

Abstract Class

Interfaces

DATE:

→ (जिसका लिया constraints, rules का लिया लॉगिक हो).

And is yes. क्योंकि इसका implement करने का automated system हो.

→ वो rules, वो conditions, वो constraints जो हमें real world

mai follow करनी पड़ती है, जो real world mai exist करती है, वो

वे constraints, issues हमें automated system mai bhi hene chahiye.

It means hum ne un constraints ko somehow apne S/W, apne project mai implement karne hain.

e.g. → For e.g., hum automate करना कि rate ~~textile cloth~~ ko

Masood textile Mills

Eik employee exist karta, वो कोई employee हो सकता. It can be manager, it can be supervisor, labour, administrator and so on...

⇒ Real world mai jo employee exist karta है, us employee ke

koi id, rank, salary, unique email address etc. होंगे. वे वो properties हैं जो कोई employee के with real world mai exist करती है. Puri Masood textile Mills mai aisa koi employee exist nahi hoga jis ki salary na ho, rank na ho, unique id na ho, email na ho.

→

For e.g., hum PV ko automate करना कि rate. PV mai koi aisa student ni hai jis ka roll no. na ho, jis ke koi cgpa na ho, email na ho.

→

Hum shape type ko implement करना कि liye IW develop करते हैं.

वो 2-d, 3-d shape हो सकती है. Puri real world mai

aisi koi shape exist nahi jis ke area na ho. Har shape

ke koi no koi area 3000 है. Puri real world mai koi aisi

3d shape exist ni kati jis ke koi ekse aux volume na ho.

→ It means ye vo constraints, ye vo properties hain, vo rules hain jo koi entity ke sath is real world mai exist koi si hain.

→ Main ap se kہتا ke hum PU ko automate karne ja raha hain and waise the team lead, project manager.

Aپکے sath 10 logo ki team team kar ri hain and you have to decide ke mese is system, is S/W mai aisa koi student exist ni kerna chahiye jis ke email na ho, jis ka roll No ne ho, jis ke cgpa na ho ---

→ Being the team lead, you have to enforce ke ap ke project mai aisa shape type ke koi object ni hana chahiye jo waise ko implement na kr reha ho, volume ki implementation na de rakh lo.

Why? Qk real world mai shape ke jo objects exist hain us aise ko bhi implement krte hain, vo volume ko bhi implement kr reha hain.

→ Being project manager, u have to enforce ke ap ke project mai aisa koi employee exist ni kerna chahiye jis ki salary na ho, jis ke rank na ho, jis ke koi unique employee id ne ho.

→ what does it mean?
jo project hum design kr raha

→ It means ke^k, design level, par, we have to make sure ke hum kuch aise rules, kuch constraints

yahan pr implement kr de jin ka jb tk ballow
na kiyे jaye tb tk us project mai un ki
existence na ho.

Project mai existence na hoga

? जब

→ Koi student type ke object exist na kro jo un rules ko follow ni kr reha. Hame:

→ Aisa koi employee type ke object exist ni kr skta, employee type ke object create ni hokta jb tk vo un constraints ko follow ni keta.

→ Aisa koi shape type ke object ni hokta jo un Sab rules ko follow na kr reha ho jo ke real world mai exist kr rete.

→ Basically being team leader/project manager, hum ne apni teams, apne developers ke liye, design level pr ye constraints is trah se implement karne hai ke jb tk un constraints ko koi user, koi developer fulfill na kri, tb tk unki existence na ha. ← Project
- It's object reside in memory kii jisके ↓
↓

Q: How we can enforce those rules, those conditions in our project?

A: In java, we can do this by following

concept of Abstract classes and Interfaces.

जो rules कि पर्याप्त हैं उनके लिए
- C++ apply

⇒ C++ mai term pass thi pure virtual function
boundary के लिए ऐसी functions लाइ
- Only declaration, only definition - (जो) वही exist
- There is no body with this function.

U can say function → pure virtual function.
without any body

- C++ के लिए जो body नहीं है उसका empty body

For eg

public void show();

Yeh ek function ki declaration
hai, body ni hoga iski.

DATE: _____

Curly braces jo hum wo karte hain function
ki body define karne ke liye, vo ni hoi.

