Michael.

Design OO

design pattern.

ANDROID

Idiom, domain (the world custom live in).

└ init(), open().

Lucene?

design pattern: Solution for what condition, (related pattern) consequence.

part of design.

~~definition~~ definition: document of a design ~~pattern~~ decision.

Singleton pattern.

They're not: resume

Global variables are bad.

access issue, public/private.

We temped to use global variables.

Xiaosong.

Static Factory methods.

Bloch's Effective Java.

dri

Readings: Chap 1.

Small. Do one thing. Strong cohesion

reading materials: refactoring.

side effect: modify the ~~var~~ var not pass into parameter.

more calls, less efficiency.

Avoid output arguments: e.g. String name = "~"; { name.toUpperCase(x)
name = name.toUpperCase;

Design: { UML diagrams
Documentations
Data structures.
Database layouts. } — Design representations.

Efficiency
Requirement } — Design Qualities

~~Main~~

cohension: focus on one thing.

Coupling: relationship between 2 modules.

two connections: static / dynamic, "run" interacting with each other. "runtime"

defined in one is another's member or inherient.

most readbility. flexibility. maintance more important

Encapsulating operations on data. Why is this on of DRY principle

promise what.

Level of perspective in OO.

Coneeptual:

Domain-level, problem-level. -- not yet considering solutions.

Specification

Solution-level, but an abstract view, interface, not internal implementation

Implementation.

php error message?

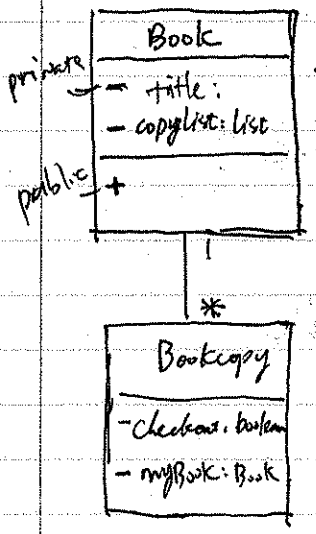
Levels of perspective in OO

Conceptual specification & Implementation.

Domain-level.
problem-level
→ not considering solution.
→ objects responsibility.

Solution-level
but can abstract view
Interfaces, not internal implementation.

online judge



the graph.
define in ~~Book~~: problem-domain
define in ~~Book~~: Specification level

So what's an objection.
*Depends on what level you're using. where you're at in development
· Conceptual:
· Specification:
· Implementation:

· Designers work at Specification (mostly).
· Analysts works at Conceptual level (mostly)

direction relationship.
△
not define in Conceptual level.

DRY — don't repeat yourself.

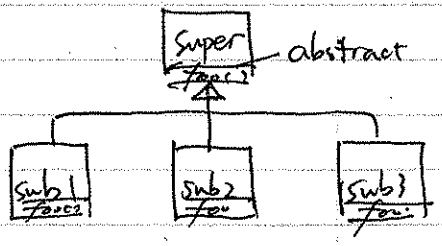
Abstract Class vs Interface. (both are type).

mixed {implem. / impl.
Can have static fields.
can have coded
at least one
unimpl

unimplement method.
Signature.

Static, ~~public~~ final.
one reference. Can change.

Shared behaviours — share
other unique behaviours.



Sandi's volunteer.

interface.

computer & clock. ↻ time., not abstract of each other.

IS-A "dash."

polymorphism ? runtime △. type's method be invoked.

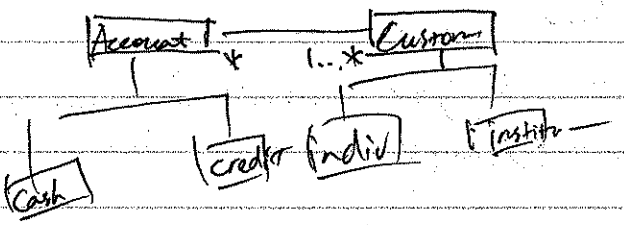
by high level abstract class.

