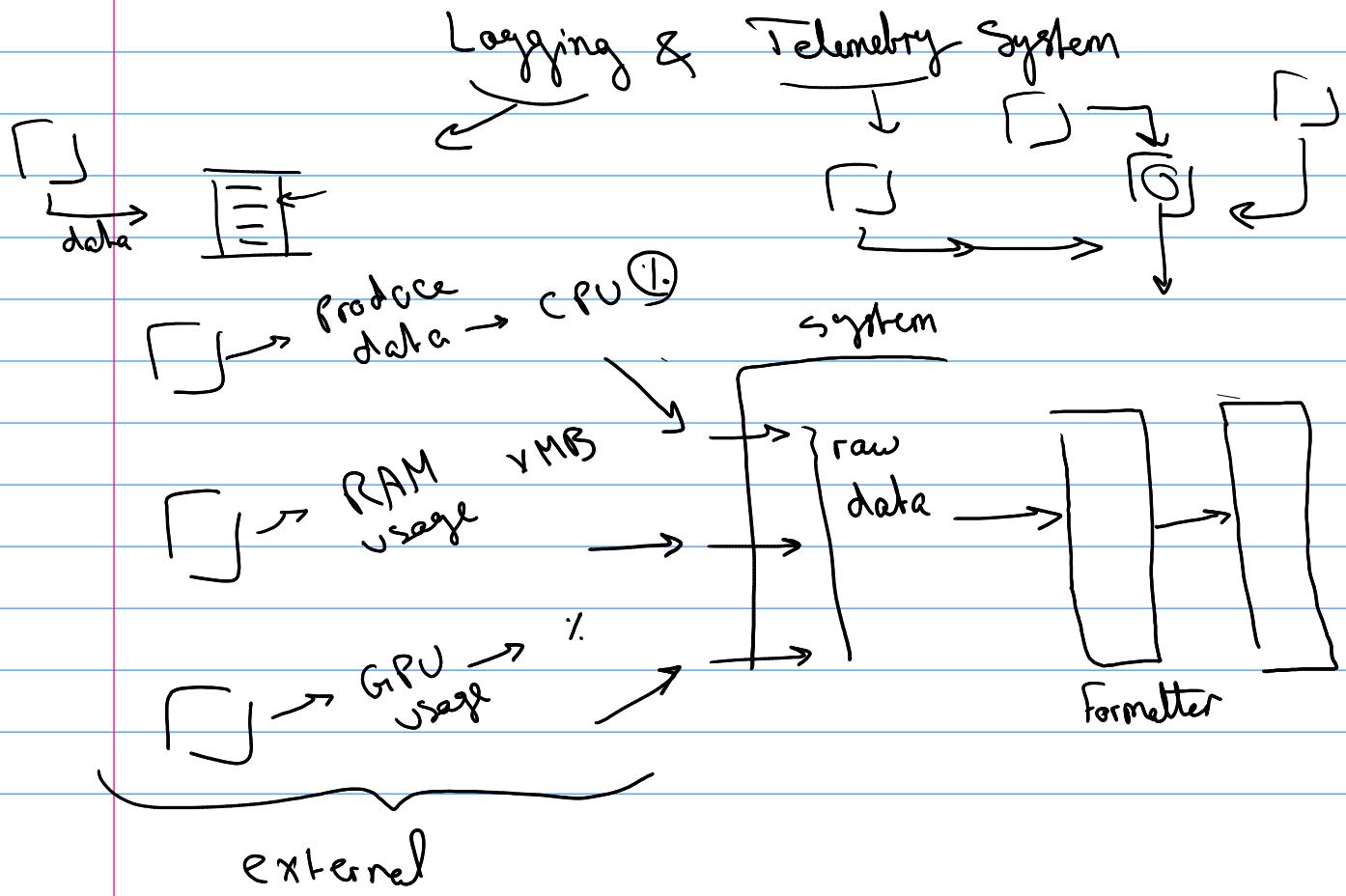
