Lecture 7 : Project : Google Docs

Page No.
Date

Problem statement : Creating a Document Editor where user can add text as well as images by following SOLID principle and make it scalable so that in future we can add more features like videos, new line, tab, etc.

Approach to solve LLD Problem

→ Top-Down Approach ⇒ Create main object first and then create small object

→ Bottom-Up Approach ⇒ create small object & then main one

(Generally used by 90% of developers!!)

⇒ Here we will use bottom-up approach

Initial Design : Bad Design

DocumentEditor
vector <string> element;
addText(string text);
addImage(String path);
renderDocument();
saveToFile();

→ We will store our text and image both here as we don't know the sequence in which user will add.

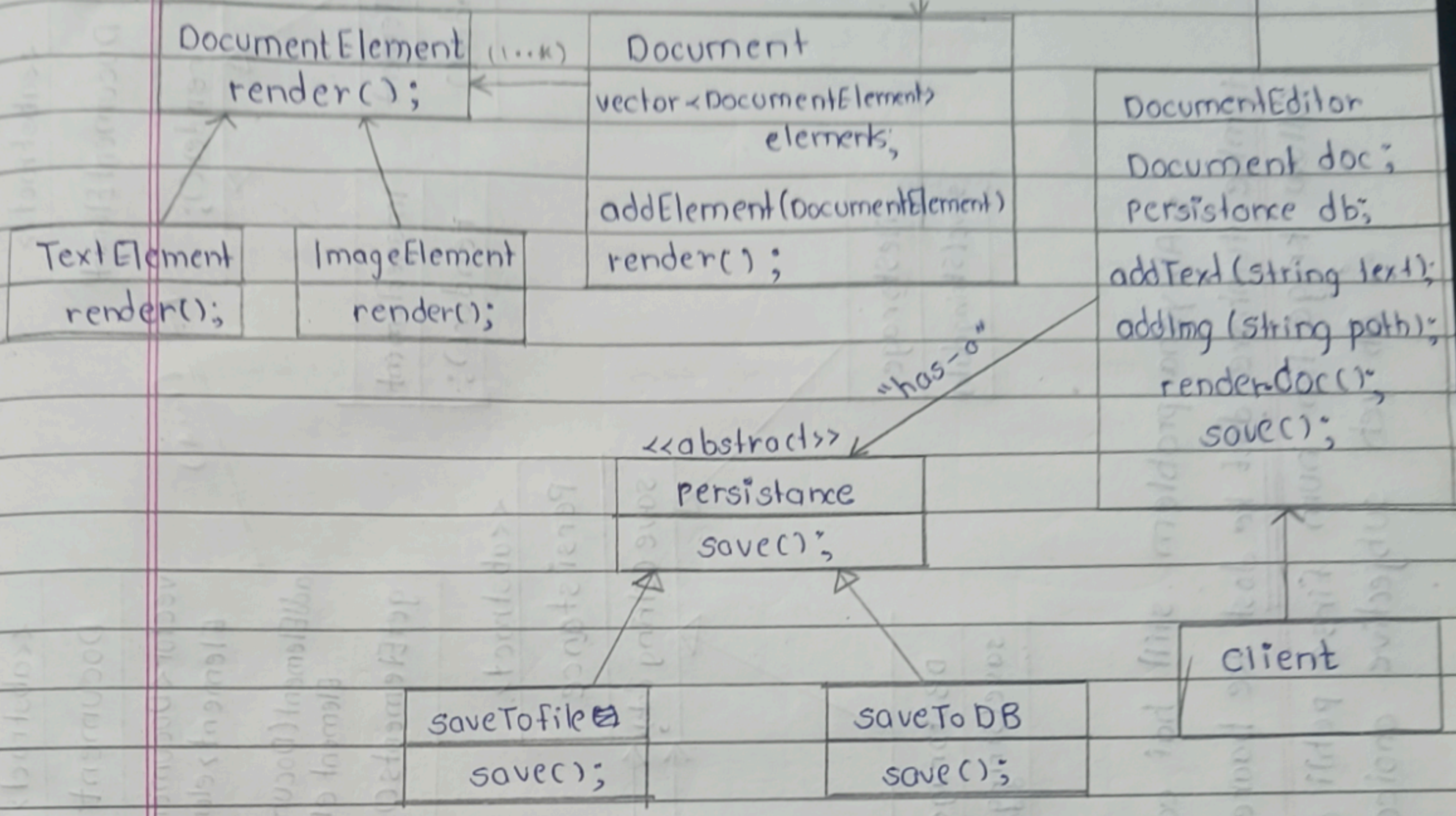
Problems :
1) Breaks OCP (x)
2) Breaks SIP (x)

