

17

NOTE - 2

1

22 | 1092021

Why multiple inheritance not supported in Java?

Diamond Issues

A : public void show() => overridden
 {
 System.out.println("Hello");
 }

B extends A
↓
Show sop(2)

extend
show ($\text{Sop}(3)$) → overriding

class D extends B, C
{
}

- ④ There will be confusion for D if to extend 'A' or 'B' or 'C'
 - ④ What is diamond problem?
 - ④ Multiple inheritance in Java is not supported through class but it is supported via interface.

Invocation of constructor in inheritance hierarchy

Constructor of sub class

Digad variable
all State is not initialised to our expectn Not added by user Progr

Java c provides super()
(args less)

missing const constr invoked
fix

added by user prog

invoked super()

... Java Insert
super();

Incomplete state
InitializaFn.

tries to call immediate
super class
atching const r
Comple state
initialised

④ subclass cannot use super class private data member but they are part of subclass.

⑤ how to access super class members?

⑥ Getters and setters

⑦ super. () → But only public are called Super

⑧ it means immediate super class, to invoke super class matching constr.

⑨ accessible from subclass only.

⑩ this. () ≈ super. () → But we can access public member

Summary: - subclass is super class + more extra + we can modify
what we got

⑪ Faculty is person.

Overriding ≈ modification

toString → is defined in object and returns
fully qualified name @ hashcode.
Object required

What is need of overriding toString?

To replace hashcode version of hashcode

e.g package bank;

before overriding Fully Qualified Name:- bank.Account @ 35454
after overriding → acct detail

Upcasting

Object o = new faculty(); no javac error ?

* Implicit type casting done by

upcasting → implicit not forced. Javac.

Sop(o) → what will it call internally?

any reference variable → it will implicitly call toString();

↳ Sop(o, toString()) → printStream → println(Object ref) ->

→ String.valueOf(Object ref) ... ref == null ? prints null b/o calling toString

Sop(o) → 2 stages

Java C → resolves method binding by type of ref

JVM → resolves method binding by type of object referencing instance.

→ we are using Indirect referencing b/o type of ref and type of object reference is different

Object o = new faculty

Java.lang.Object ↗ Faculty

Where Compiler check if toString exists?

Ans java.lang.Object and every time object will have toString :-

What happen runtime?

* Sop(f) → java C : fruit

JVM goes by calling ToString on mango class.

Object o = f :- still no error because fruit is object

4

★ ★

Polymorphism ★ ★

① one functionality → multiple forms → Carbon

fullerenes
diamond

Graphite

Method binding

② linking a method to actual method definition.

Static binding (compile time)
(early form)

How to achieve:

Method overloading

dynamic (runtime)
(late binding)

method overriding

Method overloading: Rules

③ can we add o.m in same class?

e.g. open printStream
class {
 println (int i)
 println (boolean)
 println (String)}

e.g. → Array → class

always. toString();

④ can we overload method super in subclass
→ Yes in inheritance - al.

⑤ o.m is checked by compiler. Rules. ★ ★

signature → different (no/type/both),

return type → ignored by compiler. ★ ★

Void test (int i) { }

Void test (int i, float f) { }

Int test (int i) { }

Error b/o ignored by compiler

~~Ans~~

Q Can we overload static method?

Yes. e.g. `Arrays.toString();`

JVM:- resolves method binding by signature.

`byte b;`

~~int~~

`short s;`

`long`

`float`

`go(b);`

Q If we comment `byte` corresponding function then it will check for nearest

e.g. `void test (byte b);`

`void test (short s);`

`void test (long l);`

`void float (long f);`

automatically upgraded if commented

`go(l);` → ① go to exact match
② go to nearest match

Where will long go for?

Q `go(l) long`

`void test (float f)`
`void test (double d)`

nearest

`go(f) →`

~~void test (long f)~~ X
~~void double (double)~~ X

If we comment `long` it shows java: error

test(2, 3) X
test(2, 3, 1) ✓
test(2, 3, 9) ✓

AAA.

Static void test(int a, double b)
(double int)

Ambiguity

double double

Java c

Method overloading

- ⊗ detected by Javac (early binding) via method overloading.
- ⊗ No rules on access specifier
- ⊗ Same method name etc type ignored.
- ⊗ auto upgrade to nearest.

Summary chart 9:47

Overriding

Day 5 help

can exist in hierarchy overriding only.

Dynamic Polymorphism

(late binding)

- ⊗ dynamic method dispatch.
- ⊗ decision is taken by JVM.
- ⊗ Compiler cannot resolve → blo if resolves by argument and all same argument.

what is DP method overriding

⊗ fruit f = new orange(...);
SOP(f.taste);

⊗ Compiler cannot resolve the type of object.

⊗ JVM calls subclass version.

☆☆☆☆
☆☆☆☆

- To enable method overriding click on "Virtual" keyword
- All methods in Java are virtual.
- Hence implicitly virtual

100% all are overridden → No

- Only public method can be overridden

AA Interview

Why static methods are not overridden? D-5

They are at class level hence JVM cannot resolve by object

Method overriding → is all about detecting object

- Final → also cannot be overridden.

Super-class form

Sub-class form → same method name, same args,
JDK 1.5 introduced also same; later covariance
is brought in. → or covariant type

Covariant type

Package P1
class A {

```
A getInstance()
{
    return new A();
}
```

Package - 2
class B extends A {

```
B getInstance()
{
    return new B();
}
```

Can return A or B

this will consider new method
as A is not accessible, due to
default

but same package
Covariant ref type.

* In comparison to overloading there is scope issue.

⊗ Overriding → access specifier same or wide you cannot narrow down.

protected also → error because narrower

Q) If super class has default scope?

Subclass → default, protected or public

Q) If super class has public scope?

Subclass → public

Dynamic Polymorphism

① late binding

② must have same name/argument, ret type same or covariant

③ cannot restrict scope.

Java annotation

⊗ @Override annotation.

→ use for Java and Jvm

⊗ Available after JDK 1.5,

⊗ Metadata → more info meant for compiler or JRE (Java tools)

@Override



⊗ It is annotation meant for Javac

⊗ If method level anno, that appears in a sub class.

⊗ It is optional but recommended.

public class Orange extends fruit

@Override → To tell compiler you check
public void taste() {} for rules for overriding
method

`@Override` → compile time annotation

- ↳ We place in class before method
- ↳ We write in subclass overridden

⊗ It is not compulsory but Recommended → if don't write if we make mistake in arguments → no compiler errors but we have runtime errors.

(⊗)

Event organizer :-

⊗ `@Override` → only if inherited and modified.
`public String toString() {` ✓
`}` → student

`@Override`
`public String getfirstname()` → error b/c not inherited and method.

Suitable data structures :-

`person[] participants` → can refer to
 → student
 → faculty

⊗ RHS →

LHS → Dynamic method dispatch.

Runtime polymorphism → `sys0(p)` → refer to mango

⊗ calls to string but which to call
`s1 s2 f1 f2 f3`

Orange
Appl.

for(`person p` ; participant
`toString` → `s1`

`s2`
`toString` → `f1, f2, f3`

10

event organiser

Case 4 : Display specific participant details.

⑧ Seat no are required .

→ sys0(enter your seat no);

// Validate --- valid - display details.

// to → student detail.

How to convert seat to Index

if (index >= 0 && index < counter)

sys0(participant[index]) .

AGENDA

Down Casting

sub class is super class → sub class can add additional behaviour.

⑧ add method study → student , Teach → faculty .

2/p → seat "no"

① Validation

② if student invoke study else Teach method

Case 5 : sys0(Enter Roll no)

→ if (valid)

sys0(participant[index]);

person p = participant [index];

Let p = student

P. study()

error

Javac Resove by type of ref

and ref is person and study
function is not present in person.

We have to tell Compiler
that it is of Student ?

Add study method

11

```
public void study() {
```

3. Return "firstname" ~~super~~ getname + "coucurname" + coucename;
optional.

Add Teach method

```
public void Teach()
```

return "Lastname" + getLastName() + " teaching in " + sme);
super is optional.

Main() { "Type casting" }

How to satisfy Java

((student)P). Study();

climbing down the hierarchy
for explicitly typecast

ASD 2nd year

④ down casting required for
only Compiles and not for Java RE

$\delta / P \rightarrow$	1	2	3	4	5
S_1	S_2	P_1	P_2	P_3	
6	1	2	3	4	

Z/P seat no = 1;

2/p seat no=3;  We can never fool Runtime.

→ Compiler is fooled → student(p).study() as this functn

~~* * *~~ ~~* * *~~ ~~java.lang.ClassCastException~~ is present in Compile. ~~* * *~~ Very imp.

M5Q → faculty cannot cast into student

12 When Downcasting required

when

Person p = student (a, b, 2020) ✓ works fine

Person p = new faculty (); for study fundn.

P.teach();

error
type of reference & ref is person

How to tell compiler you're searching wrong?

(faculty(p).teach()) → Runtime checking

XXX Instance of XXXX → runtime type tester.

① check for runtime type information.

by instanceof()

if (p instanceof faculty) class whose type we want to check.

LHS → runtime type checking

Returns false/true

if (p instanceof faculty)

((faculty)p).teach()

else

sys0 (invalid type).

★ ★ before down casting w/ instance of ★★

if (p instanceof student)

((student)p).study();

else

sys0 (invalid)

p = faculty()

if (p instanceof person) ↗ error as true faculty

((student)p).study(); ↗ is person

else sys0)

Object → Emp → Mgr → sales manager

Object → Emp → worker

Emp = new mgr.

- ① c instance of Mgr → runtime
 - ② c instance of Emp → True (Mgr is Emp)
 - ③ c instance of Object → True,
 - ④ c instance of salesmgr → (is manager in sales manager) false.
 - ⑤ c instance of worker → false.
 - ⑥ c is null
- c instance of Emp/Mgr/salesmgr/worker/Object
false

p is instance of faculty

p is instance of student

p is instance of person => null

Down Casting

If 2 types

If (Index Valid)

person p = participant(index);

If (p instance of student)

((student).p).study();

else if (p instance of faculty)

((faculty)p).teach();

else sys("invalid");

this also not required
but to use in assignment

Abstraction:

- ⊗ hiding details. ✓ ✓
- ⊗ unit of encapsulate in java? ✓ ✓
- * we provide abstraction by giving name of function.
- ⊗ What will happen if we remove `first()` method from fruit?
- compile time error in sub-class if we add @override ✗ ✗ ✗
- ⊗ Reasons to add in Fruit → we take common functionality and put in super-class.
- ⊗ any time you call person compiler go by reference hence shows error if function not found.
- ⊗ we always try to hide details

In C++ pure virtual functions

APP:-

```
BoundaryShop[] shapes = { new Circle(20, 40, 12.5), new
```

```
Rectangle(30, 40, 10, 15.2) }
```

Total 3 object are created

- ⊗ is there any option to skip definition in super class.

ABSTRACT or pure virtual functn in C++

Lab session

- ⑧ Grain : tasting : "grain"
 class wheat {
 class mill {

(6.4)

$$\frac{29}{14} - 2\frac{1}{4} = 8.2$$

15

Grain process () {

return grain;

a

$$27 + a \times 3^x \\ 26 + 8 \times$$

D → 6

not returning same as super

class wheat mill extends Mill {
 wheat process {
 return new Wheat();

3

10 → 30



< n^2

3.2 Assignment

option q → Display taste & fruit specific functionality

SOP/f.taste

Q

22/10/2021

$\sqrt{1296}$

1296

J

★ ★

XX

34 36

1414 +100

1414

50X ± 13

37 37

~~3ch + 3wr~~

~~3r → 3ch → 3wr + wr~~

~~3wr → 1ch + 1wr (3)~~

3

⑧ Greatest 3 digit perfect square?

⑧ Smallest 4 digit perfect square?

⑧ Greatest 4 digit perfect square?

⑧ smallest 5 digit perfect square?

⑧ Greatest 5 digit perfect square?

$$316^2 = 99856$$

=

Puzzle

12 → chocolate
 3w → chocolate
 What is max no chocolate in o

$$10 \rightarrow 10 \quad | \quad 3 \rightarrow 1$$

$$1w \quad 10w \rightarrow 3$$

$$10 \rightarrow 10w + 9w$$

$$9w \rightarrow 3wr + wr$$

$$3wr \rightarrow 1wr + wr$$

125



~~3ch + 3wr~~
~~3r → 3ch → 3wr + wr~~
~~3wr → 1ch + 1wr (3)~~

$$10 \rightarrow 10ch$$

$$9wr \rightarrow 3wr + wr$$

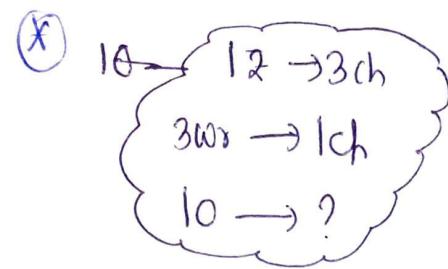
$$wr + 3wr \rightarrow 1$$

$$2w$$

(14)

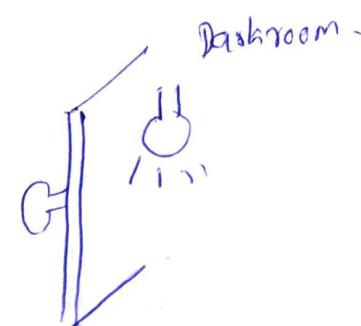
16

Take - wrapper and give back and take one check late.



(*)

1	2	3
---	---	---



to go
entry in room → Only once.

Numbers

1, 2, 10; 3 natural numbers.

1, 2, 3, 10, 20, , 00.

{0, 1, 2, 3, }

23/10/2021 which

which methods can be overridden in general?

virtual

which methods are bound early? i.e. method binding will be resolved by javac as per type of reference.

which method not early binding? private, static, final,

which method takes late?

(*) non-private, non-static, non-final, non-overloaded

what will happen.

Fruit f = new apple(); ✓ upcasting

f.taste → no java error (overridden)

Javac will find taste function in fruit

Revision from fruit assignment

Q) $f = \text{new Orange}();$

Java will check if orange is fruit is true no java error

$\text{fruit}.f = \text{new mango}();$ → early binding

$f.\text{taste}();$ runtime polymorphism

Given:-

$\text{fruit}[] \text{ basket} = \{ \text{new Apple}(), \text{new Orange}(), \text{new Mango}() \};$

To display all

```
for (Fruit f: fruitBasket){  
    System.out.println(f);  
    System.out.println(f.taste());  
}
```

Runtime:- late binding

1st iteration $\rightarrow A \rightarrow O \rightarrow M$

When is downcasting required?

Fruit $f = \text{new Mango}()$ (mango is a fruit and more)

$f.\text{pulp}()$ → Java C error

Compiler $\rightarrow f \rightarrow \text{fruit class} \rightarrow$ no pulp function

Solutn →

$((\text{mango})f).\text{pulp}()$ → Explicit down casting

$f = \text{new orange}();$

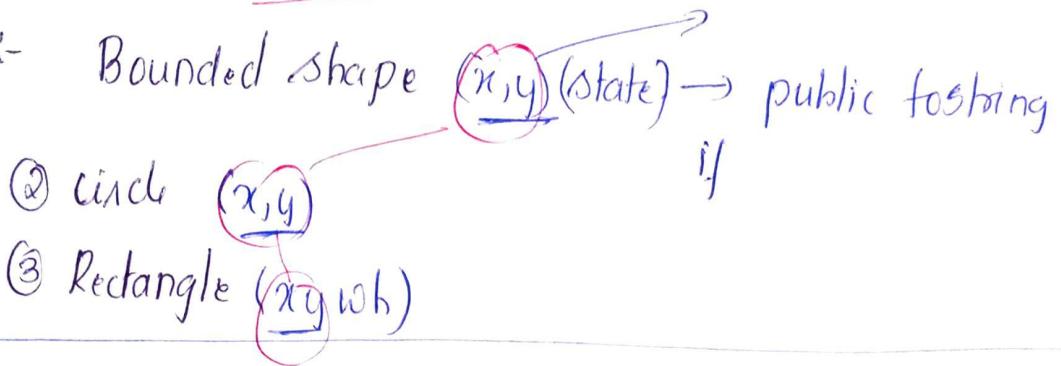
$(\text{man}f).\text{pulp} \rightarrow \text{java.lang.ClassCastException}$
 orange cannot be casted into
 mango

Solutn

(instanceof) → Runtime checking
(RTTI)

** Abstraction **

Scenario :-



Bounded shape

@override
 $\text{posting}() \rightarrow$ derives from object class.

circle $\rightarrow (x,y), \text{radius}$.

method public double area → we cannot @override.
 const $\text{super}(x,y)$

this.radius = radius;

3

Rectangle $x\ y\ w\ h$ ~~(x,y)~~

method

e.g. In a tester store different type of shapes in the suitable way

BoundedShape[] shapes = { new circle(20, 40, 12.5), new rectangle(30, 20, 10, 15.5) }

3 objects array object
 circle object
 rectangle object

for (BoundedShape s : shapes)

Error 'javac'

{
 sys(s)
 sys(s.area)}

④ because area is not defined in
 super class

Solution: - add area method to BoundedShapes.

19

```
public double area() {
```

return -1;

→ ugly looking in super
class -

unnecessary

as we are not having any particular definition hence it is
compulsory to write return -1;

pure virtual function in C++ Solution

Rules :-

→ \rightarrow BoundedShape

```
package shapes;
```

```
public class BoundedShape {
```

```
private int x, y;
```

```
public BoundedShape(int x, int y) {
```

```
this.x = x;
```

```
this.y = y;
```

3 @override → match with object

```
public String toString() {
```

```
return "x=" + x + ", y=" + y;
```

```
}
```

Circle \rightarrow Circle

```
public Circle extends BoundedShape {
```

```
private double radius;
```

```
public Circle(int x, int y, double radius) {
```

```
super(x, y);
```

```
this.radius = radius
```

3

```

@Override
public String toString() {
    return "circle "+super.toString()+"radius "+radius;
}

```

```

public double area() {
    return PI*radius*radius; import static java.lang.
        math.PI
}

```

Rectangle :-

```

public class Rectangle extends BoundedShape {
    private double width, height;
    public Rectangle (int x, int y) {
}

```

3

```

@Override
public String toString() {
}

```

3

disadvantages

```

public double area() {
    return width * height;
}

```

3

ctrl+shift+o
+

Tester public class TestShapes {

```

public void main() {
}

```

implicit add
new BoundedShapes.

```

BoundedShape[] shapes = new BoundedShape[2] {
    new Circle(), new Rectangle()
}

```

```

for (BoundedShape s : shapes)
{
}

```

3

```

    System.out.println(s);
}

```

```

    System.out.println("Area "+s.area());
}

```

3

Java error

goes for reference = shape
checks area does not
finds

Adding common behaviour in super class

2)

public double area() {

 return -1;
}

we require this much only.
but not accepted by Java
With out "ABSTRACT" keyword

ABSTRACT → $\Rightarrow \setminus \rightarrow$ Bounded shape

public abstract double area(); → now declaration is accepted.

⊗ ~~XX~~ we should make class also "abstract"

Rules: any time a class has one or multiple abstract method
class must be declared as abstract class

Syntax → public abstract class Emp();

~~AAA~~ abstract class cannot be instantiated:

⊗ Emp e₁ = new ~~Emp~~ Emp(); X

Emp e₁ = new Mgr(); ✓

Concrete \equiv non abstract

public abstract class BoundShape {}

~~AAA~~
Interior

$\Rightarrow \setminus \rightarrow$ Bound

⊗ can abstract class contain non abstract methods?

Yes.

⊗ Can we add constructor to abstract class?

yes; we will have to write constructor. to initialise private
data members.

⊗ Can we ever declare class abstract & final?

abstract class has to be extended otherwise useless

final \rightarrow cannot be extended

Q) What happens if we declare abstract as private?
Abstract class is for extend but private does not permit
hence wrong combination.

Q) Can we create a static abs class with 100% non abstract
functionality.

e.g. @HttpServlet

Yes.

Comeback for circle class

Change in circle class for abstract

Q) How we can write @Override

(X) BoundedShape[] shapes = {new Circle(), new Rectangle()};

BoundedShape p = new BoundedShape(); X

Logic behind rule.

If allowed any one would have called abstract functn which
is deadly error.

Final

Usage

① final data member (primitive type) - const

public final int data = 123;



② final methods cannot be overridden

e.g. Object class --- wait, notify, notifyAll.

③ final class:- can't be sub classed (or extended)

i.e. stopping inheritance.

public class MyString extends String { } Java compiler



23

final reference

☆☆☆.

final Emp e=new mgr()
e=worker X → Java error

Object class :- is not final, or abstract

~~AAA~~ 100% abstraction (Interfaces) ~~AAA~~

Interface: is a blueprint for implementation class.
typically it has data m & methods.

- ~~PS~~ Data members :- public static final data members
accessibl. w/o objec cannot be changed
methods: only abs declaratn

- ④ interface augment for complete abstraction and multiple inheritance.
 - ④ imposes IS-A relationship.
 - ④ cannot create object.

- Purpose :- achieve 100% abstractⁿ

Interview

 - ④ to achieve multiple inherit
 - ④ to achieve separatⁿ, loose coupling
 - ④ separatⁿ b/w What and How

What (what services to offer) How (latter part)

- ④ first advance topic → collected frame work.
 - ⑤ Java compiler add public and abstract. before function
static and final before data members

24 Syntax

default / public interface NameOfInterface extends 'Comma separated list of super interfaces.'

{

 int DATA = 100 // Public static final

 double calc(double d₁, double d₂); // public abstract

}

only interface → useless.

Implement keyword

Implementing class syntax

default / public class NameOfClass extends 'Super(s)' implements 'Single' 'Implements'

Comma separated list of interface

{

 { } // body

implement class is a

public class circle extends Shape implements Comparable, Runnable

IS-A Relation

super class

↑ extends IS-A

Sub class

sub class is a super class + more

allowed

☆ ☆

super interface, z/f₁, z/f₂, z/f₃ ...

↑ is-a
extends

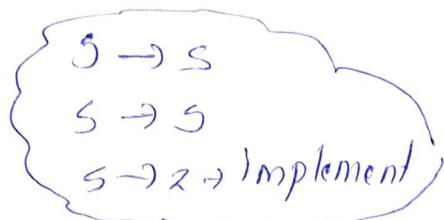
sub interface

sub interface is a super !

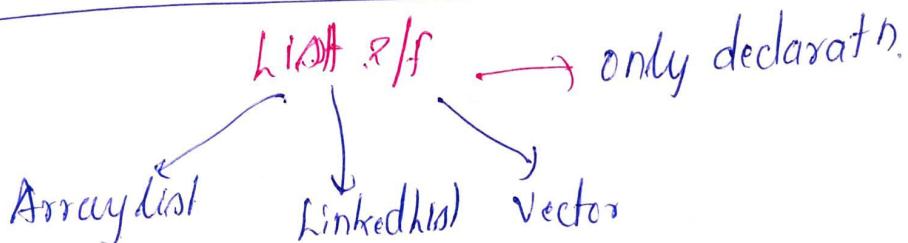
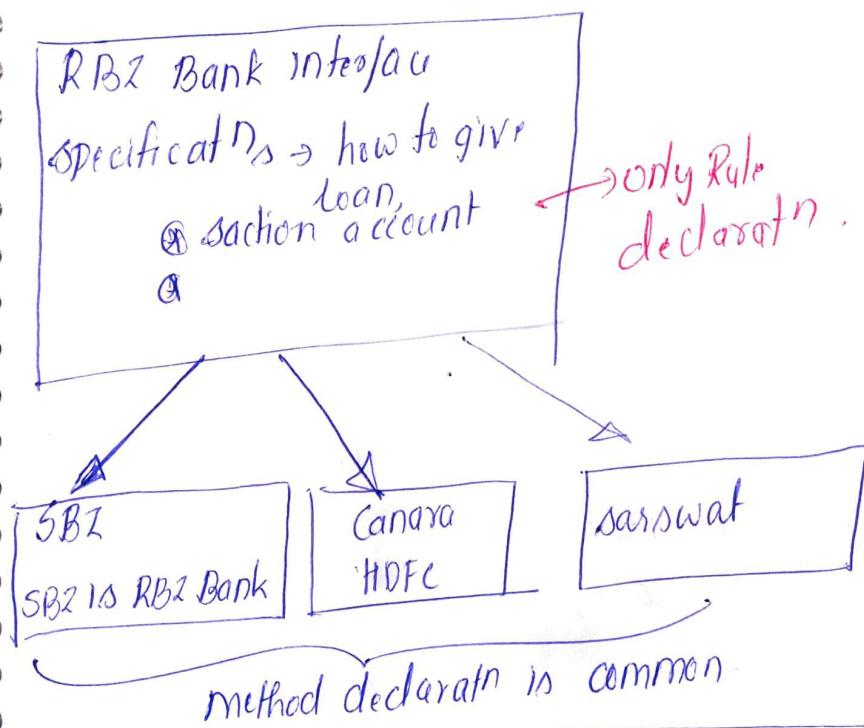
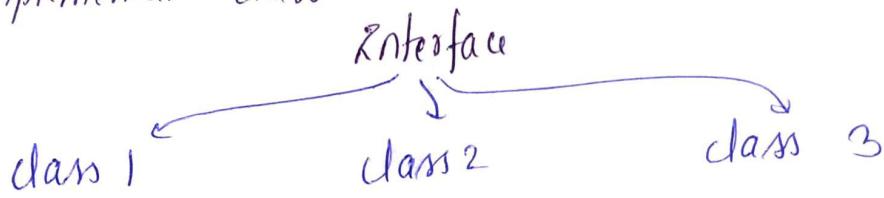
↓
if we are inheriting

Interface 1 Interface 2 Interface 3

IS-A implements.
↓
 Implementing class.



Can we have single interface implemented by multiple implementation classes?



ExamplePrinter Interface ~~7.2~~

void print(String msg);

internally added public & abstract.

Console printer
is a printer,
print declar.Print
File Printer
is a printer,
print declar.Socket printer
print declar.Step 1 Interface :- Create interface ~~7.2~~package P;
Name printer.

Package P:

public interface Printer {
 double DATA = 1234.56;
 void print(String msg);

if we don't extend then it is useless.

Step 2 class consolePrinterpublic class ConsolePrinter implements Printer {
 @Compulsory to add print method.
 if dont know → declare abstract;

Xter //

(*) Annotation to be used.@Override optional / but recommendedPublic void print(String msg) {
 same same
 { sys.println(msg + "conn") } }same / wider
access specifier.

class FilePrinter. ~~7.2~~

public class FilePrinter implements Printer {

@override.

public void print(String msg) {

 System.out.println("Storing " + msg + " in a file");

 System.out.println(interfaceConst + DATA);

can implement class Directly access constants w/o
interface name?

→ Yes we can use directly)

Class NetworkPrinter. ~~7.2~~

 {
 System.out.println("Sending " + msg);

}

Tester class

main() {
 ConsolePrinter p1 = new CPC();
 p1.print("Hi");

} ~~FilePrinter~~ FilePrinter p2 = new FilePrinter();

p2.print("Hi") ;

NetworkPrinter p3 = new NPrinter();

p3.print("Hi");

}

Ref of Ref Interface ~~test~~

Printer p; ref type → memory allocated

error

p = new Printer();

on stack because it
& method local variable.

We cannot

Create object of Interface

P =

✗ super class can refer to subclass X*
 ✗ Interface can refer to implementation class X*
 ✗ p = new ConsolePointer(); X*
 ✗ ~~is-a relation~~ X*
 ✗ Enclosed referring in java & interface X*
↳ upcasting on interfacing

p.print("new msg"); X*

↳ dynamic method dispatch X*

↳ Runtime polymorphism X*

can interface reference directly refer to any imp class inst.
 → Yes X*

array of interface X*

main() { X* } ↳ array type reference

pointer[] pointer; → array consists of references.
 LHS. X*

If objects visualized X*
 ↳ = { new ConsolePointer(), new NPrj, new FP } X*

pointer[0] → console

[1] → NFPFP

[2] → NPrj

invoke print method on all.

for(pointer p; Printer) {

 System.out.println(p.print("Hi"));

}

Can implementation class add new functionality? 29

Yes. → added in N/W Printers.

public void encrypt(String msg)

{

 System.out.println("Encrypting msg + msg");

{

Adding functionality to implemented class.

((NetworkPrinter) P).encrypt("password");

but it will give error to console printer, File printer.

SOLN

Downcasting interface

(RTTI) || instanceof ||

if (P instanceof NetworkPrinter)

P is instance of NP then →

else

System.out.println("invalid type no inheritance facility");

~~days 8~~

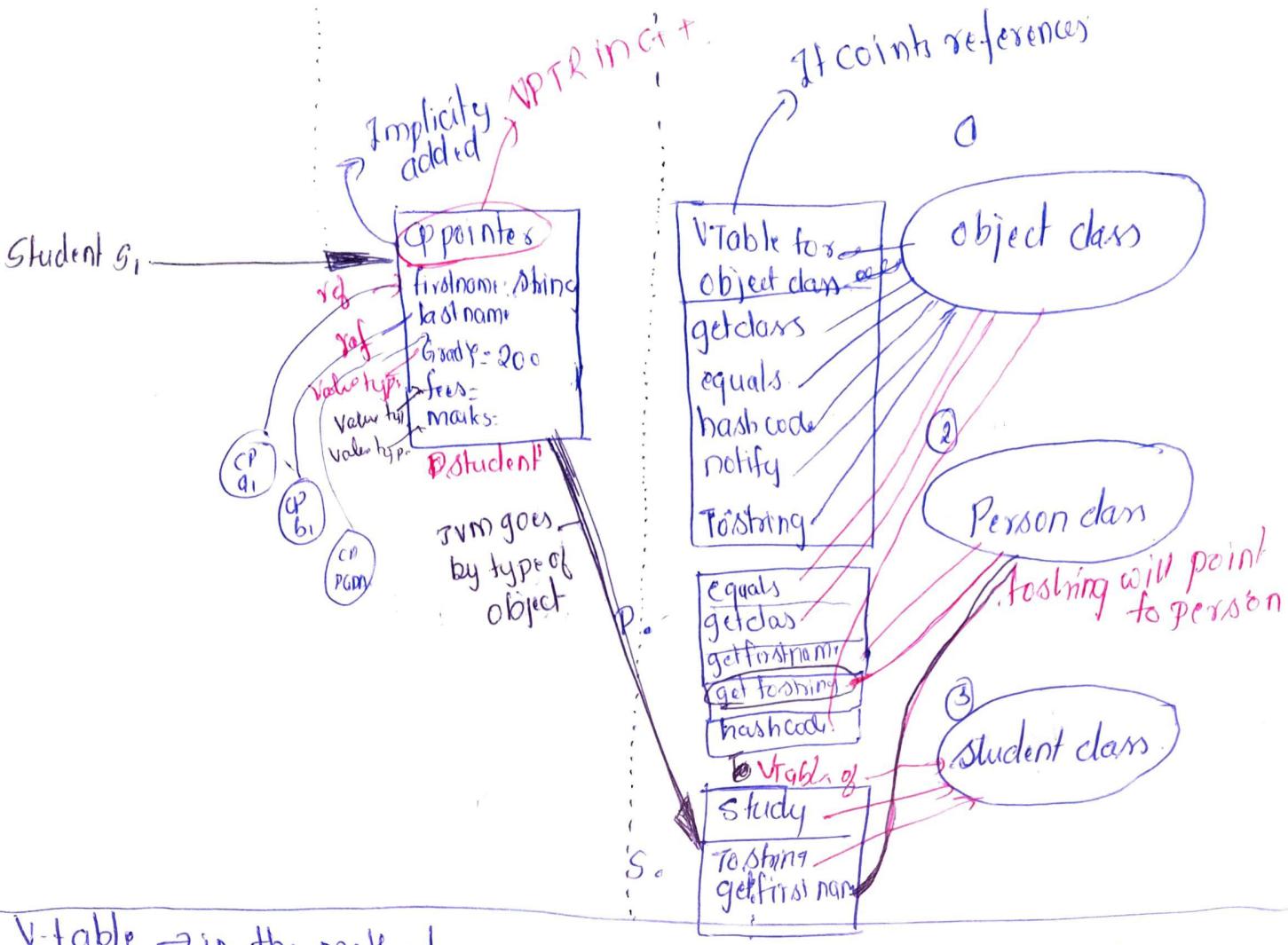
25/10/2021

memory drawing of super class

student s₁ = new student("a₁", "b₁", 2020, "PG-DAC", 12346, 80)

person p = new student("a₂", "b₂", 2021, " ", " ", 12348, 80).

