Java

Core Java

— Used in mobiles, client machines, servers

principle:

Write once, Run Anywhere (WORA)

Library: predefined classes

Java SE core

EE Adv

ME Micro/And

— interpreted

— secure

— portable

— multi-threaded

— Garbage collector

Can't access invalid location.

Scanned with CamScanner

Compile & Run

Hello.java → Hello.class

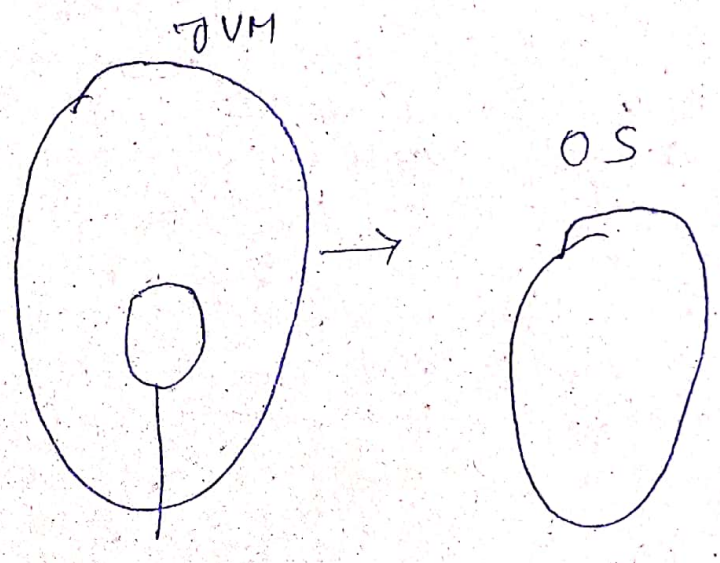
5 classes

byte code

- class
- class
- class

Hello.class

Running



JUSTintime compiler (Interpreter)

line by line gives

and intreprets again

Scanned with CamScanner

jdk

develope.kit

↓

tools to develop java programs or byte code like compiler (javac.exe), Application Launcher (java.exe).

JRE

contains

JVM

virtual machine

and java library

platform dependent

Case Sensitive

- nearly 100% object oriented.
- not possible to make a function which is not member of any class.

Scanned with CamScanner

```
public class world
{
  public static void Main (String [] args).
  {
    System.out.println ("Hey");
  }
}
```

not semicolon

public
private
protected
default

outer should be "public" or "nothing"

camel case of classes

String is a class.

class after . is using static member

out is static reference variable

and . is to point function.

println is ~~non~~ function of print stream

Scanned with CamScanner

javac name .java

java name

abstract keyword for Abstract class

Strongly typed language

premitive

char 2

byte 1

short 2

int 4

long 8

float 4

double 8

String not predefined datatype

Scanned with CamScanner

like 978 L → long

127

0x7f hex a

0177 octal

0b 0110

binary

Type Conversion:

/* */

/** */ document

widening conversion

int can be to float

Narrowing conversion

float to int

float k = 3.5 double

Scanned with CamScanner

byte to short, int long, float, double

Short – int, long, flo, double

char – int, long, flo, doub

int – long, f, d

long – flo, double

float – double

Classes / Obj

Box b1; in C++ & Box *p = new Box()

OUT with

But only dynamically in java

Box *p = new Box();

Box b1 → not object, only variable

Box b1 = new Box() ✓ (correct)

ref variable — not obj name

no pointer (*) — only variable pointing it.

Scanned with CamScanner

Static Members

static int y //static mem var

pub stat void fun() static mem fun.

(static int a)

✗ wrong

Static inner class

static class test

{

}

default - 0

to use static variable

class name . variable / func

can use only static members.

Scanned with CamScanner

Wrapper classes

99 % OOPS

int, char, float are not object classes.

There is wrapper class for every primitive data in java.

