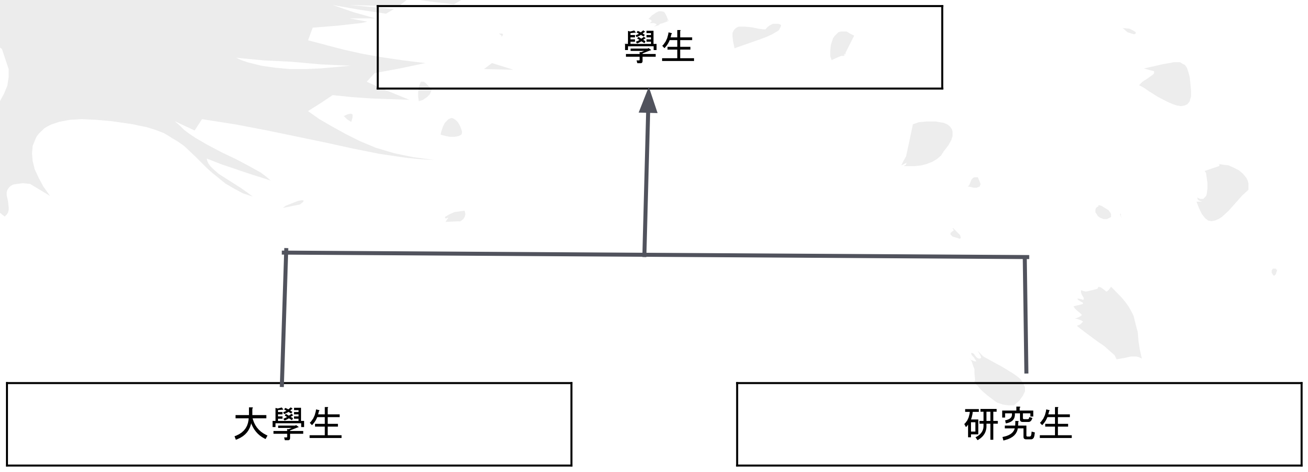
Composition

- has-a (strong)
- >0

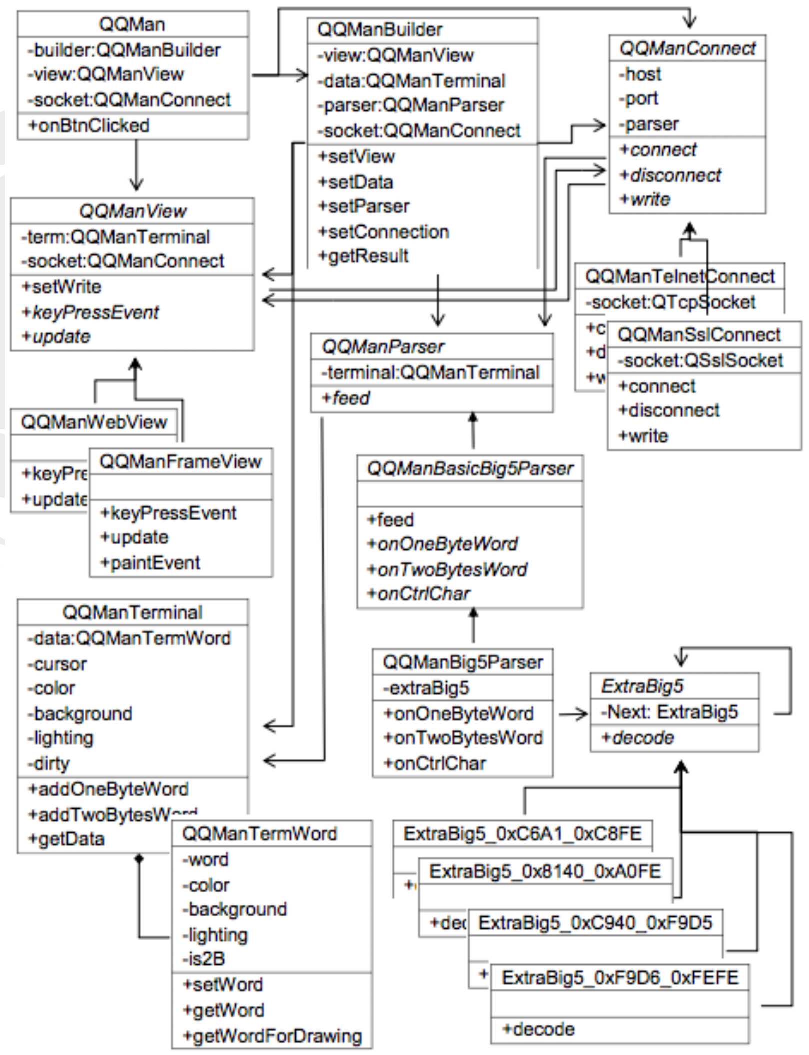


Generalization

- 繼承
- is-a



Class Diagram (Example)



QQMan by 李卿澄 & 張偉哲

Questions?

Thanks!

qcl
qingcheng.li @ qcl.tw
qc.linux @ gmail.com

@qcl (plurk/ptt/ptt2/github)
@qingchengli (twitter)