Inheritance Diamonds.

can instantiate · concrete

Seperate Abstractions

Secure.?



flexible

IEEE Mec 205 5pm

$f_1 + f_{2D}$

$f_1 - f_{2D} = f_{2I}$

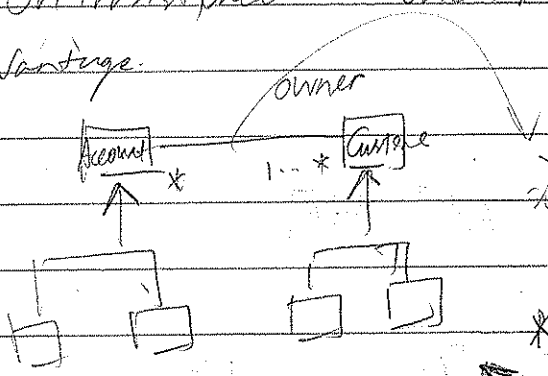
Composition / Aggregation.

HAS-A

PART-OF

On instance ↔ one class

advantage.



1..* accounts

-0.. accounts

diagram.

capture relationship

3 perspectives not the 2

Readings chapter 2 in class textbook

class diagrams developed in

a series of detail

Object domain-level, high level structure things

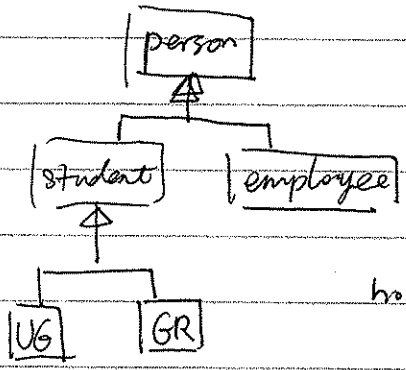
class diagram

two class ↔ association

dependency

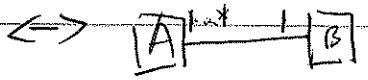
multiplicity. it makes no sense that "book is a copy of copy"

Generalization and inheritance.



multiplicity?

how many



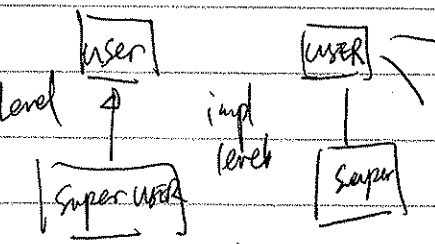
nothing about object

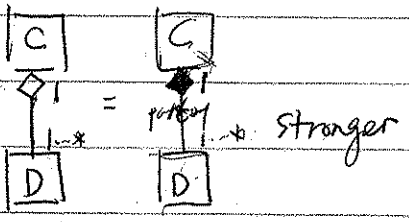
rules defining thing

Generalization <> Inheritance

pattern? what is pattern.

part of





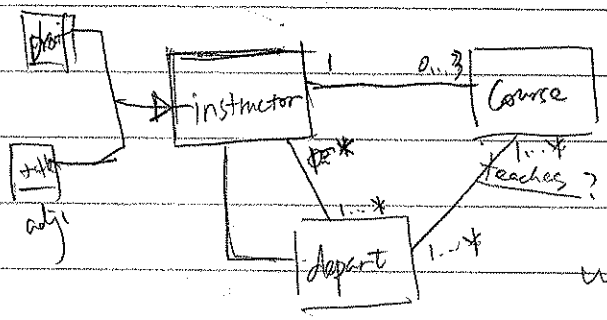
Stronger

Aggre — Compos

OCL

Compos

Bubble spinner

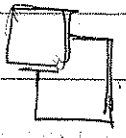


put constrains on line

white where

5.4.3.6

unidirection word



object diagram

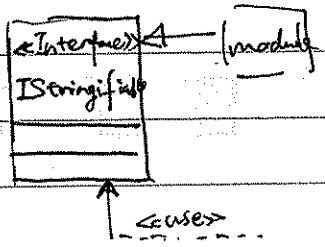
really not what looks like now

class diagram artifact

Tony

stereotypes

guillemet = <<>>



david matthew

~~Cont~~ Constraints → text goes along with diagram.

