

Object-Otiented

Chen Yuheng

Software Engineering

systematic, disciplined, qualifiable approach
to development, operation, maintenance of software

- making things
- useful
- fashioning objects
- always learning
- tractable

PROS

- must perfect
- not by ourselves
- tedious
- always testing
- outlasting

CONS

RCS: Basics & Git

RCS: Revision Control Software

Revision: also known as version,

a state of a piece of info at a specific time
as a result of some change to it

- track history and evolution
- easier to collaborate
- recover from mistakes
- multiple versions support

Repository: database of the history being tracked

Initialization: `git init`

Stage: prepare files to commit, ~~otherwise ignore~~

Commit: saving current state to history

Ignore: `.gitignore`: add the file name

diff: see changes

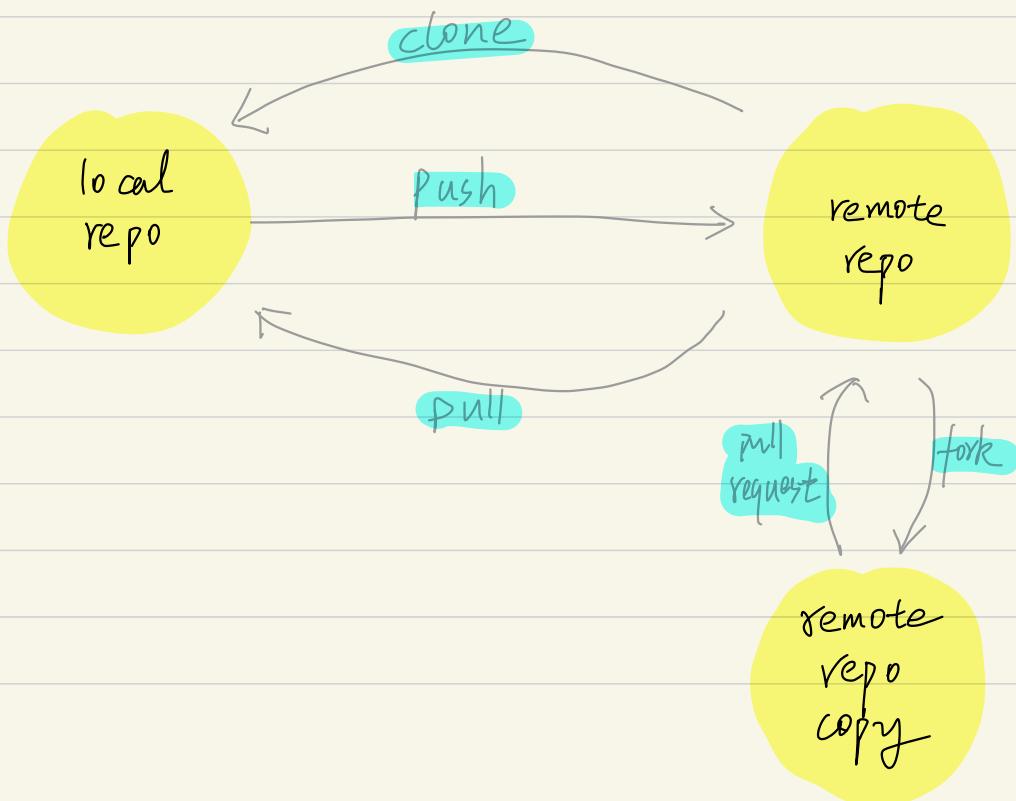
checkout: restore a historical state 查看历史状态。

HEAD label for current check out commit

tag: git tag -a v1.0

stash: 存放，暂存某些更改于 stash 中，^{然后可以}撤销当前版本的更改做别的事

REMOTE COMMUNICATION



IDEs

Integrated Development Environment

- source code editor
- compiler / interpreter
- debugger
- other tools

OOP

object-Oriented Programming 面向对象

Programming language: → datas
 → operations on datas

Programming Paradigm	Examples
OO	Java, Python, Js
Procedural	C
Functional	F#, Haskell, Scala
Logic	Prolog

- OOP views the world as a network of interacting objects
- Every object has an interface & an implementation
- Objects interact by sending messages

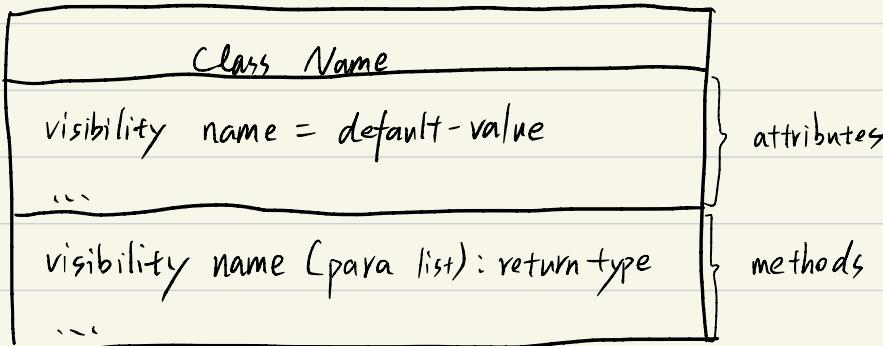
A class contains instructions to creating an object
Object is abstraction from the real world