30



V-table → in the method

Virtual Method Table → (VTable) :- possible implementation for dynamic method dispatch (may change as per JVM spec)

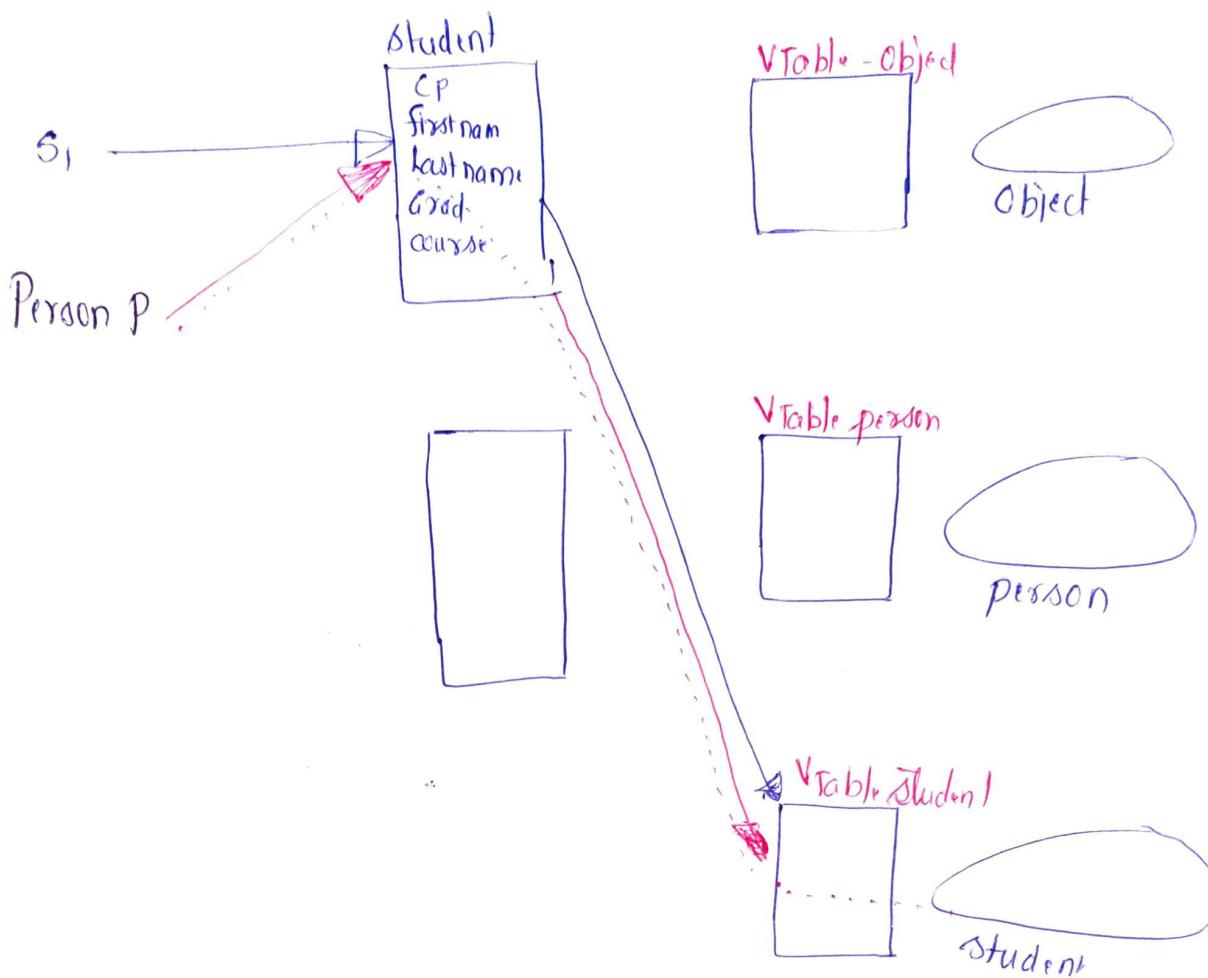
: Array of references, containing address of actual method in meta space

① `equals` :- inherited from object class

② overridden means it will point to subclass when it is defined
e.g. student class, faculty class

when the VTable is created?

is fixed for class hence at creation of class it gets created.



What will be o/p

main() {

8:46

TestInstanceInitBlock t1 = new "();

{ } . () t2 = "();

, System.out.println();

{ } → will get called before constructor.

public void printBlock(),

~~XX~~ what is interface?

① Blue print for implementation class.

~~XX~~ Advantage of ??

100% abstraction till Java 3, ① multiple inheritance using interface.
Separation: loose coupling → How part vs what part.

Till JDK 7 ↑ what does it contain mainly?
data member → public static final
method → public abstract.

Relation

② Can 1 class multi interface?
Yes

③ Interface extend from multiple IFs.
Yes

④ Class extend from multiple Super class?
No

⑤ Can impl class access interface data member directly?
Yes

⑥ Non-impl class ??, ??, ??, ??, ??

No

⑦ Demo:- Create class implementing multiple interface?

Demo!
package P;
public interface A {
 boolean isEven(int value);
}

Package P₁

public interface B {

* * *

boolean isEven(int value); \$ (60) P₁

}

package P₁

public class C {

}

public class C implements A, B {

public boolean isEven(int value)

}

No diamond problem

When will it occur?

→ it will occur when we

have 2 implementations

same

Shortcut

* P₂ 8.)

No diamond problem
in inheriting two same

Demo D₂ Package P₂

Interface A {

boolean isEven(int even);

Interface B {

boolean isEven(double value)

}

9:00

overloaded will
be detected and
need to add bar

D₃

Package P₃

Interface B {

* * * *

for overloaded → return type is
different is ignored

D₄

Pack 4

A {

int DATA = 100;

}

* * *

* * *

* * *

* * *

* * *

* * *

* * *

* * *

* * *

Data member duplicacy
is not allowed

Int Data = 200

*

*

*

*

*

System(Data)

field Data is
ambiguous.

class implement A, B {

}

Solutn

~~sysc (A. DATA + B. DATA)~~

Interview Quesⁿ

Same method name in intfs with different signature?
allowed overload.

Same method , same signature, different ret type?
not allowed → No soln.

Will you use duplicate name / datamember

Never do so

Subinterface from interface

short cut from IDE Extended interfaces.

public interface C extends A, B {

 what is inheriting from super interface?

 abstract method declarations

 a double add (double d₁, double d₂);

}

To create concrete class?

Implement all belonging to C, A, B

All inherited

behaviour is
compulsory

package PS;

public class D implements C {

 * * *
 For concrete class it is mandatory to implement
 all the implementation

Types of Interface

Marker Interface → empty interface. no method no datamembers.

package P6

public interface A { }

8.1 P-6
||

Java 8 onward → ³ functional programming added.

functional interface

P7 (@ functional interface)
public interface B { }

void show()
3 boolean isEven

error b/o no method hence
we need to add method.

error b/o 2 methods

Marker interface → No m/d

functional interface → only one m.

Interface Computable
data m → PI
Method - calculate area & perimeter

Can you compute area/perimeter of any bounded shape?
Bounded shape: Implementation class.
abstract or concrete?
State

Concrete methods

Circle Rectangle Triangle Square

Concrete or abstract class?

36

Demo

package P7;

public interface Computable {

$$\text{double PI} = 3.1415$$

double computeArea();

double computePerimeter();

۳

Impl class

```
public class Boundedshape implements Computable {
```

```
private int x,y;
```

@Override

·computeArea() {

```
return 0;
```

compute Perimetr()

return 0

3

hence make class abstract.

```
public abstract class BoundedShape implements Computable {
```

Private int x, y;

Constructors

```
public BoundedShape() {
```

$\text{th}(x-y)$

$$\text{th. } g = g$$

3

As As As

Impln of Area and Perimeter

is compulsory to be

private double radius;

```
public Circle (int x, int y, double radius) {  
    super(x,y);
```

$\omega_{\text{CP}}(x, y)$

$$\text{th. radius} = \text{radius}$$

@ public double computeArea() {
 return PI * radius * radius;
}

@ ComputePerimeter() {
}

@ ToString() {
}

Same for rectangle & {

private width, height;
public Rectangle(x, y, width, height);

Tester :: main() {

BoundedShape[] shapes = {new Circle(), new Rectangle(), ...};
Computable[] computables; } array of interface.

= n{ } → ↗ ↗ ↗

for (computable c : computables) → runtime polymorphism

if (c == sys(c))
if (sys("Area"))
first case → points to circle;
2nd case → points to rectangle;
3rd case → points to square;

Can a circle add additional behaviour:

```
public void drawArch(){  
    System.out.println("drawing arch");  
}
```

```
main() if(c instanceof circle){  
    c.drawArch();
```

→ error

soln → down casting

④ compiler goes by reference
and c → does not have
drawArch hence

④ type cast and use

Interface is over

Exception Handling

Is it error or Rerror?

runtime error detected by JVM (typically by main thread)

Why exception handling?

④ currently if we were expecting integer but z/p is double hence
due to this code gets terminated. To avoid termination & continue..
even in case of RTTZ.

④ No explicit need of for checking validation return types?

④ Separation:- allows cleaner separation.

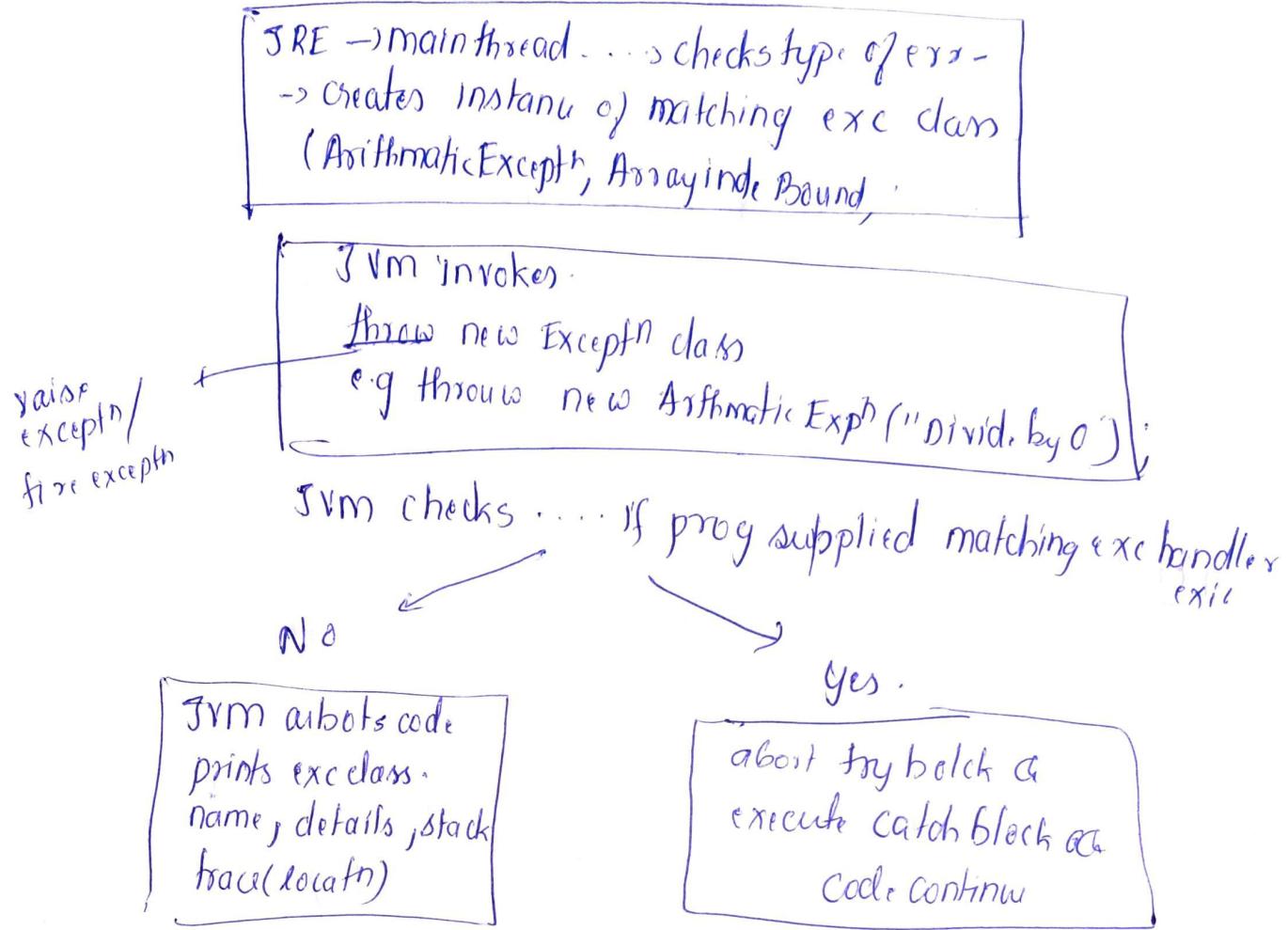
Checked vs unchecked

④ Try, Catch, finally, throws, try with resources, custom exception

Exception Handling flow

39

- ⑧ Any run time err occur (div by 0, array out of bound, null pointer file not found, unable to establish conn);



- ⑧ Who is detecting err and throwing?

JVM programmer has to do

unchecked &
checked

** Inheritance Hierarchy for Exc handling class

java.lang.throwable

Concrete
java.lang.Error

fatal/unrecoverable error

Try and catch is user hrs.

Stack overflow, out of memory

java.lang.RuntimeExceptn

Null pointer, AE

unchecked Exceptn

ZOException, parse
File not found, does not
found

Checked Ex c

Null Pointer

java.lang.Object

Throwable

Except

Runtime

unchecked

class in

flow to check checked or unchecked?



No runtime exc.

→ Runtime exception

⊗

I/O exception

Classcast exception

Popular Quesⁿ

⊗ JVM → does not differentiates b/w checked and unchecked.

⊗ Javac → differentiates b/w them

Summary : Java forces handling of checked exc upon the program. (Handling by supplying Matching try - catch block or including it in the throws clause). → then and then
 → to tell the compiler "I will not handle (method) but caller will handle".

⊗ Package → Test @ unchecked Exception {

main () {

int result = 23 / 0;
System.out.println(result);

int[] data = {10, 20, 30, 40};

System.out.println("array data " + data[4]);

System.out.println("Parsed int value " + Integer.parseInt("12345"));

noformat exception

12345

Bye (main continues)

41

NO Java C error → does not handle unchecked excps.

~~Free Thread~~
~~java.lang.Object~~

Sleep: - public static void sleep(long millis) throws InterruptedException.
→ To add

```
main() {  
    System.out.println("Before");  
    Thread.sleep(100); // checked exception  
    System.out.println("after"); // Java forces programmer  
}
```

Java C → differentiates checked or uncheck
Super → Runtime → unchecked.
class

Syntaxes

Syntax
try { } catch (ArithmeticException e) { }

✗ try { } catch (AE, NPE, IOE) { }
 catch(AE) { } catch(NPE) { }
 catch(IOE) { }

✗ Try { } Catch (NPE) { } Catch (AE e) { } Catch (Exception e) { }
 Super class can catch all.

try { } catch (AE e) { } catch (NPE | IOB / ClassCastException)
 catch (Exception e) { } catch all)

Test / Demo

Try is paired with catch or finally

int result = 23/0;

try {

 int result = 23/0;

 int data[3] = {1, 2, 3};

 System.out.println(data[10]) → AOB

 Integer.parseInt("dsf345")

}

 catch (Arithm.E) {

 System.out.println("1");

 Default handler → abort code

}

 catch (ArrayIndexOutofBounds.E) {

 System.out.println("2");

}

Shortcut to handle multiple Catches

Catch (Exception e) {

 System.out.println("3");

 System.out.println(e.getMessage());

↳ IP Input string

"dsf345")

Ultimate super class in exception → throwable

getMessage()

How to attach Debugger

To String → throwable (overridden from Object)

```
(catch Exception e) {
    e.printStackTrace();
}
```

~~How to get location of error~~

printStackTrace() → *work execute and Continue*
 catch {
 e.printStackTrace();
 }

package tester;

syso(Before)
 timer.sleep(3000)
 syso(after)

main thread

exception →

How to make thread sleep()

thread.sleep();

Never catch block empty

Javac → forced handling of checked

~~Throws :- Syntax~~ (throw vs throws)

② appears in method declarat.

e.g. Integer class API

public static int parseInt(String s) → ^{number format} NF exception

③ throws → keyword fo Javac.

Meaning → Method may raise specific exp

When is throw Compulsory ~~XXXX~~

Mandatory → only in case of un-handled (no try catch)
checked excs (not extended from Runtime exceptn)

Use → used in delegating calls

checked exceptions

main() {

System.out.println();

Thread.sleep(2000);

System.out.println();

public void main(String[] args) throws InterruptedException {

throws is used to inform java
Main method is not handling
excpt it is going to handled
by callers.)

Will the code continue to execute?

aborted

throws is not handling code.

① throws show possibility of error.

② it is for satisfaction of compiler.

e.g. parseInt(String) throws NumberFormatException.

(X)

~~** finally **~~ Java popular quest' 45

Difference between finally & finalised;
Garbage collector calls

- ① key word in exception handling.
- ② always executed except termination JVM
- ③ try { } catch (Exception e) { } finally { } 12:37

Syntax

- ④ try { } catch (Exception e) { } finally { } 12:37

⑤ In all cases finally will be executed:

is it service exception → no

try { } finally { }; **

- ⑥ Thread.sleep(1000); → unhandled checked exception
- day 8-2 /src/test/testfinally
Throws → mention what type of exception.

finally is always executed → matched or not

- ⑦ Some exception if not found will be discarded.

day 8-2 → Test finally 2

lin-27 → \$ [2].

- ⑧ If exception is not serviced then finally will be executed and normal execution is not done.

- ⑨ Test finally 3 :- normal execution goes only when interrupt is handled. (12:47)
Default handler.

finally 4 :- case if Return before finally 18:03

still finally will be called.

Ex e.g closing file before return;

write such in finally.

at last JVM will check if there is something in finally.

fill Java 7

Testscanner:

main() {

Scanner sc = new Scanner(System.in);

System.out.print("Enter int");

System.out.println("you Entered " + sc.nextInt());

sc.close();

System.out.println("main over");

Code gets aborted

if we want finally want
to execute then put in final.

Try {} finally → sc.close will execute but main over skipped

To see main() → write catch Block
main over

API

Try with resources

auto closing

g.2 / tester To avoid repeated code.

We can work multiple

Syntax

try (sc = new Scanner(System.in)) {

}

JVM will call automatically sc.close

⑧ Try block in Try with resource is inbuilt combination of Try - catch 47

- ⑧ from SE7 onward `java.lang.AutoCloseable` → will close handles automatically
⑧ represents resource such as scanner, are closed when not required
⑧ public void close() throws exception → closing resource
 ↳ automatically called by JVM

⑧ We can open more than one resource.

e.g try(Scanner sc = new Scanner(),
 FileReader fr = new FileReader())
 { }

immediately calls close resource
 ↳ catch-all
 JVM will call fr.close →
 sc.close than catch
 all gets execute

Scanner → Java docs → AutoCloseable → Interface AutoCloseable

⑧ A Try with resource best method for closing resource

Object :- Validate speed of vehicle on Highway.

Valid range → 30' to ~~100~~ 80

Vehicle speed is ~~20~~ → alert driver

100 → alert driver.

60 → ✓

Exception in ① ② → Program based Exception.

keyword used will be throw → and create own.
custom exception.

Syntax throw new SpeedOutofRangeException("too slow/ too fast");

① Handling → will be done by JVM and search for user defined exception.

3:45

② emp details → Stack → first in last out
Stack interface

③ Employee class → id, name, salary, Constructors, toString

*→ employee is not a stack

*→ ④ interface → Stack interface → push pop
stack_size constant
e.g. → push(employee e);

26/10/2021 Catch throwable is also allowed

Default handler → aborts the code

→ Name of msg
→ backtrace

Try block aborts and JVM invokes matching catch block →
and code continues

Throw throwable instance → syntax currently used by JVM
to raise system excs (eg AE, AOB, NPE, classcast)

Throwable <-- Error, Exception → it can be caught

Exception → RuntimeException <-- unchecked excs

Exception → IOException, SocketException

(
InterruptedException → checked

Checked Vs Unchecked

javac : Java forces handle the checked exc to programmer

Throw and throws

It is for
raising exception

④ JVM, programmes
can use this key
word.

④ appears in method
definitn

④ throw throwable instance

Syntax

④ JVM will abort and
search for try and catch
or finally

④ alteration of normal
flow

Meant for Javac, appears in
method declaratn

- ④ public void show() throws IOException, InterruptedException
- ④ It may throw (possibility)
- ④ if we are in single thread
- ④ it tells compiler that caller will handle
the exception

when is adding throws is mandatory?

* unhandled → no catch block, & checked exception

finally: always executed (i.e. in case of no exception as well as
exception / Return;

Exception: In case of JVM termination. ★★★★

typical use case: to i) we open JDBC connection, then after use we
have to close them, and clean up from finally,

④ cleaning of Java resources,

e.g. file handles, DB connection,
standard i/p

④ put such code in finally block.

★ ★ ★

Try with Resources

Try with Resources:

★ ★ ★

try (Scanner sc = new Scanner(i)):

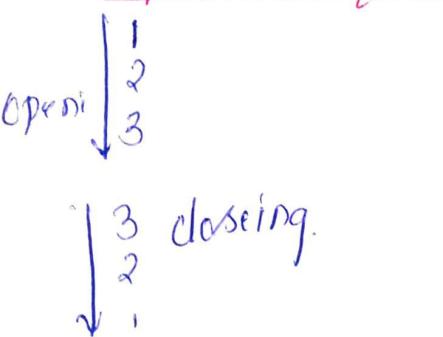
{

}

- ④ no compile error if we don't write catch block.
- ④ we can add catch block automatically JVM calls : `Scanner.close()`.

- ④ we can add multiple resource.

Sequence of closing



can we open any Resource^(class instance) using try with resource block.
No. only those class instance whose class has implemented
java.lang.AutoCloseable..

USER Defined Exceptions

- ④ In case of speed outside the range... prog will have to detect the err... create instance of user defined exc class... explicitly throw custom exc to the code

```
if (speed < 30) {  
    throw new SpeedOutOfRangeException("Ur driving slow!!!");  
}
```

if (speed < 30) → user defined exception.
throw new SpeedOutOfRangeException("Ur driving slow!!!"); → msg
default w/o msg → AAA

- ④ custom exception as checked → On → Exception
as unchecked → Runtime errors

- ④ When error msg

Exception() → parameterless constructor.

Exception(String msg) → ✓

after above msg will be display getMessage();

Step: create custom exception class.

① add parameterised constructor

② Create a separate class e.g SpeedUtils -
add static method for speed validation.

③ caller will be faster

→ function → if out of range throw exception.

day 9.1 9:04 Package custom-exception

Name: SpeedOutOfRangeException.

public class extends Exception {

How to suppress warning

@SuppressWarnings("serial");

@override

static {
min-speed = 30;
max-speed = 80;
}

public SpeedOutOfRangeException (String errMsg) {

super(errMsg) → parameterized constructor

}

Writing Validation Rules: keep it separate from tester.

package → utils

Name → SpeedUtils.

public class SpeedUtils {

Public static final int min-SPEED, max-speed;

needs to add static method to validate speed of vehicle.

access from other package → no boolean
otherwise exceptn works

if (speed < 30)

rather than using here declare it as constant

⑧ Tester able to call w/o object
5v Public static void validateSpeed(int speed) {
 if (speed < MIN_SPEED)
 throw new SpeedOutofRangeException("you are driving
 too slow");
 if (speed > MAX_SPEED)
 throw new SpeedOutofRangeException("you are driving too
 fast");
 System.out.println("speed within range");
}

gives errors because it is checked type

To change to unchecked $\star\star$

change exception to Runtime Exception()
in throws extends

Centralised one catch block is better

Tester 9:20 → main() {
 try (Scanner sc = new Scanner(System.in)) {
 System.out.println("Enter the speed");
 // tester has to invoke SpeedUtils method for validation.
 // To use directly
 // validateSpeed(sc.nextInt())
 // import static
 // utils.SpeedUtils.validateSpeed;
 }
 catch (Exception e) {
 System.out.println(e.getMessage());
 }
 System.out.println("main over");
}

Ctrl + Shift + O → to fix imports

⑧ Better Way to get error msg
e.printStackTrace();
mixed off

flow to run on Cmd prompt

→ compiled folder → .class → Test bin → cmd → Java - tester. TestCustomiz.

To get proper msg:

Attaching Debugger

- ① add breakpoint → toggle breakpoint
- ② Debug → switch perspective →
- ③ F6 → To step inside method → f5 → went
- ④ (x) →
- ⑤ f6 → To go next (f5)

** Strings **

String handling in Java AA A imp for interview

java.lang.String

- ⑧ immutable char (non-modifiable)
- ⑧ But string obj can be changed unless it final.
- ⑧ It has length as Method();
- ⑧ thread safe: they are inherently thread safe

java.lang.StringBuilder

- AA
- mutable char sequence
- modifiable
- has length & capacity
- Inherently thread unsafe
- added latter in API
-

Java.lang.StringBuffer

- legacy class → existing since start
- inherently thread safe

String Handling /

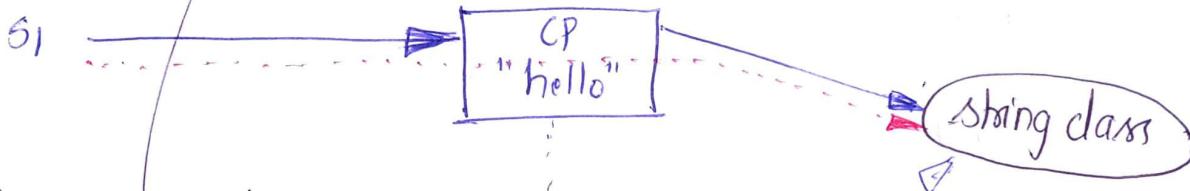
Immutability of String ✗ ✗ ✗ ✗

\equiv vs equals == vs \neq equal

Change perspective → Java.

Project Test 1 q.5c

main() {
 String s1 = "hello";
 s1.concat("hi");
 System.out.println(s1);
}
Output = only = "hellow",
new is added implicitly
Reference → Object



String s1 = new String("Hello");
s1.concat("hi");
System.out.println(s1);

No change in Ref.



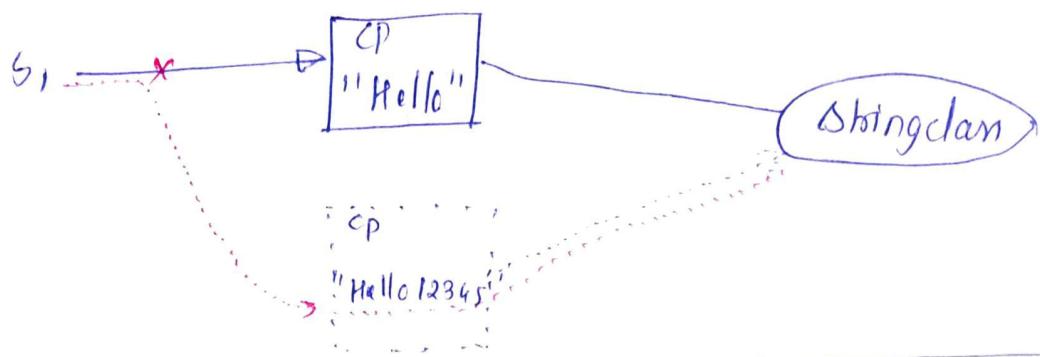
`s1 + "12345";` + is only overloaded operator

`System.out.println(s1)` Output = "Hello12345"

+ ≈ concat

memory picture

$s_1 + = "12345"$



(final) String $s_1 = \text{new String}("Hello")$

$s_1 + = "12345"$ error b/o trying to reassign
reference (final)

String $s_2 = s_1.\text{concat}("abc")$; O/P $\rightarrow \underline{\underline{\text{Helloabc}}}$

$s_1.\text{toUpperCase}()$

$\text{sys0}(s_1) \Rightarrow$ o/p will be original content.
no change

$s_2 = s_1.\text{toUpperCase}() \Rightarrow \underline{\underline{\text{HELLO}}}$

XX always new modified is created

XXX replaceAll() XXX

public String replace(char oldchar, char newchar);

$\text{sys0}(s_1.replace('x', 'y')) \rightarrow$ original Replaced

$\text{sys0}(s_1) \rightarrow$ Again original

XXX Ref equality Vs Content equality XXX

public boolean equals(Object anotherObject);

Overriding form from Object.

① same or wider ② method name & signature same.

~~***~~ compares based on "Content"

Test 2

String $s_1 = \text{new String("testing String")};$

String $s_2 = " " ("testing String");$

$\text{System.out}(s_1 == s_2);$ → false

$\text{System.out}(s_1.equals(s_2));$ → true

\Rightarrow reference / address comparison

equals → content comparison

~~***~~ Need of overridden

AAA

① To replace reference equality to content equality.
i.e sequence of character

AA

② base on case sensitive manner

equals Ignores case (s_3)

String $s_3 = s_1.\text{toUpperCase}();$

AAA

$\text{System.out}(s_1.equals(s_3));$ → false

$\text{System.out}(s_1.equalsIgnoreCase(s_3));$ → true

$\text{System.out}(s_1 == s_3);$

AAA ~~True~~

false.

* * String Vs literal : * *

main() {

String $s_1 = \text{"Hello!"}$

String $s_2 = \text{new String}(s_1);$

String $s_3 = "Hello!";$

String $s_4 = "Hello!"$

`sys0(s1 == s2) → false`

`sys0(s1.equals(s2)) → True.`

`sys0(s1.equalsIgnoreCase(s4)) → True`

`sys0(s1 == s3)` ~~xx~~ ~~xx~~ String literal pool.

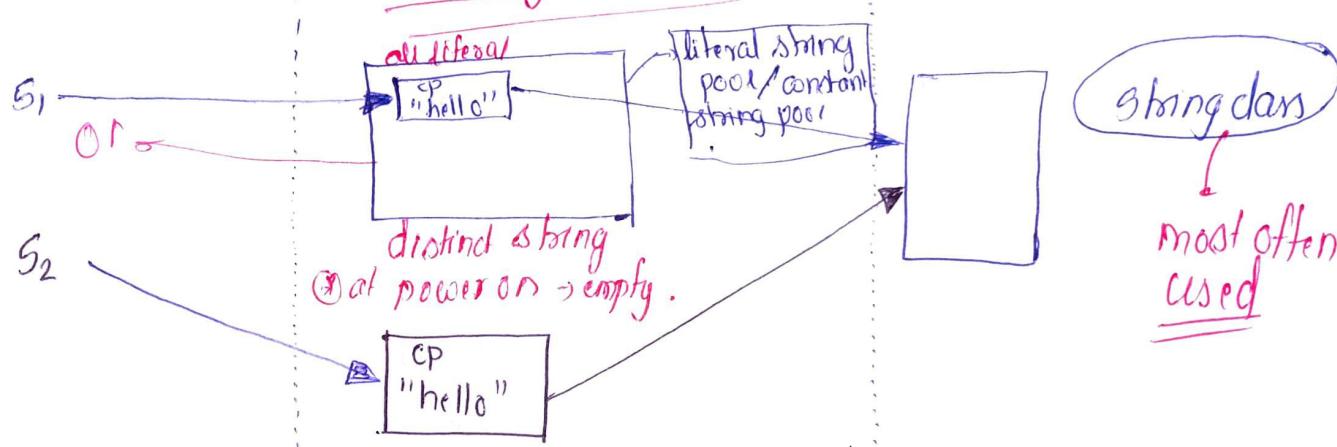
- Java treat literal and non-l separately.

`String s1 = "hello";` → literal string

`String s2 = new String(s1);` → s_2 is pointing to non literal

String s1 = "hello"; → literal string.

Interning of String



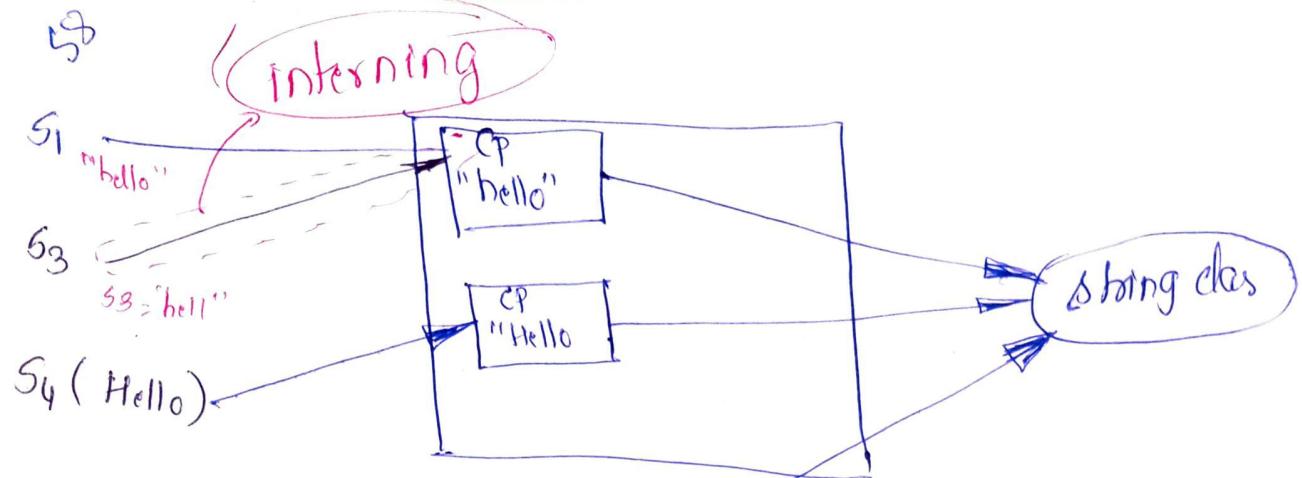
Constant String pool. (Pooling → multiple people sharing same resource to reach destination),

① JVM always checks for duplicate string.

② String class loaded ③ empty pool is created.

④ always objects on stack

⑤ only " " → will be placed in String pool.



$S_1 = S_2 = S_3 = S_4 = \text{null}$

$S_5 = \text{new String}(S_1)$

XXXX

$S_1 = S_2 = S_3 = S_4 = \text{null}$

$S_5 = \text{"hello";}$

- ⊗ candidate for gc \rightarrow ①
- ⊗ S_5 point from string pool.
- ⊗ literal string pool is not getting garbage collected and is very much intact.
- ⊗ it will go out of picture when class is unloaded from MA, i.e. when JVM terminate

When is pool created?

any time String class loaded and is empty initially.

- ⊗ always entry into pool is via equals methods
- ⊗ when is string pool marked for GC?
- only when JVM terminated.

XXXX Hashcode XXX

- ⊗ address of string is dependant on content hence we cannot see the addresses

~~*** Intern ***~~

10. 12:10 >

59

- (*) Returns a canonical representation for the string from pool

String $s_6 = s_2.\text{intern}();$ o/p = true
 $\text{sys0}(s_1 == s_6);$

Internal working

- (*) intern go to the pool and find if exist
- (*) pool means only unique
- (*) it will check 'content of "s₂"'

Adding non literal to literal pool

String $s_7 = \text{new String("test123");}$
 String $s_8 = s_7.\text{intern}();$
 $\text{sys0}(s_7 == s_8)$

- (*) intern will check "test123" in pool if not exist it will return false.