Pure virtual functions, fir se functions wali JS-C++ C=

-& M

JS-C++ function concept JS-C++ Java C=

Abstract function

Abstract Function: A function without body. A function without implementation, even though function does not have empty body (curly braces, body ni hoi).

→ Aise functions ke sath hum abstract keyword
likhe ge. • public abstract void show();

Abstract class: A class which has atleast one function or even more methods as abstract, that class must be declared as abstract class.

For eg

• abstract class Employee{

 public abstract void show();

}

class T = abstract JS class JS

abstract class JS

function JS

abstract JS

-& JS नहीं लिया declare

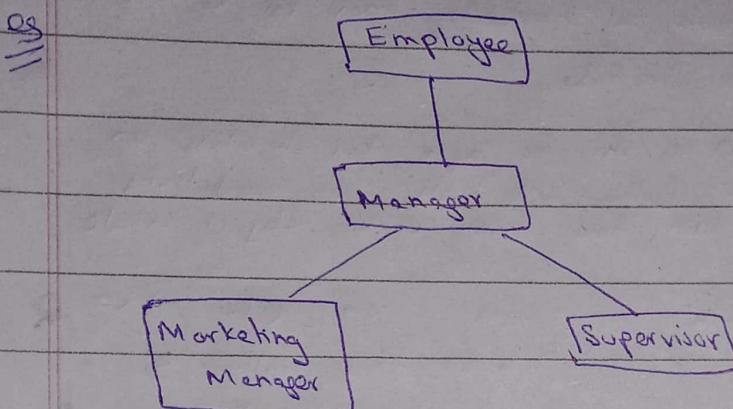
→ You cannot create object of an abstract class.

If I will try to write - Employee = new Employee();
Compiler will give me an error.

کوئلے کا object نیس بنائیں۔ DATE: _____

Q: What is the purpose of this class if we can't create its object?

A: Now it is the responsibility of the child classes to provide the implementation of abstract functions of parent class.



class abstract class Employee Employee

↳ responsibility ↳ child class (manager class) ↳

To provide implementation of all abstract

functions which are present in the parent class.

Inheritance → Parent class ki vəri functionality child class me
available hogi. ✓

inheritance of Gx2 defect function 15- patient ۱۰۷ ←

-if \exists available \cup child function \Rightarrow w \rightarrow 3

① It means manager class mai bhi ek abstract function define kar dijiye hai... To manager class hi abstract class likhi jayega.

② over-ride کو کسی کو manager class کا اکثر function کا parent class کے وہاب کر دیتی ہے۔

- Over-override - It's available in manager class abstract function
Now we can create object of manager class.

③ In-case, manager class has to implement:

ni kisi. Now it will be the responsibility of marketing manager and supervisor class to implement those abstract functions.

Q: کہ اسے کیا implement کیا جائے۔ کیا کوئی object

G: Is track hum is hierarchy mai koi ke thi object ni create kr rde.

A: Tou hum re sti thi ke agar koi bhi class un constraints ko follow ni kisi, un conditions ko follow ni kisi tou hum un ke object create ni kr skte.

→ in constraints or rules is UH-PI <= Q
→ if in UH follow is

abstract child(manager) ← → Parent (Employee) ← A

supervisor
marketing
manager

→ abstract class کوئی پڑھنے کرنے declare abstract
→ کوئی create کیا object کوئی کر

→ implement کیا rules کیا UH-PI basically is
→ کیا exist کیا realworld نہ

→ کوئی proper implementation کیا functions کیا کوئی کیا کر

→ کوئی object کیا کر

→ کیا realworld کیا کوئی لے

→ کیا existence کوئی

→ We can't create an object of Abstract class.

But we can create reference variable
of Abstract class.

For 10 Employee's

This is not an object.

e is reference variable of type Date
Employee.

→ OOP

Class allows us to define a new type.

`Em` is a variable of type Employee.

بھی کسی reference کے object کی کمی کر دیا۔

Benefit of e?

e is simple variable. It contains nothing.

⇒ In case of abstract class, the reference variable of parent class may contain the reference of child class, may contain object of child class.

abstract class 51 & child class 31 so ↵

concrete class - Book / Customer / Order / Product / Supplier / Employee

Concrete class: which has no abstract function.

