

Reflection

2:56 - 3:08

dk. of: 8
IUser - SOL User

Why required? -> To absorb common code of deriving Classes

→ If fn is abstract, will the containing class be necessarily abstract?

→ If class is abstract, will it necessarily have abstract functions?

1. Abstract class

2. Abstract fn.

BL IUser obj =
obj.add

Abstract Class

IUser add User

SOL DAL

USER DAL
add User

MyDB DAL

USER DAL
add User

Least code chngs

DAL Factory

Shared Lib

IUserDAL

BAL

IUserDAL obj =
obj = DALFactory.Get UserDAL

SOL DAL

SOL DAL: UserDAL

MyDB DAL

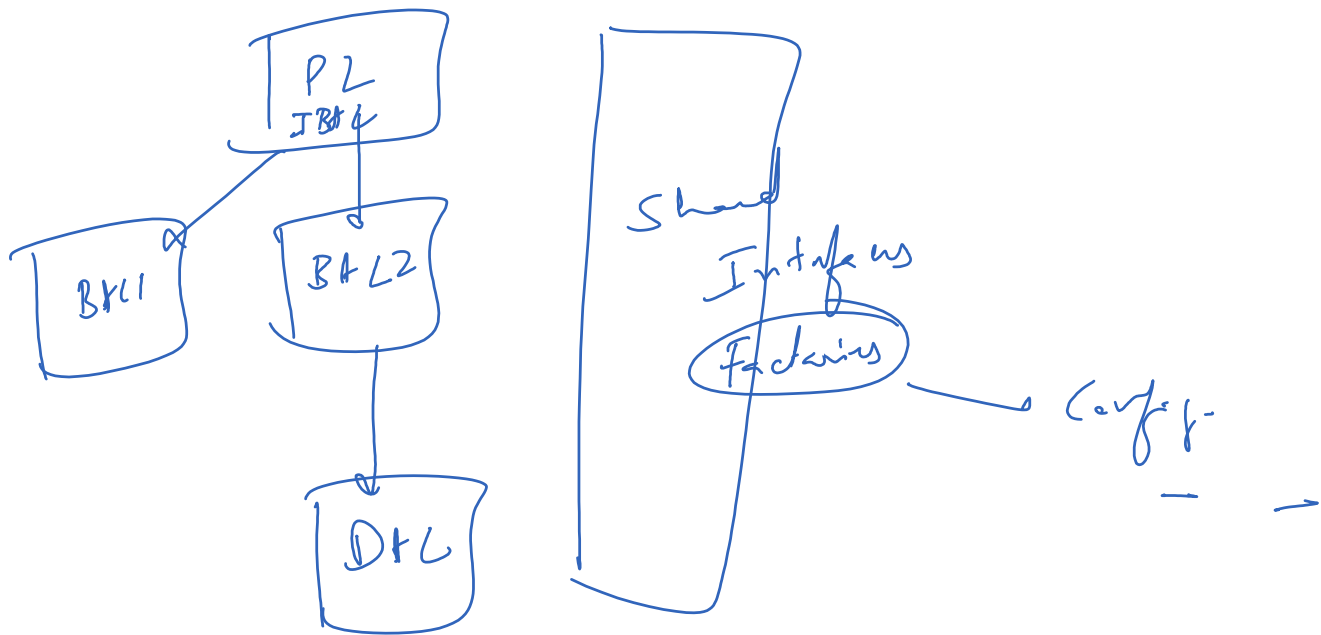
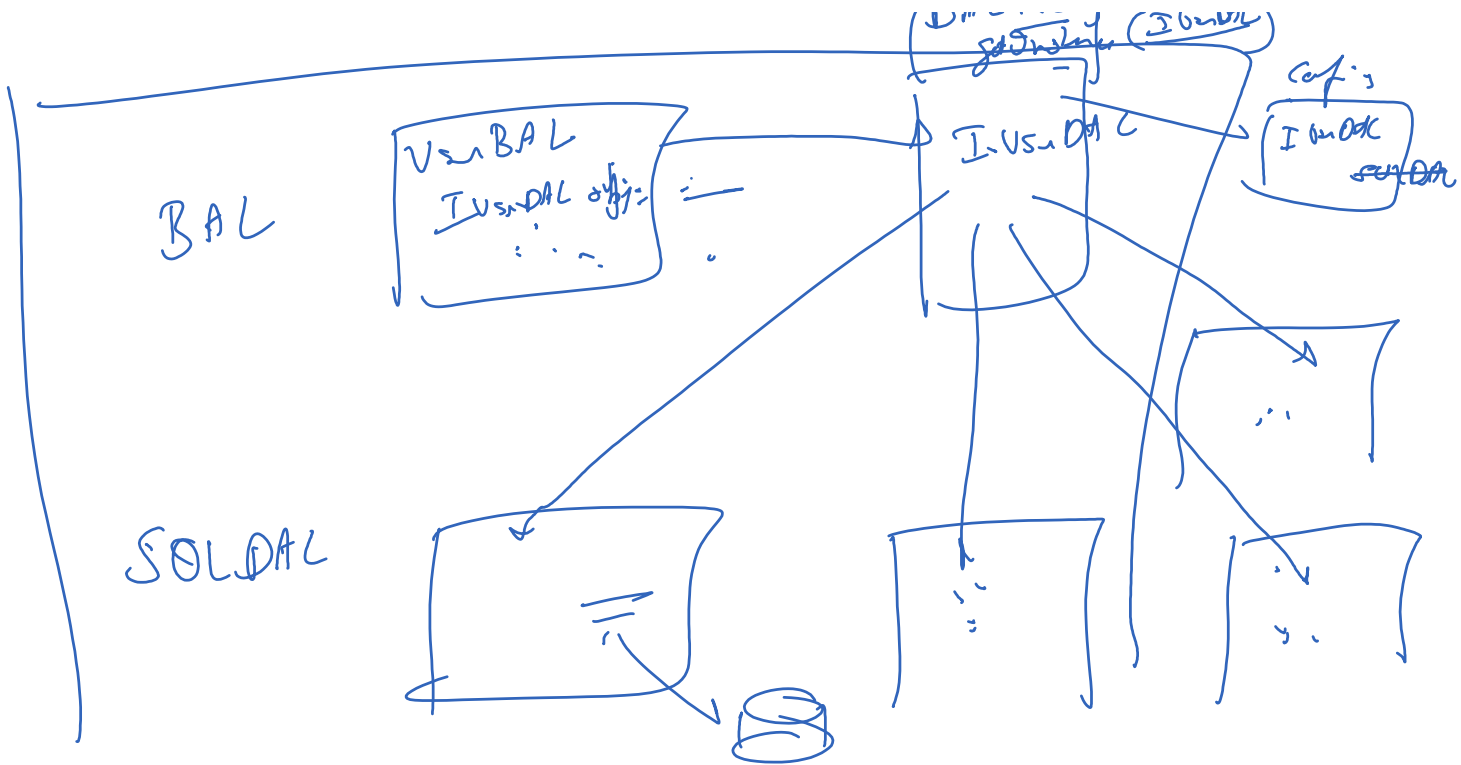
MyDB DAL: IUserDAL