(*) charAt(index)

- (*) Returns char at specific index.

main() {

working with literal for

$\text{sys0}(\text{"testing string".charAt(0) + "testing strings".charAt(1)}$

"test".charAt(0)

.length()

$\text{AT("testing string", 1)}$

As well as nondliteral

Contains

- Returns true if String contains sequence of chars

↳ CharSequence is Interface

e.g. `s1.contains("cool")` || True

`contains()`

↳ CharSequence → Interface

↳ hence can point
• to String, String Buffer

*** Indexof() ***

- ④ returns first occurrence of first char from start
- ⑤ To start from particular (int fromIndex)
- ⑥ indexof(String str);
- ⑦ lastIndexof() will search for last occurrence of string.

e.g. → `s1.indexOf("String") + " " + s2.lastIndexOf("String")`

Test:-

*** * Remember * ***
String "Compare to"

public int compareTo(String anotherString)

Returns:- 0 → $s_1 = s_2$ } lexicography (dictionary order)
-ve → $s_1 < s_2$
+ve → $s_1 > s_2$

$s_1 = \text{hello}$

$s_3 = s_1.\text{toUpperCase}();$

upper case comes
first

$(s_1.\text{compareTo}(s_3))$

hello

HELLO

→ +ve

`System("Anish").compareTo("Amiti");` > 0

Amish Amithi < o

$$S_1 \quad \cancel{z \neq y} < S_L$$

String Builder

- ⊗ represent mutable
 - ⊗ thread unsafe
 - ⊗ String Builder is recommended
 - ⊗ Constructors : default \Rightarrow length=0 capacity

String Builder (int capacity);

- ④ StringBuilder (
 ↳ StringBuilder sb1 = new StringBuilder("Hello");

Introducing memory optimisation for strong builder
blocks of mutability.

$\rightarrow \text{Sys}((\text{"contents"}) + \text{Sb1} + (\text{"length"}) + 5\text{b1.length} + (\text{"n"}) + \text{Sb1.capacity}))$

Append

~~Stringbuilder~~ ~~sb2~~ = ~~sb1.append(123,45)~~

5

length + 16 → 21

Sysso ("Sb₂S contents" + Sb₂), 111

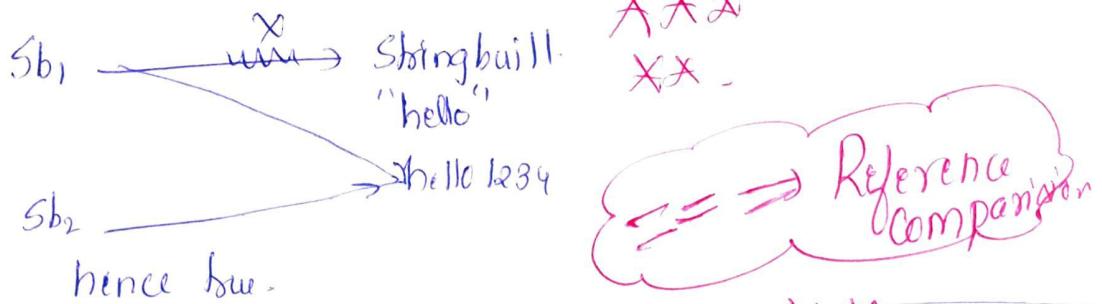
```
sys("5b1s contains "+gb2); // hello 123-45
```

↳ $\text{H}_2\text{O}(\text{g}) + \text{SO}_2(\text{g}) \rightleftharpoons \text{H}_2\text{SO}_4(\text{l})$ (exothermic)

$$\text{Systo} \left(Sb_1 = Sb_2 \right) \quad || \text{True} \quad 11012345$$

- ④ **Semimutable** when append → it can make changes on original cop

62



`StringBuilder` `SB3 = new SB ("festing SB");`

`SB4 = ("festing SB");`

`sys0 (SB3 == SB4) // false`

`sys0 (SB3. equals(SB4)) // false`

~~String builder → equals~~

~~Reference~~

2+ do not

Has `SB` class overridden `equals` method of `Object` class?
I.e. it is using Inherited version and object method based on ref equality.

everything is modified on same object

no new object is created

④ equals is inherited from object if

`sb2 = sb1.append(true).append(1234).append(1234).append("some")`

`sys0 (sb1)` → appended content

`sys0 (sb2)` → appended

`(sb1 == sb2)` → True.

Sb. reverse () ** 63
↳ Sb. reverse() → reversed content in SB,

Date time handling

AP2

java.util.Date → store current date with precision till msec precision.

Constructor :-

① implements Comparable <Date>;

Deprecated → as of next versions it may be removed

② we should not use

Date() → Default constructor.

Date (long date) → no. of milliseconds passed since
Jan 1 1970.

equals → compares two dates for equality.

↳ It overrides in class Object. **

↳ Overrides to String **.

* before *

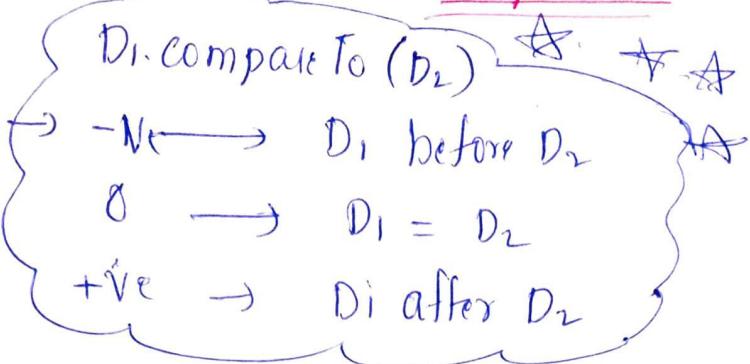
Return true → if D₁ before D₂ → True else false.
after

** D₁.after(D₂) **

if D₁ is after D₂

compareTo() **

Compare To



main() {

Date d₁ = new Date()

Date d₂ = new Date(1000);

System.out.println("Time in millisecond " + d₁.getTime());

25T.

→ System.out.println(d₂) → Epoch date → Today's date & time.

System.out.println(d₁.equals(d₂)) → false

(d₁.before(d₂)) → false (2021 before 2022)

(d₁.after(d₂)) → true

(d₁.compareTo(d₂)) → +ve no

13:34

☆☆☆

thus Jan 01 5:30:01 IST 1970

GMT IST

5½ hr

Accept Join date of emp. from user and display same.

main() {

To read Date from
Scanner -

No

☆☆

☆☆△

DateFormat.parse(string s)

Passing → String to Date conversion
Formatting → formatting (Date → String)

y → year

mm → month in digit

mmm → month in abr (JanFeb.)

mmmm → complete month name.

d → day

h → hour

m → minute

s → second

Separator → -, /

Q) Passing → string → Date

public Date parse(String s) throws ParseException

SimpleDateFormat

→ parse(str)

Format → Date → String

11-1-2022

Java.text.SimpleDateFormat

Main() {

try (Scanner sc = new Scanner ()) {

SimpleDateFormat sdf = new SimpleDateFormat ("dd-MM-yyyy")

System.out.println("Enter Date");

Shows
errors

← Date joinDate = sdf.parse(sc.nextLine());

catch (Exception e)

{
System.out.println(e);}

10/30/2021 → { 10 - 30 - 2021 }

sdf.format(joinDate)

→ Direct/straight

27/10/2021 8:00

① adding to the pool externally?

$s = "hello"$ → as already 'hello' present in pool

$s_2 = s.intern();$ → s_2 start pointing to hello

~~String~~ $s_2 = \text{new String}("Hello");$

→ Hello added to pool

→ one more in non literal

→ and non literal is returned

Date and time handling:

⊗ `java.util.Date` → represent Date & time both, till precision msec.

Consts: -1-11

`Date()`: creates date class instance.

⊗ `d.before(d2)` → True → d is before d₂

⊗ `d1.after(d2)` → True if d₁ is after d₂

⊗ Compare to → `d1.compareTo(D2)` → -ve → d₁ before D₂
0 → d₁ = d₂
+ve → d₁ after D₂

Objective to manage crude operation.

⊗ How to accept DOB? \rightarrow Date.

We cannot accept using Scanner:

Steps

① Create instance of `SimpleDateFormat`

Single copy b/o
if will act as
Converter

② `SDF = new SDF("yyyy-MM-dd")`

③ Parsing → string to Date conversion:

`Date DOB = SDF.parse(sc.next());`

67
⊗ SDF → may throw exception and must be handled b/o checked Version.

⊗ Date D1B = new SDF.parse(next())
→ if contains time and IS raw

How to format Date

Date → String

Vehicle assignment XXX

⊗ → close project → unnecessary project creates delay.

public class Vehicle {

private int chassis no;

private String color;

private double price;

@ override ToString. → Create constructor.

main() {
 try (Scanner sc = new Scanner ()) {
 System.out.println("Enter vehicle details");
 Vehicle V1 = new Vehicle(sc.nextInt(), sc.next(), sc.nextDouble());
 Vehicle V2 = new Vehicle(sc.nextInt(), sc.next(), sc.nextDouble());
 System.out.println(V1.equals(V2));
 }
}

Invokes arg blo equals is called of Object class
System.out.println(V1.equals(V2)? same; different);

⑥ If we don't override "equals" it will compare reference.

System

To get address

```
System.out.println(v1.hashCode() + " " + v2.hashCode());
```

To check depending on chassis no

Inherit stand and override it in Vehicle

→ public boolean equals(Vehicle anotherVehicle)

{

```
    System.out.println("in Vehicle's equal method");  
    return this.chassisNo == anotherVehicle.chassisNo;
```

@Override → checks for method override

What happens if we don't write @Override?

⑥ Code will work but from Vehicle method.

⑥ Then why?

replace Vehicle → object.

```
Object v1 = new Vehicle();
```

```
Object v2 = new Object();
```

```
System.out.println(v1.equals(v2));
```

→ hence adding @Override will solve the problem!

Why no error?

```
v1.equals("Hello");
```

b/c it is object

It will give runtime error

class cast exception.

SOL

check instance before down casting

```
If (o instanceof Vehicle)
```

```
    return this.chassisNo = ((Vehicle)o).chassisNo;
```

What will happen?
Vi.equals(null) \rightarrow Instance of null gives you false

69

Change chassisNo String

- ④ start Vehicle class

state: chassisNo, color, category, price, manufacturer, Date

Constr.:

\rightarrow toString \longrightarrow + manufacturer Date \rightarrow unformatted date

\rightarrow equals (overrides) \rightarrow based on Primary key;

\rightarrow SDF sdf; \rightarrow we require only one copy hence make it static

To format date

sdf.format();

VehicleHandlingException:

- ④ extends Exception;

- ④ add parameterized constructor

- ④ message.

Validation functionality, utils

add static methods for rules

④ public static ~~void~~ ValidateDate(..)
 \downarrow (return) Date \downarrow parseN Date String
 \downarrow parsing conversion do in function

- ④ parsing use static Import

- ④ handle **Centralised error Handling** \rightarrow Throws.

- ④ check before and after for checking constraints.

Throws ParseExc, VehicleHandlingException {

Tester "U7" keep user interface clean and neat

⊗ conversion and other parts in methods

try (Scanner sc) {

init phase → Vehicle array []

exit = false;

Counter:

while (!exit) {

switch ()

case 1: add vehicle;

How to avoid entire about by try block

Inner try catch :-

try {

switch

}

catch-all

Coding part: 10.2

has-a.



Vehicle class:

public static SimpleDateFormat sdf;

static {

sdf = new SimpleDateFormat ("yyyy-MM-dd");

}

make String chassisNo;

⊗ @override equals method

{ instanceof ()

return (this.chassisNo.equals ((Vehicle).chassisNo));

Custom Exception:

⑧ Vehicle Handling Exception

```
public class VehicleException extends Exception {
    public Exception() {
        Constructor();
    }
}
```

Constructor ~~XX~~

Utils →
ValidationRules {

Static → One copy

II add static method to parse n manufacture date, import from java.text

```
public static Date parseAndValidateManufactureDate(String manuDate)
    throws ParseException, InvalidManufactureDateException, VehicleHandlingException
```

II parsing

```
Date di = sdf.parse(manuDate);
```

Import static com.app.Vehicles.Vehicle.sdf ~~XX~~

↳ import

Hardcore → Boundaries

Date startDate = sdf.parse("2021-4-1");

Date endDate = sdf.parse("2022-3-31");

yyyy-MM-dd

If (di.after(startDate) || di.after(endDate))

throw new VehicleHandlingException("Invalid manufacture Date")

put in
static Block
to avoid repeat

return di;

Dont redeclare in static Block

Static {

Date startDate = "10-2-2021";

throws appears in only methods ~~XX~~

Initialisafn phase :- 10.1

xx main() {

try withen try.

try() {

switch case;

3 catch(Exception e)

case1 : // add vehicle on validafn.

if(counter < vehicles.length)
{

{
else {

throw new VehicleHandlingException("showroom full");
break;

Import static utils. ValidationRule.parseAndValidateManufactDate();

Validate date

sys0(Enter vehicle data);

Vehicle v = new Vehicle(sc.nextInt()); ~~sc~~ parseAndValidateDate(sc);

Parse and validate
in one

Year format :- 2021-6-20 (yyyy-MM-dd)

2021/6/20

Invalid pattern of Date.

Root cause of infinite looping :- ***

pending tokens, not cleared from buffer.

add sc.nextInt();

To clear of
pending to fill new
lin.

sc.nextLine();

Avoiding Duplicate (Primary key)

Validation rule :- check duplicate fundtn.

using equals method.

→ based upon Vehicle equals.

public static int checkForDuplicate (String chassisNo) {

Validated op → Storage
Ctrl+Shift + O → addition
Ctrl+Shift + I → import
→ ~~AAA~~

I/p parameter

→ chassisNo, Vehicle [],

Vehicle Array

Vehicle [],
Showroom

II Iterate over vehicle array.

II iterate over the vehicle [] → if found throw exception.

for (Vehicle v : showroom)

 if (v != null)

 if (v.equals(chassisNo))

 throw new VehicleHandlingException("Duplicate");

 return chassisNo;

always false → if (chassisNo is instanceof Vehicle)

~~AAA~~ Wrap chassis → To Vehicle
 so giving vehicle is expected.

How to wrap();

→ use Vehicle object with special constructor.

⑧ Overload only "chassisNo"

In Vehicle public Vehicle (String chassisNo) {
 this.chas = \rightarrow

In Validate Rule

Vehicle newVehicle = new Vehicle(chassisNo);
for (Vehicle v : showroom) $\star\star$
 if (v.equals(newVehicle)) $\star\star\star$
 throw NewVehicleHandlingExceptn("DUP");
 return chassisNo; $\star\star$ return valid non
 duplicate chasis

Importing all static

import static validationrule.*;

Combine Validation + parsing $\star\star\star$

Date should match format of spf $\star\star\star$

Validate category Enumeration:

Vehicle has a category ✓

Vehicle is a category X

\otimes Enum is key word in java, like class, interface.

\otimes set of related constant

\otimes car manufacturers have limited category

\otimes they are special class extends java.lang.Enum;

\otimes Compiler add too much details in Java

⊗ if combines class and interface features.

Why enum? ~~☆☆~~
popular

10.3 → Com.App.Vehicle →

private String Category;

⊗ Can be declared in class / outside class / separately.

dropdown list → enum type :-

public enum Category {

PETROL, DIESEL, EV, HYBRID, CNG; ~~optional~~

} ~~ordinal pos → 0~~
Name → petrol.

Ordinal pos 4
Name → CNG.

Rule Decompiler ~~☆☆~~



Can we extends as Enum;

No → because it is final

Javadoc → Enum. → public abstract class Enum.

Constructor access specifier → protected

Private constructor.
 Enum(name, ~~in~~ ordinal).

Enum has private constructor.

⊗ all Enum has private , →

Interface Vs enum

Interface movie constants {

int AGE_MINOR = 16

int AGE_MRN = 10

int AGE_MAX = 70

X ~~Why enum~~
④ iteration over interface is not possible but in enum it is possible.

④ Who adds value(s) in Enum;

Return type of value of

Method for iteration → Value()

Ex:

public static Category ValueOf(String s);

String → enum

↳ Parsing method to convert to enum.

④ Compiler adds additional code.

④ enum is abstract.

④ enum → name Petrol DIESEL EV HYBRID CNG
 ↳ ordinal pos.

④ Value ValueOf
 ↳ static, public, ValueOf(String s);

Return → array of enum

If user z/p Petrol → throws Exception, illegal Exception detail.
illegal argument detail

④ It is unchecked exception

Public static final Category PETROL;

→ all self type constants → static category

Category is reference type.

Static member initialiser block to initialise enum

PETROL = new category ("PETROL", 0)

DIESEL = new " " () , ,)

" " , ,)

① static member memory allocatn and followed by initialisatn.

(X) Public enum

Private Category category;

change to enum

change from string to Category.

whose to string called → for enum. it is overridden to return enum constant.

Relatn b/w Vehicle & Category(enum)

has a, used for Reusability

Reference of type: Category:

Validation Rules: add a method to convert enum

Case Sensitive
Value of

parse & validate → String → Category; + Validation.

Value of

public static Category parseAndValidateCategory(String categoryName)

return (Category) ValueOf(categoryName.toUpperCase())

}

Flow to accept Enum through Scanner

Variable - Argument

① Variable arguments → Syntax . . .

Rule : variability to last arg only.

Void doStuff (int... x) { } 0 or multiple arguments

}

Usage : ref. doStuff()

int [] ints = {1, 2, 3, 4, 5}.

ref. doStuff (ints);

String... args → In main funct

~~ref. doStuff2~~ compulsory → Parsing is optional → Int 0, 1, 2, 3 . . .

Void doStuff2 (char c, int... x) { }

ref. doStuff2();

class Test

Void doStuff3 (Animal... animals)

for (Animal a : animals)

3 .

Animal a₁ = new Horse()

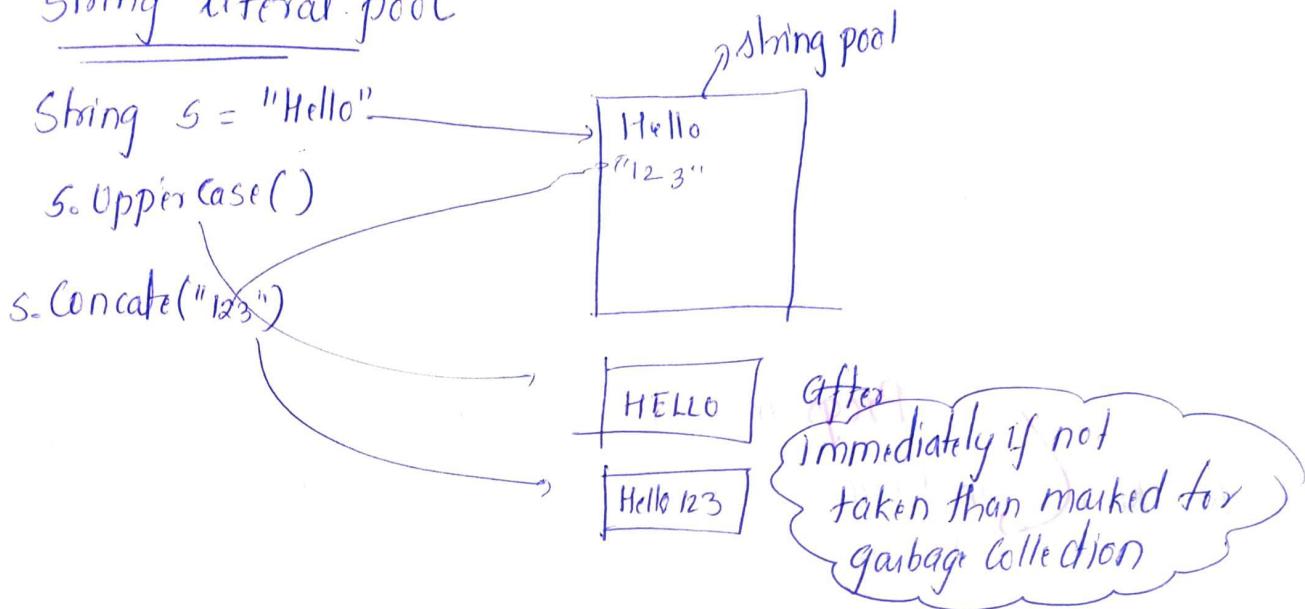
a₂ = " cat ()

Wrong Syntax

Void doStuff (int... x) { } 3

Void doStuff(int...x, char...y) X
 (String...x, chary) → X it should be last

String literal pool



Interning

* adding unique literals to the pool.

" everything will be added to pool

+ → is added to pool

Concat → no adding to pool

Q8 | 10 | 2021

How to access : enum constant

Interview Ques)

XX

which function of enum cannot be overridden?

→ equals, compareTo

which interface implements?

Comparable

To use enum constant directly

Import static com. app. core. Color X.

sgso(RED.compareTo(Blue)) --> (1)

if is latter
than RED

Q) food Bill generator using enum :-

Day 11.1 Advantages of enum :-

package `com.app.core;`

`public enum FoodMenu {`

`NOODLES, RICE, SOUP, SNACKS;`
1 2 3 4

Inbuilt prices.

`Noodles(60), Rice(100), SOUP(120), SNACKS(50);`

`private int price;`

→ Programer defined const.

④ No other class can create enum constants. → compulsory int.

`private FoodMenu (int Price) {`

`this.price = price; }` → Inbuilt Super(name, ordinal)
called by compiler.

}

How many attributes (members)?

5 (including price)

Can we add getters & setters of enum

{

`getPrice() {`

public

`return price;`

}

3

@Override

`public String toString()`

{

`return name() + " @ " + price;` *toString() returns new str*

}

Tester for the enum

8)

main() {

try (Scanner sc = new Scanner (System.in))

{

boolean exit = false;

while (!exit) {

To show
from enum

StringBuilder availableItems = new StringBuilder ("menu:");

for (FoodMenu m : Values())

availableItems.append (m + " ");

while (!exit) {

System.out.println (availableItems);

To String will
be called

Use of enum
in switch
Case

Import static
fully qualified name,

accept only
capital

Convert
from String to
enum

FoodMenu item = ValueOf (sc.next().toUpperCase());

switch (item) {

case NOODLES:

case RICE:

case SOUP:

case SNAKES: bill += (sc.nextInt()) * item.getPrice();

catch (IllegalArgumentException e) {

System.out.println (e);

① Why name enum2

① Helps to create Constants

② Iterating is simple

Interface FoodMenu

String Dosa

IDLZ

};

④ ~~(X)~~ If exception is raised Complete try block is aborted ~~X~~ ✓

★ ★ ★ Has-A relationship ★ ★

① Vehicle has chassis no

② " " color

 " " price

 " " category

Association:

③ It is relationship b/w 2 separate class using object references

Represent Has-A

④ Code Reusability.

⑤ one to one, one to many, many to one, many to many.

Car has engine | one department | authors book } faculty → skills.
many-to-one many-to-many many-to-one

Composition & Aggregation part-of reltn

⑥ both have a relation
Stronger Weaker

⑦ e.g. class a student / Bank Has a Customer.

car engines

Bank closed customer will not go away

class Bank {

 private String name;

 private String ifsc; → Weaker coupling

 private String address; Aggregation.

Composition

83

(part of or Belong To)

Human HAS A Ling (tight dependance)

⊗ person Has A address → (tight dependance).

(When a parent deleted, typically child can't exist on its own)

Nested class

Establish Has-A relath b/w Vehicle and delivery Address

⊗ We should not be able to create object of Address w/o Vehicle.

⊗ Vehicle and delivery address.

① Add new state (instance Var): Delivery Address;

Don't init it in the const (since delivery address should be assigned when customer purchase).

Nested class :

11.2 class Delivery Address {
private String city, state, country, zipcode;}

Const:
To String;
}

establish HAS-A relath b/w Vehicle and address

Add in Vehicle

private DeliveryAddress address; *default Value = Null;*
Memory = Heap;
don't add address in constructor

Link Vehicle → Address

Add non static method to assign delivery address.

public void assignDeliveryAddress (

DeliveryAddress = new DeliveryAddress (
this.address

Test aggregation

main() {

Vehicle V1 = new Vehicle ("abc-1234", 3456, ^{Color: BLACK,} ...)
Instance of delivery address.

Delivery Address adr = new DeliveryAddress (" ", " ", " ", " ")

V1.address = adr; \times not allowed \times

V1.adding DeliveryAddress (, ,)

Weak form Aggregation

sys0(V1)

How to show whether Vehicle is sold/not sold

Aggregate @ override

Why ~~Composition~~ String addres (address == null ? "No Delivery", address, toString)

To convert to string from object

return "chassis, color, category, address";

because we are able to create a object of Address therefore

How to achieve tight coupling.

We cannot

Add static method

non-static nested class

static nested \star

Types of nested class in Java

method local inner class

class {

void f() {

class

{

}

Anonymous

- ① can access private member of outer class (directly)
- ② can have access specifiers (To hide complexity).
- ③ cannot contain static members, since inner class can be ~~public~~ to ~~private~~ created only with outer class.

Why nested :

non static nested class

Ctrl+Shift+f → formatting, 85

class Vehicle {

 public class Delivery {

}

{

Test Composition :

Ctrl+Shift+O → remove unused import

Vehicle V₁ = new Vehicle();

DeliveryAddress add = new DeliveryAddress();

Java C
errors
we cannot
create w/o
Vehicle

How to link

V₁.assignDeliveryAddress("Pune", add);
Syso(V₁)

→ to print city;

city is inner member i.e private

→ Generate Getter for city inside inner class.

How to call

→ Generated Get address in outer class.
Syso(V₁.getAddress().getCity())

★ Directly
access not
possible

★ Nested methods are not allowed in Java.

★ Class file of DeliveryAddress:

Vehicle \$ DeliveryAddress.class

→ dollar -



Q When outer instance created is its inner object created automatically?

No

Q Can inner class access outer elements?

Yes

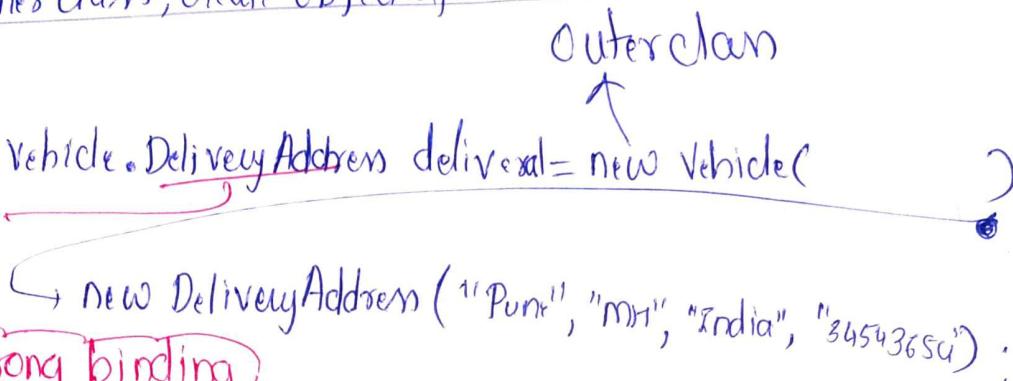
Q Vice versa is true?

No, to access inner class, create object of it

Test 2 :- main() {

To create object
of inner

Syntax → Strong binding



Adding additional option to Show room

case 3: System.out.println("Enter chassis no");
separate (finders/utils/validation) in other class by pk.

function in Validation Rule

Central Exception handling

public static Vehicle findByChassisNo(chassisNo, Vehicle[]){}

Create ve
wrap chassis to Vehicle

Vehicle V_i = new Vehicle(chassisNo);

for (Vehicle V_i: vehicles) → Return this

If (V_i != null)

If (V_i.equals(V_i)) → Vehicle is found

Return V_i;

throw new VehicleHandleException("Duplicate")

}

(Case 3 →

import static

Case 3 → findByChassisNo(Sc.next(), Vehicle);

(ctrl+I)

if check if already sold

assign to local

if (Vehicle.getAddress() != null)

Variable

Continuous
means available

throw new VehicleExp("Vehicle Sold")

System.out.println("Enter address : city, state, zip");

Vehicle.assignDelivery(sc.nextInt(), sc.nextInt());

System.out.println("Vehicle ready for delivery");

{}

* * Interview

11:54

Q) Can outer class have static & non static

Yes private int i;

private static int j; → gets memory first

Static {

j=100

{

Constructors {

{

Static test() { → can access only j (static part)}

System.out.println(j);

{

test2() {

Outer & Inner

System.out.println(i);

{

Inner class (non-static)

class inner {

Q) Can inner class contain static members:

No (IDK 1.8 onward, static final allowed);

private static final int k; ✓

Q) Can we write static initialiser block

"NO" static type don't allow any static.

Q Can we declare non static data members in inner? *

Yes private int i;

```
public inner(int i){  
    this.i=i
```

Q Can we add static method in inner block? ***

No X public static void display(){}

Q Can we add non static method in inner class? ***

```
public void fesht(){
```

```
{ System.out.println("inner non static"); i++ }
```

All are access i, j..

* along with inner class implicitly given reference of outer class

↳ we can access private members of outer class directly w/o obj.

Again switch to outer class

can outer class non-static method access inner class member directly?

NO, to refer create object of inner class and access

inner.k → final private still accessible.

* even private we access but after creating object *

What is accessible in outer static method?

Yes inner.k → only static final.

i → not accessible

* static {

inner in = new inner(848); X

X X X *

Outer.inner in = new Outer(11).new inner(22);
System.out.println(in.i);

~~A A A~~ Ambiguity ~~A A A~~ :- Interview

public outer2 {

 private i;
 pub main()

How to call



 3
 class 2nn2 {

 private int i;
 void show(int i) {
 sys(i) → local.
 sys(this.i) → inner class

C/P

30

20

To refer outer i show(outer2.this.i) ~~A A A~~ 10

How to call show method?

Outer2 outer = new Outer2(); → one object created

Outer2.Inner2 in = Outer2.newInner2();
in.show(30);

Wrapper

class equivalent to primitive type.

Why?

① Collection is growable type



② limitation → collection cannot accept primitive type

③ wrapper class has useful API's

Q5

Java.lang.Object

java.lang.boolean

java.lang.Character
char

java.lang.Number
Double
Float
long
Byte
Short
Integer
int

all are
at same
value

Integer is not Double.

Boxing conversion

from prim → wrapper

e.g. Integer ref = new Integer(1234);

Boolean ref2 = new Boolean(True);

Unboxing

int data = ref.valueof(Integer);

11.3 class Test1;

package Wrapper;

public class Test1 {

main() {

deprecated

 Integer i1 = new Integer(1234);

instead of Integer
we use this

 Integer i1 = Integer.valueOf(1234) // boxing

int → Integer
done by programmr

⑤ Double d1 = Double.valueOf(123.45) **Boxing by programmr**

* * int data = i1.intValue(); **Unboxing by programmr**

Integer i2 = 100; **Auto Boxing** earlier version gives error
int → Integer

Double d2 = 100.456;

`data = i2` → auto unboxing

~~i2~~ wrapped → int.

→ internally call `i2 = i2.intValue();`

`String s = "12345";`

`s++;`

(Java) Arithmetic operation not allowed on ref.

~~int i++;~~ } no error → auto unboxing and then increment
~~int i++;~~ Integer → int → int++.
 Integer (ref type)
 Ref ++ (all operations are performed internally)

A Interview Quesn

`int i = 123;` Widening ~~int~~

~~double j = i;~~ → Widening (auto promote)

~~Double k = 1234; // Compil. → auto boxing → int → Integer~~
 Double. ← ~~X~~

Solutn → Double k = 1234(d).

(double)(1234);

Number

Number n = 123.456F; ↗ (upcasting)

datatype

float → Float (upcasting)

Object

~~O = 123.45f~~ ↗ Upcastin

~~O = 45;~~

~~O = false~~

autoboxing float

Float is Object

boolean → Boolean →

Is Java 100% OOP?

✗ because also support primitive

✗ multiple inheritance

final

✗ open operators overloaded

Any thing can be accessed by object

Object can refer to primitive and non primitive

Generics

Generic applicable for class, interface, methods, constructor.

Why :

it is complete meant for @ compile time

illegal type casting.

all the time we were adding instance of .

makes Type safety to code;

We will not require down casting *

To avoid typecast exception.

Non Generic class :- Holder → can hold any datatype in Java

public class Holder {

private Object reference;

public Holder (Object reference) {

this.reference = reference;

}

public Object getReference () {

return reference;

}

Explicit
downcasting

Tester → create holder class instance to hold int value.
Holder hi = new Holder (100);

→ int to Integer is object
upcasting

int data = hi.getReference(); Java error

Down casting required

Explicit downcasting

int data = (Integer)hi.getReference();

Object → Integer → int ✓

Disadvantage :-

④ down casting required every time

3.15

④

Holder h₂ = new Holder("12345");

Upcasting by Java

To String

is object

String s = h₂.get Reference();

No auto boxing & unboxing

downcasting explicitly

String s = (String)(h₂ . . .);

h₁ = h₂; → no error {object is pointing to object}

data = (Integer)h₁.getReference() → class cast exception

string cannot be safely cast into integer.

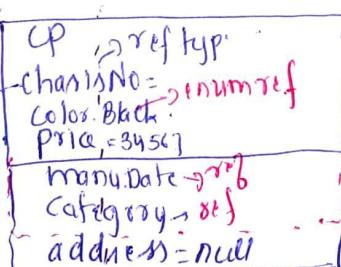
Java allowed due to same ref.