~~boolean~~ → Boolean

char → character } classes

int → Integer

value of() : static method

parse XXX () :

XXX value() :

Stores String as int, float

Integer I = Integer.valueof ("1011", 2)
binary

convert string or number (4,234)

to a object we use,

Integer.valueof ( )

Scanned with CamScanner

parse xxx()

Static func

Integer parseint ("102")

102 → int

return int

int x = parseint()

convert a string to int

int n = parseint("99");

xxxvalue() — instant value

From obj to premitive

int c = i1.intvalue();

Scanned with CamScanner

command line arguments

Java echo "saurabh" sukhla

argument

can pass many nos.

Packages

Contains diff class files

same file can't be twice in packages

package world; → first line

name

only class files stores not source file

javac -d . Hell.java

↑ directory ↑ same place

Java world.Hell

Scanned with CamScanner

we can have only 1 public class in single java file

) if no public class, any class name can be filename

Import : keyword

package pack1;

import pack2.student;

(or) pack2*;

Access modifiers.

four

private

protected

public

default.

can be for class, func, mem variables.

Scanned with CamScanner

For outer

public or Nothing

Inner

Four types modifiers

It will be public because we can use it in other packages.

If not public, we can't use

private

not access outside class.

protected

can be access by any class of same package & child class from other package same class also, subclass of same class or extending superclass in other package

default:

only same package

prot metho only by extend not by implemting

Scanned with CamScanner

Constructor

parameterized

Inheritance

class student extends person

{

}

Only one parent class,

No multiple inheritance

Supports

1) single

2) multilevel

3) Hierarchial

Scanned with CamScanner

initialization block

instance

static

only for objects, can be many

{

Init block ~~for~~ first constructor is invoked if any superclass

}

when constructer runs, his will be first to execute

like

class A

{ default value = 0.

{ Init It has no name

}

A ()

{

Init code

}

But if Super class.

A ()

{ super () ~~for~~

~~Init block~~ { }

}

Scanned with CamScanner

only for static members

static
{
}

can be called without object & main should be must executed before constructors

This / super / return can not be used

overloading / overriding

polymorphism

no function hiding in java

override (B fun override A func)

same arg, name, ret type

class A
fun1 (int a

class B
fun1 (int a, int b)

obj - f1 (5)
obj - f1 (3, 4)

This function hiding in c++ but in java overloading

we can call child fun, but not parent func

Scanned with CamScanner

Final

Final instance
static
local
class
methods

can be initialised once

Final will be blank

Final static int y

public void

{
int k → blank

Final class Dummy

Cant have sub class

Final Method

can't be over-ridden

class A
fun

class B
fun

Scanned with CamScanner

This

reference variable

instance member function only exist

to use same obj in function

like

class Box

{

fun (Box b1)

}

but B1 is not created yet so, this

In constructor also

not used in static content

Scanned with CamScanner

Super

```
class A
{
  fun1()
}
class B
{
  fun1()
  {
    super.fn1()
  }
```

Call super class constructer by like

```
A()

B()
{super("Animal");
}
```

Can invoke parent function from child by super key

and also if same variables

```
class A
{
  int z
}
class B
{
  int z; — class 2
}
```

```
Po.z=2
this.z=1
super.z=1
```

static mem in inheritance

Static functions can be inherited

```
class A
{
fun1()
}
class B
{ fun1() ← It's override /static fun1()
}
```

static fun1()

* function hiding

If function is static it is func hiding.

and if cant static hide inst meth & vice-versa cant hide

both should be same

static mem variable cant inherit.

Scanned with CamScanner

Constructer in inheritance

First parent then subclass

but calling B to A

can also call by super()

we can use in child constructer

like

```
class B
{
    B()
    {
        super();  → must be first line
    }
```

Also if

parent constructor has only arg construc

then we can use

like

```
B()
{
 super(4)
}
```

Scanned with CamScanner

⇒ constructor chaining

invocation of constructor is called constructor chaining.

→ con call con.
→ firstline must be con

we can also call same constructor

like.

```
class B
{
  public B().
    {
      this(4) ← anyone only this or super
      sys
    }

  public B( int a)
    {
      super.
    }
```

It have super because default

Scanned with CamScanner

Abstract class

keyword

which no object can be made

to generalize

to declare common characters

— can contain fields, & Methods

→ no object, but can create reference variable

→ if a class is abstract " it May or may not contain Abstract Method "

But if any abstract Method is present in class then it must be Abstract class.

