

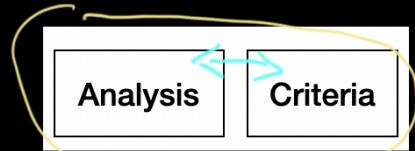
GMAIL APP.

1. Criteria

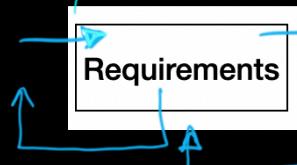
o. functionality

SDLC

defining data and operations

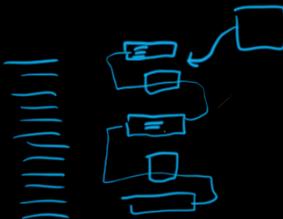


can be mixed



① Know Your Data!

2. Encapsulation



Put in

1. combining our data based on functionality

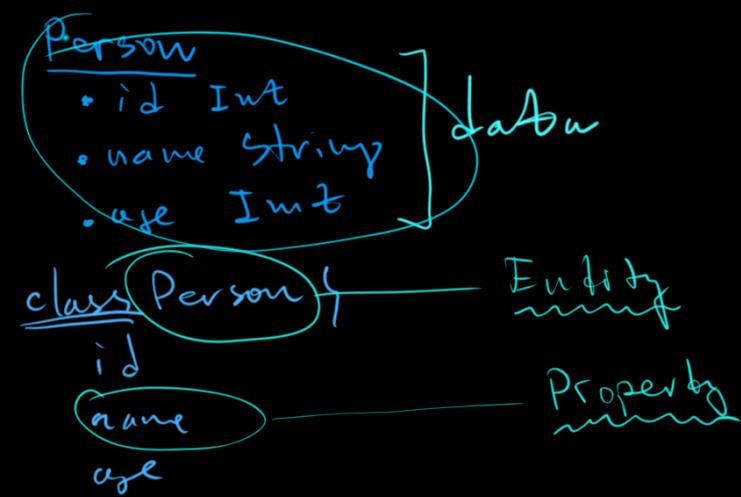
implementation only
here

5-10% time

Scaling
Fault Tol

=?

2. one of the principles OOP
→ private / getter / setter
methods to change data
data hiding
modifying ...



class Person Controller

- get By
- show All
- delete by
- ...

data manipulation

3 Design your Data

① Define your Data

② Define Operations with your Data

③ Combine Data + Operation

④ Implement It.

we implement
only when we are
ready

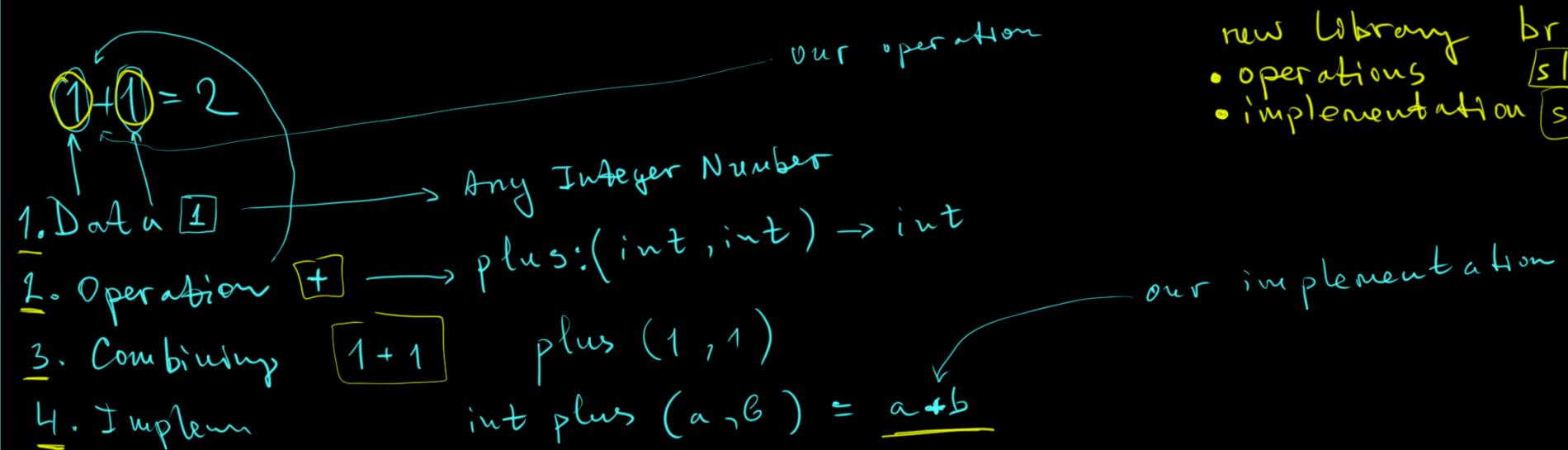
⑤ Bring New Library

⑥ Setup mapping Person → Library Type

⑦ Use library functionality

#3

new library brings:
 • operations S[lf4j-api]
 • implementation S[lf4j-simple]



Person (age, id, name)

we defined a new data type it's an encapsulation of (int, int, string)
 and giving a proper name

UserId (^{UUID} int value)

forget about primitive types
int, float, boolean, str

raw types

2. operations

- person creation
~~ummm~~

$f: (\text{int}, \text{int}, \text{string}) \rightarrow \text{Person}$

- person load :
~~ummm~~

~~(id) \rightarrow Person~~

$(\text{id}, \text{database}) \rightarrow \text{Person}$

- person store
~~ummm~~

~~(Person) \rightarrow void~~

$(\text{Person}, \text{database}) \rightarrow \text{void}$

- sort : $\text{List}[\text{Person}] \rightarrow \text{List}[\text{Person}]$

- contains : $(\text{List}[\text{Person}], \text{Person}) \rightarrow \text{bool}$

- isEligibleToOpenAccount : $\text{Person} \rightarrow \text{bool}$

- $= \text{new Person(id, age, name)}$
 ↗ implementation

p.id
p.age
p.name

$\text{database.loadPerson(id)}$
 ↗ implementation

$= \text{age} > 14$ implementation

↳ combination = writing code in terms of
 and Data
Data Operations

b.save(p);

Person p2 = b.load(id: 33);
 boolean canOpen = b.isEligibleToOpenAccount(p2);

#5
 class interface → class ↗

your code (Implementation)

Your Code

Person
 Account
 Balance
 Report

↓
 to make it working mapping

person
 log(persons)

[Logging] Library

log.info(str)
 log.error(str)
 log.warn(str)
 ...

Does any library
 know about your data?

→ NO

sort

Person → Int

Person → String
 Account → String

P → maps → logger

person → personKey → sort

- Sort in any programming language
- sort (List[Person])
- sort was no idea/knowledge how Person is constructed.

* Collections.sort(persons);

- solution: tell sort how to sort
 - ↳ exposing integer field to sort by

```
@Override  
public int compareTo(Person o) {  
    return this.id - o.id;  
}
```

this function is the way how tell sort how to sort

$$\sqrt{2+7}$$

my data is Int

+ is $f(\underline{\text{int int}}) \rightarrow \text{int}$

= is $f \underline{\text{int} \rightarrow \text{int}}$

$$\sqrt{\sqrt{x+3+y} + \sqrt{y+5}}$$

by defining TWO functions

I immediately get all combinations

person

operation w/person

combination