Compiler is not educated to look at pointing object

Vehicle v₁ = new Vehicle("abc-12345", color.BLACK, 34567, new Date(), category,

v₁.assignDeliveryAddress("Pune", "M4", "India", "845637")

DATE



enum class

Object

Vehicle

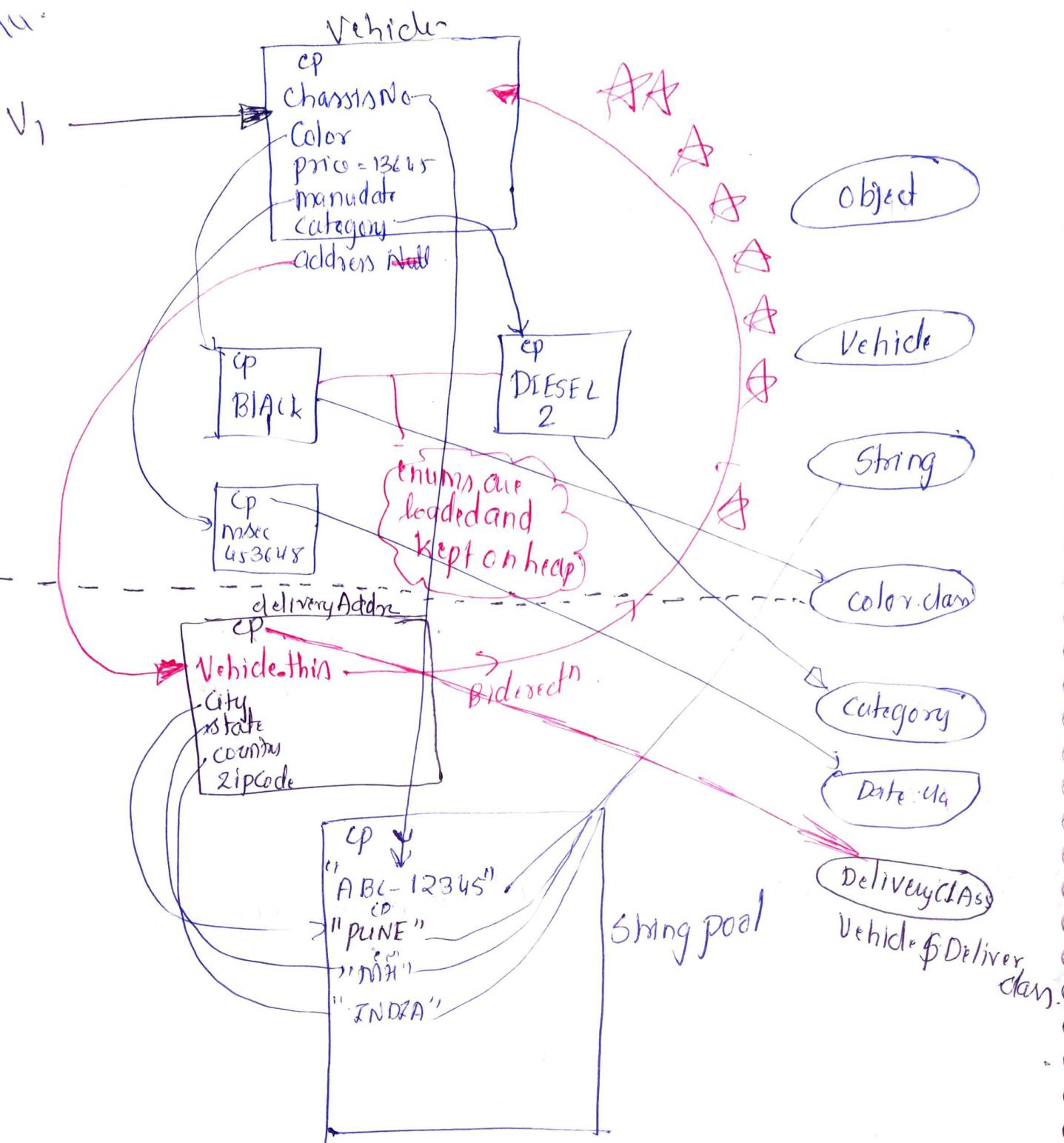
color class enum

String

DATE class enum

category class enum

String pool



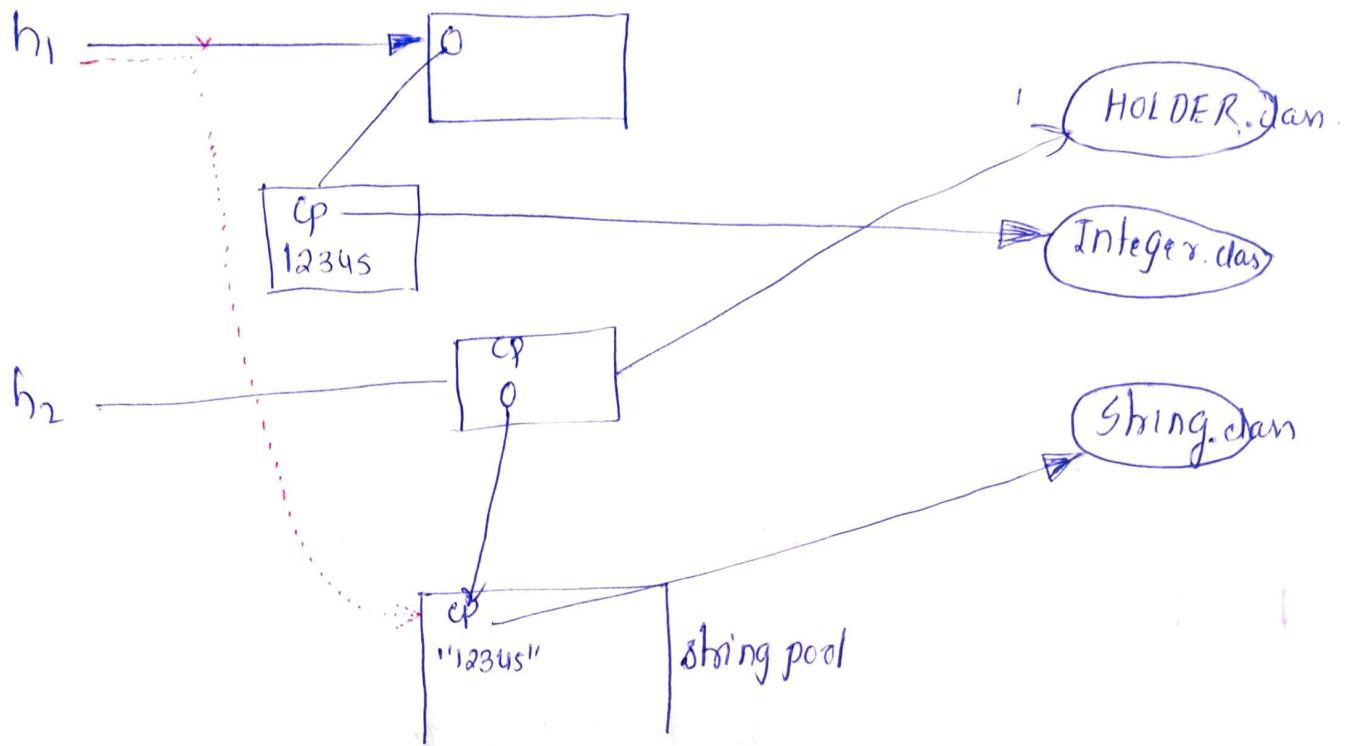
29/10/2021

95

```

Holder h1 = new Holder(12345);
int data = (Integer) h1.getData();
Holder h2 = new Holder("12345");

```



⊗ If reference variable is part of some state of instance than it is allocated in heap rather than stack.

B 12.1 non-generic → generic:

Create a class that can hold (state) any data type

public class Holder<T> {
 can be reference class only.

· placeholders to indicate what data it can hold

private Object

T o;

public Holder(T o) {

super();

this.o = o

}

AT public class Holder<T, U>

private T o;

private U o2;

USAGE

⊗ Testers without downcasting.

⊗ Create a holder class instance to hold int val

Holder<Integer> h1 = new Holder<Integer>(12345);

RHS is optional

Diamond operator

Holder<Integer> h1 = new Holder(12345)

Q any Implicit conversion?

Holder autoboxing only no upcasting

{Holder<Integer>} replaces all & Holder<T>

holder to store a string?

Holder<String> h2 = new Holder("1234");

String s = h2.get0();

get0 will also return String

h1 = h2 : error

Java C

error detected at Java C

What is change in memory

No change.

Collection Framework

(A) Can we change size array dynamically?

No

(B) Any problem while insertion new element?

No not easy

(C) Can we delete any element in array?

typically difficult,

(D) Are they sorted?

No

(E) Is searching easy in the array?

No

Case of Vehicle

Reference

Collection → group of data handled as single framework

What is framework?

① Data structure algorithms

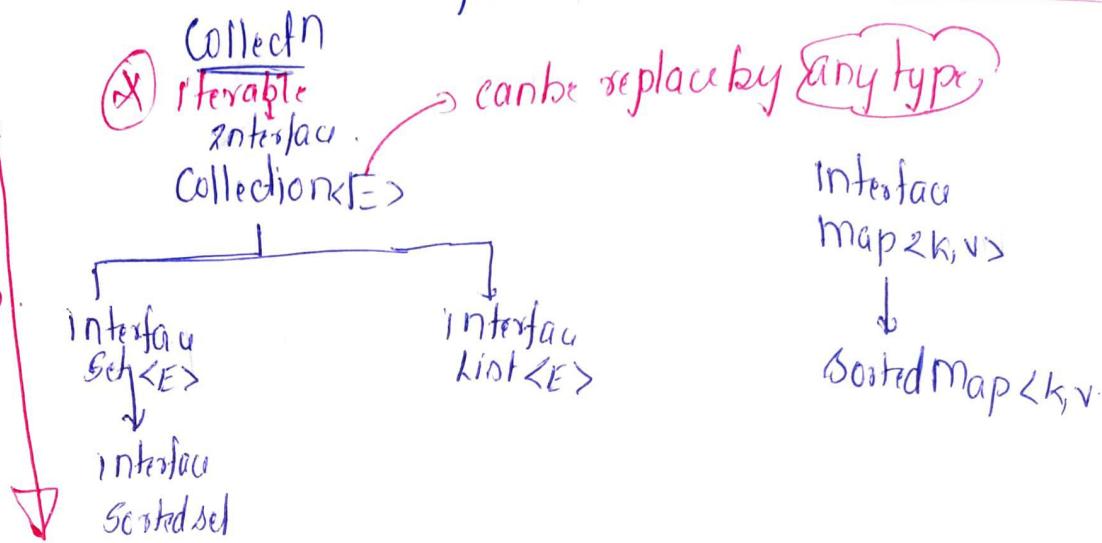
② Generalised; Ready made implementation of standard algorithms.

③ Collection framework →

④ it is extensive → it gives abstract class → we can use it extend and custom implement.

java.util

Generalisation
to Specialisation



Collection of employee = collection<Employee>;

Bankaccounts = collection<BankAccount>;

JSON: key and value pair:-

Super collct interface → Iterable.

* If any

we will be able attach iterator to implemented class.

Map is not iterable.

9.14

⑧ Iterator →

`java.lang.Iterable<T>.Interface`

Not part of collection framework
method `Iterator<T> iterator();`
Implicit iterator? `for each;`

Super Interface



`Collection<E>`: Sub interface of Iterable.
Does not specify any particular behavior.
No concrete implementation classes.
Contain method for:

`Collection<E> interface` ↗
`list`

↗
`SET`

⑧ `boolean add(E e);` ⑧ `boolean isEmpty();`

`boolean addAll(Collection c);` ⑧

`void clear();`

⑧ ↗ Clears all references

⑧ `Collection<T> iterator();` ↗

↗ Bulk removal

⑧ `boolean remove(Object o);` ⑧ `boolean removeAll(Collection c)`

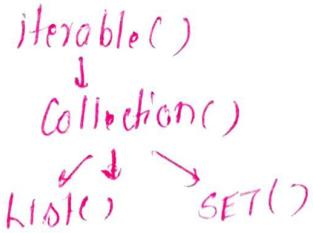
↗ Remove common elements

⑧ `boolean retainAll(Collection c);`

↗ non generic

⑧ `Object[] toArray();`

Convert Collection to Array



List<E> features

- ⊗ lists ordered collection → Order is significant (it remembers the order of insertion)
- ⊗ allows null reference → allowed
- ⊗ Allows duplicates: yes.
- ⊗ Supports index based operatn → only list supports.
will be able attach iterator to ArrayList, LinkedList, Vector.
- ⊗ Yes.
- ⊗ can we use for loop for ArrayList - ✓, - ✓, - ✓
- Yes.

Indexbased operatn

(a,b,c), (c,a,b) and (a,b,b,c) are different list.

- ⊗ the list interface inherits from Collection .
- ⊗ Important methods

Dealing with position-oriented operatn;

public void add(int index, E element); → E added at index
by shifting the element → insert

- ⊗ ArrayList is internally array

⊗ public boolean addAll(int index, Collectn c)
all the ref are inserted at specific locatn.

⊗ Public E get(int index)
return ref of element at index .

- Range → 0 to size-1; object.
- public int indexof(object element) → to first occurrence
- remove (int index)
- public object set (int index Element)

ArrayList:-

generic class

class ArrayList <Employee>;

Random Access interface

marker interface used by list to indicate constant time;

O(1) → factors

Unsynchronised → thread unsafe

Vector → synchronised → thread safe

which are fast operatn?

size, isEmpty, get, set, iterator and listIterator

Constructor:

ArrayList() → constructs an empty list with initial capacity "10"
'size=0'

ArrayList (int InitialCapacity)

Constructs empty list with capacity "10"

enough memory allocated at initial level

ArrayList(Collection<? extends E> c)

size → size + C.size

capacity → managed by JVM.

ArrayList<Integer> list = new ArrayList<>(100); 10:13 10)

① add(1) add(100)

② add(101) → automatic memory allocation
1.5 times original capacity

New project day 12.08.: package list

class IntegerList {
 main() {
 // Create empty ArrayList to hold int[] of refs
 ArrayList<Integer> list = new ArrayList<>();
 // Create int[] = sample data
 int[] data = {10, 28, 45, -1, 45, 10, 45, 10, ...};
 // Populate AL from this sample.
 for (int i : data) → implicit conversion → Non
 list.add(i); → int
 To display
 toString → return "[]";
 System.out.println("List contents via toString " + list);
 }
}

Allows duplicates

Retains order (not sorted)

Can you attach implicit iterator.

Yes.

for (int i : list) // Yes auto-boxing integer.
 System.out.println(i)

Trick Question

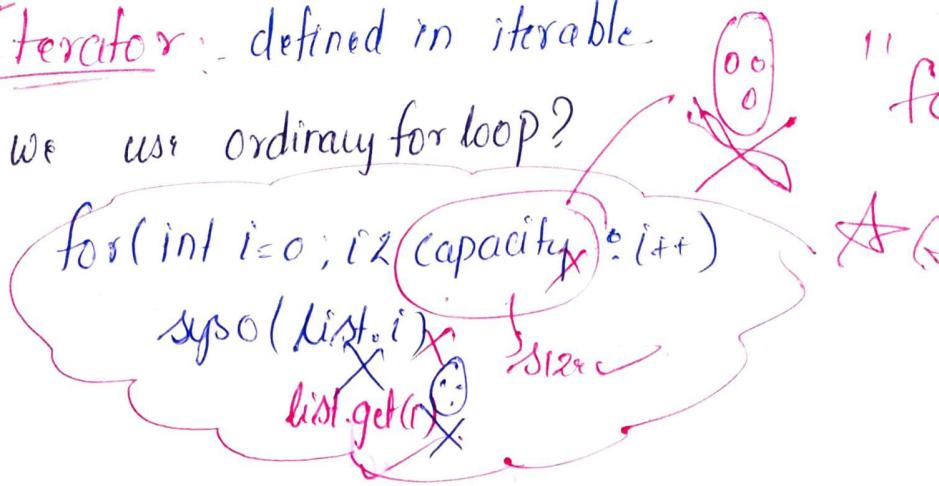
① initial contents of list?
hence

iterate over size
and size is 0

Iterator: defined in iterable.

Q) Can we use ordinary for loop?

"for loop"
for list



Explicit iterator:

iterable < T >

method = iterator < T > iterator();

return

iterator →
 Generic interface
 cursor that can transverse any collectn
 in forward manner

Method
→ public boolean hasNext();

Javadocs → iterator

* * * Difference b/w Iterator interface & Enumeration (not enum)?

Ans → Iterator

hasNext();

Returns true if the iteration
has more elements

to access next element
next();

Enumeration

longer names

⑧ Print list content using iterator?

① Attach iterator to AL

Syso (printing using iterator)

Iterator<int> list.iterator();
int i =

```
while (it.hasNext) {  
    System.out.println(it.next());  
}
```

genetic type will be Integer.

position of its is
BEFORE the first element

cursor is before first element

10 20 30

→ pointing to size

after looping position of cursor
after the last element

Java.util.NoSuchElementException

If index = size →

- ⊗ Index out of bound → thrown typically when trying to access
extra than size

`hasNext()` → checks if there is more element
`next()` → to access the next element

CRUD operatn

integers 2

`list.add(0, 999)` → shifts the current element -

$\text{sys0}(\text{list after insert});$
+list

10 Create another type of ArrayList Creating Copy

ArrayList<collection <? extends E> E)

ArrayList<Integer> list2 = new ArrayList<()>(list);
System.out.println(list2);

Recommended

To String

Two step procedure

ArrayList<Integer> list2 = new ArrayList<>();

System.out.println(list2.set(list2.size() - 1, 5555));

Setting the element

Overwrite

set() Returns old value
and used to set value

System.out.println(list.contains(10));

contains()

System.out.println(list.indexOf(10) + " " + list.lastIndexOf(10))

index of(10)

list.remove(0)

index '0' gone
all elements left

{ 0 1 2 3 ... 10
first occurrence

Index will start from '0'

Business logic:

(*) Vehicle having color, category.

Refactoring the code

sysc (Enter Vehicle Details);

Validate ALL Inputs (sc.next(), sc.nextLine(), sc.nextInt(), sc.nextDouble());

*this should return
Validated Vehicle*

Validation Rule class less code more clean.

public static Vehicle validateAllInputs (String chassis, Vehicle[] vehicles, String ch, double price, String date,

throw VehicleHandlingException, ParseException, String category).

{
 Check for Duplicate (chassisNo, vehicles); // chassis not duplicate
 Proceed to color
 Color c = parseAndValidateColor();
 Date d = validateDate();
 Category categoryValid = parseAndValidate (Category);
 return new Vehicle (chassis; c, price, d, ValidCategory)

}

Implement Collection frameworkDynamic array list:

ArrayList<Vehicle> vehicles = new ArrayList<>();

→ one object created

default constructor

constructor of ArrayList

Case 1: = Accepting array list

(ArrayList<Vehicle> vehicles)

Check for Duplicate () {

Ready made searching for Duplicate → No

fixed if (showroom.constraints (newVehicle)) {
 [→] Cells equals: Internally.

{ throw new VehicalException();

3

Case 1 :- To add vehicle.

Vehicles.add(v);

Case 2 :- for - each

for (Vehicle v : vehicles) {
 System.out.println(v);
}

(iterator goes over size
not capacity)

Contains method → internally calls equals (not capacity)

Create Sample Data

ArrayList<Vehicle> Vehicles = new ArrayList<

Collection → utils.

public class Collection Utils {

Will add a static method to return populated list of vehicle

```
public static ArrayList<Vehicle> populateSampleData() {
```

```
ArrayList<Vehicle> list = new ArrayList<>();
```

list.add(new Vehicle("abc-12345", false, "red", "red", "Pais.D9-12345"));

Import static utility to convert

45

call list in main

ArrayList<integer> Vehicles = populateSampleData();

To get details from ArrayList

Indexof, get

⑧ Get particular Vehicle Details

Case 3: // find key from - related details.

Validation Rules

public static Vehicle findByChassisNo(String chassisNo, Array<Vehicle> vehicles)

throws VehicleException

{

Vehicle v₁ = new Vehicle(chassisNo);

int index = Vehicles.indexOf(v₁);

If (index == -1)

throw error;

Return Vehicles.get(index);

case 3: sysc(Enter chassis no);

sysc(find by chassis No (sc.next(), vehicles);

⑨ Update price

, indexOf(), ~~(set)~~.^{get} setters, get, setter price, setter category

Case 4:- sysc(Enter chassis no);

v₁ = findbychassisNo(sc.next(), vehicles);

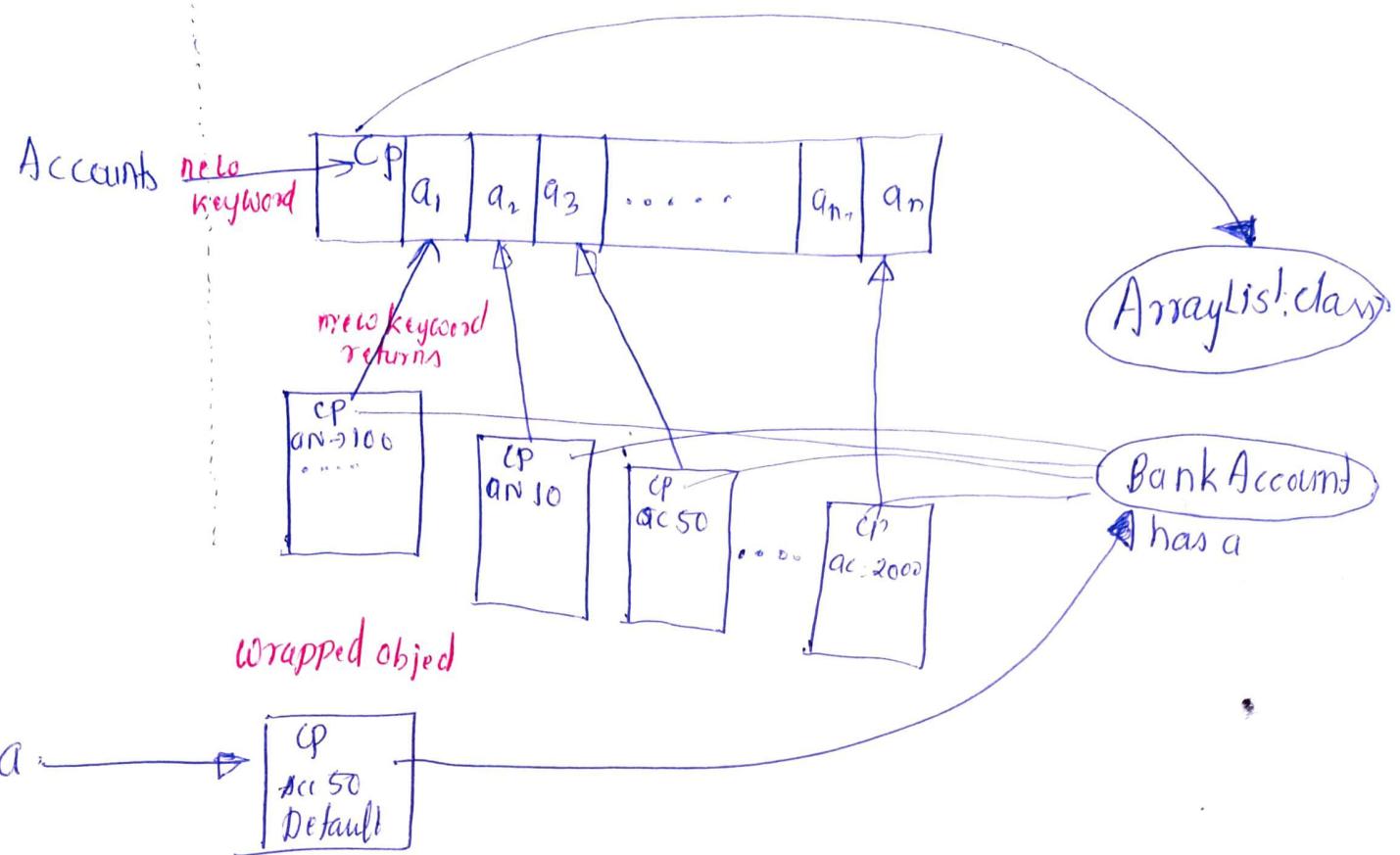
→ sc.sysc(enter new price);

v₁.set price =

29/02/2019

⑩ ⚡ Internal Working

ArrayList<BankAccount> accounts = new AL<>();
accounts.add(new BankAccount());



- * a. `contains(a)`
↳ internally iterator starts calling `equals` function

⑪ ⚡ Assignment Customer management system

constructor to primary key

```
public Customer (String email){  
    super();  
    this.email = email;
```

equally check
on email
@override

```
public boolean equals (Object o){  
    if(o instanceof customer)  
        return this.email.equals ((customer)o.email);
```

```
return false.
```

To string

Don't include password and format

⑧ public class Aadhar

private String cardNo;
private String locatn;

constructor()

toString() → We can format to string.

A {

Add instance method to link to the customer

public void linkAadhar(cNo , locatn) {

this.aadharCard = new AadharCard(cNo , locatn);

}

Custom_exception

public class CustomerHandlingException extends Exception {

public CustomerHandlingException(String msg) {
super(msg);

}

(ends with)

Validation_Rules

public class ValidationRules {

// add a static method for email verification.

public static void validateEmail(String email, ArrayList<Customer> customers) {

// check @ in email

if(email.contains("@") && email.endsWith(".com"));

Customer c = new Customer(email);

if(customers.contains(c))

throw new CustomerHandlingException("invalid");

throw new CustomException("invalid email address");

110 Tester Customer management :-

```
main() {
    try (Scanner sc = ) {
        {
            //initialisation.

```

```
        ArrayList<Customer> customers = new ArrayList<>();
```

```
        boolean exit = false;
```

```
        while (!exit) {

```

```
            System.out.println("Enter customer details");

```

~~30 | 10 | 20 | 21~~

Aptitude

Collection

④ Super interface of sets or list

④ Iterator, All(), isEmpty(), size()

Collections

class to manipulate std DS

collection framework.
Implementation of standard DS

④ two important interface List & Set, & Queue.

List

④ for loop / addWithIndex()
index of
get index of

④ List allows duplicate and null.

④ Concrete implementation classes:- ArrayList, LinkedList, Vector

④ Inheritance hierarchy of ArrayList

Javadoc → ArrayList

④ ~~(X)~~ ~~(X)~~ ~~size, get, set, iterator, listIterator~~ → Random access guarantees Constant access time.

⑤ there are no methods to detect capacity of array list?

Will you be able to create a populated AL from

① AL, LL, Vector ✓ these are collections

② HS, TS, LHS ✓^{OS} these are collections.

③ HM, LHM, TM X

add(int index, Element)
(0 to size-1) ~~size is method~~

add(E e) → appends

clear() → Remove all element

get(int index)
↳ constant time performance

if there are deletion & insertion?
link list
searching → ArrayList

Last index of (object o),

addAll(int index), Collection<?>()

contains (object o)

internally calls
equals of respective instance
& argument of equals is 'Object'.

indexof(object o);

isEmpty()

for each → implicit iterator

limitation of iterator → placed before first element.
to access the elements from back then we have to call
hasNext 500 times

④ also forward iterator.

<u>listIterator()</u>	remove(int index) → remove
listIterator(Index i);	remove(Object o) → equals

Set(int index, Element E)

which of the following will cause a structural modification of the list?

- (i) add
- (ii) remove { } ↘
- (iii) retainAll,
- (iv) SE

change in size

To array()

Why chassis cannot passed directly we need to wrap it?
if we pass chassis we will be comparing equals function on String object:

Delete vehicle detail.

① sys0(Enter chassisNo);

↳ remove(Object o), remove(by index)
easi.

② Vehicles.remove(findByChassisNo
shortcut method)

sys0("Removed" + Vehicle.remove(new Vehicle(sc.next())));

Apply discount to particular Date & Category before date.

searching key is not primary key.

manual iteration by for each.

sys0(Enter date & category);

Date d1 = parse & Validate Date(sc.next());

Category c = parse and Validity Category(sc.next());

for (Vehicle v : vehicles) → add getter
if (v.getManufactureDate().before(d) & v.getCategory == c)

enum comparison
equals & ==

v.setPrice(v.getPrice() - sc.nextDouble())

* If we give mistake
once & loop iterates
many times.

Searching In Inner class

add a purchase option

```
sys("Enter chassisNo to purchase Vehicle");  
if (Vehicle.getAddress() != null)  
    throw new exception.  
else _____ ①
```

Create object of inner class.

```
① - sys("Enter address city state country zipcode");  
    vehicle.assignDeliveryAddress(sc.nextLine());
```

case 8 sys("Enter city of Dispatch");

for (Vehicle v : vehicles)

Access inner class

city

if (v.getAddress().getCity().equals(city))

sys("chassisNo " + v.getChassisNo() + " price" + v.getPrice());

Break.

*** iterator / list iterator / ***

13.2 package tester
import static utils.

main () {
 try {

ArrayList<Vehicle> vehicles = populateSampleData();

Display all vehicles

Iterator<Vehicle> vehicleIt = vehicles.iterator();

while (vehicleIt.hasNext())

*** iterator

'sys0('VehicleIter. next());' X

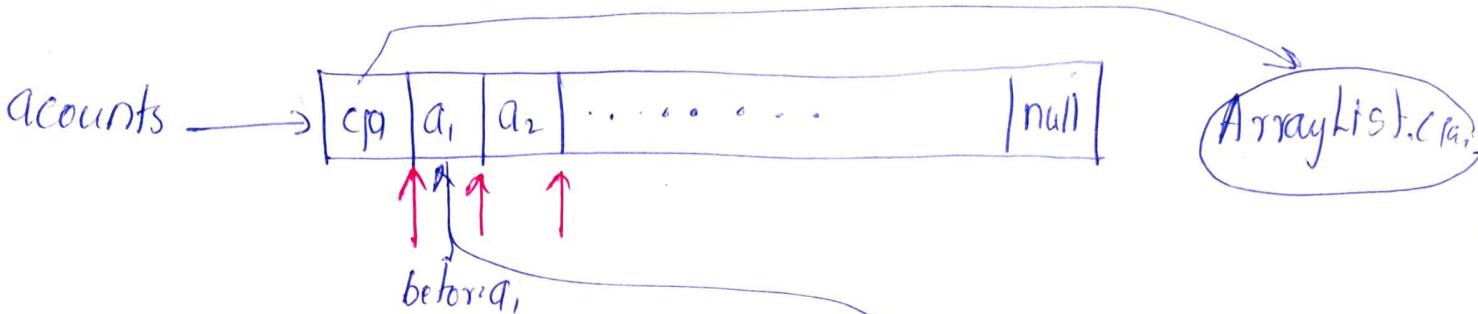
Memory picture of Iterator

Iterator < BankAccount >

accIter = Accounts.iterator();

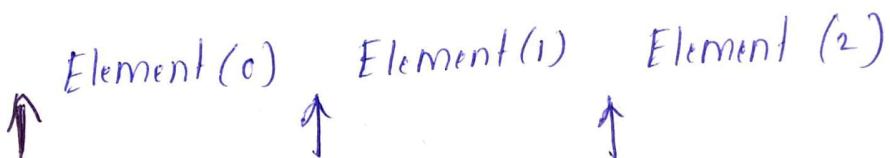
while (accIter.hasNext())

'sys0(accIter.next());'



when (hasNext), just checks if there is element yes a₁.

Next(); → hasNext → a₂, working of Next();
returns a₁ jump to



for (Vehicle v : Vehicles)
sys0(v);

Interview Point

What is requirement of iterator

→ to delete a element we get exception for each.



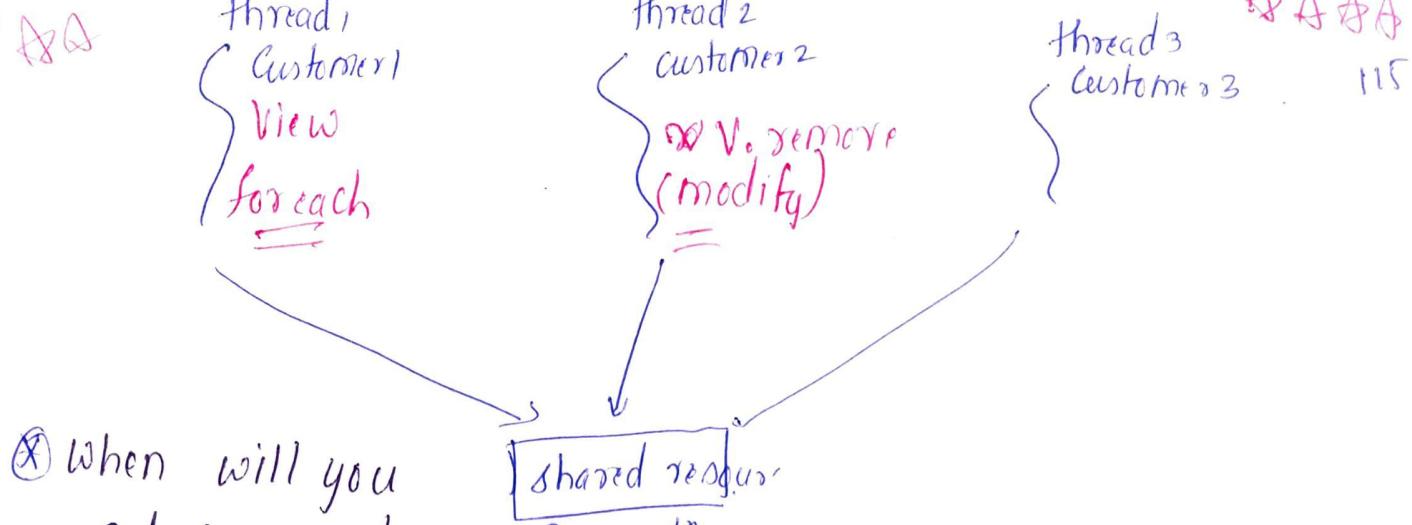
X Deleting costly vehicle

throws Runtime Exception.

if for (Vehicle v : Vehicles)
if (v.getPrice() > 1000)
Vehicle.remove(v)

Concurrent Modification Exceptn.
Java.util Java.util.ConcurrentModificationException

Concurrency (Parallel Execution)



Q When will you get Concurrent Modification Exception / fail fastness?

Ans → Java doc → fail fast →
 ① Iterator, list-iterator has behavior
 ② If the list is structurally modified when iterator is attached
 for-each
 Except iterator ~~the remove method~~
 ③ If not followed concurrent modification exception.

Solutn. → X

If r. Remove

Iterator<Vehicle> itr = Vehicles.iterator();
 while (itr.hasNext())

{
 if (itr.next().getPrice() > 100000)
 itr.remove();

} code over .

