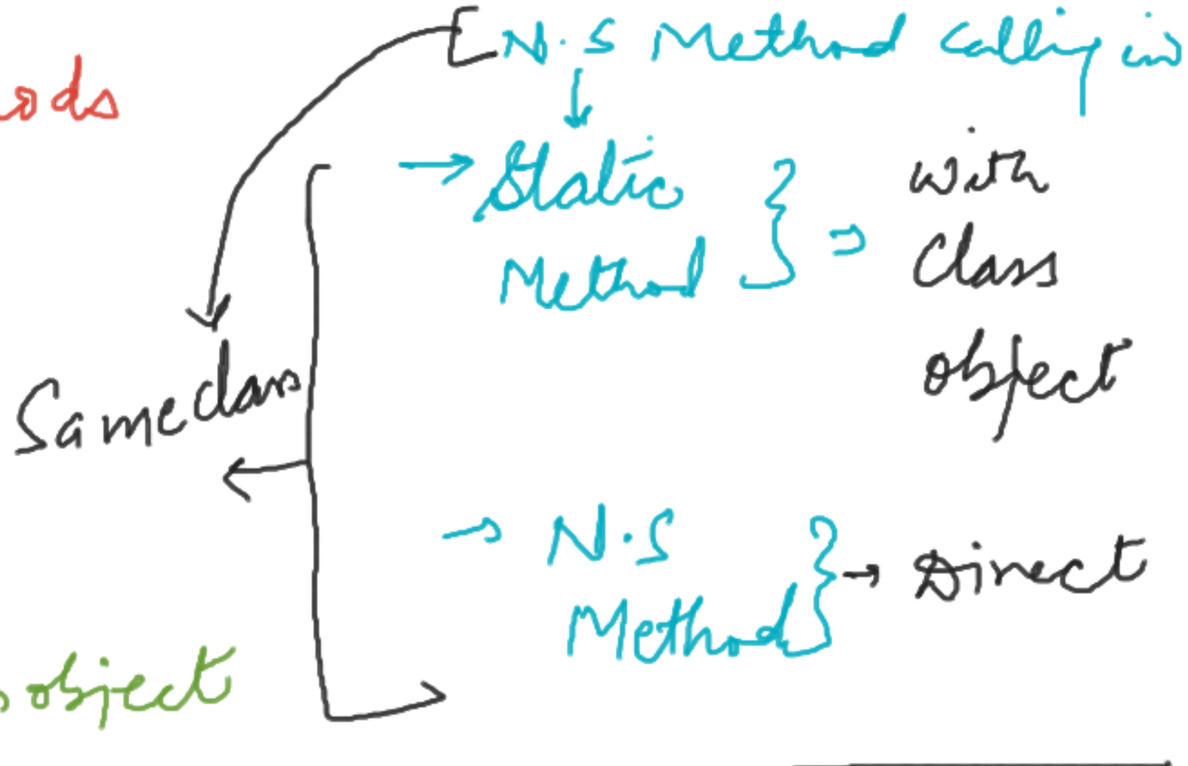
default keyword → N.S Methods



inheritance :-

Parent → N.S.M → with child's class object

Child → Static - with child's class object

→ N.S - directly

Another class → S.M - With Class object

→ N.S.M - With Class object

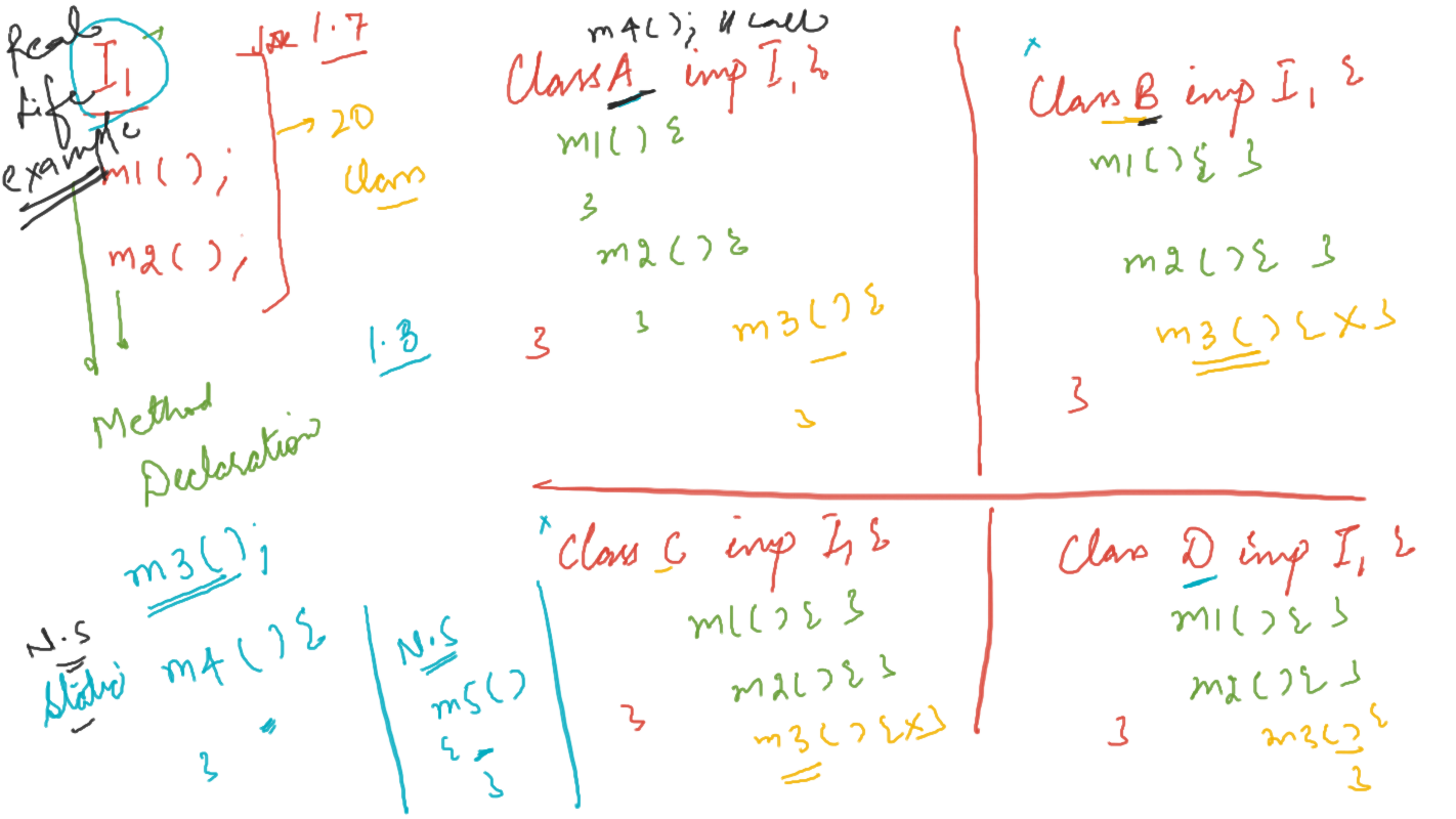
Scenario → I → default Method → N·S·M Method

↓ calls

Class → S.M → with class's object

N·S·M →

↓ directly calls


real life example
I1
m1();
m2();
m3();
Method Declaration
1.7
20
class
1.8
m4(); // call
Class A imp I1 {
m1() {
}
m2() {
}
m3() {
}
Class B imp I1 {
m1() { }
m2() { }
m3() { X }
}
N.S
Static m4() {
}
N.S
m5()
{
}
class C imp I1 {
m1() { }
m2() { }
m3() { X }
}
Class D imp I1 {
m1() { }
m2() { }
m3() {
}

Scenario2 :- I_1 → default Method → diffName – m1

I_2 → default " → " → m2

class A implements I_1, I_2 {

}

Scenario 3 → I_1 → default Method → with Same Name - m4()
(N.S.)

I_2 → " → " → m4()

class A implements I_1, I_2 {

}

Scenario 4:

