

Agenda

① Schema Migration and Versioning

Testing

① Why Testing

② TDD

③ Unit Testing

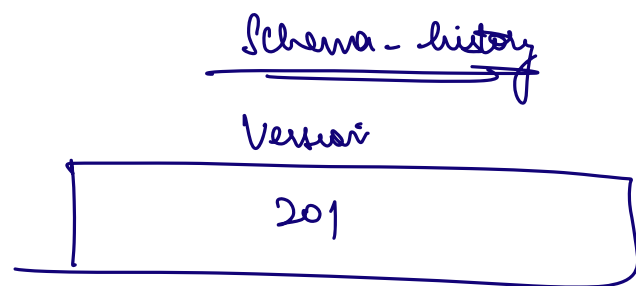
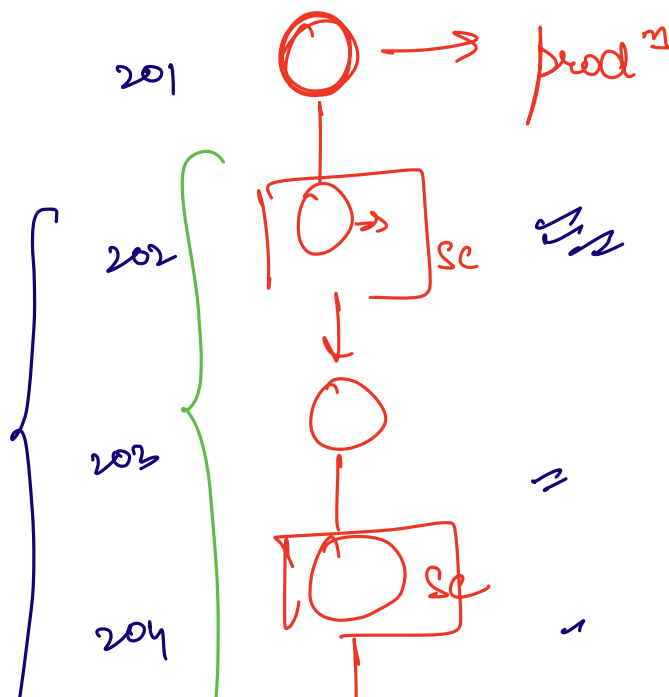
④ Flaky Tests

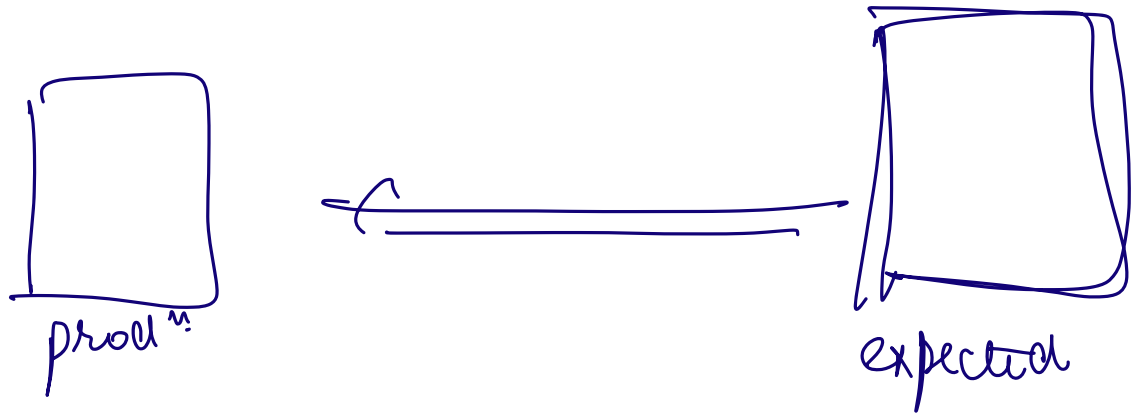
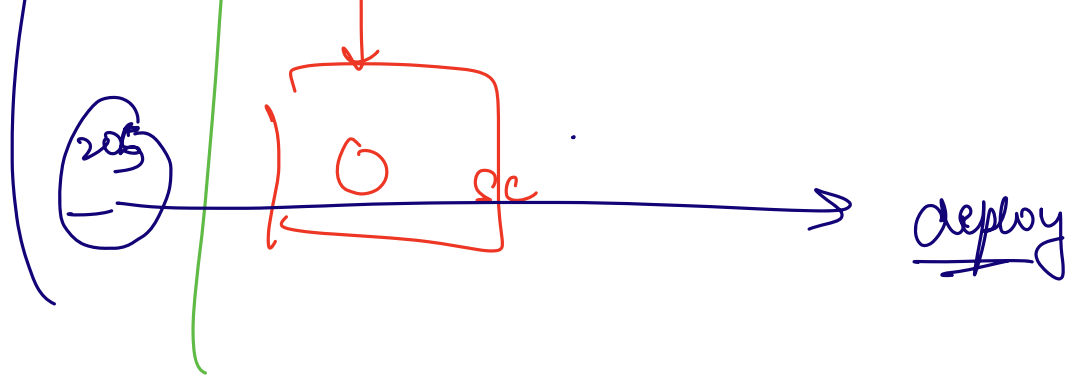
⑤ Write our first test cases
→ Suite of JUnit