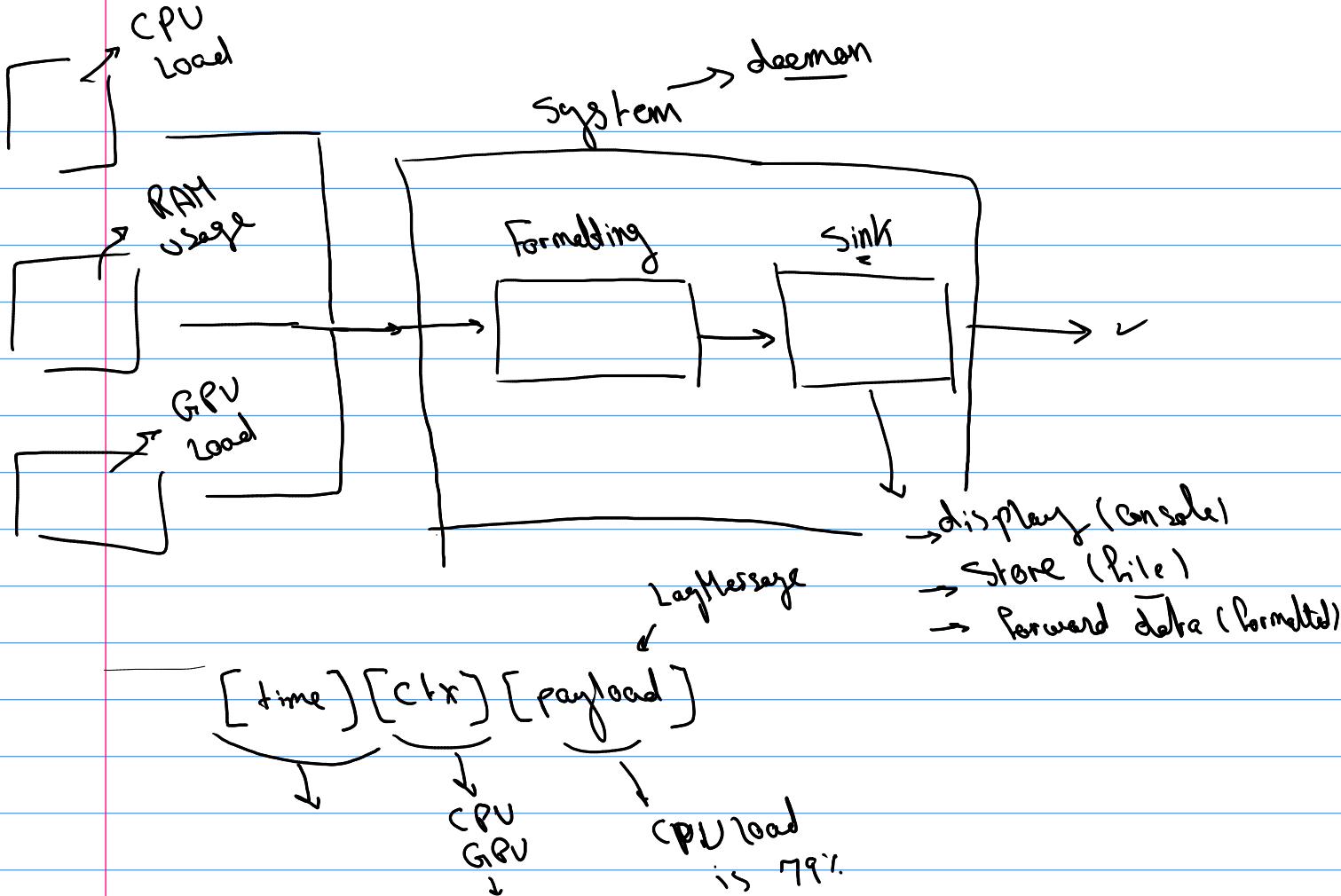


