31/01/23.　　　Software Testing.　　Dono.

Platform - hardware:

→ Java.

. Architecture Independent.

byte code.

```
┌──────────┐ ← JVM.
├──────────┤
│   OS     │
├──────────┤
│          │ ←
└──────────┘
```
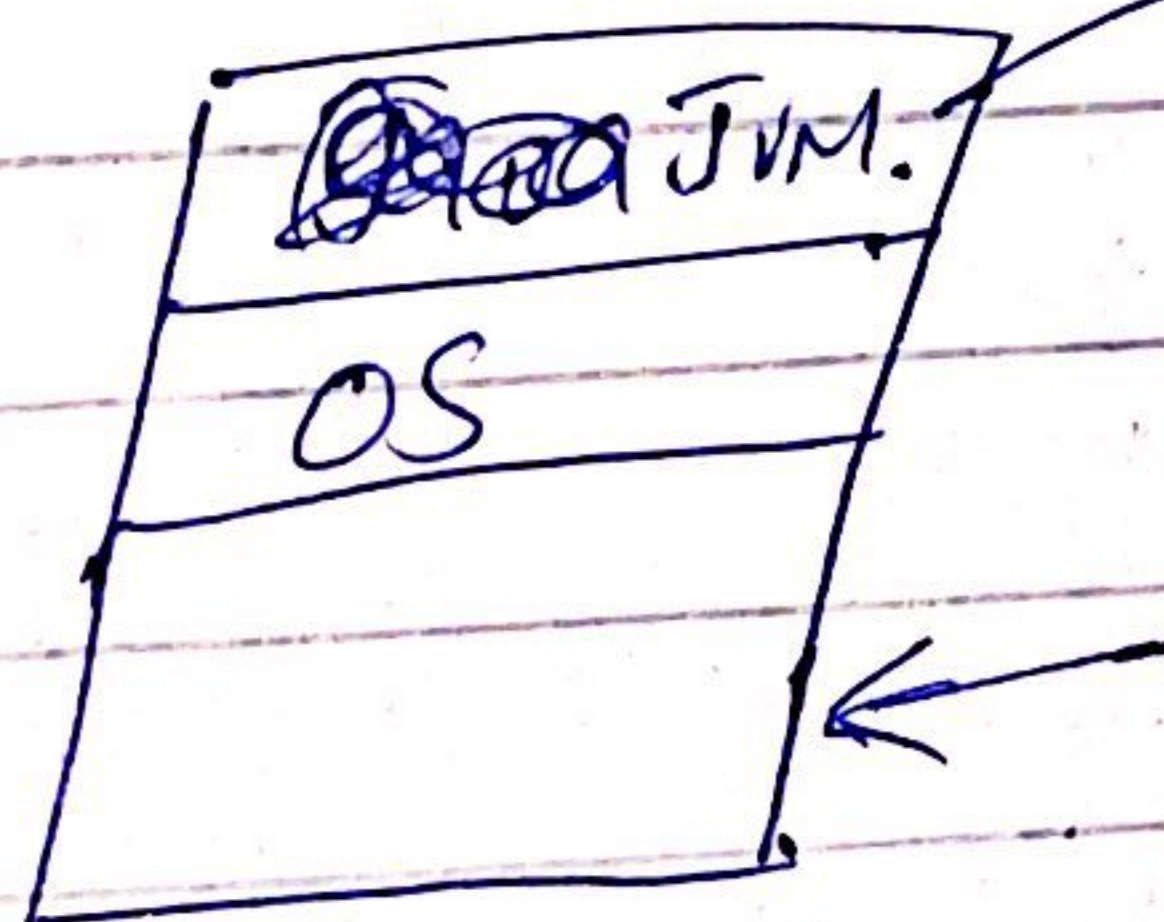
variable,
operator.
Statement

coalition
loop.

→ we do inheritance to reuse functionality.

Polymorphism:
```
Shape s = new Rect();

s.draw();
s = new circle()
s.draw();
```

→ If else can be replaced with polymorphism.

→ Java said no multiple inheritence due to diamond problem.

→ Java introduces interfaces.
　　　when you cannot make this.

→ B is type of A. we define interfaces instead of that.

→ Polymorphism can be achieved through Polymorphism.