Next Class

→ Mocking in Testing

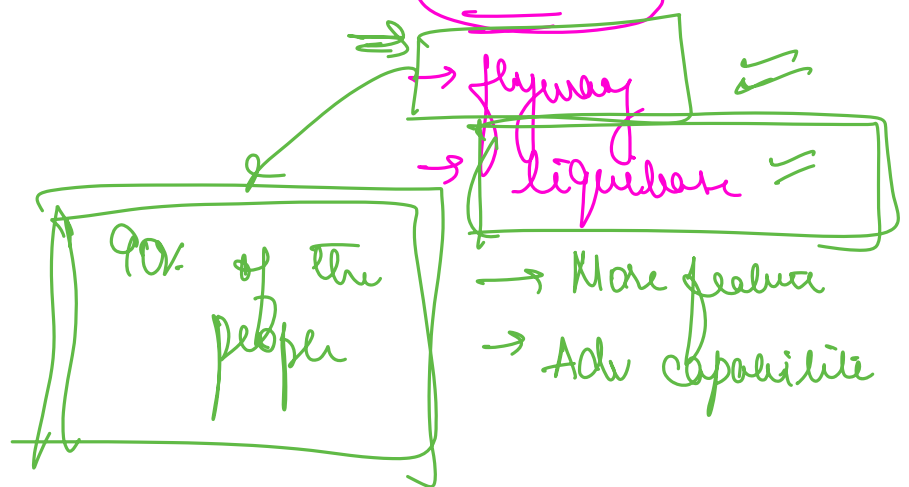
→ Web-MVC Mock Testing





Flyway

→ Schema Migration
libraries



① Start the appⁿ

② Appⁿ connects to DB and checks
current version

③ Spring sees if any other versions

Yes

Run the migration
script for those

NO

④ Spring Run Validator (model and Queries)

