

→ L

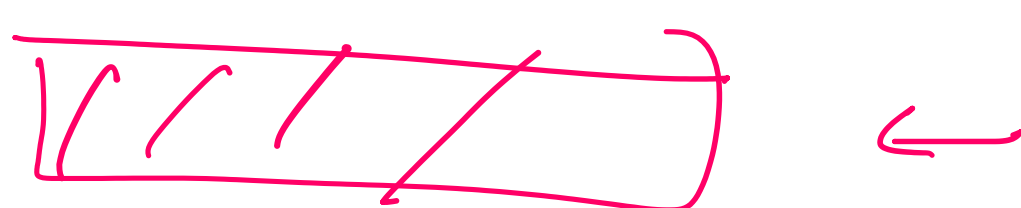
- Class dig
- DSA (hidden)

60 - 120 min

- Clean prob.
- Set of req.

- Write a working code
- Main To Test
- UI is not req.

$> 90^\circ$



Amazon (SDE-2)

Adobe

Design a Pen

Current

-> Design a **Pen** with a feature of **write()** to the user/child.

→ Anything that has nib & writes with ink is a pen.

→ Ball Pen & ink pen have diff
unit behavior.

→ Ball Pen has a refill & ink pen has ink.

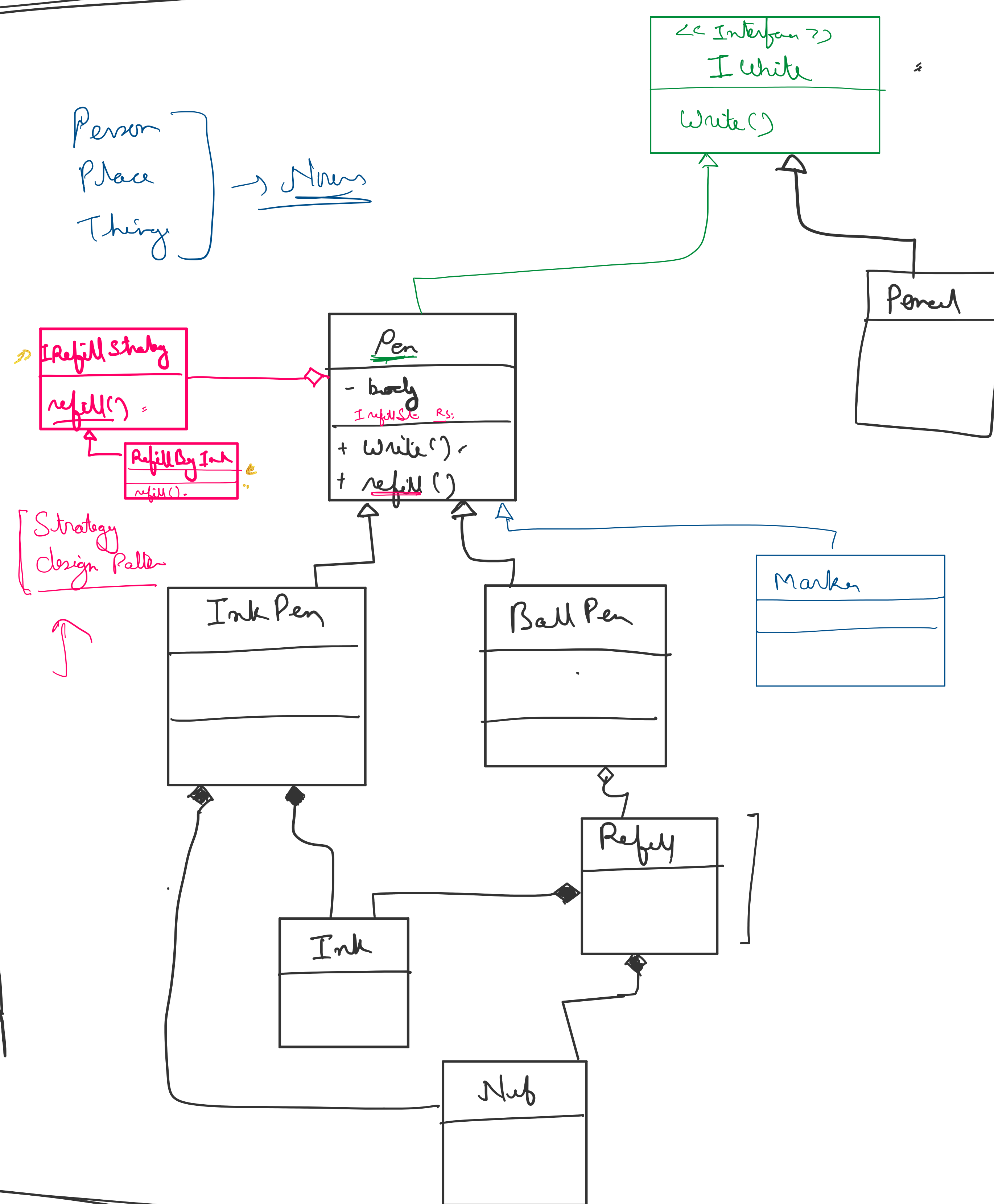
→ In future Skilled Per & Market

→ We might want to control the system for kernel, shell etc.

→ Implement refill();

Generalität

feature
scope
on
other feature