When iterator
is attached the
no modification
allowed

Test iterator 2

X X
Exception
Concurrent
Exception

because we are
accessing same
list for 2 times

list. iterator,

int data[] = { 12, 4, 2, 3, 4, 5, 6, 12, 30, 99, 44 };
 ArrayList<Integer> list = new ArrayList();
 for (int i : data)
 list.add(i);
 list.add(9999);
 System.out.println("List after add");

Iterator< Integer> itr = list.iterator();

list.add(9999);

System.out.println("List after add");

while (itr.hasNext())

System.out.println(itr.next());

A A A

115

1/6 Soln

16:50

doubt

D-13 saturd
30/10/2021

A A A

Illegal State Exceptn
A & Interview

Code :-

```
main() {
    System.out.println("original list " + list);
    Iterator<Integer> itr = list.iterator();
    while (itr.hasNext()) {
        itr.remove();
```

}

itr.remove

A

- Java docs → iterator → remove method of iterator.

① → itr.remove is used only once per b Next()

② → or we can use it if it is preceded by Next().

③ Next & Remove go hand in hand

* while (itr.hasNext()) {

itr.Next()

itr.Remove()

itr.Remove()

O/P Empty list

Runtime error

due to not followed
by Next();

Next()

Remove

Next()

Remove()

Error

because hasnext true
and next twice

Collections readm

A A A When will you get concurrent Exceptn?

④ In multithreaded single threaded system

⑤

"List-Iterator":

main() { iterator<T> listIterator; hasNext()
 ↓ Next() type forward

ListIterator<T>: list specific sub interface

of iterator<> (can be attach to list impl. classes)

How:

java.util.List method

① public ListIterator<> listIterator();

② Public ListIterator<T> listIterator(int index);

both forward/Reverse

→ before index.

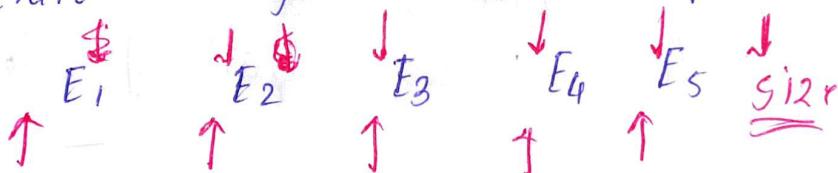
Java Docs → ArrayList → List → ListIterator.

⊗ bidirectional access

⊗ Return type is < Iterator<>

Display list in Reverse order:

ListIterator → argument based → put to size



main() {
 ↗ list
 already arraylist with int data[];

for (int i; data)

list.add

System.out.println(list);

place iterator
after last

ListIterator<Integer> litr = list.listIterator(list.size());

while (litr.hasNext())

System.out.println(litr.previous() + " ");

Position after completion will be '1' before first element.

118

Sorting :-

Sort vehicles as per chassis (asc order).

Where should you do sorting?

front end.

Test Integer Sort

main() {

 ArrayList<Integer> data;

Java does → collections

to work with

Void sort(List<T> list);

**

list is compulsory.

⊗

sys("orig list" + list);

collections.sort(list);

sys(+list);

Code over

⊗

Sorting Ex2

for

populate sample Data();

limitation of collectn. sort

catch (Exception)

Compile time error

for Vehicle v: Vehicles
 sys(v); → ordered as per insertion.

Collections.sort(Vehicles);

⊗ due to generics

Natural ordering (Tims Sorting algo)

Sorting criteria will be within
var D types e.g. Vehicle

custom ordering (Tims Sorting algo)
sorting criteria kept externally.

⊗ Steps

① involves around Comparable<T>

e.g. public class BankAccount

 impl: Comparable<T> { }

Q) Implement public int

CompareTo(T o)

define

- ve - if this < o sorting dir.
- +ve - if this > o — — —
- 0 → || = o — — —

→ Q soln of e. java c-

To support sorting AD

Insert Comparable interface.

public class Vehicle implements Comparable<Vehicle> {

Implement public int CompareTo(T o)

→ Q @override

Sorting Criteria

public int compareTo(Vehicle anotherVehicle) {

wrong
because
we cannot
compare reference
this way

System.out.println("in compare To")

Return

if (chassisNo < anotherVehicle.chassisNo)

Return -1;

if (chassisNo.compareTo(

String is Comparable)

return chassisNo.compareTo(anotherV.chassisNo);

Lexicographic Compare

A B C → abc → asd → dac → nad

natural order.

will auto invoke T's

CompareTo 'Method as per
Criteria'

120 Sorting based on price

```
if (price < another.price)
    return -1;
if (price == another)
    return 0;
else
    return 1;
```

1st NOV 2021 Day 14

Collection utils / validation Rule. → Customer management assignment 12
log in.

public static Customer validateUser (String email, List<Customer> customers):

throws customerHandlingException

② findByEmail

```
Customer c = findByEmail (email, custs);
```

// Validate password

```
if (c.getPassword().equals (pass))
    return c;
```

```
throw new CustomerValidationException (
```

) throws customer

exception.

Display customer details with card location

Case 2: SOP (Enter card location)

```
String location = sc.nextLine();
```

```
for (Customer c : customers)
```

```
    if (c.getAdharCard().getLocation().equals (location))
```

SOP(c)

for loop used because
there are no structural modifications
or else we use iterator

getter for adhar object

8:23

D-14

1 NOV 2021

part 1

Adhar not linked

```
(c.getAdhar() != null);
```

Sorting → Revision

① Sorting in java type?

② Natural order / implicit / internal.

③ Custom ordering

Difference b/w natural & custom ?

⑧ We need to give sorting criteria..

Natural ordering → implement Comparable

Sorting criteria is written in UDT in compareTo() customers.

C.O.:

Why do we go for custom ordering when NO is available?

⑧ In UDT we can place only one sorting criteria.

e.g. ↗ 1 → sort by price
2 → sort by category ↗ multiple criteria not allowed

⑧ To add multiple use custom ordering.

- ↳ email (natural)
- ↳ Aadhar (C.O.)
- ↳ manufact (C.O.)

① C.O. → sorting criteria → explicit criteria is kept outside /

→ make separate class / Anonymous inner class. ↗ Boiler plate code

C.O. → revolve around → java.util.Comparator<type of list>

Sort Vehicles as per chassis no. ↗ generic type

① Vehicle class implements Comparable<Vehicle>

② ⇒ inheriting abstract class. ↗ only one method compareTo();

③ ⇒ two option → i.e. make class abstract or implement all methods

@Override

public int compareTo(Vehicle anotherVehicle) {

 // Sorting key --> email → using ref / use compareTo not equals

 0 → V another V
 -ve → V before another V
 +ve → V after another V

 X X X X

 return this.email.compareTo(anotherVehicle.chassisNo);

③ java.util.Collections → helper class containing static utility methods.

Difference b/w natural & custom ?

⑧ We need to give sorting criteria.

Natural ordering → implement Comparable

Sorting criteria is written in UDT in compareTo() customers.

C.O:-

Why do we go for custom ordering when NO is available?

⑨ In UDT we can place only one sorting criteria.

e.g. 1 → sort by price
2 → sort by category

3 multiple criteria not allow

⑩ To add multiple use customer ordering.

- ↳ email - (natural)
- ↳ Aadhar (C.O.)
- ↳ manufact (C.O.)

⑪ C.O → sorting criteria → explicit criteria is kept outside /

→ make separate class / Anonymous inner class. Boiler plate code

C.O → revolve around → java.util.Comparator<type of list>.

Sort Vehicles as per chassis NO. ↗ Generic type

① Vehicle class implements Comparable<Vehicle>

② ⇒ inheriting abstract class. ↗ only one method compareTo();

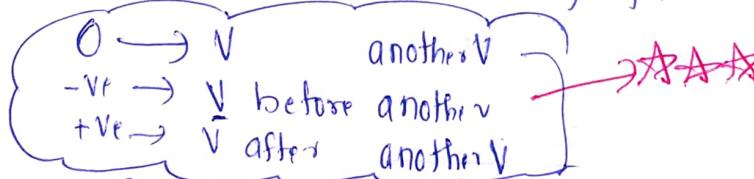
③ ⇒ two option → i.e. make class abstract or implement all methods

@Override

public int compareTo(Vehicle anotherVehicle) {



↗ sorting key → email → string ref. / use compareTo() not equals()



④ return this.email.compareTo(anotherVehicle.chassisNo);

⑤ Java.util.Collections → helper class containing static utility methods.

② public static void sort (List<T> list)

Only list implementation allowed.

e.g. Collections.sort(customers)

→ Time sorting algorithm

Internal implementation.

What will it call internally?

Invokes vehicles Natural ordering (N.O.)

based on 'Comparable' comparison function.

e.g. [v₁, v₂, v₃, v₄, v₅, ...]

↓
to compare JVM calls

v₁.compareTo(v₂) →

Sorting process

<→ v₁ before v₂ → don't swap

○ > v₁ after v₂ → swap the reference

Sort vehicles as per price.

Custom Sorting.

③ We cannot give multiple sorting criteria (i.e. chassis, price)

Solutn → Custom sorting.

④ Customer Ordering (we keep outside class) so as to avoid multiple changes

Steps in custom ordering

① Create separate class which implement Comparable<T>

② e.g. Public class AccountBalComparator implements Comparable<Acc Balana> {

14.1 (custom ordering)

14.1 (Custom exception) ordering

123

Showroom → Vehicle Testing class.

case 8: Display using criteria

9: Sort by using chassis no

10: Sort vehicle by price

⑧ check if Comparable implemented

⑨ we cannot add multiple sorting criteria.

(Case 9): sys0l sort vehicle by chassis No:

Collections.sort(Vehicles);

Display all:

for (Vehicle v : Vehicles)

sys0(v);

break.

Do not return any type of array

Natural ordering. ascending.

Case 10 Sorting by descending.

Custom ordering

ClassName: VehiclePriceDescComparator

Custom ordering. (DESC)

not comparable.

Vehicle

public class VehiclePriceDescComparator implements Comparator<Vehicle>

⑧ We inherited abstract

⑨ Implement compare(T O₁, T O₂)

Vehicle → Vehicle.

Why Compare (Vehicle V₁, Vehicle V₂) two argument

⑧ outside the class.

⑩ @Override

public int compare(Vehicle V₁, Vehicle V₂) {

 sys0("in compare : price")

 if (V₁.getPrice() > V₂.getPrice())

 Return +1

 if (V₁.getPrice() == V₂.getPrice())

 return 0;

 return -1;

In primitive so

We can use >, <, =

V₁ = 100

V₂ = 200

Q) Comparable is implemented in Double use

124 compareTo (Double anotherDouble).

④ double → primitive → Double (wrapping class) – is Comparable.

⑤ implement Double → Comparable.

V1.getPrice().compareTo()

directly not possible
without unboxing
explicit boxing required

Soln =)

V1.get
((Double)V1.getPrice()).compareTo(V2.getPrice())

Gives ASC

Autoboxing is done.

Desc ordering

Cast to step 3 → Custom implementation

Comparator Instance

Can we create instance of interface?

No, they cannot be instantiated, but interface can refer to implemented class.

public static void sort (list, Comparator<T>c)

list → AL, LL, Vector

Instance of class
where comparator is
implemented.

c → Comparator ref → implementing class.

Case 3:- Collections.sort (Vehicle, new Vehicle.PriceDescComp())

JVM calls new Vehicle.PriceDescComp()

Compare as comparison function.

class call to custom
sorting

If we use `(list, comparator)` & `comparator = null`
→ will go by N.O
~~e.g.~~ `Sort(vehicle, null);`

1:49

125

Enum (ordinal based sorting) (9:50) Doubt.

① is Comparable

② `Collections.sort(customers, new CustPlanComparator);`

Compound Criteria (Date & price)

C.O

① Create another class `VehicleDatePriceComparator` >;

② `Compare: Date is different → return`
if date is same → compare by price.

③ `Collections.sort(list, new VehicleDatePriceComparator);`

`VehicleDatePriceComparator()` implements `Comparator<Vehicle>`
@override

`public int compare(Vehicle v1, Vehicle v2)`

{
 System.out.println("in compare Date & time");

// date base comparison → Date is comparable
 int ref = v1.getManufactureDate();
 hence use Comparison function.

`CompareTo(v2.getManufactureDate());`

return ref;

only Date sorted

if (ref == 0) { same Date - compare by price
 return (Double)v1.getPrice() - v2.getPrice();

return ref;

Descending
replace v1 by v2

Ques ⑪ Date collections. sort(Vehicle, new VehicleDatePriority());

Sorting using enum (we won't many classes for every comparison)

① Create another class implements comparator <Vehicle>.

Anonymous inner class

② Instead of writing separate class implementing a comparator or writing many classes can we embed it in inner class.

Yes

→ non static nested anonymous class (nameless class); where entire class def embedded Java will give it name within single Java Statement

Outer \$1

\$2

| Default constructor

Step 1 → gone to anonymous.

Step 2 → Collections.sort(Vehicles, new Comparator<Vehicle>())

| Interface

| instance of class impl

Comparator &

{ Inner class begin

@Override
public int compare(Vehicle v1, Vehicle v2) {

 return v1.getCategory().compareTo(v2.getCategory());

}

① Petrol ② Diesel ③ EV

Sort by Ordinal no

Anonymous class Impl (enum type):

case 12 : Collections.sort(Vehicles, new ~~Vehicle~~ Comparator<Vehicle>())

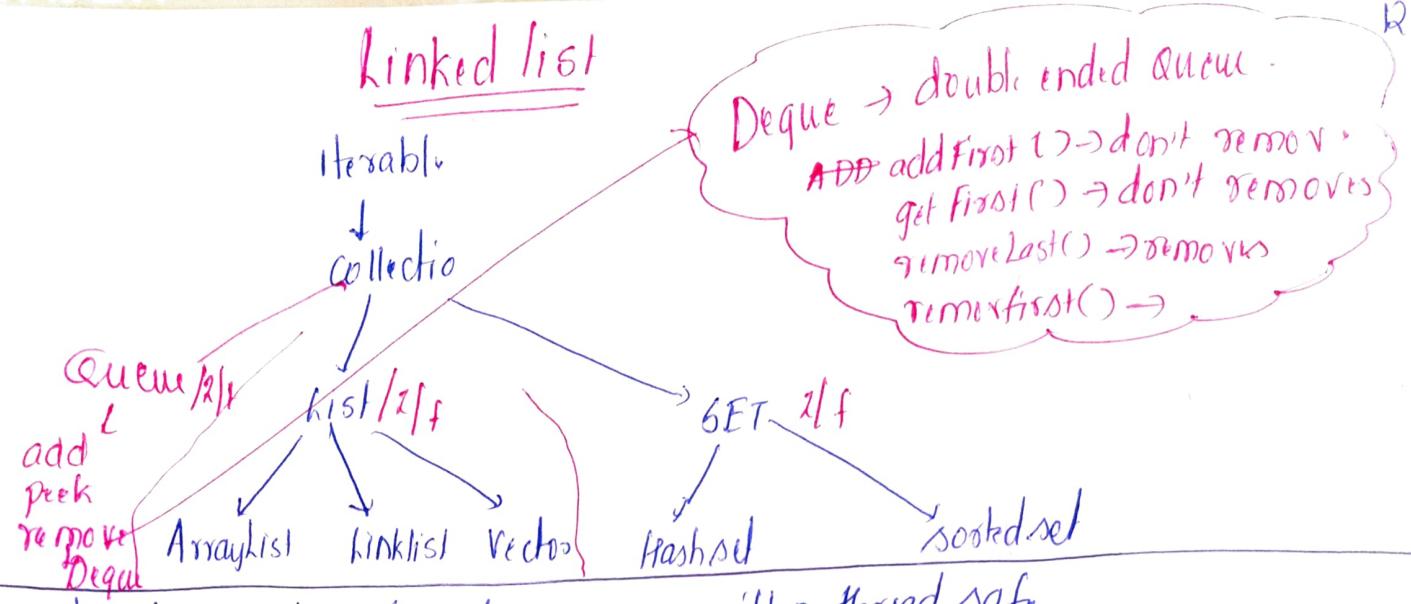
@Override

public int compare(V1, V2) {

 System.out.println("Anonymous category")

 return (V1.getCategory().compareTo(V2.getCategory()));

}



Vector: legacy class (old class) were written thread safe.
array is added later (thread unsafe)
use internally lock

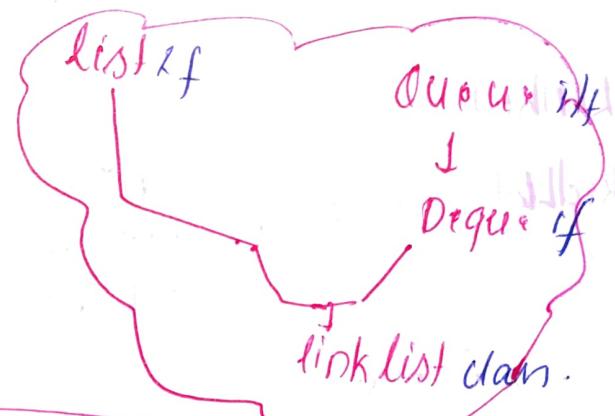
⑧ ArrayList will perform better
limitation of AL

⑧ many insertion & deletion makes it time consuming.

⑧

Queue linked list vs AL

- ⑧ AL is not queue or deque.
- ⑧ link list is list + Deque
- ⑧ LL is newly added
- ⑧ thread unsafe
- ⑧ performance & unsafe



Java docs: LL → iterable, collection, list, Queue, deque.

Deque → Summary table.

Insert → addFirst(E e)

↳ if space is not sufficient → illegal state exceptn.

Offer first(E e)

⑧ Returns true if the element was added to this deque? Ret: false

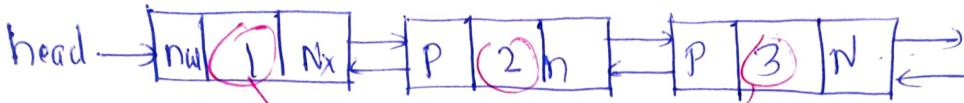
- ✘ removeLast()
 ✘ throws exception if not present
NoSuchElementException

posILast()
 returns false X
 returns → null value

④ No Random access in LL

- ✘ LL class uses Doubly Linked list

✘ Data As Reference



more memory for LL than array list

- ✘ Data means it is a reference type

linked list infers?

Iterable → collect → list
 collect → our → Deque → LL.

To increase 19.2 package → linked list

Linked List → Construct list, if empty linked list

LinkedList<String> l = new LinkedList<String>(); no capacity factor

Java docs → AddAll (collection<T>, T...).

④ Safe Varargs → type safe to variable arguments

T... → zero element, comma separated,

④ boolean addAll(collection<T> T, ...)

④ collections.addAll(l1, "one", "two", "three")

sys0(l1)

12:15

129

Iterator <String> itr = l1.descendingIterator();

while(hasNext()) {

 sys0(itr.next());

 l1.add(0, "hundred"); } Index based insertion.
Without shifting.

pollFirst():- don't throw exception.

pollLast gone from list

addFirst → void returning, l1.offer("1"), l1.offerLast()

l1.clear() → removes all element from list (size=0);

pollLast() → returns null value;

Interview

Q) ArrayL vs linked list Riadm Day 14

Search → array L Difference

Similarity

Insertn/Deletn → LL List implementn

Memory → AL fail-fastness

point

Interview

Q) When to use L & AL?

Generics: Generics

Generic classes → ~~AL, LL~~

" " ~~Intef~~ → ~~Collectn, iterablr~~

Generic Method: - a method which has its own type parameter

It can exist in a non-generic class → yes.

e.g. class collections → generic →

↳ 9 Arrays class Java does.

`asList(T... a);` → Generic

Returns a fixed size list backed by specified array.

Syntax :- type declaration → sits b/w method modifier → return type.

`asList(T... a)`

→ no argument ✓

→ T₁ T₂ T₃ → ✓

>Returns Varargs to list.
gives fixed size list

Code → ^{14.2} Generics

main() {

 // Java.util.array

 // Public static <T> List<T> asList(T... args)

 // Create fixed size list of integers containing 10,20,30,40,50;

 List intlist = Arrays.asList(10, 20, 30, 40, 50)

Keep at Interface

→ primitive

→ autoboxing.

`sys0(intlist)`

`sys0(intlist.get(intlist.size() - 1));`

`sys0(intlist.set(1, 12345));`

`sys0(intlist);`

→ replace 20 by 12345
and returns 20

`intlist(9999)`

error no increment

`intlist.remove()`

in size

→ no decrement

Can we use `ArrayList` to create a growable linked list → yes.

131

linked list constructor

`LinkedList()`;

`LinkedList(Collection)` .

Code

`LinkedList<String> Stringlist = new LinkedList();`

`List<String> list = ArrayList.asList("1", "2", "3", "4", "5");`
→ fixed size.

~~size~~

`Stringlist.add("dfgh");`

`list.add("asdfg")`

→ gives error
unsupported exception

`Emp ← mgr ← salesmgr.`
package generates.

`public class Emp {`

`public void computSalary() {`

`}`

→ Emp.java

`class Worker → Emp`

`@override`

`computSalary() {`

`}`

`class TempWorker → Worker {`

`@override`

`computSalary() {`

`}`

(No. 2)

`Emp e = new Mgr();` upcast

`e.computSalary();` → Mgr sal.

`E = new TempWorker();` → E.ComputSalary → Worker

Q2
ArrayList<Emp> l1 = new ArrayList<>();
ArrayList<mgr> l2 = new " " " RHS deduced as Emp
l1 = l2; javac error. RHS due as mgr

Object o = new mgr() up casting.

ArrayList<Object> l3 = new AL();
l3 = l2 } Java
l3 = l1 } =

Generics → runtime polymorphism does not work

Generics
main() { → Test 1.java
 Emp e = new mgr();
 o = " " TempWork();
 ArrayList<Emp> l1 = new AL();
 emps

 <mgr> mgrs = new AL(); → Generics don't work
 emps = mgrs

AL<?> objs = new AL<>();

 objs = emps, → both does not work.
 objs = mgrs

④ Inheritance :- does not work similar way. *

④ Soln → Wild Card Syntax *

? → if in wild card in Generic → place holder
 can be replace by any type.

Ans → RHS → object type
AL<?> l1 = new AL<>();

AL <Mgr> l₂ = new AL<>();

↓ from previous.

Inheritance in Generics

133

l₁ = l₂



? = unbounded wildcard

AL <?> emps = new AL<>(); → AL of object

AL <Mgr> mgrs = new AL<>();

AL <?> objs = new AL<>();

objs = emps;

objs = mgrs;

fixed → Growable
vector

To tell compiler it is a list put ?

class Collections → shuffle → p. s. v shuffle (List<?> list)

→ class Test3 {

main() {

Vector<Integer> v₁ = new Vector<>(Array.asList(1, 2, 3, 5, 6, 7, 8, 9, 10));

v₁.add(0, 999);

System.out.println(v₁);

fixed

Generic

Create nonempty Growable vector by

Copying the inter-type of ref from fixed size

No error

shuffle thr. ref from vector.

shuffle

List of Vehicle

Collections.shuffle(v₁);

System.out.println(v₁);

Shuffled values

EMP
fruit
Date

Customer management system assignment

public static Customer findByEmail (String email, ArrayList<Customer> list){

wrap email to customer

Customer c = new Customer(email);

int index = list.indexOf(c);

34

```
if (index == -1)
```

```
    throw new CustomerException();
```

```
return list2.get(index)
```

Public static Customer authenticateCustomer(String email, String password,
throws CustomerHandlingException {
 array list)

```
Customer cust = findByEmail(email, list);
```

```
If (cust.getPassword().equals(password))
```

```
    return cust;
```

```
throw new CustomerHandlingException("invalid Pass")
```

change password?

```
else if (change Password)
```

```
Customer validCustomer = authenticateCustomer(sc.next(), sc.next(), cust);
```

Display customers with plan

```
for (Customer c : customers)
```

```
    c.getPlan == customer.plan;
```

Display customers with address location.

2/11/2021 manager is object but arraylist of managers is not

array list of object

Generics Inheritance is Different

We use ? (wildcard).

Q) What does ? represent?

It is wild card in generic syntax. can be replaced by Any type.

Q) What will happen in ? place?

```
AL<?>l1 = new AL<?>();
```

```
AL<mgr>l2 = new AL<?>();
```

```
l1 = l2 ✓
```

Q) ? = unbounded wild card ?

what happens if we replace ℓ_1 by T ?

135

$\text{AL} < T > \ell_1 = \text{new AL} < > (s);$ || Java Error.

$\text{AL} < \text{Mg} > \ell_2 = \text{new AL} < > ();$

$\ell_1 = \ell_2$

method.

$T \rightarrow$ we cannot use it, it is used in generic interface/class.

14.2 Test 2 8:26

① T vs $?$

$?$ → represent wildcard, can be replaced by anything.

$T \rightarrow$ specific type

Inheritance $\rightarrow ?$

? in shuffle

shuffle

unknown type, integer, double, Fruit

public static void shuffle(List<?> list)

Vehicle,

only list allowed

② Create a fixed size list of integers & shuffle?

Collections.shuffle(Arrays.asList(1, 2, 3, 4, 5));
~~sep(list) → shuffled list.~~

A LIST?

Ordered fixed size
void returning.

reverse

Java error

P.S.V reverse(List<?> list);

~~LIST<Double> list = Array.asList(1, 2, 3, 4, 56);~~

~~AL<Integer> list = new ArrayList();~~
~~list = Arrays.asList(1, 2, 3, 4, 5);~~ 1d, 2d, 3d

Collections.reverse(list);

reversed o/p.

Int * * * *

① Autoboxing → int → Integer.
Integer cannot convert
Double.

We can apply reverse to fruit, vehicle, ?

Yes

Any Any

② Write a static method in GenericsUtils class? to print elements of
any list (AL, LL, Vector) of Any type (Fruit, vehicle, customer)?

class GenericsUtil {

ANY(AL, LL, VL)

pub. static void printListElement(List<?> list)

* * * Any Any

(Fruit/vehicle/cust)

P. S. V. ~~first invoke to compute salary (List<Emp> emps)~~

for (Emp e : emps) {
 e.computeSalary();

↳ add a static method which can accept any list of any type
↳ emp(Emp/ngr/s) & compute salary
↳ add a static method which can accept any list of any type
↳ Emp.java

↳ unbounded wild card

ArrayList: In certain cases we can replace T → ? and some cases

↳ generic interface

: (.., ii + 0) or ho
for (T o : list)
 change name
 u/p

↳ public static <T> void printList(List<T> list);
↳ Ambiguity
↳ printList(List<Object> list);
↳ Both can work if I use
↳ convert to Object

All preferences correctly

: ("P", "C", "B", "D") == ("a", "b", "c", "d")
↳ list <String> list2 = Alwaysолоси(3,4,5,6);
↳ list <Integer> list2 = Alwaysолоси(1,2,3,4,5,6);

printList(Element1);
↳ double <Double> list1 = Alwaysолоси(1.1, 2.2, 3.3, 4.4, 5.5);

main() {
 list

X → object

↳ equivalent operation

↳ will be applied by object

: (ii + 0) or
for (Object o : list)

(for (Object o : list)) or

X

320

Above is Non Generic method.

137

Test 3 → main() {

 Create fixed size emp list.

 List<Emp> list1 = Arrays.asList(new Emp(), --, --, --);
 InvokeComputeSalary();

// mgr is in Emp.

 Create fixed size mgr list

 List<SalesMgr> list2 = Arrays.asList(new SalesMgr(), --, --);
 InvokeComputeSalary(); → Javac error

☆☆☆☆

? Extends E

? Super E

Improved version

P S V invokeComputeSalary(List<?> emp).

? Extends E
? Super E

(Calling Bounded?)

? → wild card is generic syntax.

? Extends E → represent upper bound → any type E or its subtype

? Super E → represent lower bound → E or its supertype

* We use upper bound

(AL/LL/VI)

P S V invokeComputeSalary(List(? extends Emp))

{
 for(Emp e : emps)

 e.computeSalary();

Mgr/Sales/Work.

T with Upper or lower bound

public static <T extends Emp> void invokeComputeSalary2(List<T> emps){
 for(Emp e : list)
 computeSalary(e);

138

ArrayList Constructor 3rd hyp.

AL (collection) < ? Extends Emp > C)

A A

Emp

→ It creates populated "AL" of Emp type, from AL / LI / VL / HS
 LinkHashSet / TM (map). any collection implementing class.
 , from Emp or any subtype A A

Q will you be able to populate AL of Mgr from AL < EMP >

④ AL < workers > ✗ ✓

⑤ AL < sales Mgrs > ✗ ✗

AL < Fruit > ✗

AG < object > ✗

LHS < noHR Mgrs > ✓

⑥ Collection < ? Extend Mgrs > C,

is salesmgr in mgr ✓
 is fruit in mgr ✗

Copy (→ collections)

↳ ? Extends Fruit
 → List of fruit or its subtype

List < ? super mango
 → fruit, mango or object is allowed

SET (No Duplicates)

Iterable<T> 2/f

Collection<E> 2/f

SET<E> 2/f

HashSET<E> class

→ unordered & unsorted (not possible to sort)
 Based upon hashing algo

- hashCode, equals

(U U)

Linked HashSET<E>

Ordered But Unsorted
 uses singly linked list
 Doubly.

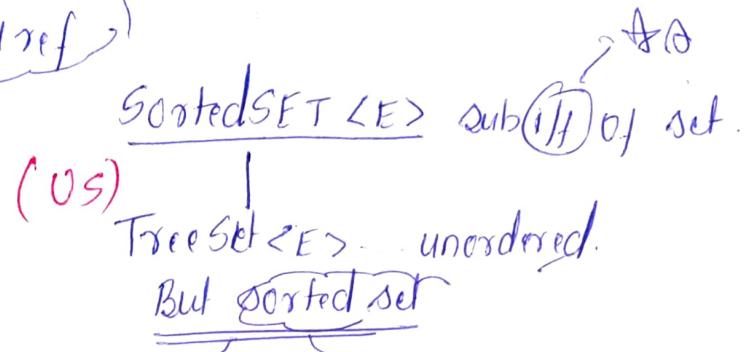
(O O)

Sorted SET

- ⑧ SET → no index based
 - can be traversed by for each, iterator
 - for not allowed
- ⑧ Allows single null ref

☆☆☆
☆☆☆

remember all feature



natural ordering

custom ordering
Comparator compare

① Comparable (comparator)

⑧ only list supports indexes

⑧ set don't implement any extra method

⑧ only single null reference ✓

⑧ fastest set → HashSet: (b/o no ordering, no sorting);

TreeSet (collection <? Extends E >) c)
ArrayList

↓
Emp
↓
m, v, h

calls comparator
JVM

TreeSET (comparator <? super E > comparator)

empty TreeSet, custom ordering, Emp,

function

IntegerSet: HashSet Constructor

main() {

// create empty HashSet to store integer type ref.

HashSet<Integer> hs = new HashSet<>();

Q4

`int data = {10, 12, 10, 23, 1, 45, -78, 10};`

populate hashset from int array();

~~b4o~~ `for(int i : data)`

`hs.add(i);` //auto boxing int → Integer
//display hs.

~~3~~ `System.out.println(hs);`

create populated hs from fixed integer list.

`List<Integer> list = Arrays.asList(10, 20, 10, 20, 30, 4, 5, 6, 7, 100);`
// create populated hs

~~Hashset<Integer> hs2 = new HashSet<>(list);~~ → changed to growable
Constructor `HashSet(Collection<? Extends E> collect)`.

// Can we attach iterator to hs?

`Iterator<Integer> itr = hs2.iterator();`
`while(itr.hasNext()) {`
~~System.out.println(itr.next());~~

As once iterator is declared we can not allowed to structural modification.

15.2 Emp →

~~class Emp {~~

~~private int id;~~

~~→ name;~~

~~salary;~~

~~Generators:~~

~~Constructors:~~

{

`Main() {`

`HashSet<Emp> emps = new HashSet();`

`Emp e1 = new Emp("P01", "abc", 1000);`

$$e_2 = (101, abc, 1000)$$

$$e_3 = (101, \underbrace{abc}_6, 1000)$$

emps. add (e₁) T T
(e₂) F T
(e₃) F T ✓

sysof

All duplicates are added

Hashset is not behaving properly

because equals not overridden

Override equals

public boolean equals(Object o)
 {
 System.out.println("equal")
 }

Sysolinqual

if(o instanceof Emp)

return this.id == ((Emp)o).id; again.
return false; keeping duplicate.
} even not calling equal

return false;

again.

keeping

keeping duplicates.
en not calling equal

even half calling equals

HashSet → is a database based on hashing.

flashing → encryption → = Converting any Java object into Integer.

hashcode()

return int(hashcode)

* object to int conversion

* only supported for hash tables

☆☆☆☆☆☆

Q3 Is there any contract on hashCode()?

- ④ If two objects are equals (object) method, then calling the hashCode method

1

$\star \Delta$ gives distinct values.

10

142 Contract not followed (15.02 Test Emp Oct 2) 11.47
→ same object produce same hash code Day 15
21/11/2021

@Override
hash Code



shortcut → @ equals & @ hash code

If overrides the inherited hashCode method, to follow the contract b/w the hashCode and equals.

If same object must produce same hashCode to achieve this contract we override hashCode

```
public int hashCode() {  
    System.out.println("in hashCode");  
    return 10;  
}
```

```
public boolean equals(Object o) {
```

gives same hashCode for same object

- ① goes to hashCode → always called
- ② if not found enter to equals → if true → Duplicate element.
- ③ if

Which methods do JRM calls for HashSet?
equals & hashCode.

Test Different object may produce different hash
same object must produce same hash

Contract → assign same hashCode to same type of object

for correct working HS?

implement equals & hashCode

Contract exist → same object must produce same hashCode

Q) Why people come to hashCode if we are comfortable in List? 143

* If List \rightarrow 1000 \rightarrow 1000 searches.
 $100000 \rightarrow 100000$ searches.

It is O(n) factor

Q) Constant time performance. not guaranteed

optional part of contract

↑ speed

Q) It is not required

different object \Rightarrow calling hashCode on each object is
 $o_1, o_2 \rightarrow$ producing different int

Contract b/w equals & hashSet

`ref1.equals(ref2)`

true. `ref1.hashCode()` must
be same as `ref2.hashCode()`

mandatory

false \rightarrow optional But recommended
ref hashCode diff from obj.hashCode
improve performance of hashing base

Flow :-
① Identify pk and override equals?
② Use the same data members in overriding hashCode.