→ Layered Application:

```
        _____
       |   PL   | ↓
        --------
       |   BL   | ↓      interfaces,
        --------  ↓      IDAL
       |        | ↓
        DAL
```
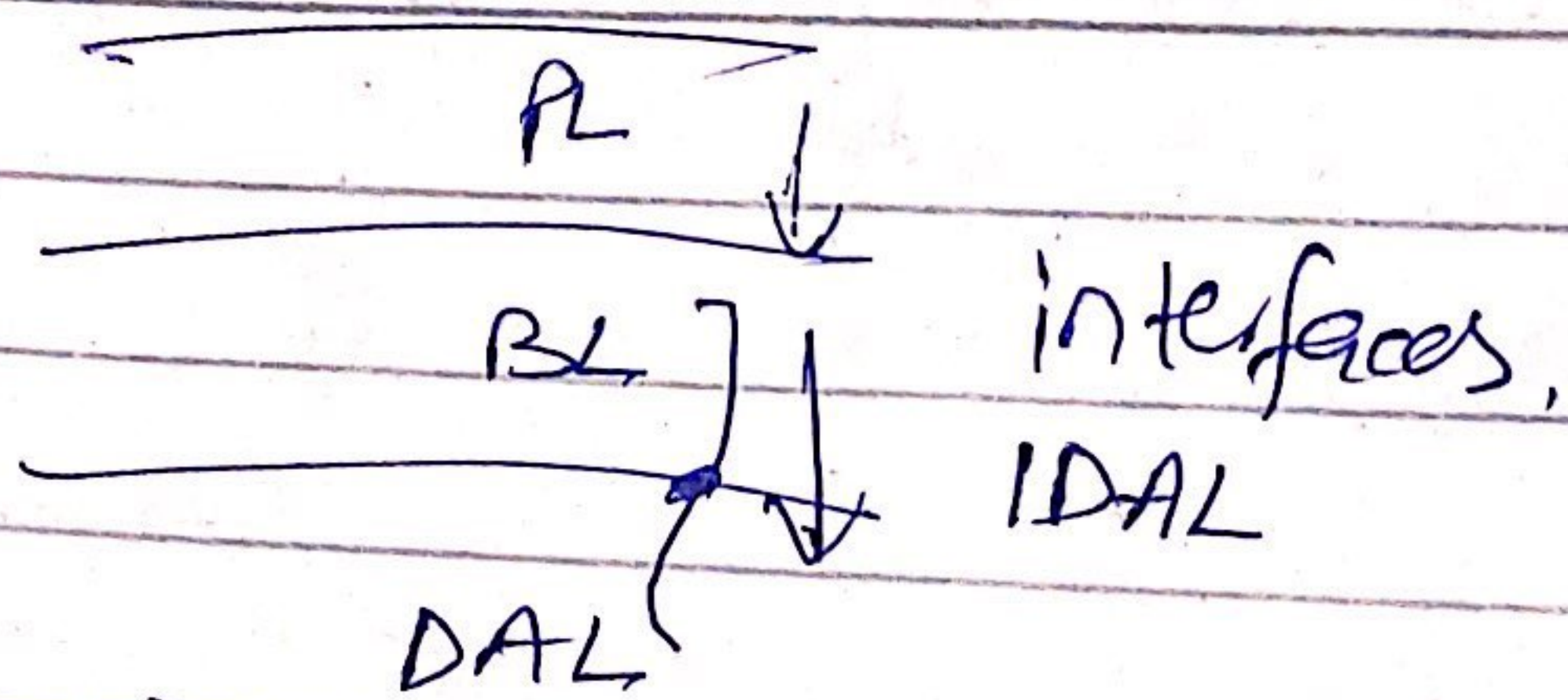
▭ mysql   Now i want to
         change mysql to
         nosql now I have
         to change Just one
         layer.

→ coupling can be reduced by removing concrete
  classes with interfaces.

→ if we have to separate two layers then we
  use interfaces instead of placing whole
  object to other.

→ Interfaces are used to define contracts.

→ Collection. Database columns results object
  and it is stored in collection to make
  it more generics. Suppose if we replace
  files with database then it will still work.

→ Exception: try {
                    ——— code actual
               }
               } catch( ) { if exception This
               } catch( ) { if This
               } finally ( ) {
               lazmi pat

assert
~~assertion~~ These are used in code for pre and
post codndition.

→ Use assethion for error handling.
→ JRE 8.
→ JUNIT 5.
→ Eclipse.