Not implemented : ① LIP
② DIP
③ ISP

we can replace it by child class

Page No.
Date

Better Design

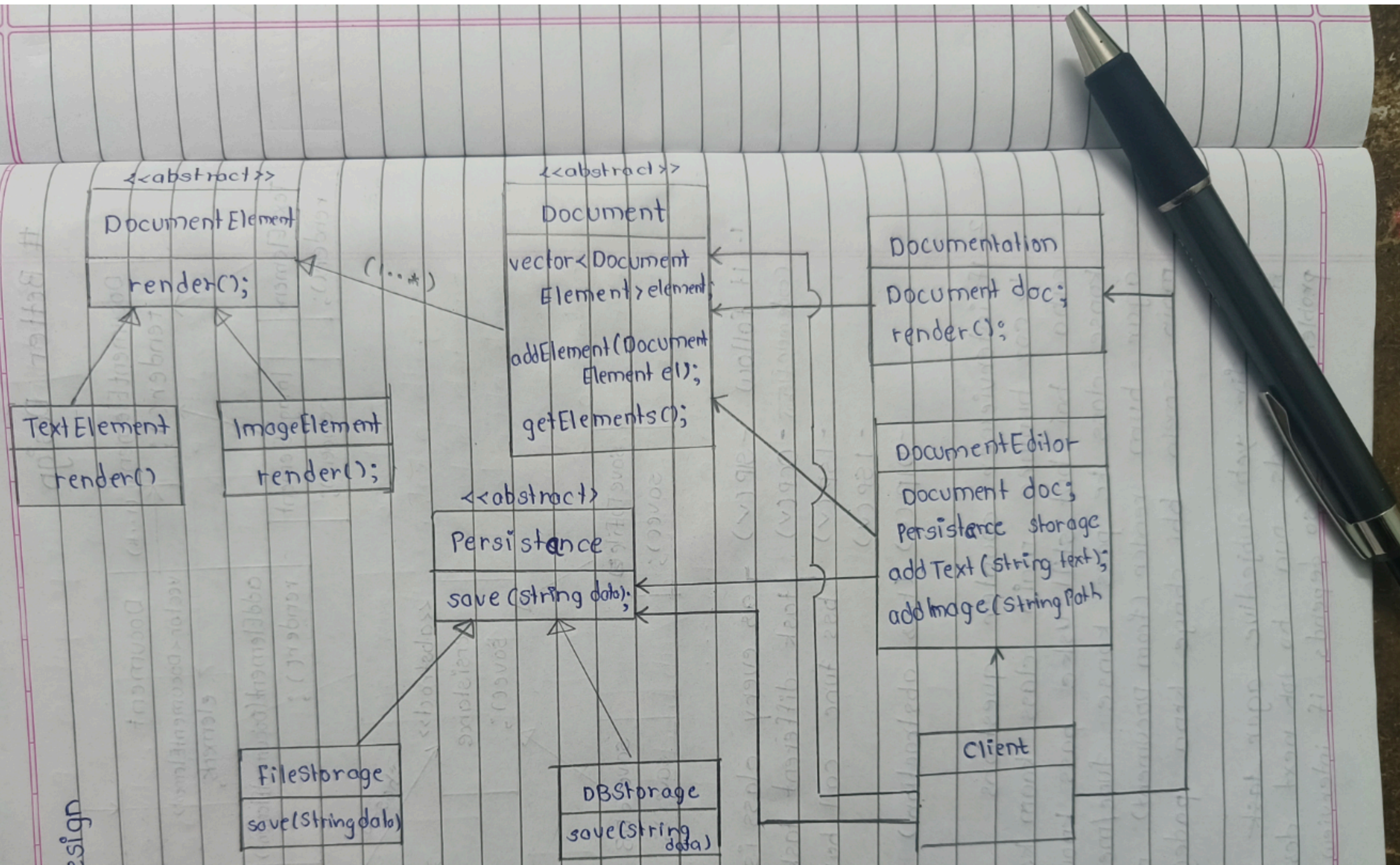


1. It follow - SIP (✓) - as every class is working on single task different task ko delegate kiya hai
   (code mein dikhai) - OCP (✓) - bss func call hai
   - LSP (✓)
   - ISP (✓)
   - DIP (✓) (has abstraction)

2. If interviewer counter questions
   ↳ coz humare main class kaam ko delegate toh kar raha hai but still uske pass knowledge hai ki konse class ke pass konse func/methods hai aur agar hum render (from Document) nikal de to humme main ismein bhi chnge krna padega

↳ Ab fir yeh subjective agar ~~task~~ main ke pass main methods naa ho toh next design ab usmein bhi problems hai (so depends if interviewer approves this)

Optimal Design



- Ab yaha problem still hai called Principle of Least knowledge isko mtlb hai tumhe tumhare dost ke dost ke baare mein nhi pata hona agar hum yeh principle follow krege toh hum firse pehli design pr ayege
∴ Ab yeh subjective choice hai ki konsa follow kre.!!