CPP Project

Modern C++ → applied



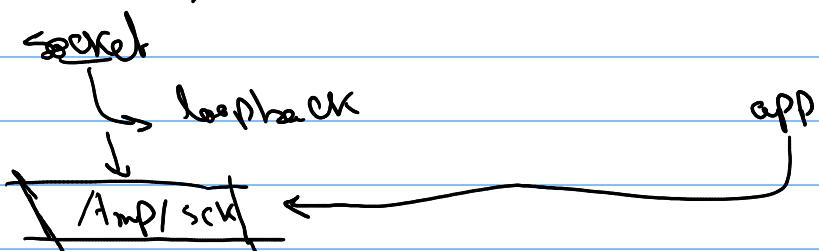


Shell Script

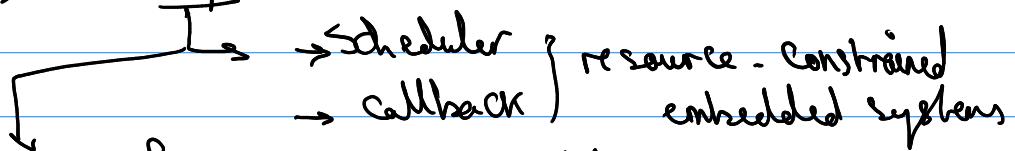
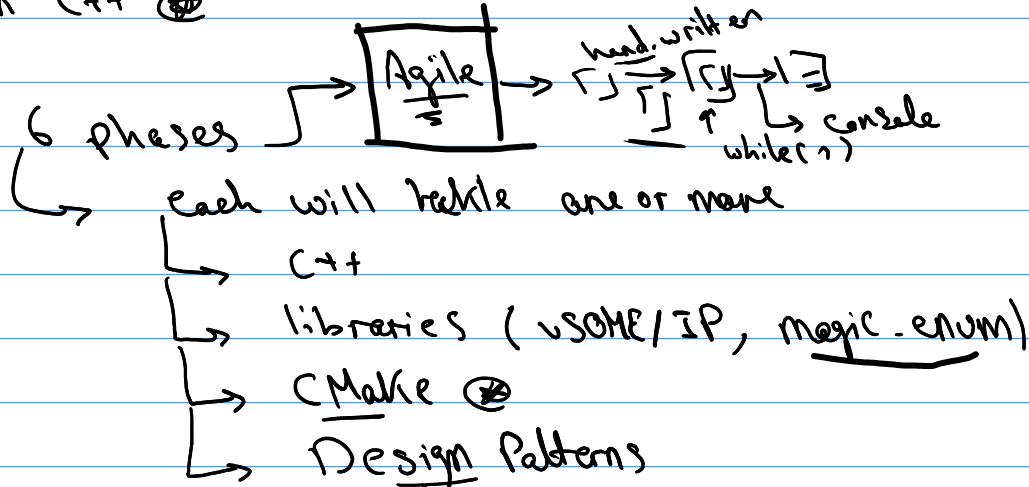
↳ randomly generated data in while (terminal)

↳ some ip service

↳ offer data



④ Modern C++ ④



best practices for common SW problems

- †) Creational → SinglTON
- †) Structural → private constructor →
- †) Behavioral →
- †) Archi → IMUOM, MVP, MVC

Design Idiom

language-specific best practice

Rule of 0
std::vector

RAIIS

Resource Acquisition Is Initialization

⇒ OP mark per phase
OPTIONAL

mech team -

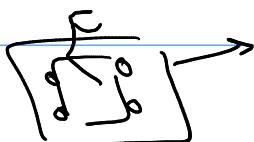
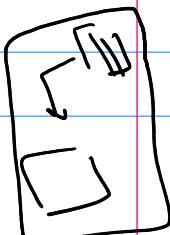
elec team -

sw
embedded -

RGS

↓

actuation



Grading & Rewards

Per Phase

C++ ✓
CMake ✓

Minimum

- ① at the end
↳ Doc (README + class diagram)
- ② at least one
class unit tested GTest

Her list

To score → reward

.FIDL

France gen

Bazel

Optional

GTest

DP

✓ stub

✓ service

- Modern C++ (req)
 - CMake (req)
 - UML Diagrams ② (PlantUML) (req)
 - Unit testing using GTest (req)
 - SOME IP (req)
- } *