For eg Manager class is concrete class

اگر ایک abstract function ہے تو کوئی بھی

You have ~~the~~ new Manager();

e is containing
object of
child class.

This is an object of management theory.

Employee class ke direct object hi bna skte
gk vo abstract hei.

Let's reference variable of Employee class may contain object of child class.

→ And that will be concrete class.

3. اگر کلی جا سکتی ہے class مرف اگر class statement ہے
 function ہے جس میں کوئی نہیں - اسی نہیں abstract class ہے - کوئی concrete
 نہیں ہے abstract

- abstract class Employee{

```
    public abstract void show();  
    public abstract void display();  
}
```

DATE: _____

abstract class Manager extends Employee{

→ display valo function jo parent class mei hei

No due to inheritance child class can't override base

Tou manager class mai aise function exist keta hai

jo abstract hai... Ager koi ek class mai ek

function thi abstract ho tau ve class ko abstractнического.

↳ implementation func. ↳ display ↳
↳ concrete ↳ in ↳ abstract , manager ↳
↳ class ↳

→ Ager manager class implement ni koki abstract
function parent ka tu child classes ki responsibility
uski supervisor, marketing meyes

hai ke un ko implement kre.

→ Assume marketing manager is us abstract

function to implement methods for no concrete class

bin jayo gi. Supervisor ne implement ni kiya thor uske

object we create no sets.

→ After manager class we abstract function ki implementation

provide kredi to uski child (marketing manager, supervisor)

mai - vo rules follow kune ki gaddat ni.

Q ni zoooyet? Qk jk yahan functions ki inheritance

hagi tuv vo functions inhosit hoge jo override ho chuke hei.
Vo over-rided konse hoge? Jo concrete functions hoge. Ab qk vo
functions abstract ni hei, concrete hei tuv supervisor aur marketing
manager mai apko vo implement karne ki责务त ni hei.

اگر کم جائے تو ان کو دوبارہ over-ride کر سکے ہیں لیکن اب ان کے اور یہ rule نہیں لا اپنے ہو گا۔

⇒ Abstract class ke ander date members bhi ho sakte hain.

Abstract class ke under functions thi ho skte.

the implementation of `g`. `g` concide with functions with

-نیوج burchids یا -available

- abstract class Employee{

int x;
public void process()
{
} }] → Q: क्या ये functionality
child class mei
available hogi?

```
public abstract void show();
```

public abstract void display(); A: Yes, it should be

Here, Due to
inheritor's
undiscordancy (chate)

concrete hi T_0 , vo

child class mai
available today.

① Eik simple class hai jis ke name MyClass hai aur

abstract sketch he is up to.

exist abstract display something
functional shape or.

Eik concrete function stages

hai show ke go sirf eih

line display kr reha.

concrete का एक class (ग)

- U^{v} abstract \Rightarrow set
functions

- public abstract class Myclass{

public abstract void display();
method();

```
public void show()
```

System.out.println("This method
is declared in Abstract class");

② Ek child class hai MySubclass. MyClass se inherit hoga hei.

- displaySomething() ye abstract function tha, usko main yahan implement, override kr rhe hain. ye parent class mai abstract declared hai aur isko yahan override kr rhe hain.

- display() function ka thi yahan override kiya hai.
- show() ke function due to inheritance yahan pr available hoga.
- Main function: child class ka object banaya. show ke function due to inheritance parent class se is child class mei available hoga.

• public class MySubclass extends MyClass{

```
public void displaySomething()  
{
```

System.out.println("This method is
'displaySomething' and declared in abstract class");
}

```
public void display()  
{
```

System.out.println("This method is 'display'
and declared in abstract class");
}

```
public static void main(String args[])
```

```
{ MySubclass b = new MySubclass();
```

```
b.display();
```

```
b.show();
```

```
}
```

```
}
```

→ Run it -

Ques: Ye jo reference variable yahan banaya hai,
apne MySubclass ke ni karna. Apne karna hei
MyClass ke.

Parent class ka reference variable create karna
hai aur vo object of child class create kar
rhe hoge ⇒ MyClass b = new MySubclass(),

Aur phir check karna ke, hum abstract class ke object ni ban skte
tum kya hum log uska reference variable create kar skte hain?
Vo reference variable kis trah se child class ka object ke
contain hoga? Aur unke functions child class ke hain?