But in C++ it must be implented to have virtual function

- Abstract Method is no definition must end with semi colon ;

Abstract int foo () ;

can have arguments.

Scanned with CamScanner

we use abstract method to make override compulsory in child class.

Interface

like class

can have private protected in inner class or inner interface

interface name

{

only declaration no body

}

interface

can't be final

→ In interface all variable public

only public	cant be other

public & abstract → Methods

fields should be

int x → public static final

we can also initialize now.

and

class implements II {

only two modifiers of declared outside

public, default.

Scanned with CamScanner

dont have constructor.

abstract class has constructor, to initialize non static members.
public

interface has only static

Multiple extension is possible

One interface can ext multiple

Class B extend A implement D1, D2, D3

Possible

interface reference can be made.

diff btwn Abstract class & interface

→ abstract have any access modifiers
but interface only public members

→ interface only declaration

→ abstract class can have static or non static

→ but in interface all variables are static & functions are non-static

Scanned with CamScanner

→

Abstract can have final or non final

→ no const in interface but in abstract present

27

Taking input from keyword

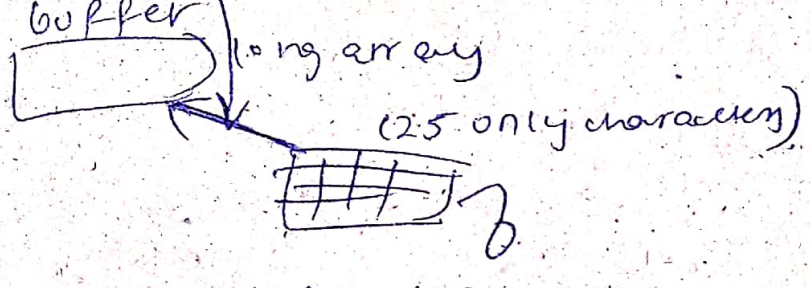
Scannar is final class.

part of java.util.

System.in reference variable

↑ Class

if we recieve from file change system.in



Scanner has next() function;

Scanner sc = new scanner (System.in)

x = sc.next()
float

line — multi strings

next — one word string

Scanned with CamScanner

28. Arrays

int arr[5]; //error

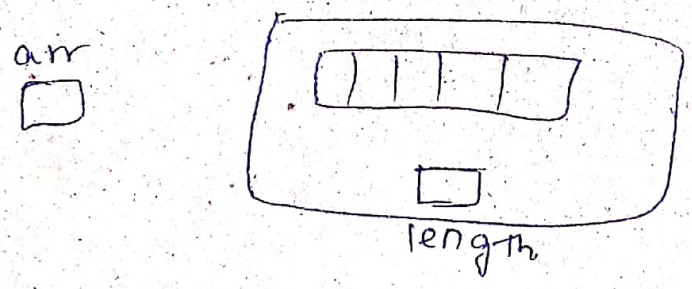
int arr[];

reference variable

int arr[] = new int [5];

int [] arr

only dynamic



arr[0]

arr.length

→ Cant mention size & variables sametime

like

int arr[] = new int [2] {2, 4, 6, 8} ✗

[] {1, 2} ✓

int arr[] = {2, 4, 6, 8, 10} ✓

Scanned with CamScanner

all have zero values

int arr[]; — has blank

arr[0] → accessing blank is error

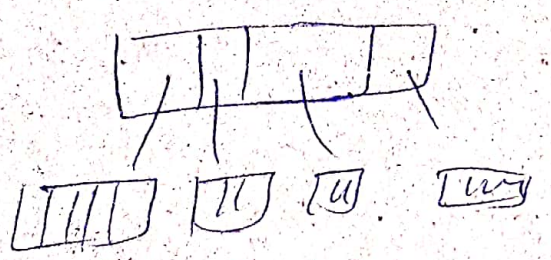
arr = new int[5];

29. 2-D array

int arr[][];

[][] arr; — ref variable

= new int[4][5].



int a[][] = new int[][] // error

int[][5]; // error

int[4][0]; ✓

Scanned with CamScanner

int [][] = new int [4][]

a[0] = new int [5]

can do like this. ✓

a[4][]

a[0][0]

access this is wrong. null pointer exception

String class in java.

in java.lang.string

It is final

It is not predefined datatype

immutable class

string once created can't be changed on same reference

String str1 = "my";

Scanned with CamScanner

2f

Str1 = "My" ——— □

Str2 = "My" ———↗

refers same

and

Str3 = "My name" + "is Bond"

refer same object

to have new object

→ Str3 = new String ("...")

and we can use '+'

String or char array.

