Sunday, 25 February 2024
9:20PM

Abstraction

↳ Process of hiding the implementation and only showing functionality to the user.

↳ Focus on "what" instead of "how"

↳ Abstract classes

↳ Interfaces

1) Abstract class

↳ abstract

↳ some methods

↳ cannot create an object for it

↳ has abstract & non abstract methods / method bodies

↳ can have static methods

↳ can have [illegible]

↳ can have final methods

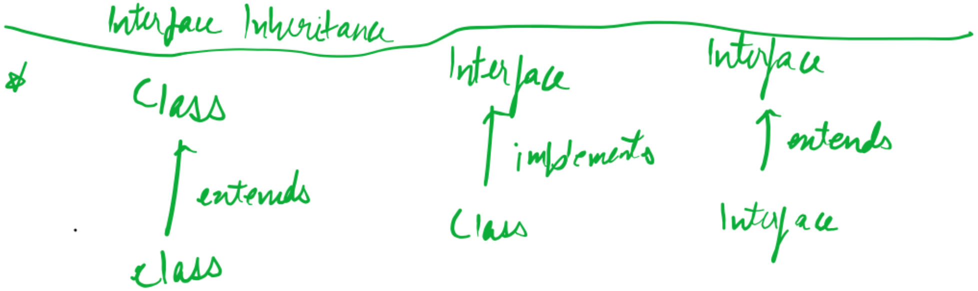
↳ May or may not provide complete abstraction.

2) Interfaces → 100% Abstraction

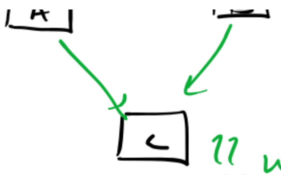
↳ Blueprint of a class

↳ It will only have abstract methods but no implementations.

↳ cannot instantiate interface.



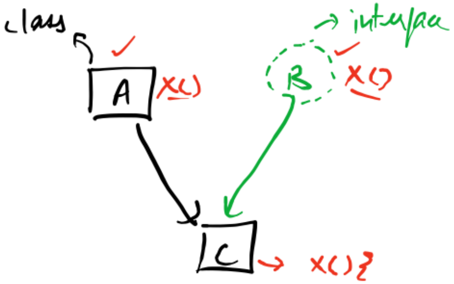
Multiple Inheritance in Java:



?? which x()
will c choose
→ Not decided

⇓

Use interfaces to
support this



```
→ x(){
     // ∵ c needs to
        implement B.
  }
```

· Abstract class

→ Abstract + non abst methods.

→ Do not support multiple
inheritance

Interfaces

→ only abst methods

→ They support multiple
inheritance

→ static, final variables

→ final, non-final, static
non static variables

→ can implement an interface

→ extends

→ 0-100% abstraction

→ Interface cannot implement interface/abstract

→ implements

→ 100% abstraction

$(a+ib)\ (c+id)$

$ac + i(ad+bc) - bd$

$ac - bd \rightarrow r$

$ad+bc \rightarrow i$

$10 + 10\ i$

$1000 + 10000i$
$- 100i - (1000(-1))$

$= 2000 + 9900i$

PIZZA BILL GENERATOR---> https://vimeo.com/780838079/cfc70c4521?embedded=true&source=vimeo_logo&owner=182753595

Github --> https://github1s.com/acciojob/pizza-billing-tushargarg2000