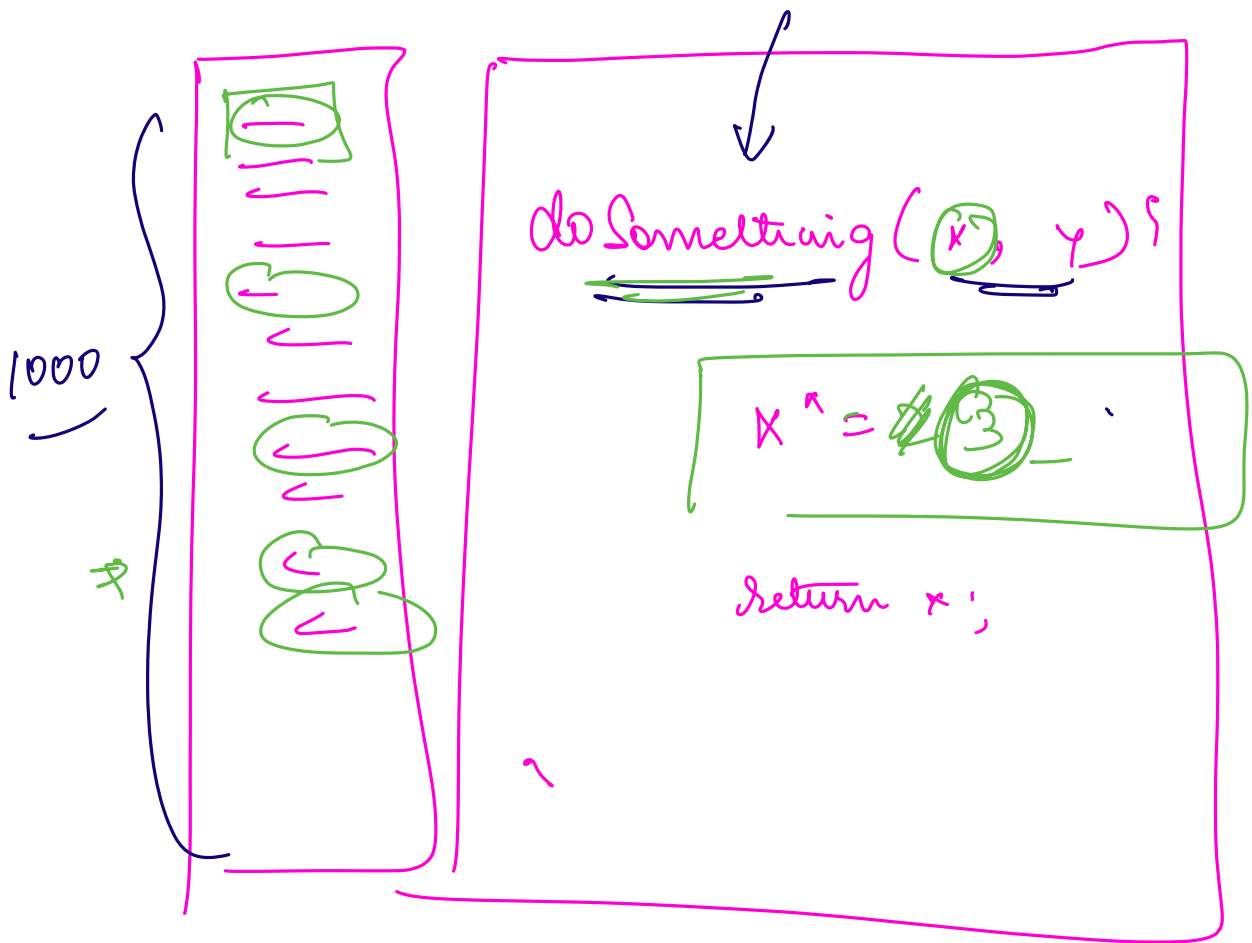
Yes

Appⁿ is ready
to serve req

NO

{ Throws error
and appⁿ
stops.

Break till 10:25



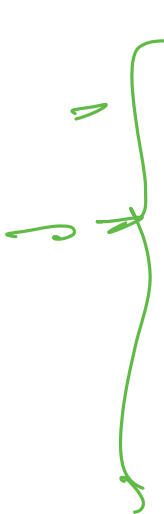
B. java

get Number Of Road lanes (int lanesPerSide) {

total lanes = doSomething

(lanesPerRoad)