S1 == S2 references T

S1.equals(S2) strings T

S1 == S3 F

S1.equals(S3) T

Scanned with CamScanner

String Methods

toUpperCase()

toLowerCase()

s1.toUpperCase()

indexOf(int ch)

lastIndexOf(String)

i = ('m')

(int ch, int from index)

('c', 2)

Starts from '2'.

returns '-1' not found.

compare

equals()

equals ignore case

compare To

COM ← 77

COM — 109

a

i

— neg

dictionary

+ ve

opp to direction

s1.substring(4);

last string (

substring (2 , 4 not inclu)
Start

computer
01 -- 4

32

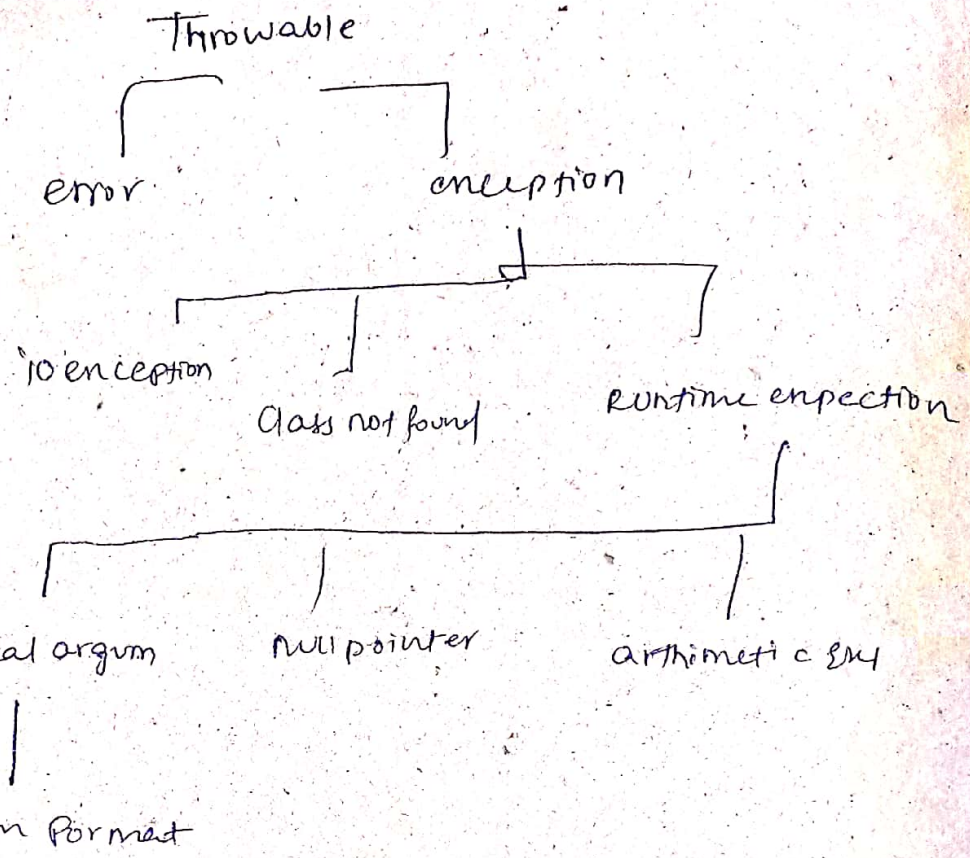
Exception handling.

can handle by default

default throw & default catch
default throw & our catch
our throw and default catch
our throw & our catch

→

Scanned with CamScanner



Class ka object created & throws

Using blank variable is compile time error

s1 = null

s1.length

null pointer exception

Throwable provides getMessage() to retrieve exception

It has string variable

Scanned with CamScanner

checked

Unchecked exceptions

Unchecked are runtime exceptions

array out of bound

null pointer

are subclass of java.lang.RuntimeException class

which is subclass of exception class

```
try
{

}
catch ( exception type parameter )
        class type
  {
  }

finally
{   executes all time
}
```

No finally in C++.