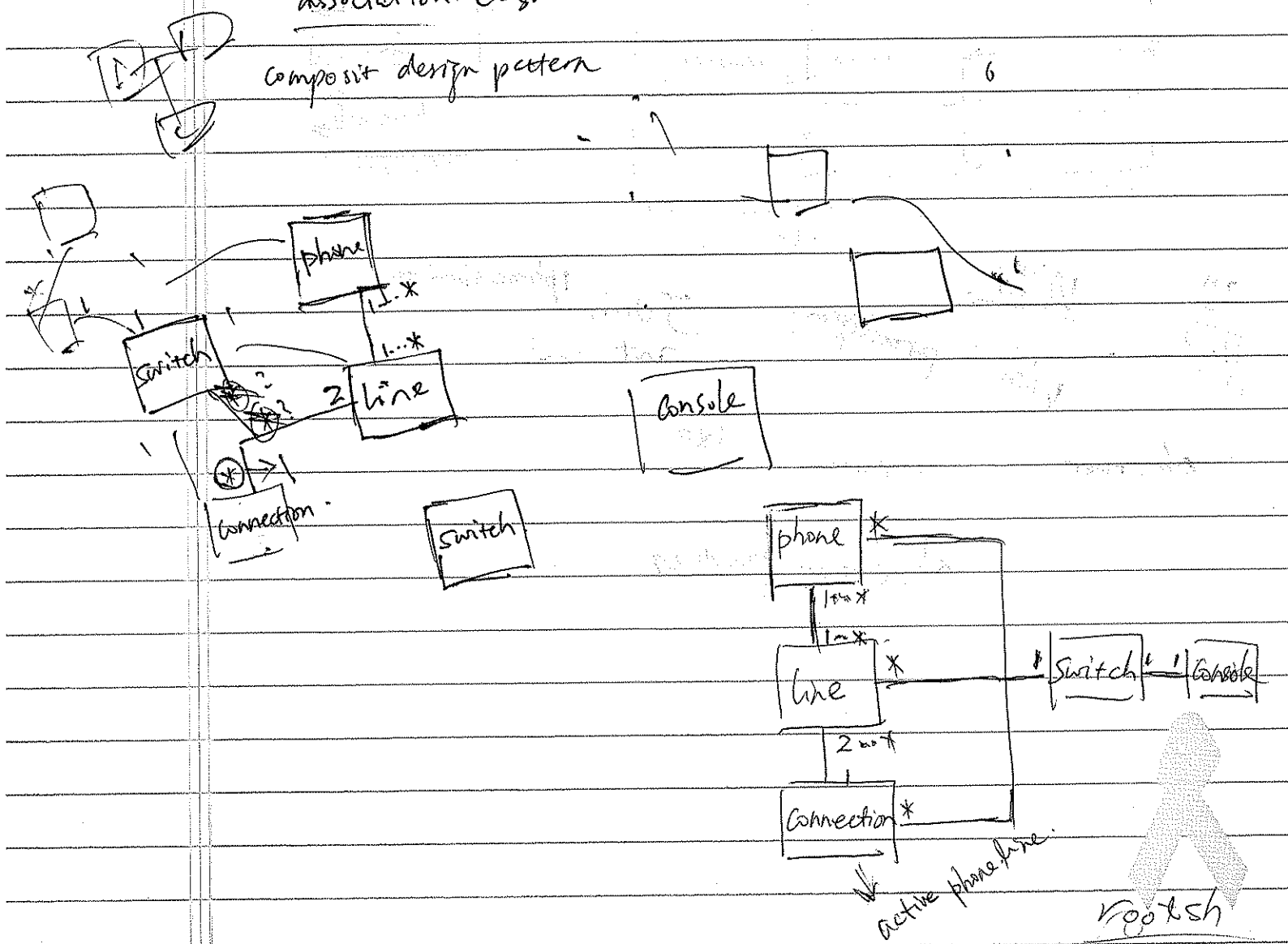
use sparingly

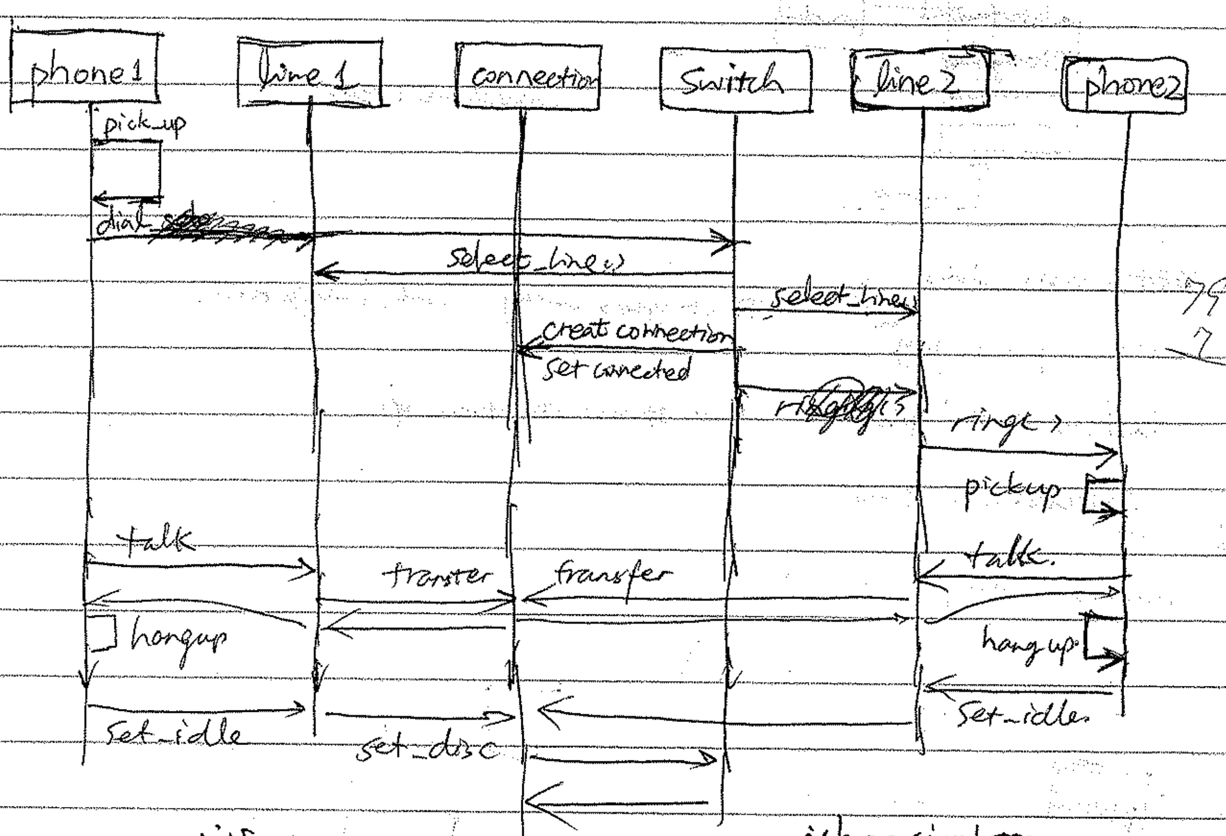
slash → not implement. dup other relationship

association class

composit design pattern

6





2.75 / 1.7 / 17.95 / 17'10

VISIO.

visual paradigm.

iphone simulator

JUnit

ant. xml.

GUI

observers observes.

observer pattern