07/02/23.

get Person.
search Person(nam)

→ depedency inversion
→ Now That will use interface and not
concrete object.
→ Now IDAL will be used in PersonBO.
   Pehle   PersonBO —→ PersonDAO.
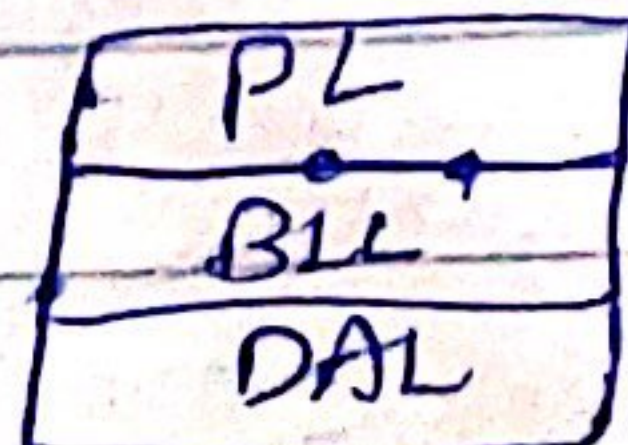   concrete

                       depends
   Now   PersonBO ——→ IDAL.
                         ↑ implements.
                     PersonDAO

   PersonBO now depends on   abstraction.
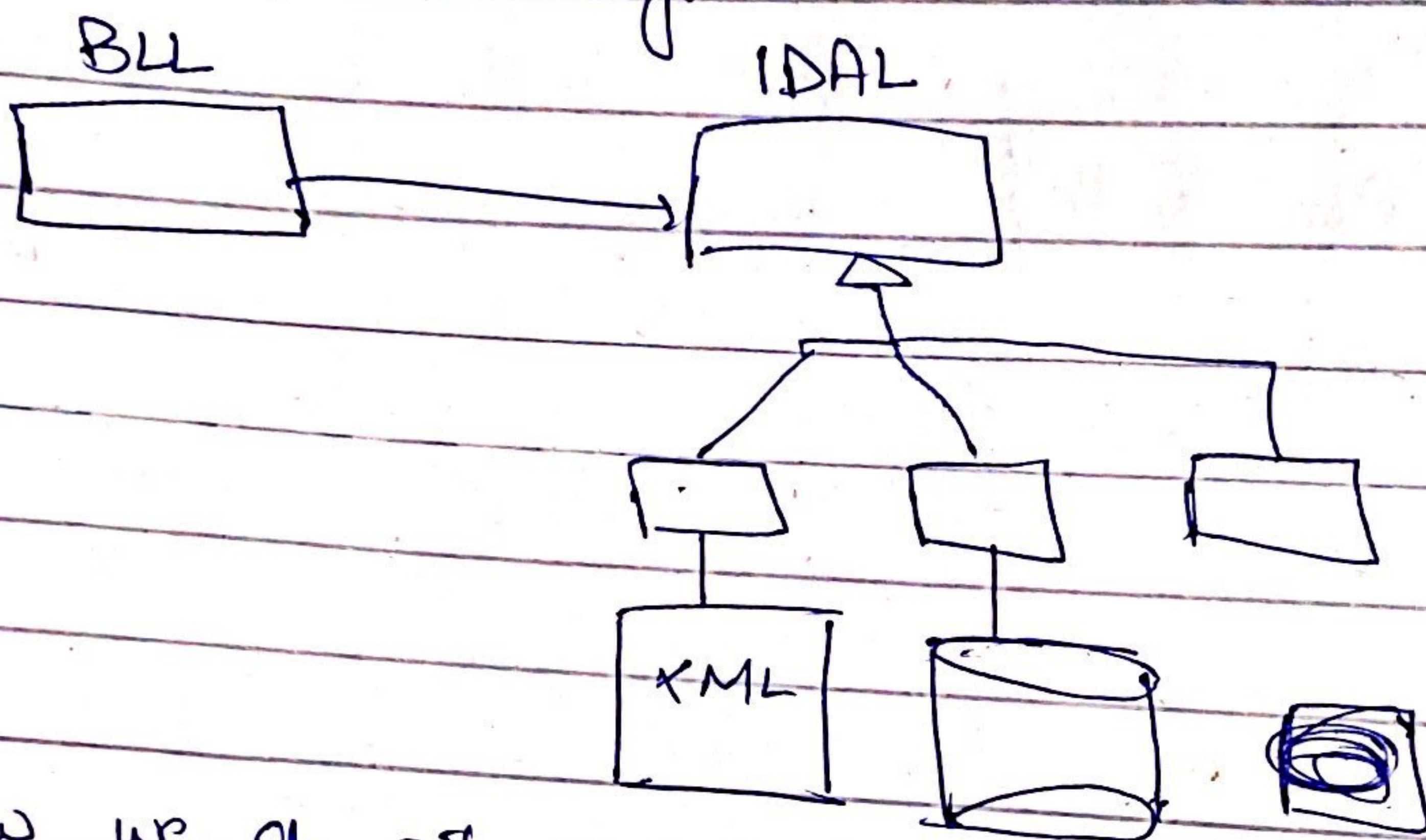
→ classes should depend on abstraction and
not on concrete classes.