Scanned with CamScanner

if exception happens no lines will be read

catch (ArithmeticException e)

34 Explicit throw.

throw <throwable instance>;

can pass msg to exception object constructor

like.

throw new (ArithmeticException ("Insufficient"));

catch ( )

{ +(e . getMessage)

}

35 Throws keyword (use in checked Exception

Checked Exception Compile time

forces user to deal with exception

Scanned with CamScanner

IO Exception

SQL Exception

IllegalThreadState Exception

checked at compile time

So, ~~use~~ must be caught or reported to be thrown

So, we write or try java

try
catch

function() throws <Exception type>

like

public static void main (String[]args) throws IO Exception
↑

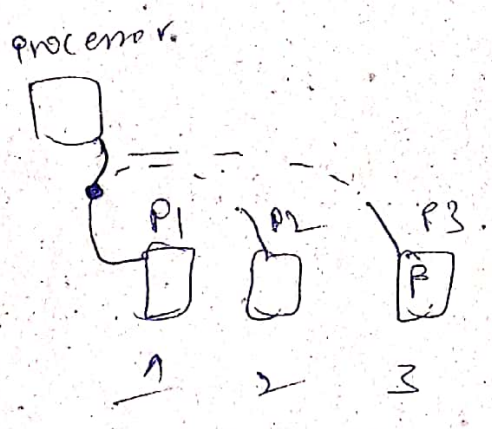
Scanned with CamScanner

36 **Threads**

independent path of execution within a program

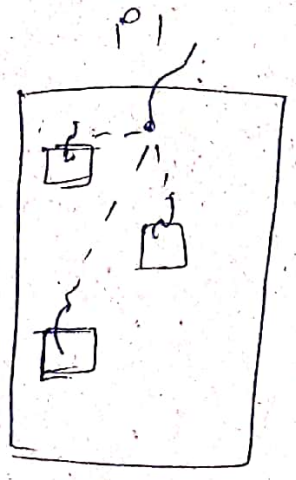
→ many threads can run simultaneous

→ Multithreading → 2 more task executing concurrently



scheduling

we feel like concurrent but it is in order gives sometime to everyone programs



also run parallely if more than 1 processor

Scanned with CamScanner

java.lang. Thread class

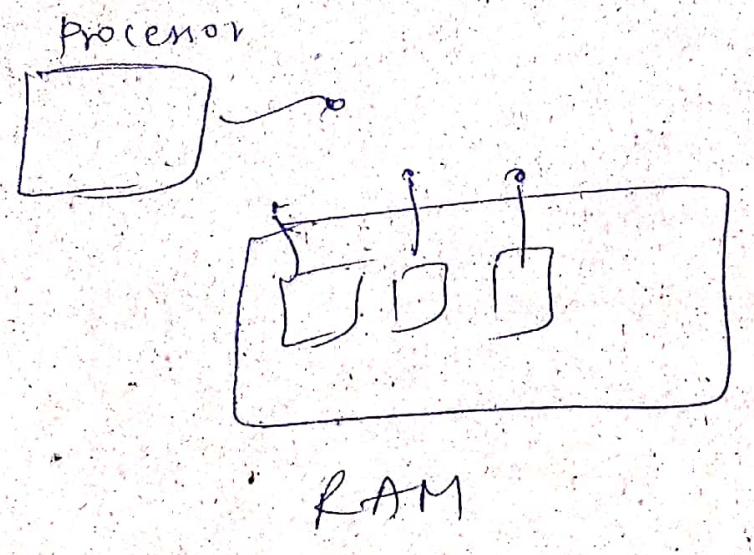
Thread class.