Code :- override equals

(id)

True \rightarrow Yes

false \rightarrow distinct hashCode

↑ performance

public int hashCode()

{ System.out.println("in hashCode"); }

return id;

{ Optional Contract }

Constant time performance

144 equals \rightarrow id hashcode = 10 \rightarrow horrible hash function

Emp e₁ = (101, abc, 1000)

e₂ = (10 →, →)

(50 →, →)

(50 ←, →)

System (emp.add())

())

())

())

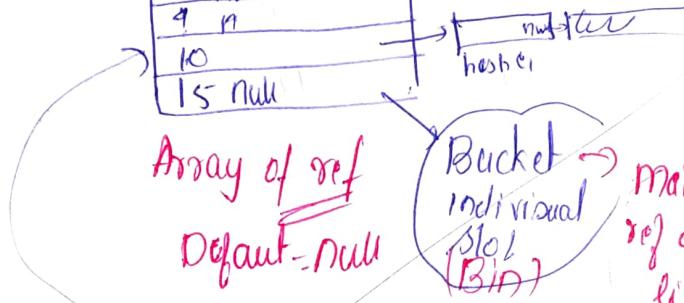
Bucket

initial capacity 16 load factor = 0.75

Heap \rightarrow Hash tabl.

HashSET

0 null
1 null
2 null
3 null
4 null
5 null
6 null
7 null
8 null
9 null
10 null
11 null
12 null
13 null
14 null
15 null



JVM calls

Hashing Algo

followed HashTable

① Hashcode (JVM)

e₁.hashcode

② JVM checks if

bucket is empty or not

empty

null check

④ no further

call to equal

add()

b True

non-empty

(not empty)

emp.add(e₁);

① e₁.hashcode \rightarrow used to identify tabl.

② JVM invoke hashcode function (JVM)

Bucket id = e₁.hashcode() % capacity;

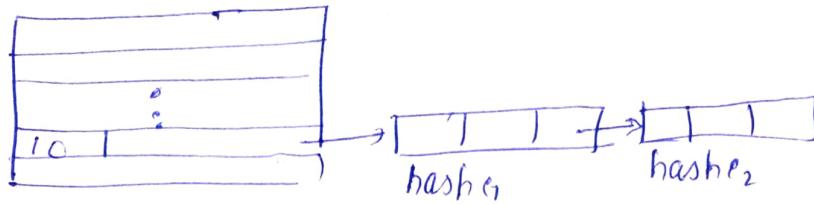
provided

$$10 \% 16 = 10$$

③ Buck id = 0 to capacity - 1

* Hashing algorithm for non-empty bucket

145



hash collision

- Q emps. $e_2 \rightarrow$ ① if will call $e_2.\text{hashCode} \rightarrow$ ② non-empty
 $e_2.\text{hashCode} \% \text{capacity}$ ③ check by equality (equals)
10 \% 10 = 10 ④ $e_1.\text{equals}(e_2)$

True
duplicate
return false

False
Non-duplicate
add()
return true

- Q $\text{emps.add}(e_3)(50, \text{abc}, 100) \rightarrow$ non emp

- ① b $e_3.\text{hashCode} \% \text{capacity}$ ② equal
= 10 ③ equal
④ add(e_3).

Load factor

- Q When Hash set getting filled (0.75) \rightarrow 75% full then jvm will rehash complete table \rightarrow

reha \rightarrow size of hash table * total ref exceed current load factor

Rehashing \rightarrow creates appx 2 times bucket (2^{powers})

and jvm invokes rehashing algorithm, to copy ref from old hash table and delete that old hash table.

Internal working of hash set

Q and hence jvm tries to achieve const time performance.

146 Hash Map

99% map used
in web

Never →

$\text{Map} < \text{K}, \text{V} \rangle$ Interface

- ⊗ key value → double generic type
- ⊗ nested interface → entry ($\text{Map}, \text{Entry} < \text{K}, \text{V} \rangle$)

⊗ No Duplicate

⊗ key → distinct value can be duplicate

⊗

HashMap

→ unsorted & unordered
maps

→ Based on hashing algo

→ Ensures constant ($O(1)$).

for -

put, putIfAbsent, get, remove, count
key, size thread unsafe

LinkedHashMap

sorted

⊗ To get Populated map

→ upper bound
key or subtyp: Bank account
saving ✓

HashMap ($\text{Map} < ? \text{ extends } \text{K}, ? \text{ extends } \text{V} \rangle$ m.

↳ HS / LHS / TS → error → V or subtyp.
↳ HM / LHM / TM ✓

⊗ Create new empty map to store account details ↗

HashMap< Integer, BankAccount > hm = new HM<>();
key ↴ return type ↴

size = 0 cap = 16, L.F = 0.75

⊗ Create new account ↗

A PZ Hashmap ↗

put(k key, V value)

**

147

- ↳ already exist → replace old value reference by new
- ↳ if new → inserts

⑧ keys is primary (equals on key).

⑨ new project → maps

TestHashMap() main () {

 HashMap<Integer, String> hm = new Hm<>();

 System.out.println("added".+hm.put(1001, "abc"));

(1001) :
 (1002) :
 (1003) :
 (1004) :
 (1005) :

System.out.println("hm: "+hm);

System.out.println(hm.size());

Deduce from left

if RHS Diamond is
not present

Max from
any list

⑩ object j is not a fruit Java c

⑪ ?super melon ← fruit ✓

?super men ← new linked list

⑫ if we don't write in RHS it is
deduced from LHS

⑬ melon ← watermelon Java c
(?super melon)

AddAll (→list) APZ

AddAll(Collection<? extends T> c)

All v2

↑↑

melons.addAll(wmelons) → No error

fruit.AddAll(melons)

{cannot on
Reference}

Generic Question

Q Write a Method to find
a max number from Any
list of any numbers (Z/F/Double)
n returning it to the caller?
Code:-

package utils;

public class GenericUtils {

solve using ?

bound wild card

① ? cannot
be in return
findMax type

public static <T> T

return null.

2 Tmax = list.get(0);

for (int i=1; list.size(); i++)

 if (list.get(i) > max)

Number

Imposing rules on

limitation of ?

In above case, ? is not valid.

SOLN

public < T extends Number & Comparable < T > > T findMax

(List < T > list)

```
{ Tmax = list.get(0);  
for (int i=1; i<list.size(); i++)  
if (list.get(i).compareTo(max)>0):  
    max = list.get(i);
```

return max

3.0

All Java class implement

Contract

Assignment on library

Book class → isbn(string), category(enum), price(double),

implement → hashCode equals contracts.

```
public class Book {  
    private String isbn;  
    private Category bookCategory;  
    private double price;  
    private Date publishDate;  
    Author name;  
    private int quantity;
```

@override toString; constructors,

Fullfill Contract Criteria

```
@equals public Boolean equals(Object obj) {  
    System.out.println(book eq)  
    if (!obj instanceof Book)  
        return isbn.equals(((Book) obj).isbn)  
    return false;
```

@Override hashCode

```
public int hashCode() {
    System.out.println("in book hashCode");
    return isbn.hashCode();
}
```

Tester :- main()

```
try (Scanner sc = new Scanner()) {
    // Create empty HashSet
    HashSet<Book> books = new HashSet();
    while (true) {
        System.out.print("Enter book detail:");
        System.out.println("Added " + books.add(sc.nextLine())));
    }
} catch (Exception e) {
    e.printStackTrace();
}
```

window → preferences → content → Java content areas → Advanced → scroll down.

Restore def.

* hashCode → if empty → add
everytime new hashCode → everytime empty

Aa BB
Ba CB
Ca DB
Da EB.
BB BB Aa BB.

Same hashCode
but diff string →

→ Aa → BB
→ Ba → CB
→

Lab → 2/11/2021

5:30

3/11/2021 Flashing →

huge
data



small indexes

To add the ref in Hash based data structures? contact 2
hashCode & equals.

Q) Where are hashCode and equals coming from?

① Object → equals → True → reference equal
False →

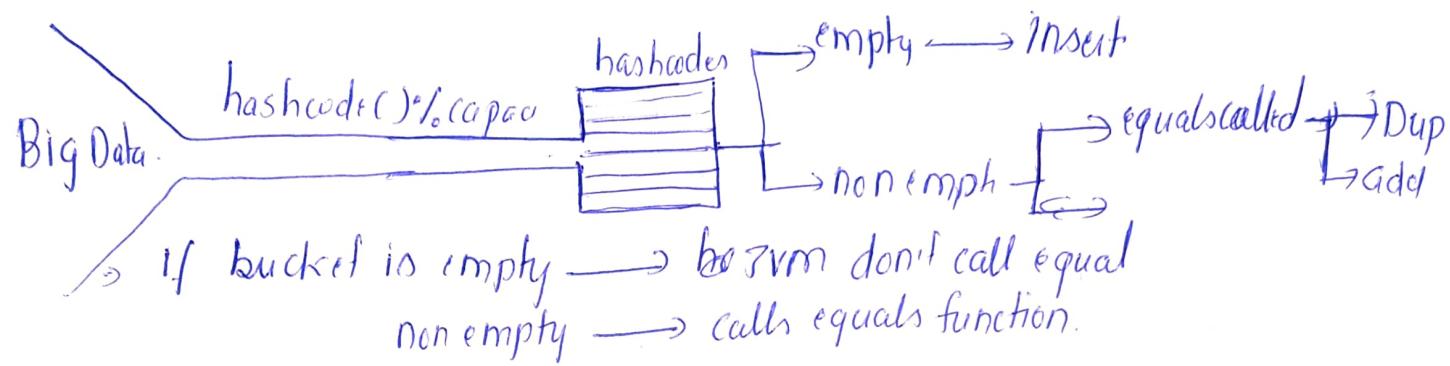
② Object → hashCode → address → internal representation of memory location

③ Contract

↳ mandatory → equals & hashCode → same equal object must produce same hashCode

↳ optional but recom →

④ If two reference books ref1.0.equals(ref) → True → same hashCode
False → different hashCode.



In case of Duplicate → put() → replace old value.

optional contract → unequal object should produce (for improvement of hashtable) distinct hashCode

How to implement Contract:

① Identify Pk and override equals

② and use same Pk field in generating hashCode.

Hint → use prime numbers in generation of hashCode.

Prime numbers in
Generation of hashCode.

e.g. → String, Wrapper → Integer, Double, Date, LocalDate...

↳ Any java class has already followed
Contract.

④ They have implemented both parts.

Map Overview

root if $\rightarrow \text{MAP} < k, v >$ it is called Entry.

JavaDocs $\rightarrow \text{MAP} \rightarrow k \rightarrow \text{distinct}$
 $v \rightarrow \text{may be duplicate}$

Map.Entry(k, v) \rightarrow outer.inner() \rightarrow Entry(key, value);

MAP API \rightarrow statistically nested 2/f

Map implement class \rightarrow hashmap, linkedHM, Trees, unordered.
 (vs VO) (vs O) (S, O)
 ordered ordered unorderd

Sorting \rightarrow NO \rightarrow Comparable
 CO \rightarrow Comparator

HashMap<key, value> hm = new HashMap<>();

(*) To get populated HM

To get populated

HashMap< Map< ? extends K, ? extends V > M

\rightarrow size=0, initCapacity=16, lf=0.75

BankAc \rightarrow saving
 \rightarrow current
 \rightarrow loan

BankAc \leftarrow Saving
 \leftarrow current
 \leftarrow loan

If we want populate BankAc can we do it using?

Yes:

Steps to c. Map

public V put(K, V);

Duplicate k \rightarrow Replace // returns old value reference.

Different k \rightarrow new entry.

HM<String, Customer> hm = new HM<>();

HashMap for Customer

duplicate email → replace old customer (Danger)

existist email →

Safer version

public void putIfAbsent(k, v):

exactly like put → if new entry, returns null.

existing entry → Won't replace → returns old customer

put(k, v)

New Entry
→ enter

existing
→ replace

putIfAbsent(k, v)

Won't replace
- return old customer

Simple book library

java.time.LocalDate

④ Const time performance & powerfull API

HashMap

Guide n searching by P.k.

④ HashMap<String, book>

④ & which methods are overridden? ★★☆☆

hashcode &

→ ~~no~~ not required because key is String.

④ prepopulated Map. data(Put)

⑧ Java 8 → Java.time.LocalDate

Local Date → it holds only date & is it is immutability.
Inherently thread safe.

⑨ API = LocalDate.now(); (2007-12-03)
(yyyy-mm-dd)

(SDF not required)

⑩ Parse is inbuilt.

Parse → String to Date

No except checked Exception

for Month > 12 → No exception.

Quick fix.

of() → int year, int month, int day.

Book-class

```
private String isbn;
private Category category;
private double price;
private LocalDate publishDate;
private String authorsName;
private int quantity;
```

→ Constructor

→ to string

→ No need to implement hashCode because isbn → string.
and inbuilt contract

7
(chr1 + 1)
enum

FICTION, TECHNOLOGY, YOGA
ROMANCE,

Collection utils: - populate maps

java.util → Collection utils

P.S.V HashMap<String, Book> populate Sample.Date()

{ // Create empty map

HashMap<String, Book> map = new HashMap<>();

String "Added" + map.put("ISBN", new Book());

Size = 9 L.F = 0.75

Tester:

```
main() {
    try(Scanner sc) {
        HashMap<String, Book> books = populateB();
        Myso(books);
        boolean exit = false;
        while (!exit) {
            sys0(add a book); 2. display ... 100. Exit)
    }
}
```

Skeleton Code

putIfAbsent

(ctrl+i) → import local variable

(ctrl+f) → find & replace

exception handling

Scanning

→ I/P → Book ISBN
does not exist → remaining book details
exist → prompt for quant
new book enter

putAll()

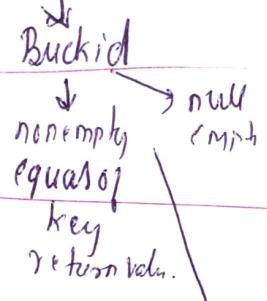
e.g. map1.putAll(map2);

public ~~Object~~ get(Object key) → O(1)
Value

Return → Value
not exist → null

How O(1)

map.get(k₂) → Value → hashcode() → k₂.hashcode()
map.get(k₁₀) → null



boolean containsKey(Object key) → O(1).

Return → True if contains; else false.

boolean containsValue(Object value) → O(n)

* if sys0(Enter Bbn);
String isbn = sc.next();

If (books.containsKey(isbn)); {

sys0("Enter quantity");

Books.get(isbn).setQuantity(sc.nextInt());

Book b = books.get(isbn);

b.setQuantity(b.getQuantity(1) + sc.nextInt());

X user
less

3 else {

System.out.println("Enter other details: cat, price, ...");

Book b = new Book(isbn, valueOf(sc.nextInt()), toUpper()), sc.nextIntDouble();
 parse(sc.nextInt(), sc.nextInt());
 books.putIfAbsent(b, books.get(b));
 books.put(b);

We already checked contains
 Instead of using (contains + get) → get → found → Book, not found → null

```
if (get != null);
  Book b = books.get(isbn);
  if (b != null) {
    System.out.println("Enter quantity");
    b.setQuantity(b.getQuantity() + sc.nextInt());
```

return break

case 2 → DisplayAll() (iterator used for LLAD)

We cannot use for, → it's not index based

~~for each~~ → no iterator

Limitation of Map: ~~for each~~

- ① Since map doesn't extend collection no iterator allowed
- ② Searching & sorting & removing as per key based criteria
- ③ for searching/sorting/removing as per value based criteria involves Converting map
- ④ Converting → Map → Collection view

~~How to overcome?~~
 Convert map → collection view.

- ① → get key-value pairs (Entry) of ref from Map:
 Map & f →

(11:08)

How to extract key types ref from a map?
 Set<k> keySet

Q How to get values type of reference from map?
↳ public Collection<V> values();

e.g HM< Integer, Accounts >

⊗ // can you iterate over the map directly // No.

SOLN → Convert map → Collection<V> (public collection)

for (Book b : books) → java → its hash map. $\langle V \rangle$ Value

SOLN → for (Book b : books.values())

Map → Collection<V> sys0(b); To iterate over HM.

Issue of book

* If ISBN → if available → if 0 → no book
if ! 0 → reduce the quant.

Custom exception:

{ BookHandlingException.

⊗ Case 3 sys0(Enter ISBN):

ISBN = sc.nextLine();

b = book.get(isbn);

If (b == null)

throw new BookHM.Exc("Book doesn't Exist")

If (b.quantity == 0) { currently unavailable. }

throw new Exc("Book unavailable");

If book availb.

b.setCount(b.getQuantity() - 1);

break;

Return a book.

2/ If ISBN found update the quantity suitably.

Case 4: If ret the book

sys0("Enter ISBN");

isbn = sc.nextLine();

b = books.get(isbn);

throw new Bookhandling ("invalid");

157

b.setQuant(b.getQuant(c)+1);

Remove old books :

Remove key based.

public Value forremove(Object k)

if found → remove complete entry.

if not found → returns null.

Case 5: remove book from library

sys0(Enter ISBN):

isbn = sc.next();

If (Book.remove

b = books.remove(isbn);

if (b == null) {

throw new BookhandlingException ("invalid ISBN");

sys0("Book removed");

Display book isbn for all books by specific author & category
(value based searching)

Conversion
to Collection

① Value based conversion is sufficient.

② Collected Iterable.

for (Book b : book.values())

if (b1.getCategory().equals(author) && b1.getCategory() == bookCategory)

sys0(b1.getisbn());

Display book isbn, qty n price for published after a Date.

sys0(Enter date)

LocalDate date = parse(sc.next());

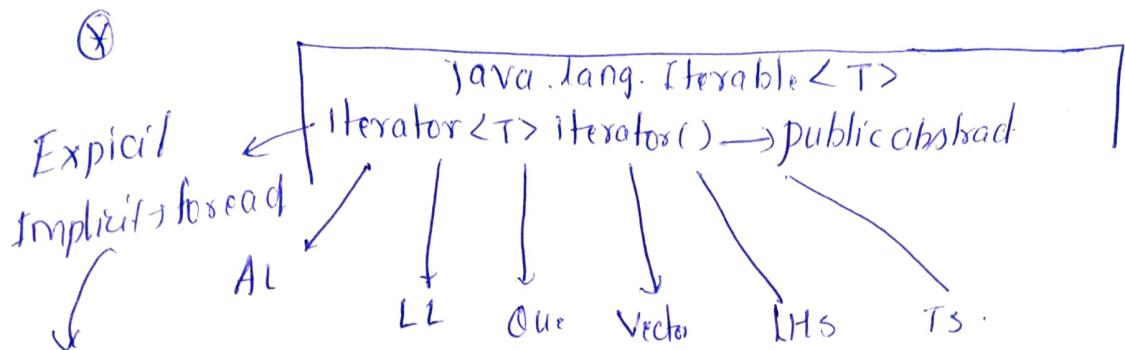
for (Book b1 : books)

b2.getPublindate().isAfter(date),

sys0(b1.getisbn() + " " + b1.getQuant() + " " + ());

15th Java 8 New features 12:14

⑧ Default → in interfaces
Default default method → it can appear in interface
 till Java 7 → we have public/abstract for backward compatibility
 Java 8 → allows non-abstract method.



* Java 8 wanted to introduce additional method void forEach,
 (Consumer<? super T> consumer)

9 Interface formulae

```

double calculate(double a);
default double sqrt(double a, double b) {
    return Math.sqrt(a+b);
}
  
```

public class myclass implements formulae {

3. @ ~~only~~ double calculate(double a);
 sysout("1")
 return (a*2)

16.3 Java 8 f.

⑧ it is optional for implementation class.

another formula class -

```

--- n ---
--- n ---
--- n ---
  
```

return (a/2);

It can implement class
override inherited def

159.

```

package P;
public interface A {
    void show()
    default double add(a, b) {
        return a + b;
    }
}

```

Interface

```

class C {
    @Override
    public void show() {
        System.out.println("in abstract impl.");
    }
}

```

class C

12:40 **Important**

package P;

```

public interface B {
    void show()
    default double add(a, b) {
        return a * a + b * b;
    }
}

```

Interface

Diamond Problem
ambiguity.

If Java forces imp class to override duplicate def method impl.

A. Super.add(3, 4);
B. -1;

soln. soln.

Q8/11/2021 Is it compulsory to override default method?

No can we extend it?

Yes
 → `@Override`
 default void show() {
 System.out.println("in c show");
 }
}

How to access A interface.

A.super.show();

How to add static methods in Interface

How to access interface.methodname → Implicit keyword → public.

17.1 P.3 → adding static method in interface

static void show() {

System.out.println("in static method of inf");

160 → Implementation class
class implements interface {

Default → can be overridden

@ public double calculate(double a) {
 // so (must implement abstract)

@ override
 @Override
 public double sqrt(double a, double b) {
 default
 }

@ Override

public static void show()

{
 // static method

Redeclare static

static void show() {
 // in static method
}

Java error.

Static methods can be
redeclared

④ If is compiler specific
early binding

Test math() {

 formula f1 = new formula();

 f1.calculate(f1.calculate(12));

 f1.show(); *error* How to access redeclared
 definition.

 formula.show();

 MyFormula.show();

Static method in interface → Default accessibility Public

Functional Interface:

161

⊗ An interface with single abstract method is called F.I.

e.g. Runnable, Comparable, Comparator, Iterable, Consumer, Predicate.

② Functional Interface → at interface level.

XX functional interface can be substituted by lambda expression.
X Method references, or constructor reference.

⊗ Which are valid functional interface?

public interface A { double calc(int a, int b); } → Yes SAM

public → B extends A {} ← Not extending A.

public → C extends A { void show(); } → X ↗ abstract meth.

public → D {} → X marker iff.

public → E extends A { default void show(); }

Static void test() {} ✓ SAM.

① abstract + Any no default
is allowed

Lambda Expression :

its derived from lambda calculus.

⊗ We cannot define function/method without class or object.

⊗ We cannot pass Argument → function /: X not possible,
return → behavior X in Java

Why λ expression

⊗ Objective →

① Create your own functional interface n use it in 'λ'

e.g. perform any arithmetic operatn on 2 double values & return result.

P-4 → Interface → Computable
↳ public interface Computable @ Functional Interface

// add Single Abstract Method

double compute(double d₁, double d₂);

⑧ Create to add static method

→ ~~class is~~
Interface is sufficient

→ Computational iff

⑨ static double performAnyOperation (double a, double b, } SAM

{

return operation.compute(a, b);

}

Computable
operation

P-4 → Interface → Computation Utils →

(15:41)

public interface Computation Utils {

// add a static method to set result

static double performAnyOperation { double a, double b, Computable,
} return operation.compute(a, b); } operator()

P4 → Test1 :- Main () {

System.out.println("addition " + performAnyOperation(10, 20, new Adder()));

P4 → class → adder → adder implements Computable {

{ return a+b;

⑧ subtractor class implements Computable.

How to avoid creating Many classes.

⑨ Anonymous inner class.

~~new~~ → Sys0 ("subtraction" + performAnyOperation(100, 20,
new Computable() {

```
    public double compute(D1, D2)
    {
        return d1 - d2;
    }
}
Multiplication of 2 numbers
    return d1 * d2;
}
```

flow many class files → 2 anonymous class.

⑩ always we need to create object or inner anonymous

what is λ exp?

⑪ it is concise, anonymous function.

(argument → (body))

Multiply '2' no using 'λ exps'.

Sys0 ("Product" + performAnyOperation(100, 20, (double c, double d) →
syntax (method args → body)
Test 1 → P4 →

Args }
body } { return (c * d); } 3

More Shortcut

Sys0 (, , , (c, d) → Return c * d;)
(c, d) → c * d

⑧

Arrow function →

Division

`sys("Division") performAnyOperation(20, 80, (a, b) → (a/b));`

functional Arg



primitive

behavior

Higher order Function/methods.

⑧ Functions who can accept/return functional argument

iterable → `ForEach(Consumer<? super T> action)`

e.g. `for (T t : this) { } F2n/f`

3

Java docs → `java.util.function`.

Consumer -

All Functional interface can be replaced by λ exp

has single abstract method → `void Accept(T, t)`

P4 → Test2

`main() { }`

// Create a fixed size list Integer & display it using
Internal iteration : Higher order function

→ Main method Arg is
functional If and
this SAM or specific
behavior

⑧ Arrays.asList(12, 3, 4, 5, 6, 14, 67, 45);

ctrl+I → assign to new local variable.

// use Anonymous inner class

`List<Integer> list =`

`list.forEach(new Consumer<Integer>() {`

`@override`

`public void accept(Integer t) {`

`sys(t)`

`});`

Anonymous clas-

⑧ Py → Test2 → replace by ' λ '

list.forEach($\lambda \rightarrow (\text{sys}(t))$);
 action can be anything.

Conversion to ~~non-f~~ ' λ exp'

⑨ Higher order function

⑩ Remove everything in ~~an~~ anonymous. fit the signature

⑪ then catch return → put in behaviour

Test 3 → P4 Create a list and remove all odd

list<Integer> list = Arrays.asList(12, 3, 4, 5, 6, 24, 67, 45);

for-each → will fail b/o structural modification.

collection → removeIf

default boolean removeIf (Predicate <? super E > filter);
 Accepts functional Args E or super type.
function interface.

Function &/f Predicate

↳ functional interface

↳ boolean test(T t) →

↳ boolean test (t % 2)

↳ True - will not be removed
 ↳ false

Collection

list.removeIf (new Predicate<Integer> () {

return t % 2 != 0;

};

print list using fl Function.

Anonymous

list.forEach(i → sys(i + " "));

}

X expression

166 list. removeIf($t \rightarrow t \% 2 != 0$),
list. forEach($i \rightarrow sys0(i)$);

function iff -> if is wrapper to hold SAM

Map: - \rightarrow for Each ()
 \rightarrow cannot iterated

Java 8 onward →

for Each BiConsumer<?super k, ?super V> action),
 ↳ key & value pair
 ↳ performs to all entries

Interface Biconsumante $\langle T, U \rangle$

$T \rightarrow \text{key type}$

$v \rightarrow$ value

④ if it is Functional interface \rightarrow SAM \rightarrow accept(object, object)

→ void 'accept' (T& , U)

~~A Java Interview~~ Java 7

perform the display on map.

possible from Java 1.8 by `forEach(BiConsumer<T, V> accept(K, V))`

17.1 → SRL → Com.app. core & util

→ Online shop ↗

utils → interface → static populate data

static

T7-1 → Test P4 :- main()

// Get populated map of products.

17:34

Map<Integer, Product> map = populateMapFromList(populateData());

Action supplied will be performed on all Entries.

map.forEach((k, v) → sys0(k + " " + v));

AAA

VS → Earlier style

(Imperative)

external iterator

Functional style of programming
(declarative style →)

AAA

Conversion to Collection View:

① keySet() } Import static Map.Entry;
② values() } Imperative style

③ for(Entry<Integer, Product> e : map.entrySet())
 sys0(e.getKey() + " " + e.getValue());

Point only Values AAA

map.forEach((k, v) → sys0(v));

for(Entry<Integer, Product> e : map.entrySet())
 sys0(e.getValue());

Custom Sorting

Old approach for sort.

P4 → Test 5 :- List<Product> list = populateData();
Sort as per date

Collections.sort(list, new Comparator<?> {

}

list.forEach(p → sys0(p));

→ HOF (High order function)

~~(6)~~ Collections.sort() → it is HOF only declaratn
Comparator → 'λ' Expression → Single abstract method
ki Collection.sort((P₁, P₂) → (P₁.compareTo P₂))
*** -
How to Reuse F. Programming

P17 → P4 → Test 6 → Functional literal:

Cop & Test 5

int data = 1234; // integer literal

String s = "hello";

upto java 7 only primitive literal allowed.

from Java 8 :

// Functional programming style → it treats functions
are 1st class citizens.

// Functions can be passed as method arg or can be returned

Q: Can we assign complete funct to a variable?

Yes

Comparator<Product> = (P₁, P₂) → P₁.Date().compareTo P₂.date();
Product Comp
functional Behaviour of
Comparator()

Collection.sort(list2, Product Comp);

using above API (higher order functn)

169

Chaining of more than one Function

Collections.sort(list, productComp)
list.forEach(p → sysOut(p))

Collections.sort(list, ProductComp).forEach(p →)

X
Chaining is not possible.
Sort Return → void
void.() → not possible.
not possible
{filter & display} ✓
Questn.

Accept product category from user remove all product from specified category n display remaining list?

* soln imperative → for, iterator.

* soln Declarative → RemoveIf → Predicate → return

Global variable

soln

Java 8 Streams → chaining of functions

Collects

VS

Streams

* Analogy → movie → ① download

② watch

* AL, LL, SL, → created on heap

| → movie → streaming
| → net required
| → buffers → streams

* actually stores element.

| Does not store elements.

* support add/remove

| not present

* support external iteration

| only internal iteration

① ~~190~~
source → filter → sort → map → collect.

② Streams can be created from collections, Lists, set, long, double,

③ Stream operation are like intermediate or terminal.

Java docs → Stream

Interface Stream<T>

int sum = widgets.stream().filter(w → w.

9/11/2021 Day 18

{ write without IDE }

'X Expression:- any function to add conciseness to code.

④ Functional literal: assigning

→ examples

Display all emp(list) using declarative(function) style.

→ emplist: empList.forEach(emp) → sys0(Emp)); fun.

⑤ Display all emp(Map) {empid, employee}

empMap.forEach((empid, employee) → sys0(empid));

⑥ Remove underperforming emp from emplist → Performance Index

emp.removeIf(~~(empid, pi)~~ → (pi < 10));
employee

→ statefull

getter of performance index
state of emp
is changing

sort employee as salary ~~AA~~ (use functional literal)

w/o touching emp class 8:35

~~Collections.sort~~

Comparator<Emp> sortsalary = (emp₁, emp₂) → emp₁.compar(emp₂.sal))

Collections.sort(emp, sortsalary);

more shortcut in 'λ'

8:uy

171

Method references:

⊗ only $s \rightarrow \underline{\text{sys0}}(s)$, simply calling



system.out :: println()

example: emplist.forEach(emp \rightarrow sop(emp));
emplist.forEach(system.out :: println);

e.g List<Integer> list = Arrays.asList(1, 2, 3, 4, 5, 6, 7); ~~list.forEach(i \rightarrow sop(i));~~

list.forEach(system.out :: println);

is not simply calling function
hence not possible to further
conciseness

Function
interface

Display the integer on the same line space separated?

list.forEach(i \rightarrow sys0(" " + i))

is not simply calling function
hence not possible to further
conciseness

Display Squares

i \rightarrow sys0(i^2 " ");

not possible to convert

Sort \rightarrow list \rightarrow as per price

Collection.sort(list, (P₁, P₂) \rightarrow P₁.getPrice().compareTo(P₂.getPrice());

Comparing

Comparator

More short form

functional interface

Comparator.Comparing (Function < ? super T, ?

Returning Comparator

172

Inter Function < T, R >

(9:06) ✓

T → Type of R/P

R → type of Result

SAM → R apply(T, t)

R/P → T

o.

Sort as per price

Collections.sort(ProductList(P₁, P₂) → (Double) P_i.getPrice Comparator())

Collection.sort(ProdList, Comparator.comparing(p → P_i.getPrice()));

Collection.sort(ProdList, Comparator.comparing(Prod_i::price))

Referencing price in product)

Day 18.1 → P₁ → Test 1

C and P com.app.con
d.
util.

main() {
 Ø get populated product list from util.

 list2Product > list = populateData(); ↗
 sort manufacturer data in Asc manner.

List.sort()

default void sort(Comparator<? Super E>c)

list.sort((P₁, P₂) → P₁.getDate().compareTo(P₂.getDate()));

list.sort(comp). ↗ comp p

list.forEach(system.out::println);

or in a simpler manner →

Product → Pr.getDate

Comparator<Product> Comp = Comp = comparing(

list.sort(Comp);
list.forEach((sys.out::println));

functional P will give
more points

173

Comparing + method referencing :

Comparisons < Product > Comp = comparing product
P → P.getDate()
get Date()

equivalent to

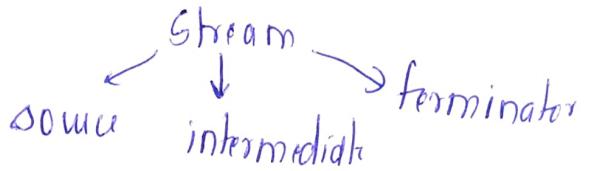
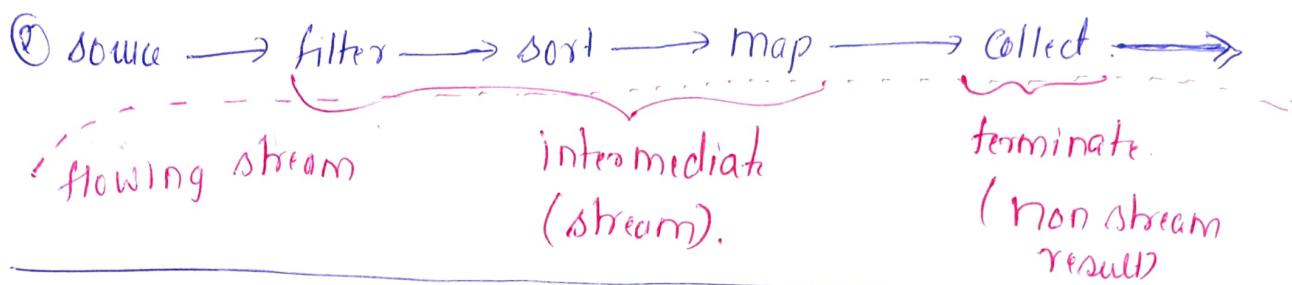
Java 8 Functional streams.

Collections

- ④ collections actually stores the element (i.e. refres)

Streams

- ① doesn't store all the elements represent only structure of opn



④ java.util.Arrays (Java docs)

public static IntStream stream(int[] array);
primitive type
Stream

P2 → Test1 : Main() {

// conv attach IntStream(primitive type) to int[] array;

int[] data = {23, 1, 20, -20, -12, 45, 123, 20, -10, 64, 782};

// Arrays.stream(int[].data);

IntStream in = Arrays.stream(data) → Source

P2 → Test2 → main { }

 || 1. Create int[] → display content

 for each (int consumer action) → Java docs stream
 { → T/P → int.

Terminal → b/o Return type is void : chaining further not possibl.