I_1 → default Method → with diff name - m1

↑ extends

I_2 → " → " → m2

Class A implements I_2 {

3

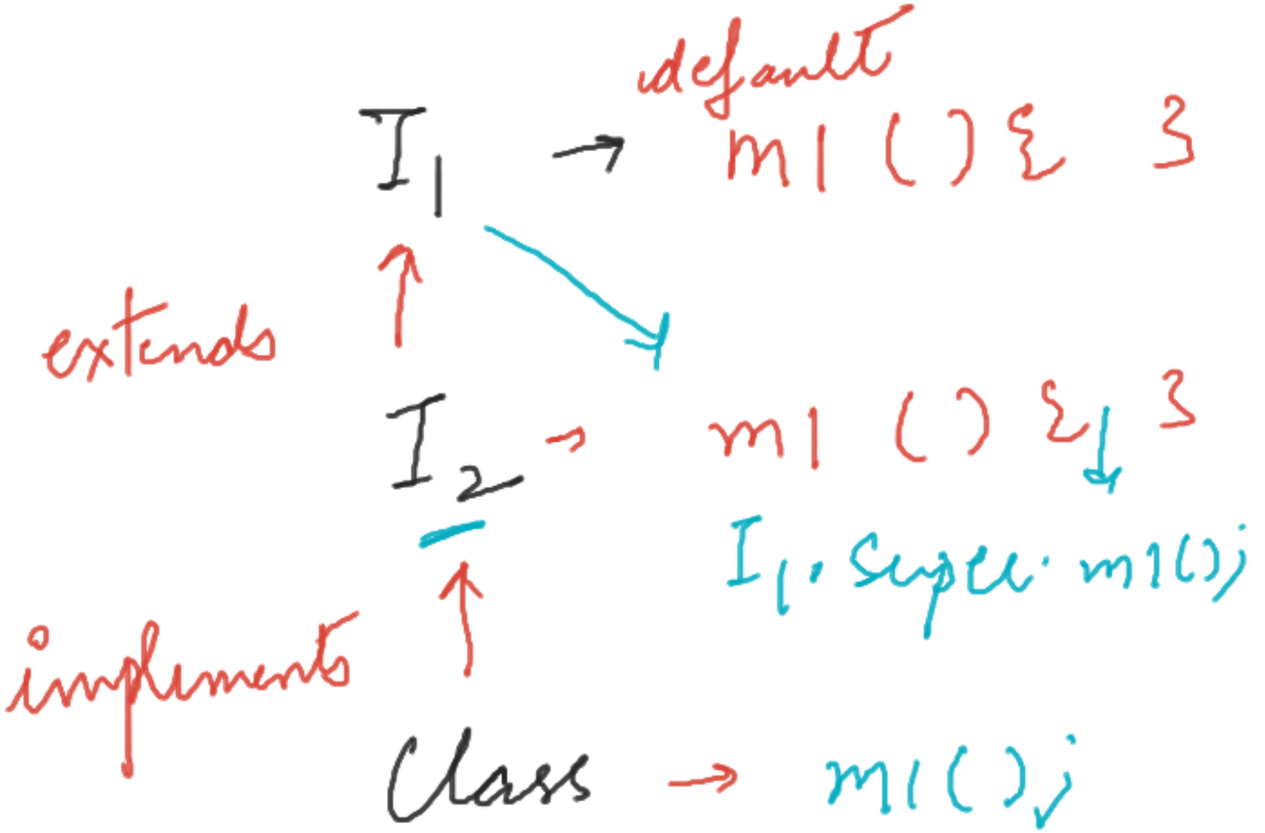
Scenario 5 :- I_1 → default Method → with same Name - m5()

↑ extends

I_2 → " → " → m5()

↑ implements

class A



Scenario 6 :- I1 → m1() 3

↑ implements

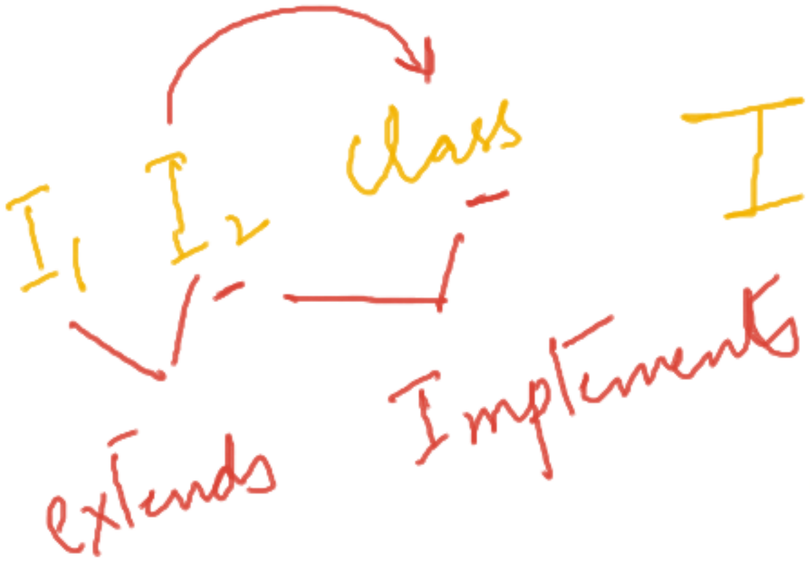
Abstract class → abstract / N- abstract

↑ extends

Class =>

abstract class - implements Interface

Class A extends abstrclass



Interface2 extends Interface1

Class 2 extends Class 1

class implements interface

Scenario 7:

X { I1 → m1

abstract class → abstract m2(); non-abstract m3(){}

class A extends absclass implements I1 {

}