Ways

- implement Runnable interface
  (java.lang.Runnable)
- By extending Thread class (java.lang.Thread

By Runnable

→ Create thread
→ attach code
→ execute thread



Scanned with CamScanner

java

area() ✓

paint() ✓

Sum

1. create thread
2. code
3. Run

Thread t1 = new Thread ()

Thread constructor requires "runnable interface variable" java.lang.runnable

class A implements runnable

{ run () ← imp

{

Runnable r1 = new A()

t1. start()

override run method

thread t2 = new Thread(new A()

= new Thread(new A()

Creat Thread & pass class

Scanned with CamScanner

ends by then run() method ends

Thread class (only extend Thread).

java.lang.Thread

```
class A extends Thread
{ public void run()
  {
    =
  }

A o1 = new A()
o1.start();
```

diff btwn runnable & Thread is

runnable → it can have interface & super class

Thread → it can have only 1 super class

Scanned with CamScanner

Thread States

4 States

New thread

runnable

not runnable

dead

New

run

not run: when user input

start() } only in new thread state
stop()

calling any method besides this will cause Illegal Thread state Exception

(Runtime Exception)

sleep() → not runnable
for specific time & resume

suspend() ✗

wait() → waits for free resource
or other thread

Scanned with CamScanner

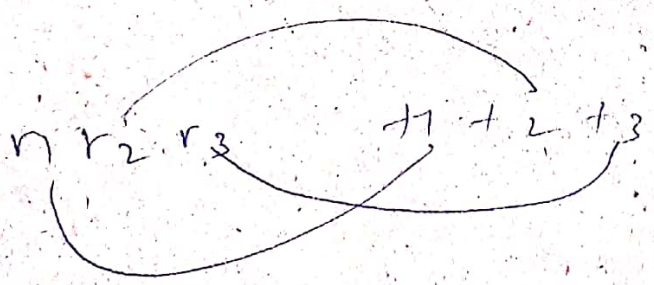
Resume () for suspend()

notify all() or notify() for wait()
should be done by object
which holds resource

dead state
run() comple or stop() This state comes

40

Thread priority



high priority means access to resources

5 default

setpriority() to set

getpriority() to get

Scanned with CamScanner

MIN - PRIORITY(0)

NORM - PRIORITY (5)

MAX - PRIORITY (10)

Static final integer constants are in Thread class

4)

Syncing multithreads

two thread access same resource may have problem

bal = 1000

bal > rembal | bal > rembal

t1 | t2

clash!!! corrupt

Scanned with CamScanner

So we sync

Account → pass this reference

customer 1 (a1)

customer 2 (a1)

To sync we write

synchronized (account)
shared
resource

{

}

Scanned with CamScanner

File handling

java.io

Stream is source or destination
(flow)

Streams are :

byte, character

- input / output of bytes
- in / out of characters

01011011

011011 011 11

C A R

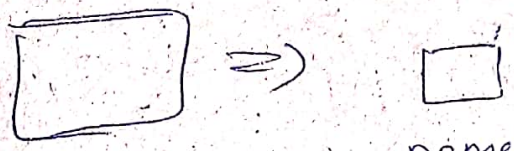
Scanned with CamScanner

File class

g:/drive

g:/java pro/name.txt

path



Java programs name1.txt

name
size
read/write

info about file is in file class

characters

java.io.File

File f1 = new File("pass string path")

f1.canwrite()

.exist()

.getname()

length()

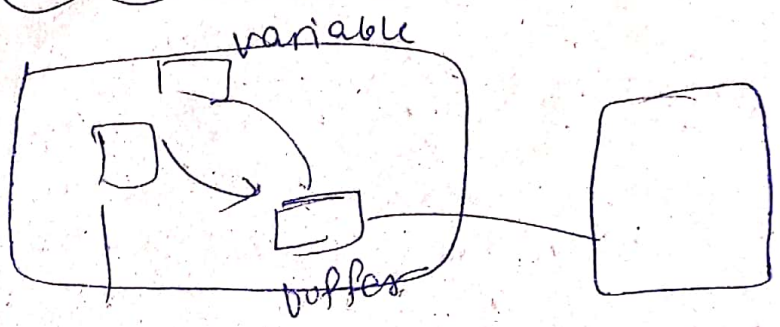
Scanned with CamScanner

f1.create new file().