IUserDAL
Abstract = ~~Concrete~~
MyDB DAL

Server

DAL Factory (IUserDAL)

Conf: 3



Interface IEmployee P
 (x data members) → instance
 void work();
 void makeAttendance();

5

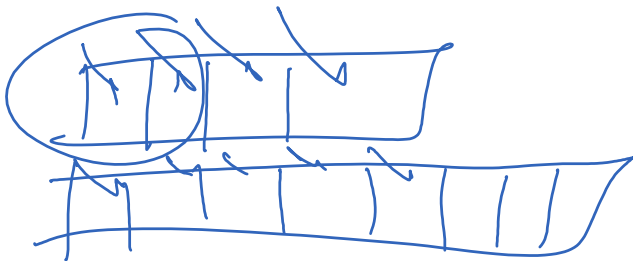
Interface

List <T>

ArrayList <T>

LinkedList <T>

for add remove
get set



I_1
 \uparrow
 I_2

$f_2()$

$4:32 \rightarrow 4:42$

I_3

$f_3()$

$C: I_2, I_3$

$f_1()$ $f_2()$ $f_3()$

I

$d1$

$f_m()$

I $d1 = \text{now } C()$

$o.d1$

$C: 9$

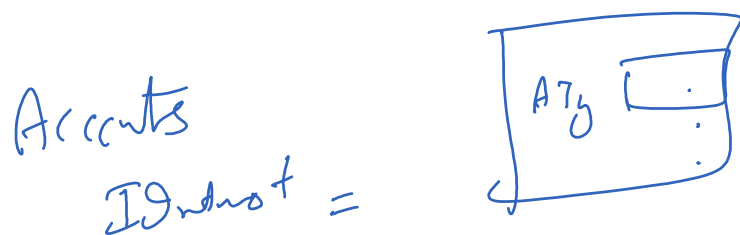
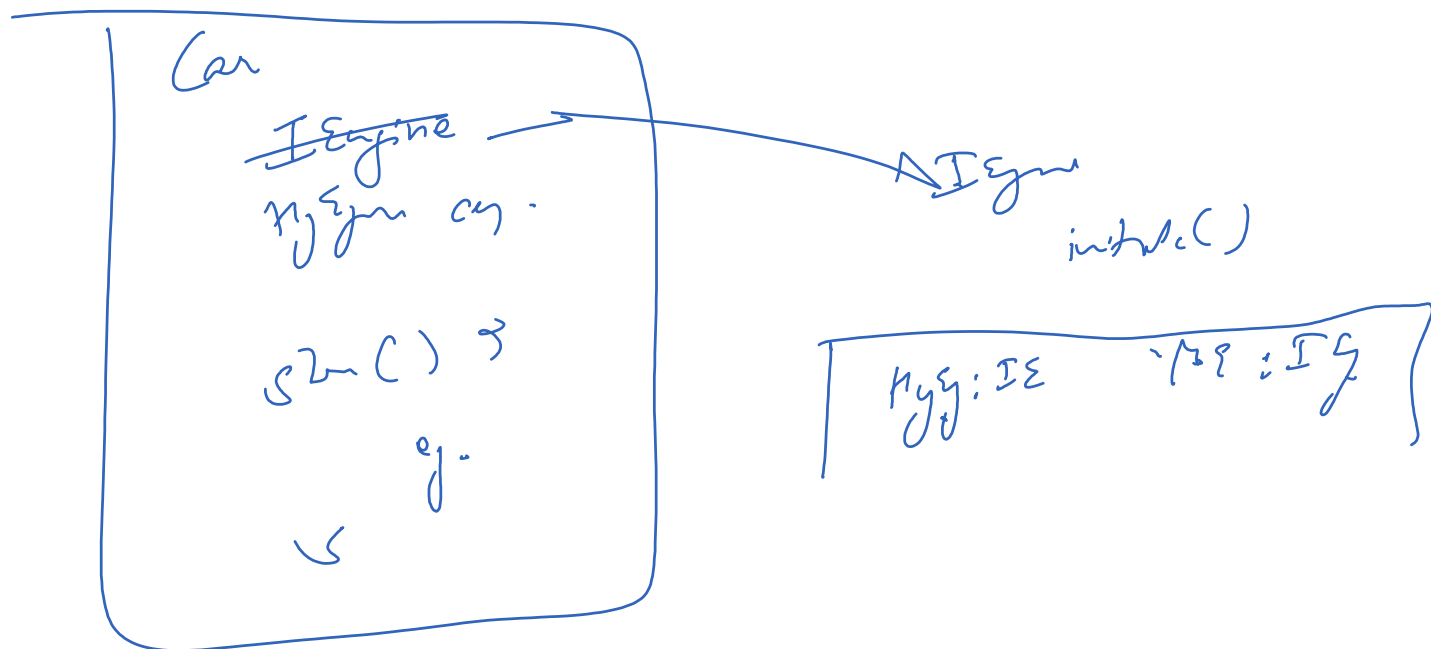
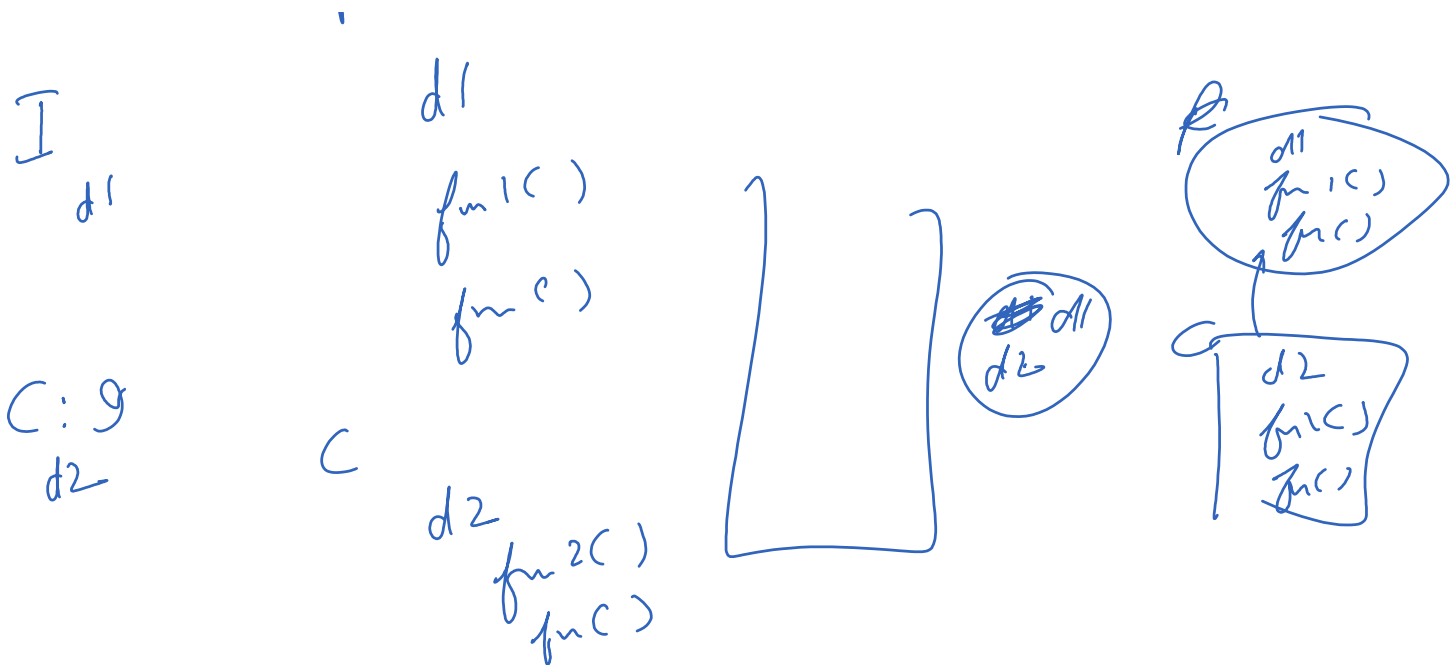
$f_m()$ \uparrow