- Ques:**
1. Can we declare an abstract function as private?
 2. Can we declare an abstract function as static?
 3. Can we declare a constructor in an abstract class?
If yes? when it will be executed?
 4. Can we declare a constructor as an abstract?
 5. Can we declare a datormember (i.e., variable)
as an abstract?

→ Parent class (abstract class) ref variable = object of child class

And then calls,

(Polymorphic call generate known for
deck if it is possible?)

6. Can we declare a class as an abstract, even if
it does not have any abstract function?
If yes, then what would be the behavior of
this class?

→ Feed back (vahan saare ans hai).

⇒ Interfaces

constraints of design for its purpose is Abstract class
level define E.g., rules of apply
51 - abstract E.g., of functionality of class (51)
functions all - (51). functionality of class will be
- available of body E.g., 51.

51 abstract nahi mein kya hoga [Interface] ←

51 aur 51 abstract functions hain 51 hain - N

51 purpose to abstract class 51 is 51's purpose 51

syntax wise 51 - 51 concept wise 51 - 51 difference

- N

⇒ Interface ko likhne ke liye ap log class ki bajaye interface ke keyword likhe ge. Uske baad interface ke name (interface Shape)

② In interface, all methods are abstract.

→ जैसे यह concrete function नहीं हैं तो वे अब्स्ट्रेक्ट होंगे।

DATE: _____

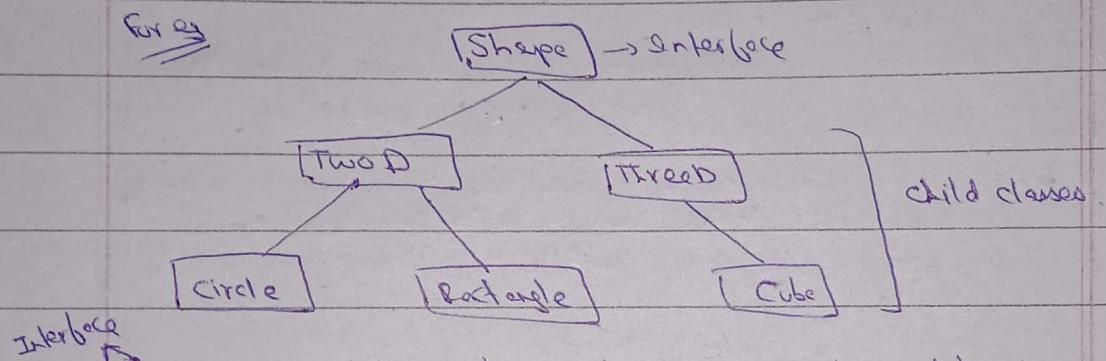
→ जैसे यह by default लॉगिक फ़ंक्शन हैं तो वे अपर्याप्त होंगे।

functions without body ← [- void show();
- void process();
- void display();] → जैसे यह define concrete function हैं।

③ By default, all these functions are public and abstract.

So, unke sath likhne ki zaruri nahi hai..

④ Now, it is the responsibility of the child classes in hierarchy to provide the implementation of all these functions.



→ Assume kilete, shape class mai ek abstract function hai show() ka. void show()

→ TwoD class mai ek abstract function hai jis ke name Area hai void getArea()

→ ThreeD class mai do function hai. void getArea()
void getVolume()

① Interface mai sare functions abstract hoga. Koi concrete function nahi hoga.

② Agar koi class interface se inherit hesi hai
& inherit के shape के TwoD class में.

Keyword to implements के लिए inheritance का use

• class TwoD implements Shape {
} }
In case of interface & ↗ use

interface class (जैसे यह)
→ ↗ inherit के
Keyword to implements
→ ↗ use

Q. 2. اے interface نے اے responsibility کی TwoD class اے
اے implement کر کے ان سب کے اے functions
well & اے اے implement کر کے ان کو class اے اگر ①
good
implement کے ال class اے اگر ② DATE:

اے اے abstract کے class اے 3 بھروسے

- abstract class TwoD implements Shape

Q. 3 کے وک Abstract : Q

functions اے ال class اے اے اے Inheritors : A
اے کے class اے اے - اے abstract کے اے available
کے class اے اے اے abstract کے function
اے abstract class