45//

Read from File using

File input Stream



fileinputstream

object

FileInput stream (File file)

(String name)

read() function

return -1 if it reach eof.

Char(i).

reads() gives unicode

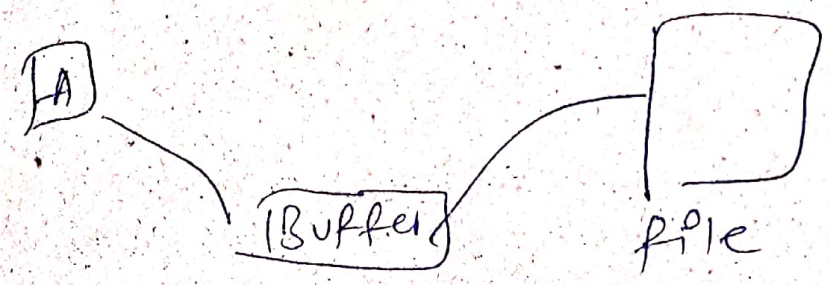
we type

Cast to char

Scanned with CamScanner

write by

Buffered writer class.



Buffered writer (writer out)

subclass (writer out, int sz)

we should pass file writer object

new Buffered writer (new FileWriter
("files/...")
, true)) ;

↑
appending
to not
delete
existing
data

f. write ("comp")

close ( ) ;

Scanned with CamScanner

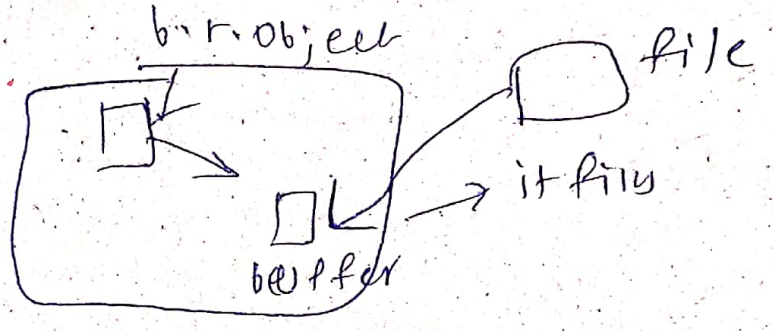
U.7

reading from file by

buffered reader

)

object



read from buffer.

Bufferreader (Reader in)

(Reder, in, size)

B.read = new B.reader(new file reader

(file/...)

file reader

Subclass

while (ch = br.read() != -1)

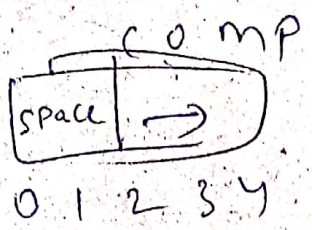
string readline ()

line of text

int read (char[], int off, int len).

2 4

no of letters



Garbage collection:

new student();

compul in java.

not compol in c++
sorry

all are dynamic in java

auto delete dynamic object in java, is called

Garbage collection.

when nothing points to it

By JVM

3 threads

Scanned with CamScanner

main() | thread scheduler() | garbage collector

and we can stop some time before collecting by finalize()

also forefully call garbage collector

System.gc() | RunTime.getRunTime().gc()

only request

Memory leak ← solves

Scanned with CamScanner

finalize()

No destructor in java

runs before Object destroys

lang.object

protected void finalize() Throws Throwable

{

}

call only one time for an object

not chained like destructors

we can explicitly call

finalize for superclass like

finalize()

{

super.finalize()