 ——— (X) Always, stream(data), || printStream.

 for each (System.out.println);

P2 → Test3 → Create AL<integer> populate it.

 Convert if seq stream & display.

Javadocs collection → Streams.

Stream Collectn,

ArrayList<integer> list = new AL<>();

 list.stream() → Returns Stream of integers → Arrays.asList(1, 2, 3, ...)
 collectn → attach stream
 Sequential Stream

Chaining

→ Test3

 → (X) How to add parallel stream
 System.out("Parallel Stream");

 list.parallelStream() & for each (i → System.out(i + " "));

Unordered

O/P

P2 → Test4 → Create Stream of ints b/w 1 - 100 & &
 & display even

⑧ Main() {

⑨ intStreams _____ ⑧

Range (int startInclusive, EndExclusive) 1 to 100

Filter → intStream filter (n) Predicate predicate)

⑧ intStream containing (1 to 100)
 intStream.rangeClosed(1, 100).filter(i → i % 2 == 0);
 for each (System.out.println); Intermediate.

Test 5: It created intStream from fixed size list (unsorted) of integers, sort n display.

Javadocs → intStream → intStream sorted()

AL < List = AL (ArrayList());
 list.sorted().forEach();

Stream of integers → list.stream().sorted().forEach(i → System.out.println(i));
 Intermediate.

Test 6: Display all product name of particular category, exceeding specific price. X/P → categoryName & price. (stream, filter, for each)
 e.g. → product list;

list → Stream → filter → forEach()
 (refs Stream) =

Main() {

try (Scanner sc = new Scanner()) {

list = populateData();

System.out.println

Category Category = Category.valueOf(sc.next().toUpperCase());

176 `list.stream().filter(p → p.getProduct() == category)`
creates new list which contains.

returns stream of products
Stream

`filter(P → P.getPrice() > price), forEach(P → sop(P.getName()));`

terminal operation

The diagram illustrates a data flow starting from a 'stream'. The stream is first processed by a 'filter catego' operation, which branches off into two parallel paths. Each path then passes through a 'filter price' operation. The final output of the second 'filter price' is labeled as 'sop'.

How to display empty product category

Count → long count()

Test 7 → main() { Display all nos b/w 1-100
 1) src - filter - reduce ↗
 ! specialised
 reduce

Sum → IntStream → int sum

```
int sum = intStream.range(1, 100).filter(i -> i % 2 == 0).sum();
System.out.println(sum);
```

Display Avg of all odd no 1-100

Average → optimalDouble average()

OptionalDouble → containers which may or may not contain value.

⊗ empty stream \rightarrow no average \rightarrow will not hold double value

⊗ use `getAsDouble()`

Test - 8 → // Display average of all odd nos. b/w 1 to 100

`double avg = intStream.range(1, 101).filter(i → i % 2 != 0).average()`

- `getAsDouble()`; → throws "Exception NoSuchElementException"

Optional Double → `orElse(double other)`

Test - 9 Display avg b/w user specified range.

- ① In case of empty stream.

```
try (Scanner sc = new Scanner()) {
    int start = sc.nextInt();
    int end = _____, _____
```

`Double avg = intStream.rangeClosed(start, end).filter(i → i % 2 != 0).average()`

- `orElse(1000)`

→ returns '0' in case of empty stream

Test 10: 8. Display sum of product price (list) prices of specific category. If category name o/p average price.

|| list --- stream - filter - stream<Products> → Mapper (category) (stream result)

Java doc → Stream → filter → Stream<Product> → MapToDouble → avg
No arg in Stream

Mapper → product → Double

MapToDouble ★♦

get As Double/
but orElse

⊗ `list = populateData();`

`list.stream().filter(p → p.category == category).mapToDouble(p → p.getPrice()).average()`

→ average() · orElse(0);

78 Value() returns collection

collection is not a list ✓

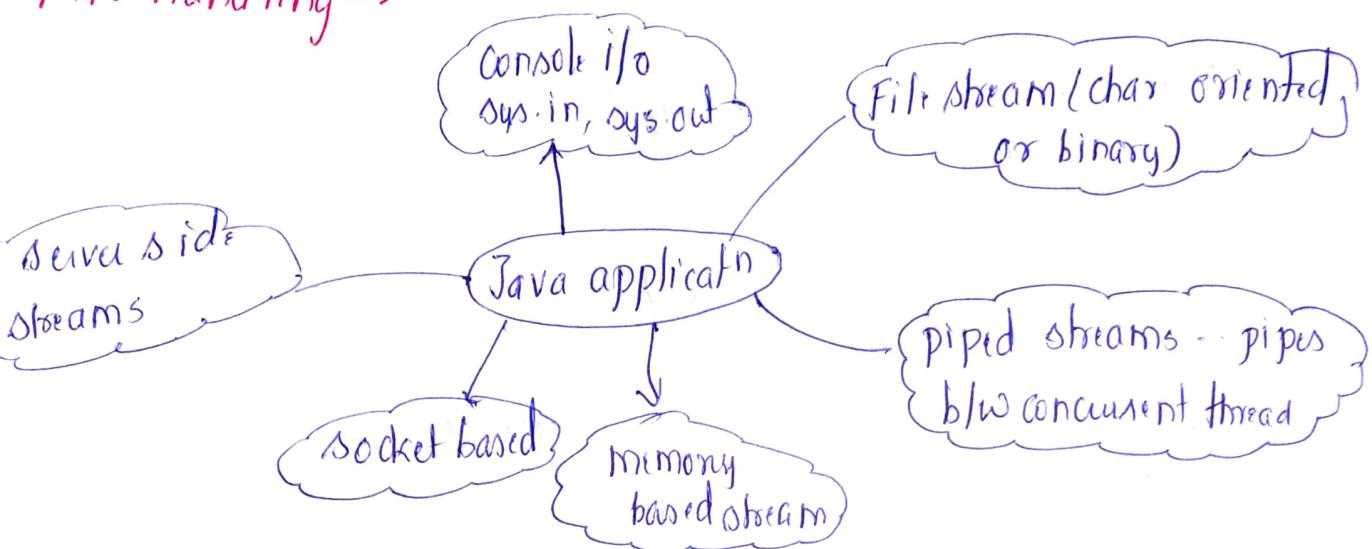
list is collection ✓

⑧ ArrayList (constructor) → to convert to arraylist

Java docs → LinkedList & Collection c).

Input / Output Programming :-

File handling →



Java applicn ← Data Source (Node stream)
I/O stream

Java applicn → Data sink (Node stream)

Types of Streams

mandatory

Node stream

⑧ Device handling stream

file z/p stream, PipedZpstream

SocketStream.

Buffering

⑧ → Buffer → Device
Java App

⑧ Buffer for all Devices
is present

filtering (12:34)

⑧ auto conversion to
binary.

⑧

Hierarchy:

12:20 40

179

(java.io)

- ⑧ Any runtime error instead/will be... **Java.io.IOException**
- ⑧ Java.io API --- blocked API, any read/write operatn causes it to block the blocking of inv

Byte data streams
Binary

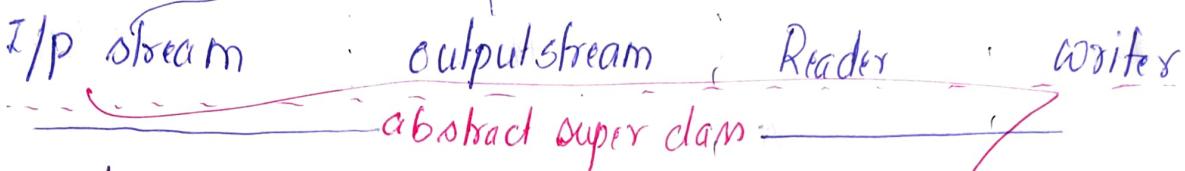
char data streams

DTU is char

- ⑧ for read/write operatn
data transfers unit in byte

- ⑧ will read 2Bytes and converted to char

- ⑧ No conversion



Java docs → java.io → InputStream.

write (byte [] b)

write (int b) → only LSB written.

→ Reader

Concrete classes a).

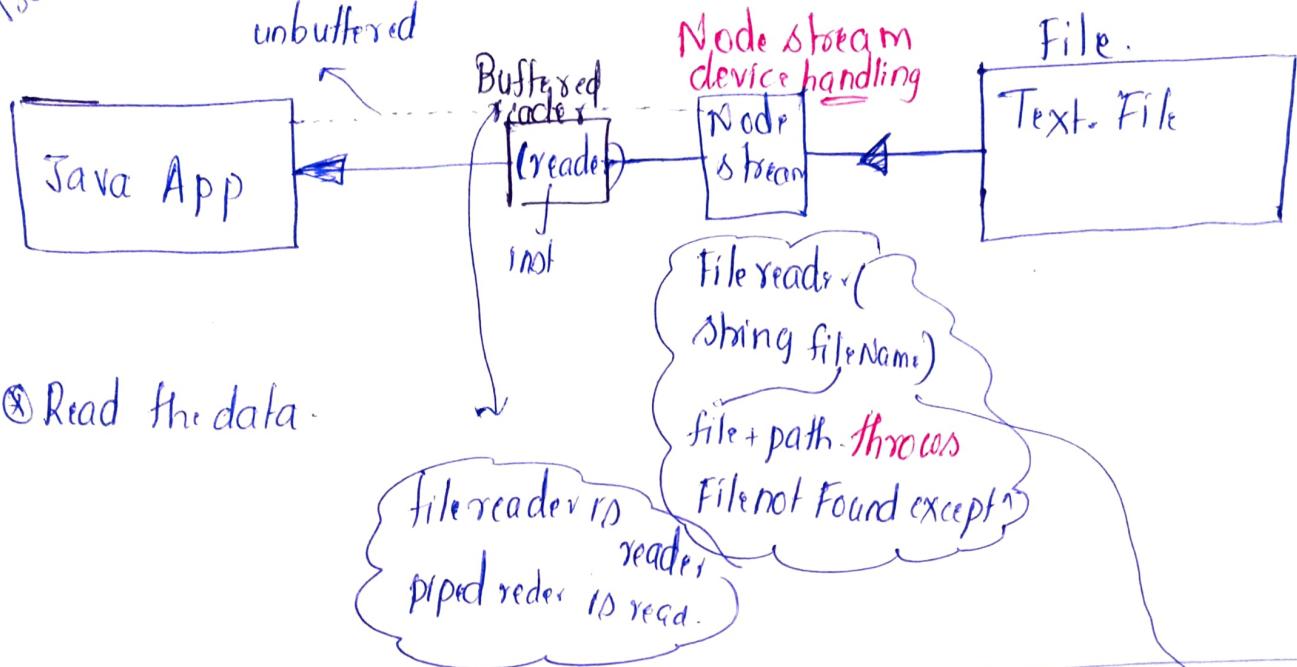
- ⑧ Node streams are always unbuffered.

FileInputStream

FileOutputStream, FileReader

FileWriter

Q) Read data from text file using buffer, till EOF, display
unbuffered



Q) Read the data.

Java code → BufferedReader br = new BR(new FileReader());
public String readLine() throws exception;
End of line → \n → delimiter.
EOF → Returns Null
String s=null;
while(s=br.readLine() != null){
 System.out.println(s);
}

day 18.2 → char[] → BufferedReader compulsory.
I/O based main() { } Java ←-- BufferedReader ←-- Node Stream
1st br = new BufferedReader();
2nd sc. close() → try (Scanner sc) { }
3rd sc.close() → 1st BufferedReader br = new BufferedReader(new FileReader());
→ read file data till End of stream/file.
String line=null
while(line=br.readLine()!=null){
 System.out.println(line);
}

Functional style of code

Java docs → lines → BufferedReader.

Test Buffered Function

```
→ stream of string  
br.lines().foreach(system.out::println);
```

10/11/2021

MAX

list.stream().filter(s → s.getSubject().max(Comparator.comparing(Student::getGPA)).get().getName()); → after terminal operation.

Revision :-

java.io → read/write methods → Blocking Methods

I/O → IOStream → Console
(data transfer)
→ file handling

To read data from src → java app (I/O) stream
into java.
(O/I) stream
out from java.

node stream → device handling streams
→ unbuffered

FIS, FOS, FR, FW, PIS, POS, PR,

buffering → in case of large data transfer → independent of the device
stream

BufferedInputStream, BufferedReader,

Filtering or Conversion Streams :-

To binaryData, primitive types / objects.

e.g. DTS, DOS, OOS, ODS

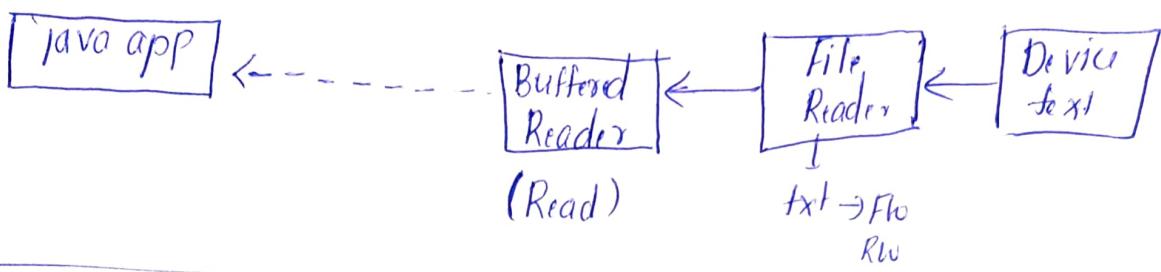
Which are types of stream in ZO?

① Byte oriented → z/p stream "java"
→ o/p stream "java"

② char oriented → Reader
→ write read to java
 write from java

File handling Cases: FIS, FOS, FR, FW.

Chain of Stream



try {
 BR br = new BR(new FR(sc.nextLine()));
}

} catch (All

Java docs → **BufferedReader** (functional)
lines **lines**

Object → Read data from text file using buffer, till EOF, filter
lines length > 50, upper case.

try {

 br.lines().filter(s -> s.length() > 50).map(s -> s.toUpperCase())

}

Converter to Upper

19.1 → char IO → Test BufferedReader.

try (Scanner sc, BR br) {

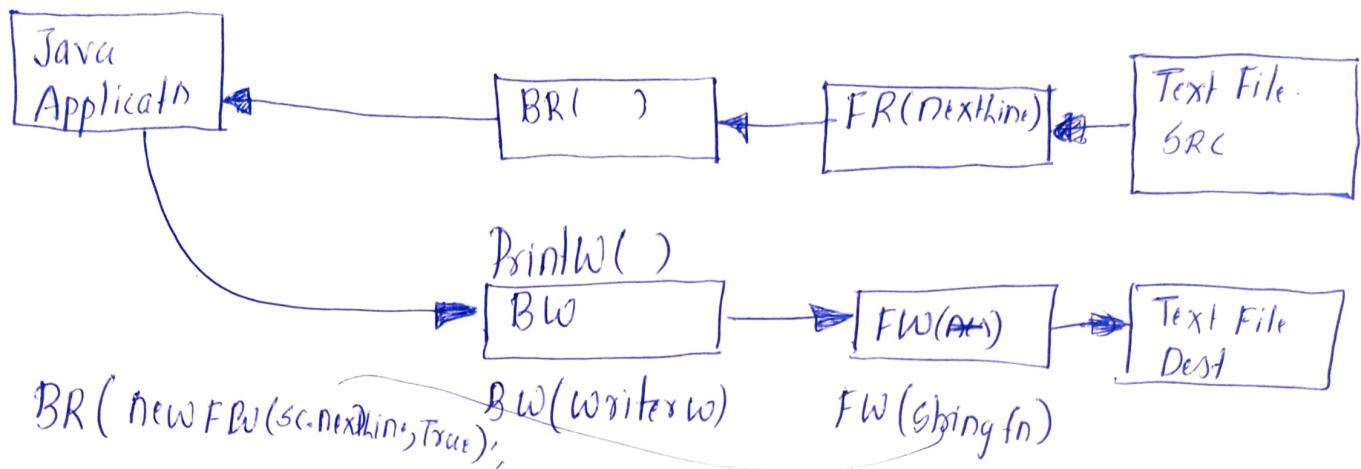
 BR br = new BR(new FR(sc.nextLine())) ;

br.lines().filter(s → s.length > 50).map(s → s.toUpperCase()); 183

for each(syso: println);
syso("success file read");

Objective → Copy file using buffer

x/p → src file name
dest file



FileWriter - constructor

overwrite → FW(string fn) —

Append → FW(string fn, boolean append);

BufferedWriter Methods (methods are not powerful)

Void write only char arrays,

SOLN → printStream (Binary Based) ⚡⚡⚡

Best Buffering + Powerful

PrintWriter (Character Based)

⊗ Const PrintWriter(writer out); → overwrite.

⚡⚡⚡⚡⚡

BR br = new BR()

Append Flag
Auto flush

PW pw = new PW(new FW(destfile, true), true)

180
ArArArAr

Why PrintWriter and not BW?

BW

PW

not user defin.

gives binary

powerfull API

O/P

* we can define size

(*) limited API

(*) can send data to
char streams

(*) does not support
auto flushing

(*) `printWriter(OutputStream out)`
↳ only Buffer + storage

(*) support auto flushing.

Flush → content is sent directly before buffer is full

+
close → (send + close)

buf - full →

19.1 → char.io → Buffered File Copy

main()

try/Scanner ; BufferedReader = new br(new FileRead(sc.nextLine()));

printWriter = new printw(new FileWrite(sc.nextLine()));

I/P O/P ⇒ Buffered char o/p :)

{ for Each()

br.lines().forEach(s → pw.println(s));

Directly calling method
on s we can use references

br.lines().forEach(pw::println);

X A A Interview
A A A

Take a populated product list → accept product list → sort them date
→ save to text file using buffer.

Design

① core classes → product, category

② custom exception →

③ validation rules

④ Collectionutils → Collection utils

⑤ ZOutils → populated Data → list

⑥ Tester → add static method to return sorted filtered & sorted product to caller.

→ s/p → product list, category

o/p → stream or list (Collect) .

To collect a stream in Collection:

I/O util → add static method to save product details to the text file
buffered manner
s/p →
o/p → void

19.2 → copy com.app.core & utils from 18.2

① Collection utils

Add static method to return sorted & filtered

static Stream<product> filterAndSortProducts(List<product> products, product, category)

Return productList.stream().filter(p → p.getCategory() == category)
• sorted(Comparator.comparing(p → p.getDate()));

19.2 → ZOutils interface.

{ static void storeProductDetails(Stream<product> productStream) }

19.1
 1) Java app \rightarrow PrintWriter \rightarrow FW \rightarrow Text file
 Text data \rightarrow PrintWriter(fileName)
 $\text{try}(\text{PrintWriter pw} = \text{new PrintWriter}(\text{fileName}))$
 {
 1) write product details \rightarrow P \rightarrow PW.println(P)
 ProductStream.forEach($(\text{Product p}) \rightarrow \text{pw.println(p)}$);
 }

19.2 \rightarrow SRC \rightarrow Tester \rightarrow Test Collection and ZO.

main() {
 ① import static utils.CollectionUtils.*;
 ZOUtil.storeProduct();
 ————— (8) try(Scanner sc) {
 {
 1) get populated product list.
 List<Product> list = populateData();
 Category category = parse(sc.nextLine());
 method chaining sys0(fileName to store product);
 String fileName = sc.nextLine();
 filter and sort(list, category);
 Store product(list, fileName);
 sys0(file stored);
 }

Binary Streams:

Which node streams?

↗ File Z/p Stream will replace FileReader.
 O/P Fwrite.

Java docs → `FileInputStream`.

Const `public FileInputStream(String name) throws FNF`

Methods : `read()` → portable → platform independent.

`read(byte[], b)` reads upto `b.length`.

`read(byte[], b, int off, int length);`

Objective → write java app to accept product from user & store it in a binary file.

④ Enter and display it.

Tester : OR

`Bin2OUT` → store details

{

1 Java app → `bufferedInputStream` →
→ `fstream` →
→ `(String fn)` →
Binary fil.

3

javadoc `FileInputStream` →

`write(byte[])`

~ programmer needs to convert.

Auto Conversion Streams & filtering

→ need → reduce the burden of P.

javadoc → `DataOutputSteam`.

Converter ~~A/A~~

④ Converter from primitive → binary

Construct → `DataOutputSteam(o/p out)`

InputStream ~~A/A~~

Method → `writeBoolean()` any binary

`writeByte()`

`writeChar()` → 16B.

`writeInt()` → 4 bytes are extracted

input stream from
any file or socket
or database

④ WriteUTF (String str)

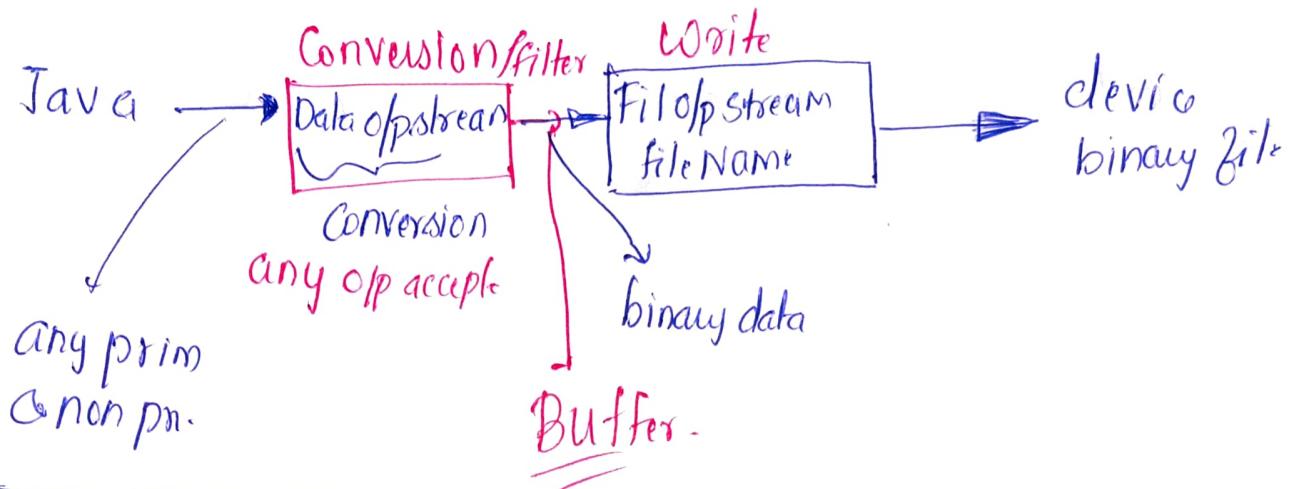
UTF-8
1char → 1byte

WriteChars (String s)

↳ UTF-16

I/O diagram.

Object → store → product detail of single in binary file



19.2 → utils → interface BinZooUtils {

 Static void storeProductDetails (Product, string filenam)
 {
 try (DataOutputStream out = new Dos (new FOS (filenam))) {
 int id, string Name, category Date price.
 id → out.writeInt (P.getId()); → id → 1byte → Binary
 out.writeUTF (P.getName()); → UTF8.
 out.writeUTF (P.getCategory().toString());
 out.writeUTF (P.getDate().toString());
 out.writeDouble (P.getPrice());
 }
 }

19.2 → Test → testData I/O Stream

```

Main() {
    try(Scanner s) {
        System.out.println("Enter Product Details");
        Product p = new Product();
        System.out.println("File Name");
        StoreProductDetails(p);
    }
}

```

Read the Data :

Java → D2S → F2S → Binary.

Data I/O Stream

Const = DataInputStream (Input Stream in)

Methods : readBoolean(), readInt, readUTF();
readDouble()

{ZIP Comming from devices}

Java.io.File → path to folder.

Confirm if file exist.

Const → File(string pathname)

Methods → exists().

Check if folder or file

isFile();

④ Readable file

CanRead();

Object Restore product :-

Q) ~~Validation~~ → regular file → read permissions APZ → java fil.

19.2 → UHs → Binzootil

Static Product restoreProductDetails(String fileName) {
Create File class instance

File f1 = new File(fileName);

If (exists(f1.exists()) & f1.isFile() & f1.canRead)
}

// Validation success.

try (DZS in = new DZS(new FileInputStream(f1)))

}

return new Product(in.readInt(), readInt(...))

}

}

19.2 → Tester Test dzs data I/P stream:

main() {

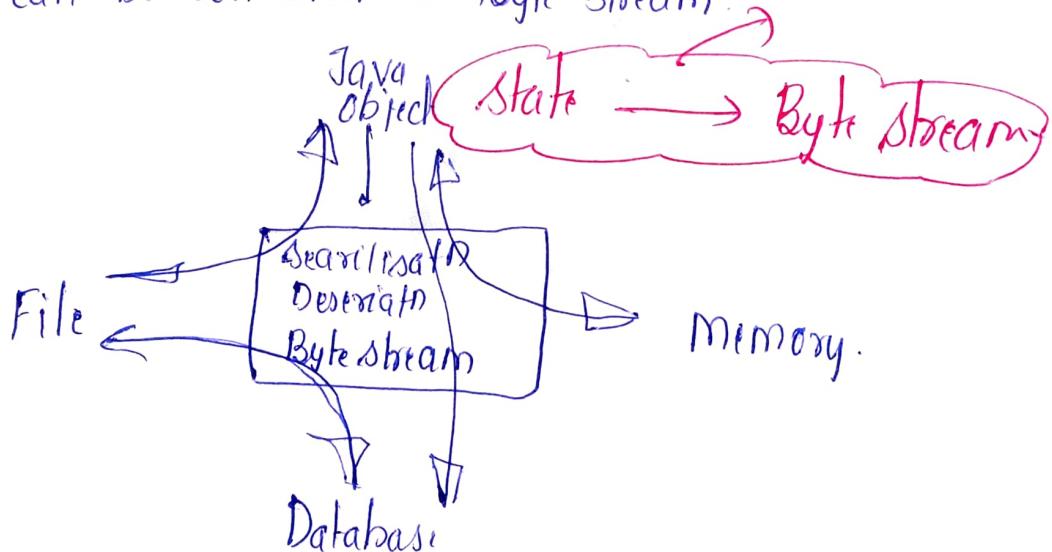
System.out.println("Enter file name");

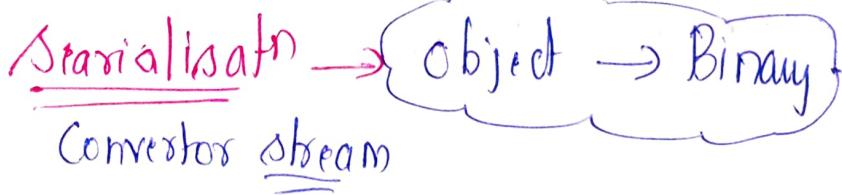
System.out.println(restoreProductDetails());

}

How to Overcome this tedious task

Any object can be converted to Byte stream





Why S&D

To achieve persistent

Persistent → storing info permanently (state of object).

Data o/p, object o/p & f.

Object o/p stream.

* this class performs serialisation.

* extracting state of object and converting to binary

* Serialisation stream → state in binary form.

State of object: - {non static & non-transient data members}

Object o/p stream (output stream out)

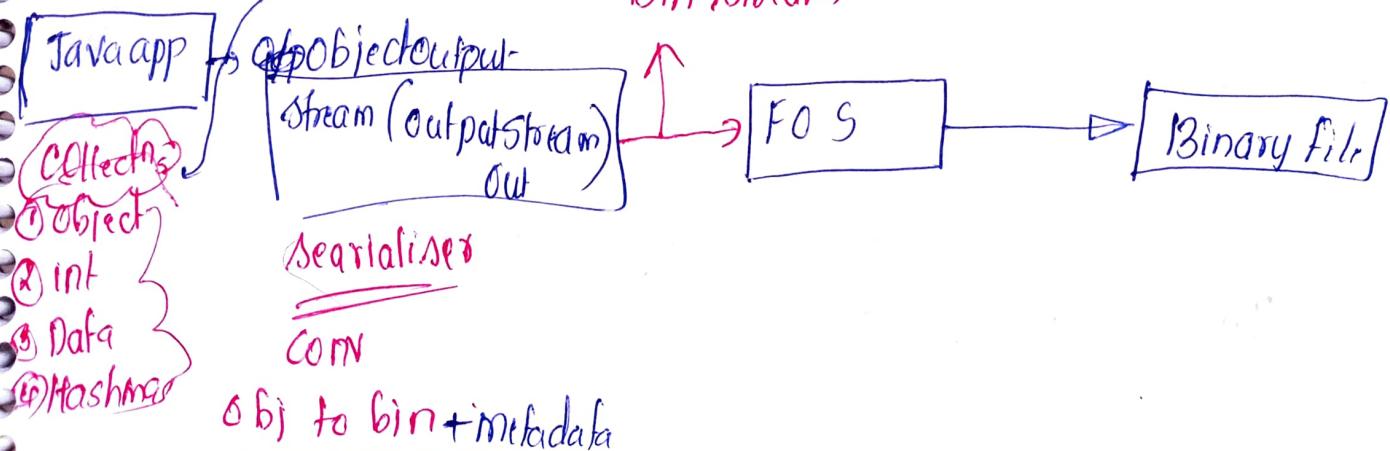
↑ destination o/p binary file.

* public void writeObject(Object o);

De-serialisation → reconstruction of java object.

Object: store object product map in bin file using serialisation.

Bin Stream



Most of the cases we don't require Buffers:

Java cod.

Out = new ObjectOutputStream (FOS(filename));

19.3 Serialisafn de serialisafn

Copy → com. app. code →

SRC → interface → utils → SerializableUtils()

static void storeProductDetails (~~Map~~ Map<String, Product> products, filename)

{ // Java App --> OOS → FOOS → bin

try (ObjectOutputStream out = new OOS(new FOS(filename))) {

}

Javadoc → OOS Out.writeObject(products);

}

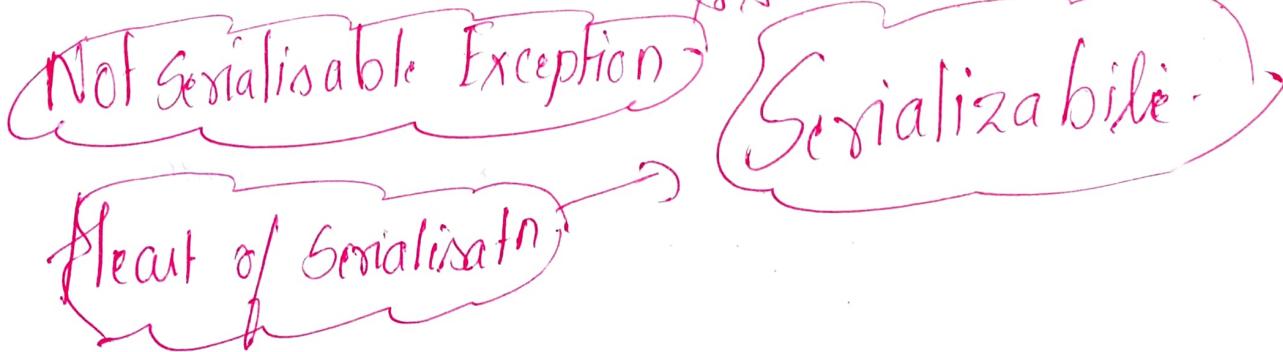
Tester → StoreProducts

try (Scanner sc =

sc.nextLine();

storeProductDetails (PopulateMapFromList(), sc.nextLine());

}



① JVM checks if entire argument is serializable.

② All or nothing → aborts by exception

'java.io.NotSerializableException'

Serializable → no method/data member is marker interface.

↳ Its job is to provide runtime marker for serialisation

Java docs → Serializable.

parent → serialisable → child automatically + ~~for~~

④ Map → key type → Product → Check Sequence
 ✓ serialisbl. integer ✓ serialisbl. X not serialisbl. =
 of Serializable

⑤ Serialisation = transaction in DBT

Metadata → Binary code → Readable text:

at non-static → methods along with metadata



19-4-5 Deserialisation → src → utils → Deserial20

add static method → Why → because of

static Object restoreProduct(String fileName)

File fi = new File(fileName)

If (fi.exists() && fi.isFile() && fi.canRead()) {
 try (OOS oos = new OOS(new FIZS(fi))) {

ObjectInputStream in = new ObjectInputStream(new FileInputStream(fi));

return in.readObject();

Object input
stream

read object

Tester TestDeserial Restore Product

System.out.println(restoreProduct(sc.nextLine()));

194 => Class NOT Found Exception A A

binary ---> java object.

- ⊗ JVM finds no class loaded hence class not found because only state is serialised

↑
Binary file Both state + Behavior
↓
Binfile A A A

Userdefined type class must be passed.

- ⊗ class enums must be added A A

⊗ Why every object cannot be serialised? A A A

- ⊗ no encryption security leaks

- ⊗ serialisation on one machine and deserialisation on other machine where classes are UDT are not available.

11/11/2021 Revision

D-20 r⊗Ans:

A A A Interview

- ⊗ what are prequisites of de-serialisation?

① classes (required) (Behavior)

② supply .class file of "UDT", supply in runtime class path

⇒ else class/Class Not Found Exceptions A A A

- ⊗ Next step → Instance creation in the heap?

provide .ser file (metadata + fully qualified Name + state)

Best way of sharing Behaviour → (JAR)

195

How to create .jar?

Command line / EIDE

Methods in Deserialization?

readInt(), ..., readObject(), → throws ClassNotFoundException,

Invalid class Exception (Serial Version UniqueId)

Chain of I/O stream?

Java app ←-- OOS ←-- FTS ←-- Bin File

19.3 → Serialization

19.4 → Deserialization → Delete com.APP folder → run

→ JAR file creation

→ Right click → com.APP.core → Export → Java → JAR File

→ export destination → d:\dependancies\jar → Finish

Compal

Copy .jar file → right click → open with Winzip →

Manifest.info →

Command prompt:

8:56

D:\> jar tvf dependancies.jar

19.4 How to add Jar

→ Right click → Build path → Configure Build Path →

→ Libraries → classpath → add external jar → goto folder of jar

Transient keyword :-

Objective: suppose application should not store product

19.6 manufacture data

Skipping A component from serialisation (Date / id / password)

serialisation → state → non static & non Transient

Transient → meant for JVM to skip serialisation

Deserialisation → Transient & static ⇒ initialised to Default (null)

What is use of transient? ~~to persist~~
to persist partial serialisation.