⇒ Suppose, ye class eik aur abstract function declare
kr deta hai, void getArea(); • abstract class TwoD implements
Shape

⇒ Now as per hierarchy, it is abstract void getArea();
responsibility of Circle class

to provide implementation of • class Circle extends TwoD
all functions.

implements show اے کے ←
→ getArea ← کے
→ getArea ← کے

show() { }
getArea() { }

concept اے exact اے ہی اے اے interface ←

اے 2 کے
کے rules

اے constraints اے 1
کے follow

• interface Shape {

void show();

abstract class TwoD implements Shape

{

abstract void getArea();

class Circle extends TwoD

{ show(); }

{ getArea(); }

⇒ function when declared in interface, there is no need to write public or abstract with it. There is no issue even you write it. But by default, they are public and abstract.

Now it is the responsibility of child classes to provide the implementation of interface.

Statement in Book: Whenever, a class is implementing an interface, basically they are signing a contract.

And the contract is, now this class is responsible to provide the implementation of all the functions in the interface.

⇒ If skip any function in a class then this class will be declared as an abstract class. It is not object of class (जबकि वह object का नहीं होगा)

⇒ If class is part of hierarchy (जबकि वह object का होगा) then it must implement all the functions (जबकि वह object का होगा)

Otherwise,

It is not object of class (जबकि वह object का नहीं होगा)

- If override & show function then it enforces (Sheep class)

- If override & getArea then it enforces (TwoD class)

getArea or show is the responsibility of Circle class.

It is not object of Circle class (जबकि वह object का होगा) it implements

functions of responsibility (cube class) जैसे कि वह करता है।

1- show
2- getArea
3- getVolume - यह implemented करता है

यह वही करता है जो एक को करता है जैसे कि वह cube class करता है।

In memory object का होगा जो एक abstract class का होगा

जैसे कि वही होगा

⇒ Interface bhi vahi hai kuch differences ke sath.

⇒ All data members in interface are public, static & final.
By default.

Public: They can be accessed from anywhere.

Static: They can be accessed without creating the object of this class (interface).

Final: They are constant. their value can't be changed.

→ By default, all determinants in interface.

Q: Actual difference b/w abstract class & interface?

→ abstract function ↪ abstract class

ویژگی های کلاس abstract functions interface اور

فی بات کو some اور different کا کیا ہے؟

① We can't create object of interface but we can create reference variable of interface.

We can write,

Shape is

```
s = new Circle(19)
```

This reference variable will contain object of child class.

5 methods \hookrightarrow L (JS) & child class implement

A: Actual Difference:

\Rightarrow Interface provides u the concept of

multiple inheritance,

There is no concept of -~~is~~ ~~is-a~~ if multiple inheritance in Java.

But plz remember,

Java provides the concept of multiple inheritance in Java using the concept of interface.

- \triangleleft concept \triangleright scenario \triangleleft interfaces \triangleright ...
- \triangleleft executable

How? → Multiple inheritance
ke
Many possible ways

① One interface may extend from other multiple interfaces.
inherit

- interface I1 extends I2, I3, I4

- کسی implement کرے کے class کو کسی بھی ایسے

کے implementations کی responsibility class کو provide

- I1, I2, I3, I4

- class Text implements I1

{

// I1

- کے scenario کا

// I2

- کے responsibility Text class

// I3

// I4

- کے methods کے interfaces کے ان

- کسی implement کرے

→ Other inheritance here.

Tum kya I2, I3, I4 in teeno ki functionality I1 mai available hoga? Yes.

- کسی کسی implement کرے کے class کو جب ایسے

methods کے وہ ان I1, I2, I3, I4 oo کے available hoga

- کسی کسی implement کرے

- کسی کسی skip کرے کے method woh کسی class کے ان کے

- کسی کسی object create کے Text class

- کسی کسی partially implement کے interfaces کے class کو ایک ایسے

- کسی کسی inherit کے نئے new function کو کاہی

- کسی کسی available class کے abstract function oo کے ہوں گے

- کسی کسی کو اور کسی abstract کے class کو اور کسی

- abstract class Text

implements I1

{

}

- کسی کسی object create

→ Eik interface oik interface se hi inherit hokte hain.
Eik interface koi class se inherit ni hokta.