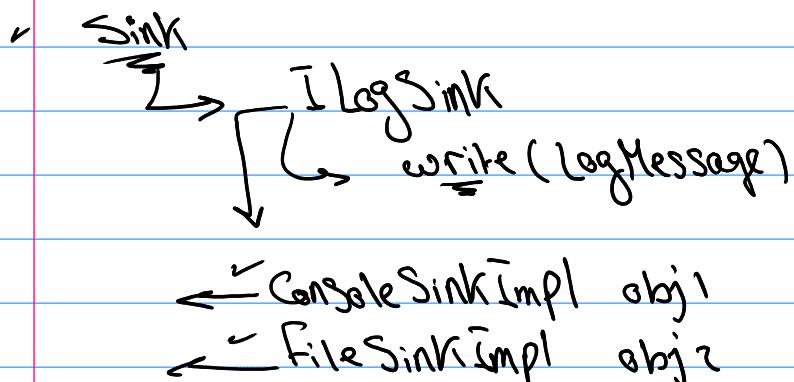
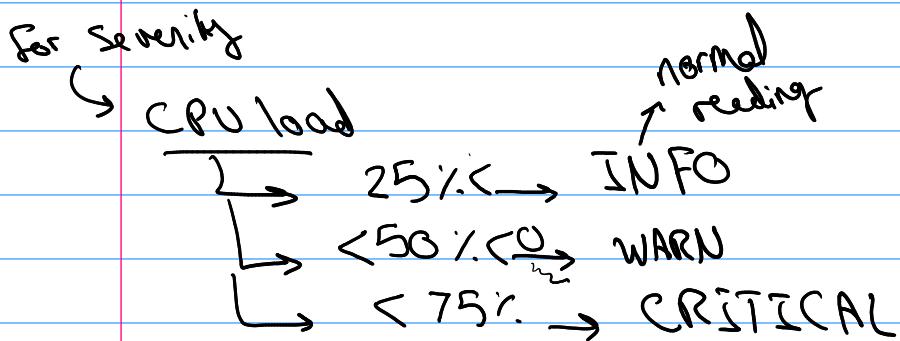
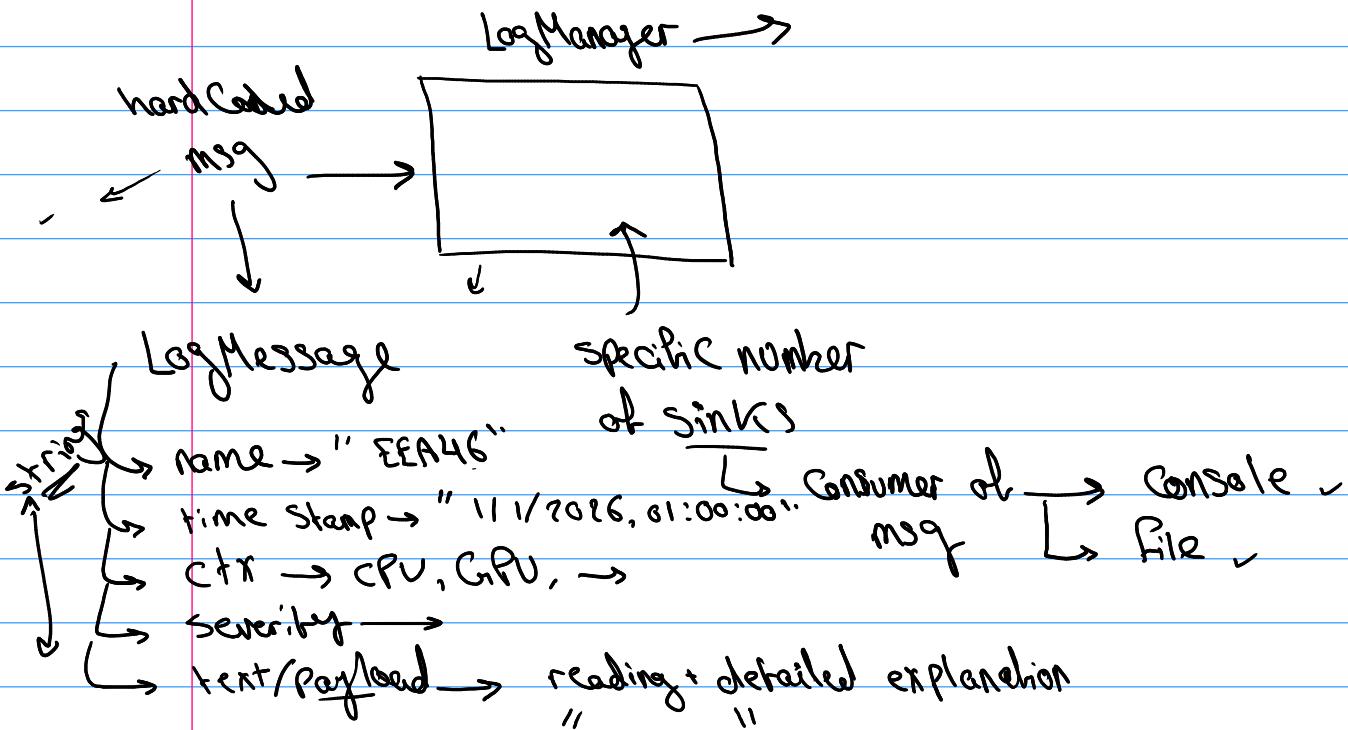
- DP
- Bazel
- Common API