19.3 → Date → transient.

private transient LocalDate manufactureDate?

Invalid class exception : → updated jar was not provided,

SerialVersionUID → Serializer

Serial VersionUID → ~~to~~

④ Soln → old compiler produced Serial2D

What is serial version UID?

e.g. serial ver: java.util.HashMap

Recommended: click on warning

① add → transient keyword → Add default SerialVersionUID.

② Create jar again

D20 → README deser & serial.

Serializability

Demerits

① Java technology only.

Difficult to

maintain cleanformal

State of the object

is insecure

② May lead to security leaks.

Method overriding — same name, same, Return type → same or subtyp.

③ Overriding form of the method (in sub class) can't any new or Broader checked exception.

④ What will happen.

⑤ Sub class i.e. can throw same exception or zero exceptn
Cannot throw broader checked exceptn (or new exceptn)

Class B

Overriding
meth

A → ZOExceptn ✓

B → A → File not foun ✓

A → ZOException

B → A → Exception

Multithreading

⑥ What is multitasking? & Why?

many tasks at a time

To effective utilisation of up
to ensure that one task doesn't
effect other task

⑦ notepad + STS + chrome + paint, to improve efficiency.

Process vs Threads

Process

Swapping unit = entire process

⑧ program in execution

⑨ Cannot controlled by JRE/JVM

Thread

Finer control, process can be subdivided into parallel, concurrent
runnable task

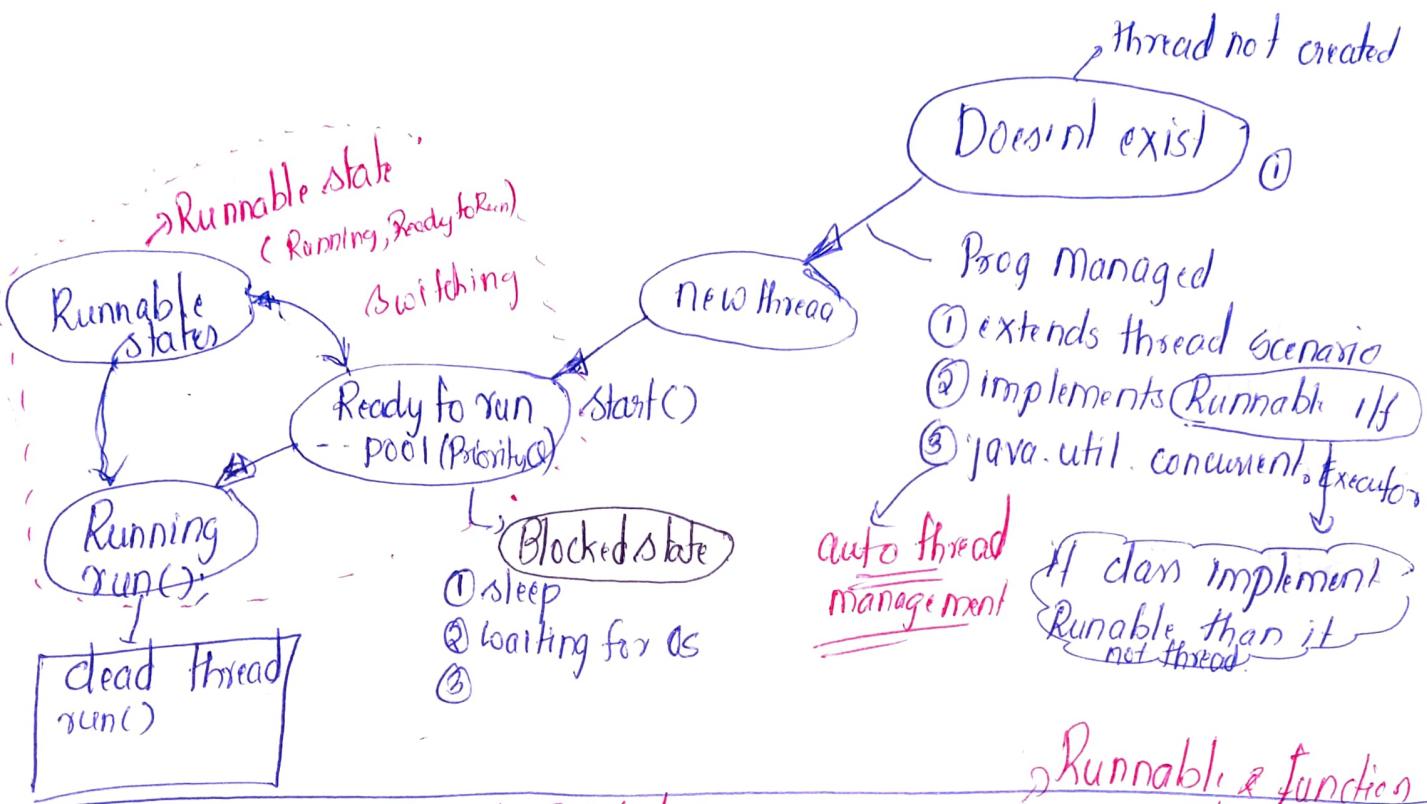
can be controlled partially

Context switching → heavy weight
light weight, since all threads belonging to same process share common add space

Stack → 1 per thread

heap → 1 per process, shared across all threads.

Thread State transition diagram



thread not created
Doesn't exist
Prog Managed
 ① Extends thread scenario
 ② implements Runnable If
 ③ java.util.concurrent.Executor
auto thread management
 If class implement Runnable than it not thread

Runnable & Function

Thread Related API

① Java.lang.thread impl
 class Runnable
 ② Concrete impl class (Blank impl/n)
 run(): blank
 empty run() does nothing and returns

Java.lang.Runnable : functn 2/f
 SAM
 public void run(); will be invoked by JVM to thra B:L (business logic) / execn logic

User defined thread extends thread class no Java ext

e.g. Public class MyThread extends Thread {
 @Override
 public void run()
 {}
 {} BL

Thread

(10:21)

199

Constructor → parameter less

What will happen if you create class (Thread) and do not write run?

no error, blank code runs of parent

→

①

Not a thread we need to
create it later

implements scenario

e.g. public class MyRunnableTask implements Runnable.

@Override Java C error

P. v. run() { } B.L. 3.

Thread API

start point →

Constructors → Thread(); to give name

Thread(String name);

Thread is runnable

④ Constructor of thread class implements

Thread(Runnable instance)

Thread(Runnable instance, string name)

↳ interface referencing pointing impl class

interview

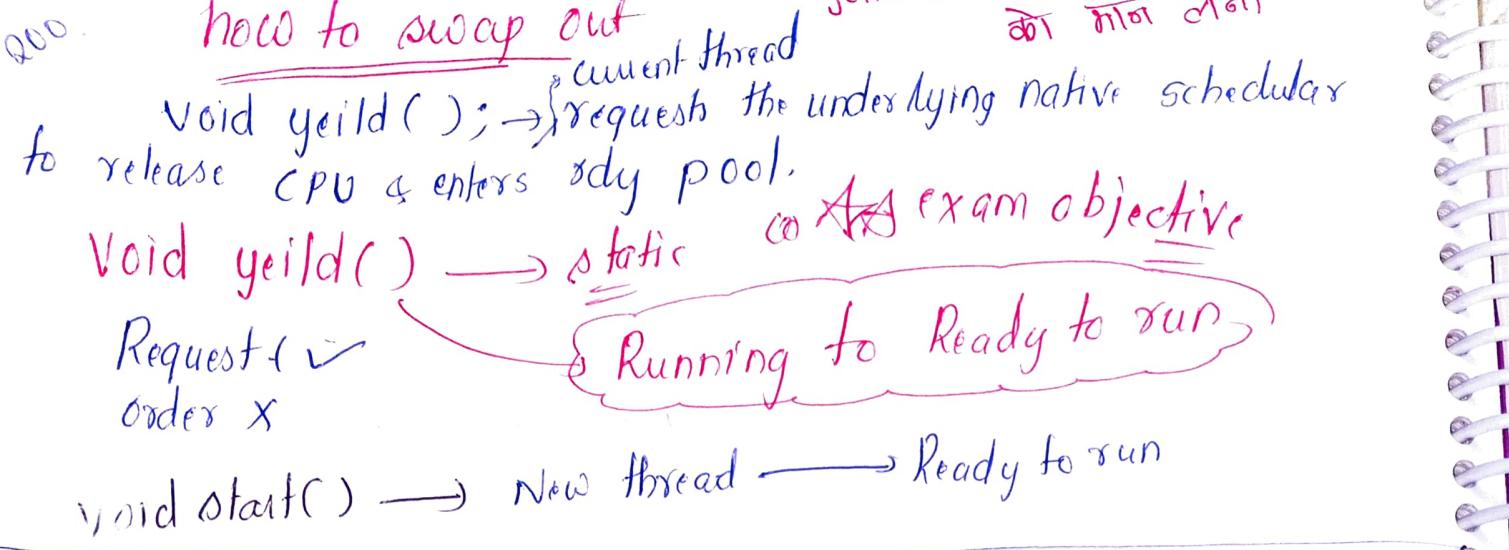
Methods of thread class

Void start() → (New → Runnable)

* → if sometime executed → IllegalThreadStateException

④ Round Robin → fixed time given to all thread

④ preemptive → preemption → lower priority are drawn out

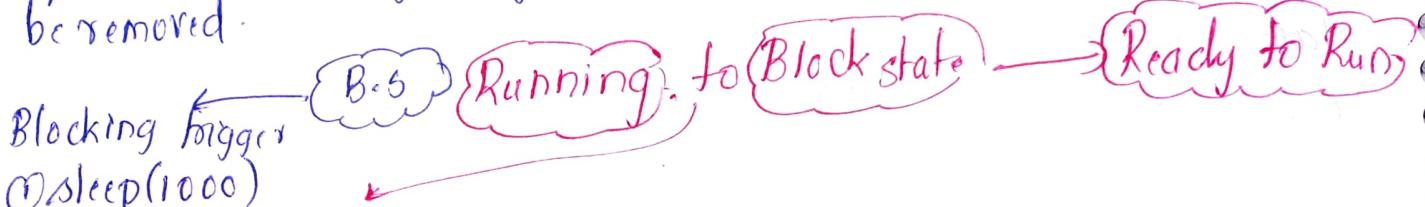


* After execution

Dead state → when run() returns.

*** Blocked State ***

- ① Up should be always busy, hence in Block situation thread should be removed.



- ③ join (waiting for a thread to finish)

- ④ blocked on monitor(lock) trying to access locked resou.

- ⑤ Blocked on I/O = java.io is blocking I/O.

Objective → Testing Concurrency

UStack
1 MA, HA

Main thread (look for psv main())
parent thread.

t₁ child (thread)

t₂ child (thread)

t₃ child (thread).

Day 20.1 → P1 → thirds - NewThread

201

p. class NewThread extends Thread {

public NewThread (String name) {

super(name);

start(); = Start ✓

@override

public void run()

{

New state

↓

Runnable (Ready / Running)

X

⑤ public void setName(...)

public String getName()

priority scale → 1 to 10

Thread class const → min_prio = 1 max_prio = 10, Noo_prio = 5

⑥ public void setPriority(int prio)

*** public static Thread currentThread() gets invoker (current) thread ref.

⑦ public String toString() → overrides & also name, priority, thread groups

→ sysout(getName() + " started");

for (int=0; i<10; i++)

sysout(getName() + " exec # " + i)

thread.sleep(500);

sysout("over");

try {

catch:

sysout(getName());

+

Tester → main() {

sysout(getName());

sysout(Thread.currentThread());

Create multiple threads.

NewThread t1 = new ... (One)

t2 -

(Two)

t3 -

(Three)

Java c → because tester
is not thread.

202

System.out.println(Thread.currentThread()); // 4 threads (1 parent + 3 children)
 It mains business logic
 parent child

for (int i = 0; i < 10; i++) {

System.out.println(Thread.currentThread().getName() + " " + i);

Thread.sleep(1000);

↳ Illegal thread state exception block trying to start dead

* When will JVM terminate? (12:32) D-20 / 11/11/21.

* When all ~~user~~ daemon threads over than will JVM terminate?

* When all

→ 120.1 Thread 2 swap sleep → of child or main).
 delayed will be last terminal.

* Parent thread has died first Orphan threads

We should not keep orphan threads

↳ because parent opens database, socket, but orphan requires resources

How to achieve

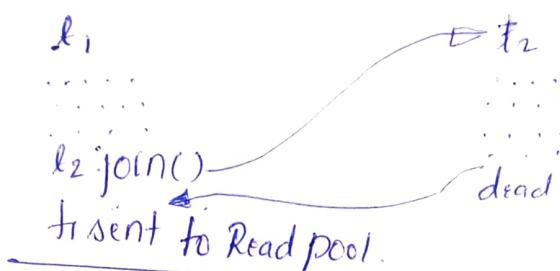
public void join(): throws InterruptedException.

Blocking method API

* causes invoker thread to block till specified thread gets over



⊗ When in t_1 → reaches $t_2.join$ → t_1 Blocked on $join$
 (t_1 is waiting for t_2 to complete exec) → unblock
 after t_2 execution $t_1 \rightarrow$ Head. Ready pool.



on unblocking thread returned to Readypool

Q0.1 → threads.

sys0("main waiting for child thread to complete execn")
↳ t1.join(); → main waiting for t1,
t2.join();
t3.join(); } main waiting for t2,t3.

B1 ^{main} Blocked on join.

(extend & implement
↳ Hand)

Implement Scenario (can come together) =

Thread 3 → MyRunnableTask (it is not thread)

→ extends → implements.

→ Remove conductor

→ @override

getName → thread.getName();

`currentThread().getName()`

Tester → myRunnableTask: task = new MyRunnable();
To create a thread(); Runnable -> only math thread

attach thread to Runnable
Thread t₁ = new Thread(task, "one"); Runnable thread = 1
New → Runnable
t₁.start
= 1, =
Runnable → 4
New = 2

Q) When in $t_1 \rightarrow$ reaches $t_2.join \rightarrow$ Blocked on join

t_1 is waiting for t_2 to complete exec
after t_2 execution $t_1 \rightarrow$ Head.

203

t_1 $\rightarrow t_2$
 t_2 \rightarrow dead
 $t_2.join()$ \rightarrow t_1 sent to Ready pool.
 20.1 \rightarrow threads.

t_1 Blocked on join
 $t_1.isAlive();$
 $t_1.join();$ main waiting for child thread to complete execn
 $t_2.join();$
 $t_3.join();$ } main waiting for t_2, t_3 .

Blocked on join (extend & implement)
Implement Scenario can come together

Threads \rightarrow MyRunnableTask (it is not thread)

\rightarrow extends \rightarrow implements.

\rightarrow Remove constructor

\rightarrow @Override

getName \rightarrow Thread.getName();
 $\quad\quad\quad$ currentThread().getName()

Tester \rightarrow myRunnableTask: task = new MyRunnable(); Runnable ->
 To create a thread(); $\quad\quad\quad$ @only main thread.

attach thread to Runnable
 Thread $t_1 = new Thread(task, "one");$ Runnable thread = 1
 New \rightarrow Runnable
 $t_1.start$ Runnable \rightarrow 4
 $= 1, =$ New = 0.

→ thread 4 → Tester → Anonymous - Inner class
main() {
 System.out.println("Current Thread: " + Thread.currentThread());
 // Create a new thread with name "t1"
 Thread t1 = new Thread(new Runnable() {
 public void run() {
 System.out.println("Name of thread: " + Thread.currentThread().getName());
 }
 });
 t1.start(); // Start the thread
 t1.join(); // Wait for the thread to finish
}

→ Asynchronous nature.

threads 4 → test2 →

Thread t1 = new Thread() → Run
System.out.println("Name of thread: " + Thread.currentThread().getName());
Body of run.

Public void interrupt() → interrupt(ublock) threads blocked on sleep/join/wait

* threads 5 → Adding interrupt signal

line → 21, 22 → System.out.println("Main sending interrupt signal");
Main is sending to t1. 23 → t1.interrupt() → Is not blocking trigger.

⑧ my RunnableTask → t1 → thread.sleep(1000);

11/11/2021 (3:45) How to generate Random number. Sometimes Problem

Javidoocs → Random → Random()

(100 to 200) → Instance.

r1.nextInt(100) + 100 → 0 to 99
r1.nextInt(100) + 100 → 100 to 199

sleep(0)

① return is compulsory

②

12/11/2021 Running → Block

① sleep → throws interrupted exception.
e.g. `t1.run();
Thread.sleep(1000);`

Throws interrupted Exceptn

① join() ② sleep() ③ wait()

Day -> 21 Blocking Vs unblocking.
↳

Unblocking methods

Summary.

205

after 1sec → ready

what
↳ t2 interrupting t1,

t2.run(); → t2 → Runnable

t1.interrupt(); → t1 → unblocked

↳ after interrupt

↳ toy block is aborted.

↳ ready pool.

① sleep

natural
↳ interrupt() → sleep break

↳ unblocking methods.

② Blocked I/O

→ Data available or write operatn complete.

e.g. → sc.next() or system.in.read() → causes the invoked thread to block on I/O.

t2.run();

t1.interrupt() →

{ Cannot be unblocked using interrupt }

Continues to be blocked,

↳ I/O blocker

③ Blocked on join → ① specified thread completes executn.
join()
join(long ms)

② interrupt

③ timeout elapsed.

Q. 2 join → t2.join(1000) →

t1.run()

t1.run calls t2.join(1000)

t1.run:

try {

t2.join();

System.out.println("join completed");

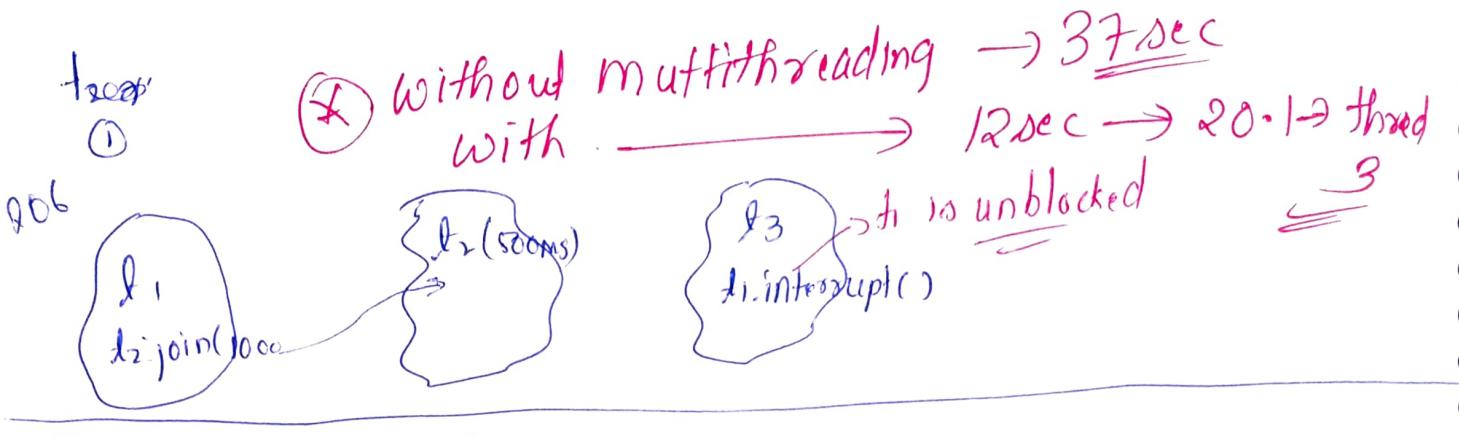
catch (Exception e) {

Simultaneously

↳ t2 gets (deadlock) 500ms

t1 is immediately unblocked

block → ready



3 ways of implement

- ① Create separate class
- ② Anno class
- ③ Lambda expression

What will happen?

- ① It extends thread & don't override run():
do nothing, executes parents run().

- ② implements Runnable & don't implement run() method?

Java corr. → compulsory to implement Interface Method

What will happen?

→ main thread will run()

instead of calling start(), run() method is invoked?

→ no error internally as method start() started.

its invocation will →

- ① cause a new native thread or extant to be created (by os)
- ② cause the run method to be invoked on that!

Objective: create a multithreaded App for saving student detail.

in 2 text file (taken from the map of students): in student_gpa.text : sorted by gpa & in student-dob.text sorted by dob) using 2 different thread concurrently?

① Thread

↓
file 1

Threads → main

② Thread

↓
file 2

Sorting & writing

- dop based sort task → ④ students.
- gpa based sort task
- ① Cons
- ② State,
- ③ run

⑤ get sample data

⑥ Scanner

⑦ child thread(Runnable)

⑧ Constructors

⑨ Run.

Main

$\{$ doB_sort() S subjed_sort
 hm \rightarrow collectn \rightarrow stream \rightarrow list
 write it in text file.

Parsing functional Argument

Day 21 \rightarrow copy from day 19.assignment

21.1 student collectionUtils \rightarrow add static method to sort students info in map \rightarrow sorted list.

static List<Student> sortStudentsGPA(Map<String, Student> map)
 $\{$ Stream not possible

return map.values().stream().sorted(s1, s2 \rightarrow s1.compareTo(s2)).
 Collectn unsorted Comparing(s1, s2)

Collect terminal
 \rightarrow Collect / collectors.toList
 \rightarrow New list

WARNING

Static method to sort as per dob. \rightarrow just add functional Arg

\times static List<Student> sortStudents (Map<String, Student> map,
 $\{$ Comparator<Student> comp)

return map.values().stream().sorted(comp).
 3.

37 sec
12 sec

package utils \rightarrow interface Text2outils $\{$ Textfil.

static void storeDetails(List<Student> students, String filName)

$\{$ Java api \rightarrow PrintWriter \rightarrow FW \rightarrow Textfile

try { PrintWriter pw = new PrintWriter(new FileWriter(fileName)) } $\{$

students.forEach(s \rightarrow pw.

pw.println()

package → runnable tasks. Comparator. comparing()

20th GPA Sorter Task implements Runnable \Rightarrow GPA Sorter Task implements Runnable
 @Override add private map & filename;

Construct() public void run() {
 System.out.println(currentThread().getName());
 try {
 save & save sorted student detail as per GPA.
 }
 }

Main() → Single threaded Only.

Store Details (SortStudent(students, (s₁, s₂) → (s₁.getDOB(), s₂.getDOB())))

Another Runnable Task

Tester → TestCollection to thread.

```

    Main() {
      try (Scanner scanner) {
        // get populated sample map
        Map = populateMap(populateList());
        // accept file name
        System.out("Enter file name");
        // Create Runnable Tasks & Thread
        Thread t1 = new Thread(new gpaSort(map, SC.next(), "GPA"));
        Thread t2 = new Thread(new dob(
          t1.start();
          t2.start();
        System.out("Waiting for childrens");
        t1.join();
        t2.join();
      }
    }
  
```

Thread safety

Objectives: Consider join bank account shared b/w multiple us.

- ① check balance
- ② update balance
- ③ Lack of atomicity
- ④ lack of thread safety.

21.2 package uit.com.app.core → Bank account → Rename & joint Account
 1) update Balance ← 500 → 500
 System.out.println("Upadating bal: " + currentThread.name());
 balance = balance + amnt;
 System.out.println("cancel updates after th."

Thread.sleep(20) // simulating practice delay for forcing context switch
(running -> blocked) 20%

⑧ Main blocked on join

How to tell Java to do it Atomically

→ balance = balance - amt; // customer decided to cancel
System.out.println("update over " + currentThread().getName());

Applying thread safety: Synchronization:

multiple threads accessing a common shared resource: joint Account

Threads: customer1 (update) & customer2 (check)

{ main · parent / system thread · use thread · no
daemon thread
create single instance of joint A/c balance = 5000;
Create runnable task, attach thread, start.
Wait for complete execution.

{ JointAcc instance
c1 updater
⑧ state: has a reference to JA
⑧ parameterized const
run():
while(true){
 jointAcc.update(500);
 sleep(...);

{ c2 = checker task

run():
while(true){
 jointAcc.check();
 if, contin.
 otherwise → ~~exit~~ terminate
 System.exit(1);

package → Runnable task → UpdaterTask.

{ private jointAccount acct;
 constructor

 System.out.println("update by " + currentThread().getName());

}
 @Override
 run(){

 System.out.println("current thread started");

 try{
 while(true){

 // c1 thread update balance

 acct.updateBalance(500);

 Thread.sleep(17);

Checker Task

@

while(true) :

{
 c₂ thread: check balance & validate
 double balance = acc.checkBalance();
 if(balance != 5000)

{ invalid sys(Invalid Balance !!!!!)

System.exit(1);

}

Thread.sleep(3);

→ add to
detect problem
earlier

Tester → Test Shared Resource -

// Create joint account instance

JointAccount a₁ = new JointAccount(5000);

// Create tasks, attach thread, start

Thread t₁ = new Thread(new UpdateTask(a₁), "C₁");

t₁.start(); → (checker (a₁), "C₂");

t₂.start();

sys(main waiting);

t₁.join();

t₂.join();

sys(many errors)

3

→ later or sooner
due to a RACE code
Aborted Concurrency.

Synchronised Solution { Critical Sect
* ↑ prone to error

public synchronized void updateBalance() {

Piece of code prone to error → synchronised,

public synchronized double checkBalance()

Readme → Race Condition synchronisation.

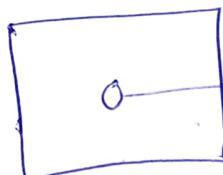
Atomic instructn.

synchronised → method or code block level



box with finey entain'

① lock / monitor can be associated with any java object.



equal to width of single thread

② when does thread need to acquire lock

if its invoking either synchronised methods or code synchronized block

③ multiple synchronised? yes.

④ Blocking trigger : unable to acquire lock (enter monitor)

under which cond'n lock is not release

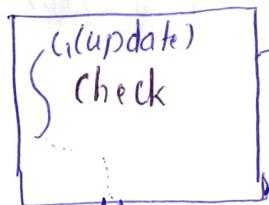
sleep, join, yield, notify or encounter context switching it holds any locks it has - it does not release them

locking

mechanism

Synchronisation Internals:

① C_i is owner of monitors



monitors associated with shared resource (e.g. joint Account)

② Blocked on monitors

main blocked on join

C_i updates
run() acc. update
synchronised
C_i blocked on sleep
inside monitors

why lock: since using synched method

Any thread has to acquire a lock (= enter the monitor)
while calling synched method

C₂ checker

C₂. acc. check

synchronised

① first it has to acquire lock

② Blocked on monit.

⊗ All three are blocked → dead lock

Q1 → runnable → c_2 will ~~acquire~~ lock
After realising c_2 is unblock
lock
{ and c_2 is runnable}

c_2 blocked on sleep inside monitor.

After releasing lock it is not compulsory that waiting blocked will get monitor.

Synchronised at Block: (shared resource ref)

{ synchronised (shared resource ref) }

{ * * * * }

⊗ Why '2' ways? → Block

method → complete method is locked hence worst the performance. and identify critical section and guard it in synch block.

⊗ add thread unsafe in safe manner in sync "Block"

Q1.3 → synchronised Block

⊗ objective → given code is unsynchronised, (thread unsafe)

⊗ Apply thread safety externally.

→ update task t

object of which class
synchronised (acc)
{ which is to be made so

acct.updateBalance(500);

→ checker Task → synchronized (acct) {
 double Balance = acct.checkBalance();
 if (balance != 5000) {
 }
 }
}

213

Services →

21.4 → no-xfc Emp →
utils2 → ,

what will happen if we use directly ~~the~~.run()?
directly code will be controlled by main

④ What is meaning of Runnable? state.
task may be in Ready Pool or running.

System.in.read() → Blocks thread on. System I/O.

④ Rapid fire sheet → written (16 question)
loom video → 3 question

13/11/2021 Synchronized :-

D-22

→ Can a single thread acquire multiple lock?
Yes. when thread access two critical sections.

④ Blocking trigger: Blocked on monitor.
unblock trigger → owner of lock releases lock.

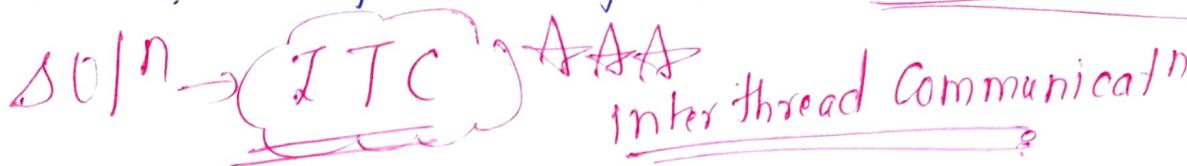
Javadoc → Thread.

Sleep, join, yield,

cpu will be release but dont realase lock

21.4 Problems → producer getting more chances than consumer
→ Data loss
→ consumer getting more chances than stale data

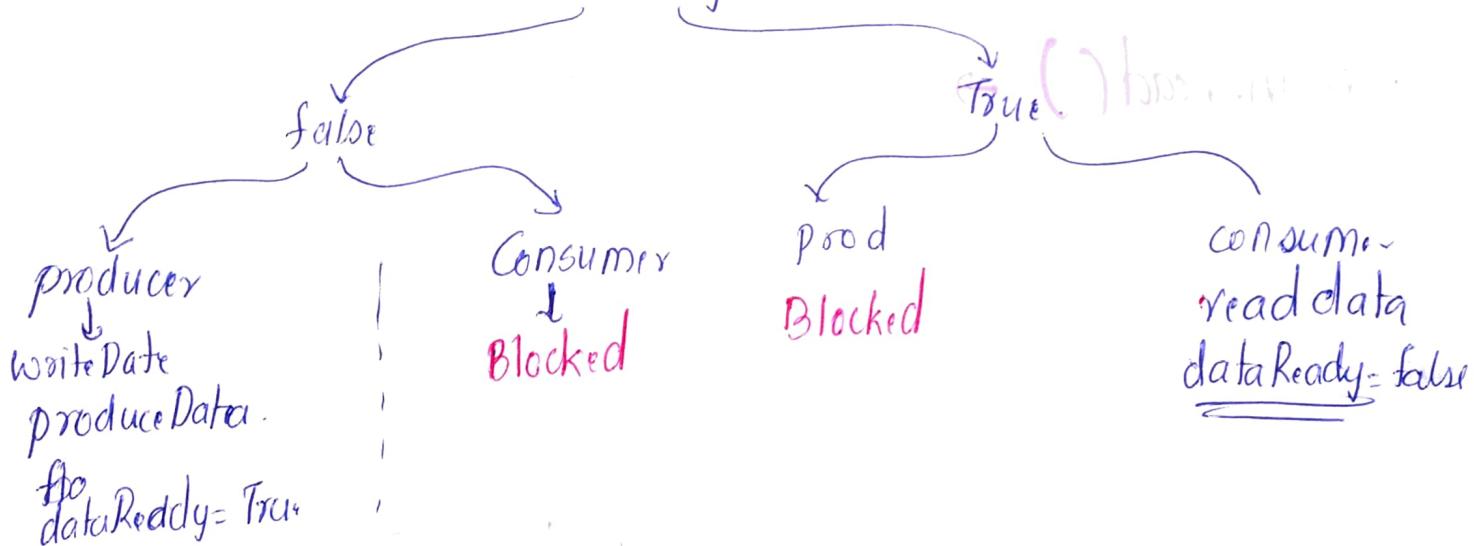
Cause:- lack of sequencing logic (Inter thread Communication)



since there is single producer n consumer : boolean flag:
dataReady :

④ add data ready flag in utils. class.

dataReady =



22.1 → copy 21.4 → Rename → itc-with-sleep

utils 2 {

private boolean dataReady;

writeData {

while (dataReady) {

 Thread.sleep(30);

}
 e = ref;

 sync (writeData);

 dataReady = true;

```

readData() {
    while (!dataReady) {
        Sleep(100);
    }
    sys("Read Data "+e);
    dataReady=false;
}

```

3

↑
leads to dead lock

Main → Block on join
 Consumer → Blocked on Sleep inside monitor.

In case consumer is running faster --

What will happen if it gets 2 consecutive chances : dataReady=false
 main → blocked z/o

Consumer :- blocked on sleep inside monitor

producer :- synched method: writeData → blocked on monitor: outside

After pressing → main blocked on join

requirement → release CPU as well as Block

interrupted

WAIT()

Sleep, Wait, join

methods of Object : (ITC)

① public final void wait() throws InterruptedException.

↳ wait will release both processor & monitor & wait outside

IllegalMonitorStateException →

If the invoker thread is not an owner of the monitor
 (i.e. if it's invoking neither a synchronized method nor a block)

→ When you don't have monitor and you are asked to release

216
22.1 ITC with wait → utils 2 →

write() wait();

Read() wait();

↳ Again Deadlock.

↙ consumer → Blocked on ~~no~~ wait outside monitor

main → blocked on %

producer → DataReady = True.
 blocked on wait

both C&P blocked on monitor outside

Notify()

→ we cannot call without synchronised
↳ if unblocks (wakes up) exactly 1 thread which has
invoked wait on the same objects monitor.

NotifyAll() → wakes up all threads on same object.

ITC with wait + notify

22.1 → write wait();

 notify();

}

Reads wait();

 notify();

{

15:49
13/11/2021

Read Data() →
notify() →

at last.

 notify();

{

till code is blocked

P /Consumer → still blocked on wait() outside monitor.

Sol:-

TestProducer(producer.java)

syn sending (interrup signal to thread blocked on wait)

```
if (t1.isAlive())
    t1.interrupt();
```

```
if (t2.isAlive())
    t2.interrupt();
```

t1.join

t2.join

blocked on wait
object class API

unblocking wait()

```
notify();
notifyAll();
interrupt();
Time out elapsed.
```

~~⊕ ⊕~~ Daemon VS User Thread ~~⊕ ⊕~~

User

④ created by default (i.e. by
any of the thread class const)

④ user thread is running prevent
JVM from termination

e.g. main, all we created

Daemon

④ are created using Thread class

④ public

④ they do not stop JVM from
termination.

usage → servicing thread

↳ e.g. → Garbage Collector

~~⊕ ⊕~~ Java Appln will terminates if and only
if all user threads are over