OOPS - 2

Thursday, 9 May 2024 9:07 PM

Encapsulation

↳ wrapping up the data and methods in a single class

OOPS

↓

Security

[private int age;]

⇒ Access Modifiers → (4)

→ private : only inside the same class.

→ default : within the same package → collection of classes.

→ protected : default + child class (can be outside the package)

→ public : any where

→ private ↳ getters and setters

↓ ↓

returns the val used to set some value

→ class BankAccount {

public int balance;

private

↳ public void setBalance (int balance) {

if (bal < 10000)

balance = balance

}

}

BankAccount kunal = new ();

kunal.balance = 10k

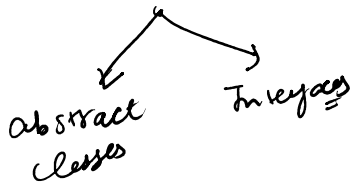
Abstraction

↳ hiding the details and showing only the functionality

[System.out.println("Hello");]

↓

Abstraction



Abstract class

→ abstract and non-abstract methods;

→ you cannot instantiate a abstract class.

Interface

↳ 100% abstract class

→ Cannot instantiate

→ final and static

value will not change.
It has to be initialized when declaring.

Inheritance

↳ getting the properties from the parent.

Parent → Child

- Class extends class
- Class implements Interface
- Interface extends Interface

Child object

↳

Child() { calling Parent }

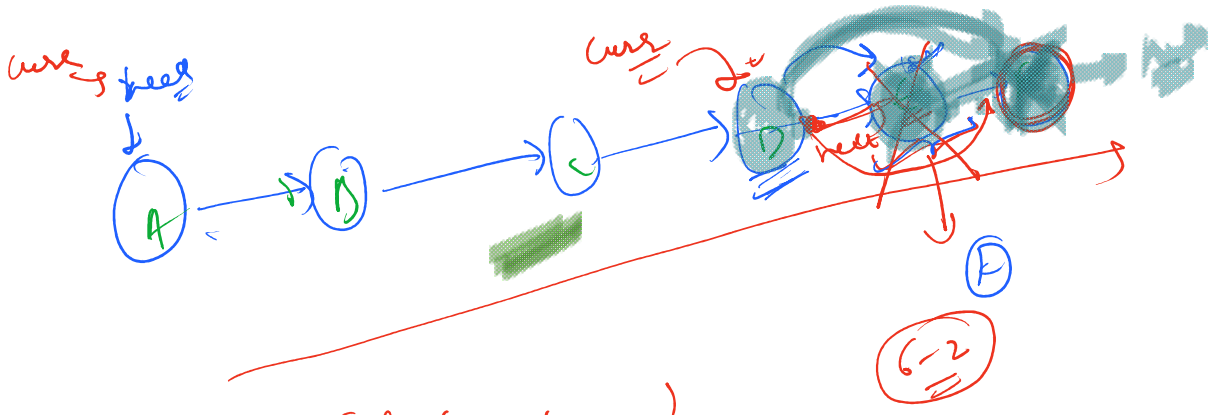
{ Child() ' calling Parent()
}

→ super() → parent constructor

→ super.func()

→ this.func() → methods of current object

→ this() → my own constructor



{ for( ) curr = curr.next

→ curr.next = curr.next

curr.next = C

curr.next = E.next

curr

6

null

curr.next