Naming a project

Commitment

Phase One

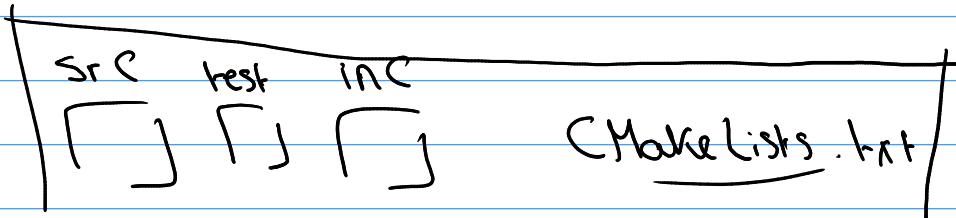
Core Synchronous Logging Foundation



`ILogSink* pIrr = &obj1
obj1->writer();`

LogManager

```
addSink( )  
vector<sink> { }  
vector<LogMessage> →  
writeToAll( )  
iterate over all sinks  
& let them consume msg
```



friend

operator = ()

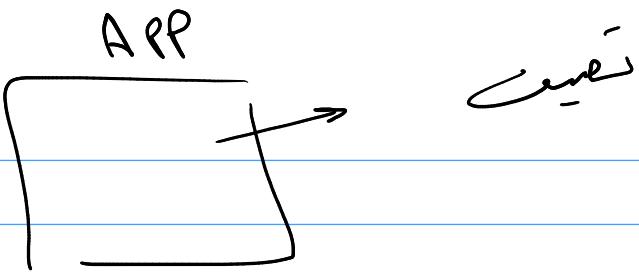
* Bonus

operator <<

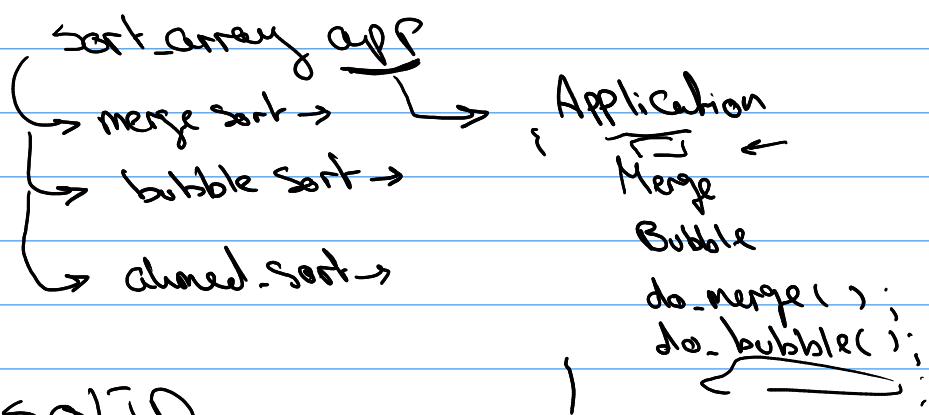
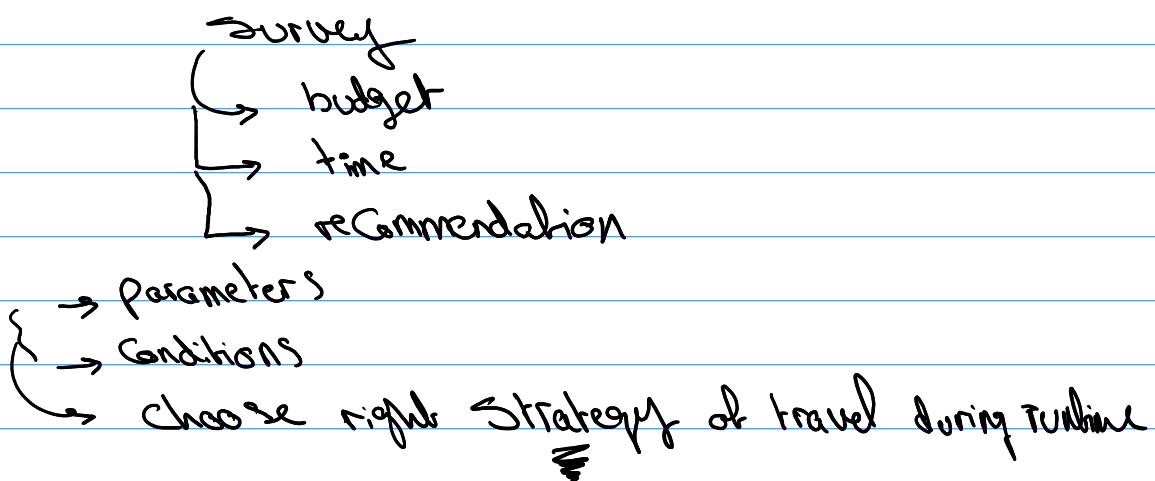
LogMessage msg{ };