}

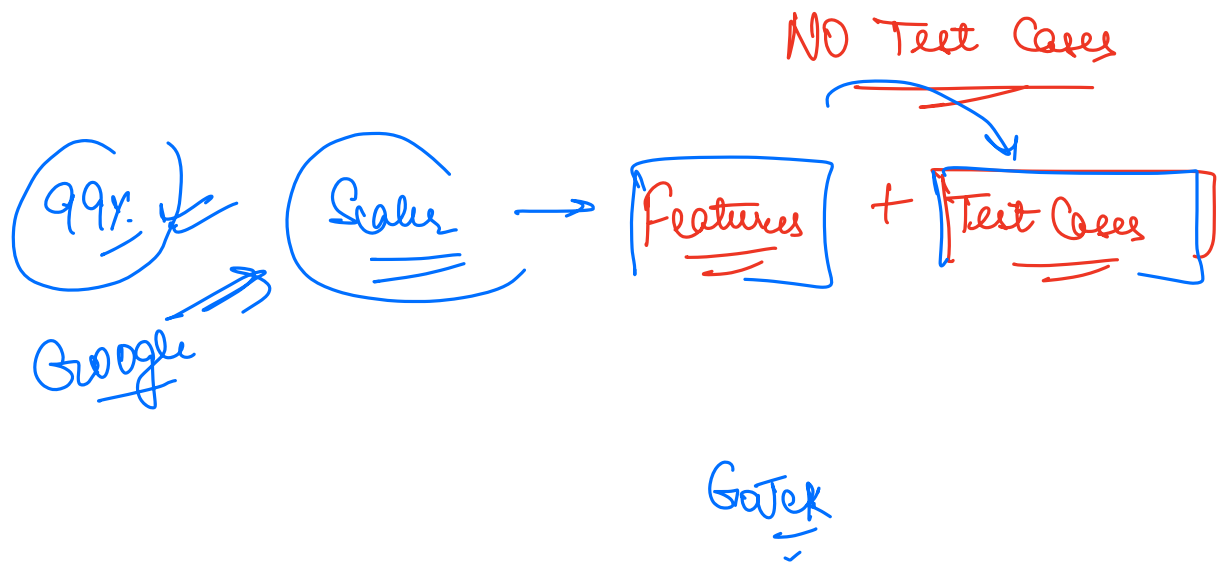
 `int total Married People (total Distinct Pairs) {`
`return doSomething (total Dist Pairs)`

`int n`

Testing is important

- ① cost of a change becomes high as the size of codebase increases.
- ② manual testing may not take care of all bugs/ issues.

Test Driven Development



```

class FibonacciTest {
    fibonacciCalculator calculator;
    
```

```

    test 10 {
        if (calculator(2) != 1) {
            // exception
        }
    }
}
    
```

>

Solution {

Solution(list<int>, int, String)

== // to fill.

}

}

Solution Test {

= Solution({1, 2, 3, 4}, -, -)

}

↓

Tax Calculator - calculate Tax (3000000, 18%, 0, 50)

Tax Calculator

- set Earning (30000)
- ~~set~~ Tax Rate (18)
- set Saving (0)
- set Soc (50)
- calculate()

⇒ Client

GoTFR / Thoughtworks X

TDD

→ Make dev more empathetic to user

Com

→ slow
→ clumsy
→ boring.

Flaky Tests

```
Printer implements Runnable {  
    int n;  
    run() {  
        print(n);  
    }  
}
```

main() {

T1 → Printer(1).start();

T2 → Printer(2).start();

}

Test Case {

String op = main();

check if op == "1 2"