→ Advantages: Low coupling,
           dependency ~~inversion~~ injection.
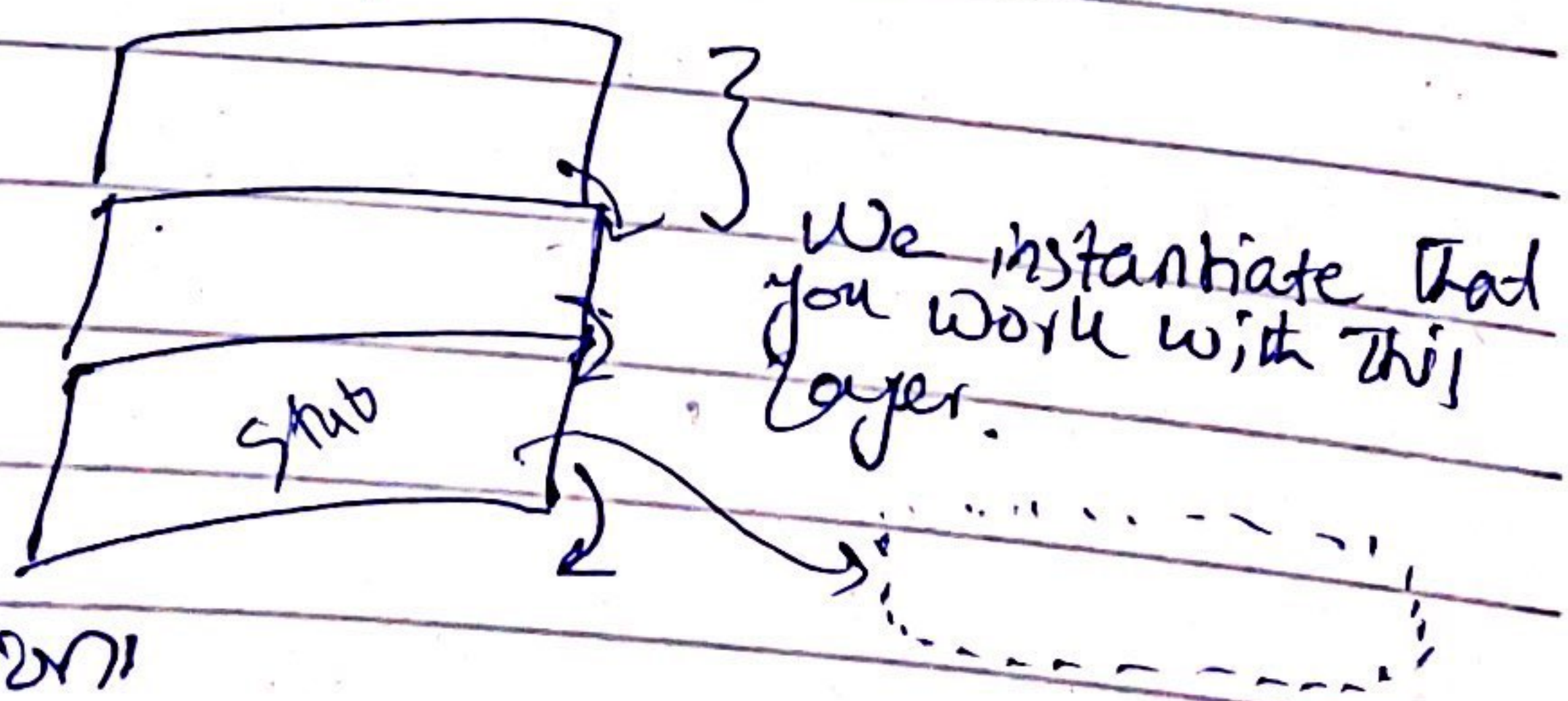
T

T

→ If you have done, you can introduce stubbs.

→ It will help in stubbing.

BLL   IDAL

XML

→ Now we do not have to create class objects.
We can a have Test cases for IDAL

                    In
⇒ Interface
→ Also connect   Reference interface Types.
   to dal to pl.
   09/02/23.

We instantiate that
you work with this
layer.

stub

→ Dependencies Injection:

→ Regression testing: Run all Past Test cases because
   some may fail due to new.

→ we will use stub and not exact
   implementation for Testing.

Advantages of stubs.

→ Stub use ~~Gote~~ Krein Kuke noshta error DAL ma ho.

→ Second DAI is not in access.

Bar Bar Query Krni prtin ha,

May DAL testing Kor or Kr sha ho.

→