$\left[\begin{array}{l} \text{Muehl} \end{array} \right] \rightarrow \begin{array}{l} P_n \\ P_n \end{array} \rightarrow \begin{array}{l} P_n \\ P_n \end{array}$

$\text{str} \rightarrow \begin{cases} I \\ - \\ 0 \end{cases}$

$n=3$

Surze / dimension \rightarrow $n \times n$ (n in a.m.)

- Multiplayers or AI \longrightarrow Computer players

No. of player \longrightarrow 2 play

= Move undo? \longrightarrow

Confirm the rules / feature / what is it

API

→ Create / Launch gem

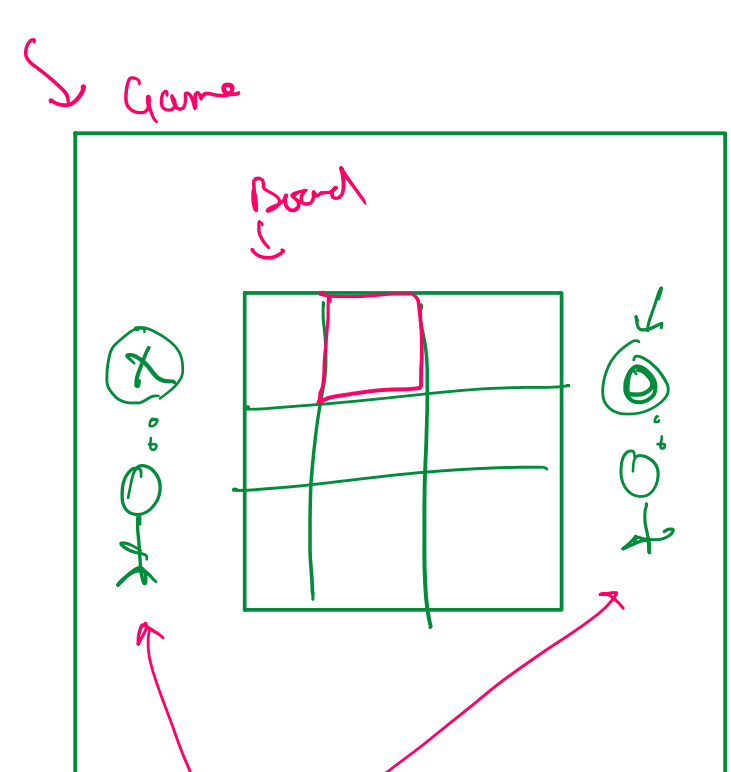
Players make moves turn-by-turn

→ IS union?

Stop the game if no moves left
or
there is a winner

① Choose difficulty level of computer plays (Easy/Med/Hard)

② We might want to update the winner selection algo.



Player

Board (init())
2 Players ↔ Symbols