std::cout << msg << "\n";
→ [] [] [] [] []

→ [EEA216] [1/1/2000, 13:00:00] [CPU] [INFO] [load is 2.5]
when printed
on terminal
OR
written in a file

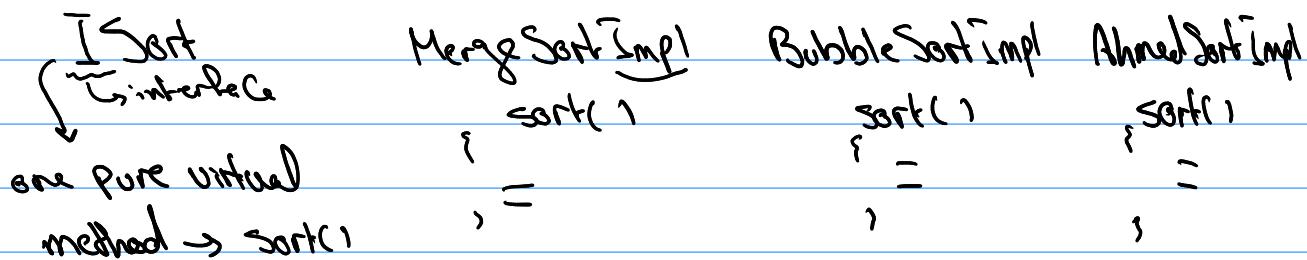


x way of transportation



SOLID

Open for extension
Closed for modification



Application

ISort

} init according to certain logic

Design Pattern \Rightarrow Strategy

⊕ Optional
For Phase 1 ⊕

↳ Behavioural

Design Pattern

friend → merge - sort(); → MergeSort
 ↳ bubble - sort(); → bubbleSort
 ↳ merge - sort(); → mergeSort

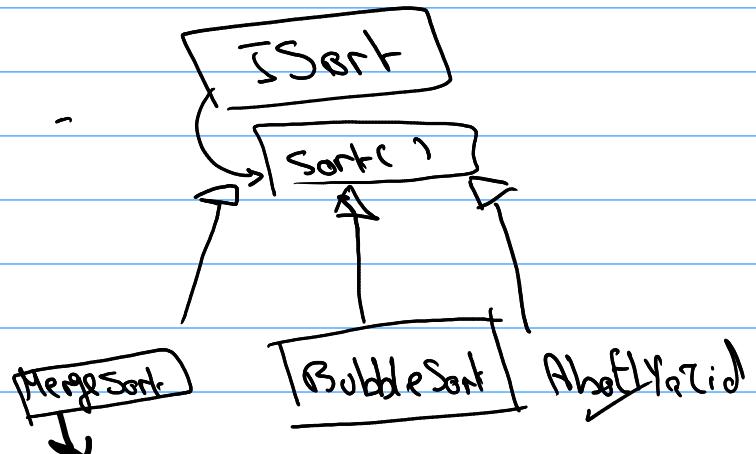
class SortArray <
 BubbleSort bubble;
 MergeSort merge;
 array; <
 SortArray() merge - merge(); ~

SortArr (enum Alg)
switch (Alg)

 case added

Strategy ✓

class SortArray
 ISort* sort
 SortArray();
 sort - arr();
 sort - ptr -> sort();



⊕ virtual bool sort() = 0;

MergeSort obj
SortArray
 ↳ MergeSort
 ↳ MergeSort
 ↳ ISort