Encapsulation protects implementation from unintended access

- package to a self-contained unit
- hiding informations

UML: Basic Structure Diagram

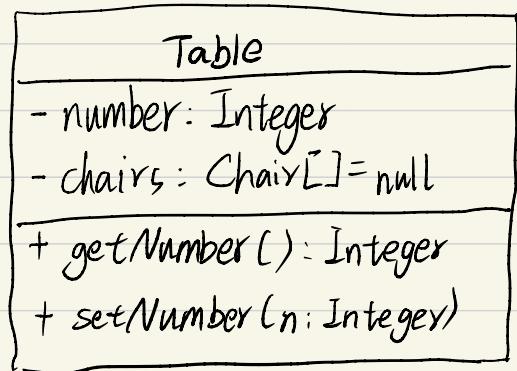
Class Diagram



visibility:

- + public
- private
- # protected
- ~ package private

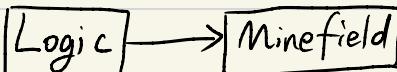
An example
→



Association



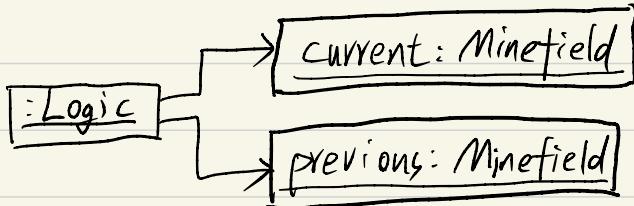
Navigability: awareness



逻辑知道雷区



人与狗相互知道

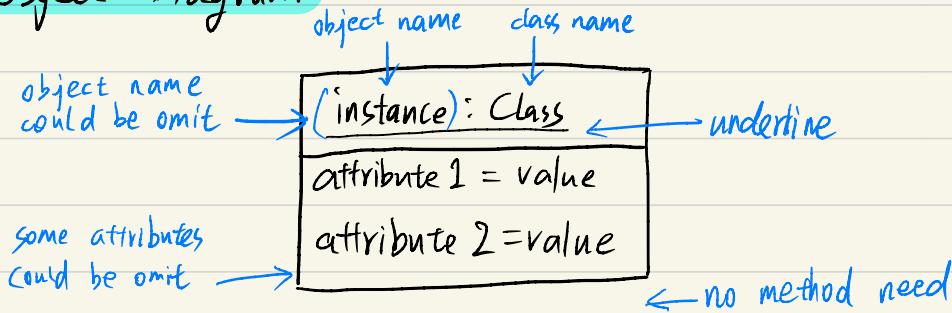


Not only for classes
also for objects

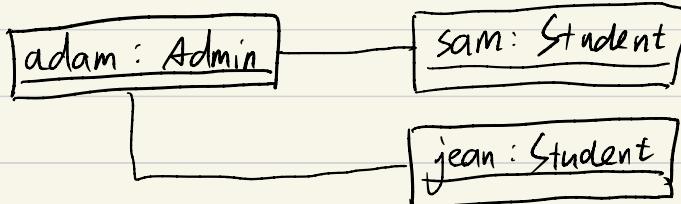
Labels 注解用



Object Diagram



Associations



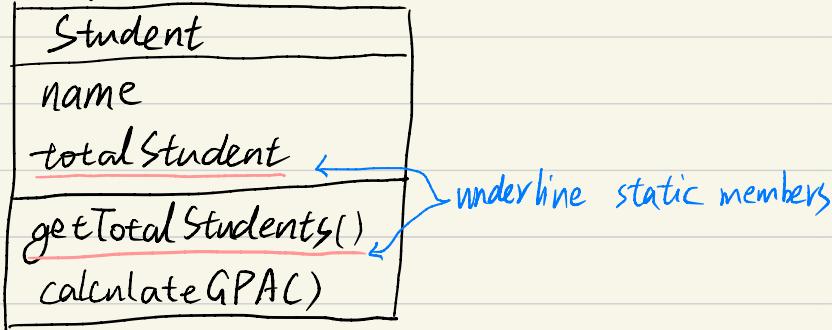
Java

Class-Level Members

Also: static ...

- class-level attributes
- class-level methods

In class diagram, underline them: (also example)



- use static final to define constant

Enumerations

enum

Varargs

methodName (Type ... name)

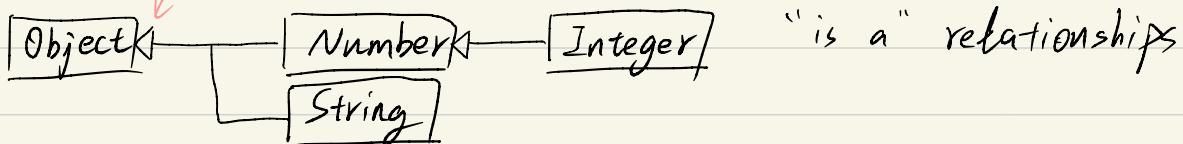
OOP: Inheritance 繼承

Base Class : parent class, super class



Derived Class : child class, sub class, extended class

triangle



Multiple inheritance is available in C++, Python,
but not supported by Java