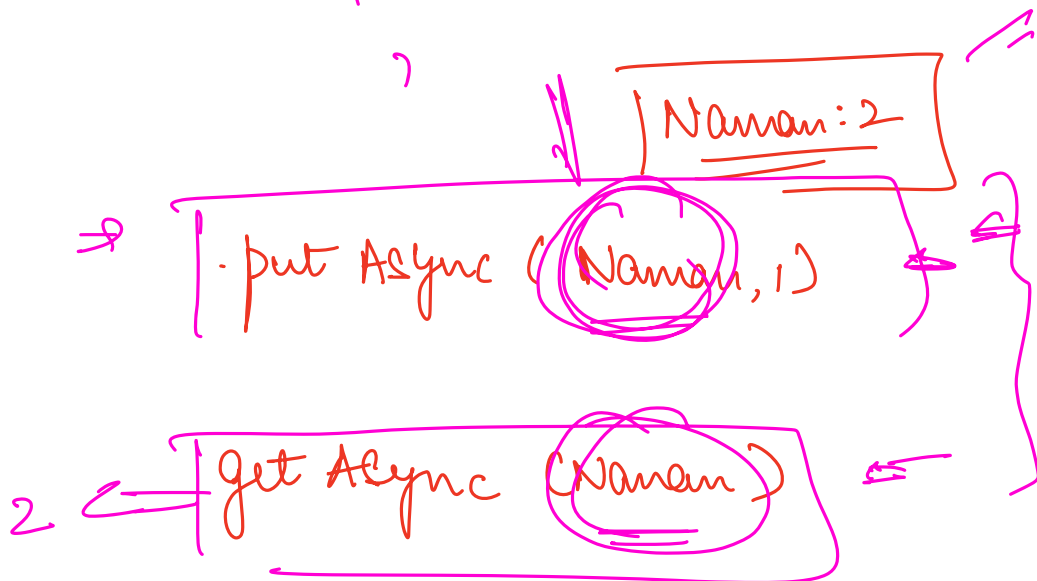
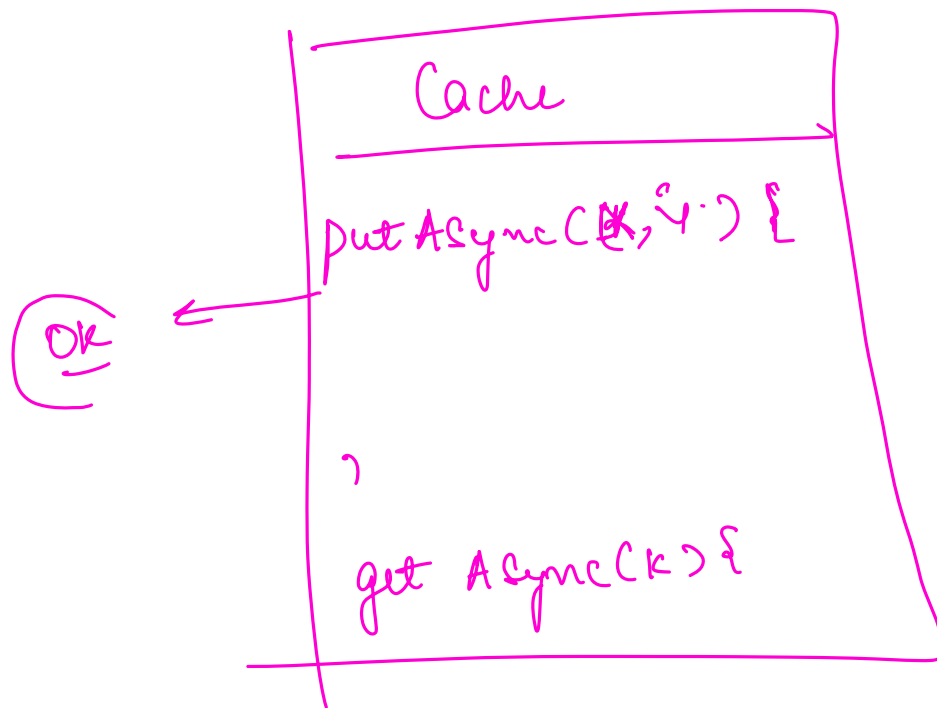
}

"No Flaky Tests"

Flaky Test \Rightarrow test that is indeterministic
 \Rightarrow that may pass sometimes
and fail sometimes.

typically \rightarrow

- ① Concurrency
- ② randomization



→ What kind of test cases.
→ How to write

- ① What is TDD
- ② What test cases to write
- ③ What is unit testing