→ multiple interfaces

• کسی کو inherit اور interface کے class کا لے

- class Text implements I

- کسی کو کسی inherit اور class کے interface کا لے

- interface I extends Text X (wrong) DATE:

② Eik class multiple interfaces ko inherit kar saki.

• کسی inherit پر directly کے class کا لے

- class Text implements I1, I2, I3, I4 • multiple interfaces

- کسی کو working scenario کے سامنے کو کسی کو

- کو different syntax کو concept

• فریکو کو inherit interface کے scenario کے -

- کو implement کے class

- کسی کو directly implement کے واروں کے class کے کو -

Ye koi multiple inheritance hai.

③ Eik class ek waqt mai koi class se extend bhi ho

skti awa usi waqt koi interface ko implement bhi kr skhi.

کسی کو inherit کے class کو کسی کو کسی کو کسی کو

- کے interface کو کسی کو implement کو نہ

- class Text extends sampleDemo implements I

- کو multiple inheritance کو نہ

Using interfaces کو کسی کو کو multiple inheritance کو کسی

Diamond Problem:

Scenarios \Rightarrow multiple inheritance

- class Text implements I1, I2, I3, I4

{

}

Assume کو in sub interfaces (I1, I2, I3, I4) mai

ek common method hai jis ke name show hai.
void \rightarrow abstract.

• interface I1 {

 void show();
}

interface I2 {

 void show();
}

interface I3 {

 void show();
}

interface I4 {

 void show();
}

Wala:

interface (اے جے) =

کو ایک نام کے

- show کے function

I's (اے جے) Text class

- (سی) implemented کے

text class کے

کو ایک responsibility

کے functions کو ایک class کے

functions کے implement, over-ride

- کے same signature - کے same

- کے multiple functions کے ایک class کے

- کے Text class کے ایک functions کے

کے اسی کا طلب کے signature کے same کے اس کے

کے functions کے ایک Text class کے

کے show کے

• class Text implements I1, I2, I3, I4

{
 void show();
} --- *multiple functions*
}

\Rightarrow object کے referent variable کے object کے I1

child class کے referent variable کے اور

proper implement کے I2, I3, I4 contain

- ΣΔ τελος

```
ref = new Text(1,
```

1Q: Can I write this statement?
⇒ ref.show();

line:

Yes, I can write this sentence.

2Q: ~~Can~~ I write this statement too?

I2 ref:

```
ref = new Text();
```

ref.show(),

\Rightarrow Yes, we can write this

Ex. rab - o { call → show it in rab

Σ - variable & I type

Show function

3Q: 13 ref:

806. show(); 09 < 806 14:

• obituaries show 5-23 - e variable & 13 type

⇒ But note, Text class mai tuu kum ne show ke eik function likhe has.

N diamond W lava N E S W claims
postm

- void shout, it is a void function which is

referent variables الک ۳، ۴ ک ۵۱ اور ۱۵

1. 5. 1.  call for

overload. We can function is IS to Text

- use statements illegal 31, 2Q, 3Q 5

This is diamond problem in Java. It is interface
argument between community & against Java &
DATE: _____

Community پر behaviour کے Java 3. ←
method یعنی کہ یہ دلائل ہیں ہے جو اس کو وہ کی بھی
ہے same name کے signature same
H- implementation پر ایک بھی خاص interface
کے درجہ تک اور جائز ہے اور
کو انہاں کو کر دیتے ہیں اور میں کوئی
نہیں ہے diamond problem

→ Jeje ke against community is Microsoft . Csharp
ne is ko proper answers kiyo.

لیٹریلی کوئی دیکھنے کا
لیٹریلی کوئی دیکھنے کا
لیٹریلی کوئی دیکھنے کا
لیٹریلی کوئی دیکھنے کا

- class Text implements I1,I2,I3,I4

```
{ void TA :: show()  
{  
    -- -- --
```

```
void I2::show()  
{  
    -- --
```

```
void I3::show()
{
    -- -
}
}
```

DATE: _____

کروں calls 310 1Q, 2Q, 3Q پہلے بھی اپنے

call ہے is responsible to generate ہے

بھی کوئی claim کرے گا Java میں ←

نہ لے لے گا still کے Java

There is no diamond problem in Java.