\rightarrow

P

$d1$

$d1$



(obj. calculus)

IT calculus

SJ

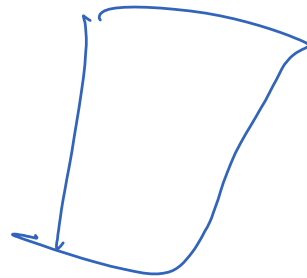
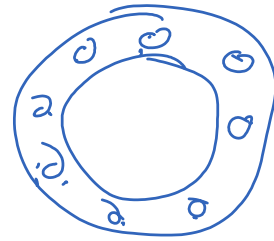
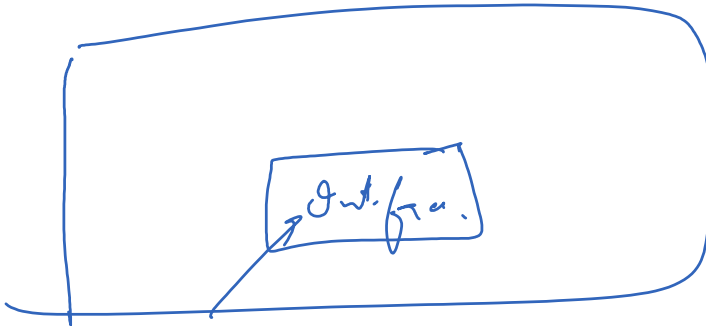
CG

RJ

—

—

—



Modularity

.dll

flexibility

security

IT

|||||

n Equ: IT

|||||