{

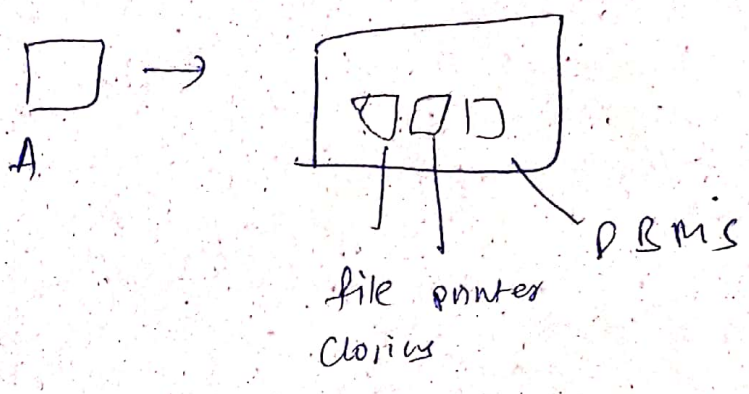
}

~~Runtime~~

Can call forcefully

Scanned with CamScanner

Dynamic obj



(These should be closed connections)

So, without finalize it does not close, but object destroys

nothing refers to →

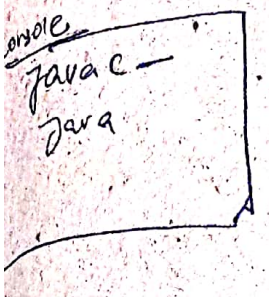
So, we release or close functions by finalize

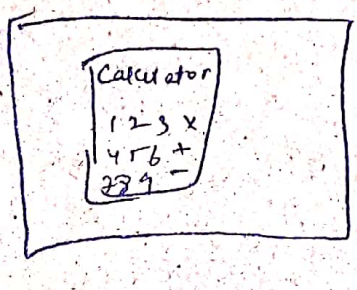
GUI

Graphical user interface

&

console based programing





Two API's

(1) AWT (Abstract windowing Toolkit) org

(2) Swing (fresh) major
(AWT+)

packages

AWT → API has 12 packages

2 packs — java.awt java.awt.event.

Scanned with CamScanner

Java.awt contains classes

Component classes (Button, TextField, Label).

Container classes (Frame, Panel,)

Layout Managers classes (FlowLayout, BorderLayout, GridLayout).

Custom graphics classes (Graphics, Color, & Font).

Java.awt.event (to use Mouse & other)

actions package

event classes:

→ ActionEvent, MouseEvent, KeyEvent, WindowEvent.

Event listener interfaces.

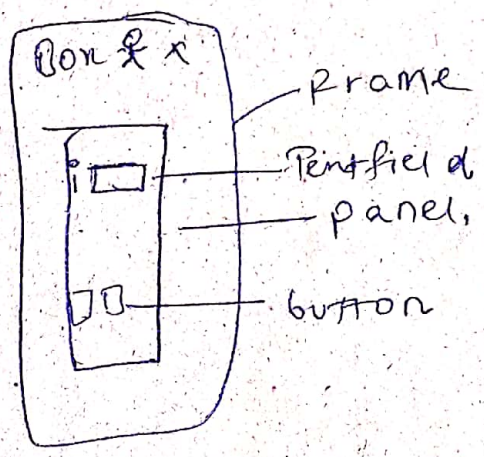
Action listener, Mouse listener, Key listener, Window listener

Scanned with CamScanner

frame toplevel container

frame has title bar (icon, title, minimize)

panel is rectangular area to group components



frame f1

panel p1

Button B1
B2

p1.add(B1)
(theo1 label( )

container has method ~~read~~ ( )
add (component c)

Scanned with CamScanner

Nested class

like Methods.

class inside class

```
class outer
{
static/ class inner
{

}
```

static inner class
non-static inner class

```
outer.inner obj = new inner()
                  outer.inner()
```

for static class

no need for outer object

For nonstatic

Outer Obj is compulsory.

first ref variable

```
outer.inner o1 = new inner().
                 Out.new inner()
```

Scanned with CamScanner

static class can access static variables

inner class can be private / prote / public / def

protected in java	in c++
access in same package for all subclass / class non-subclass / but not for diff package ~~Sub/non-sub~~ class	

~~protech from~~ creating outside package

Scanned with CamScanner

An anonymous class.

→ it will be always inner class
No name

class B
{
class

it should be a child class.

can store parent class ref variable.

class A
f1()
f2()

A o1 = new B()

class B
f2()

we can call only 'A' functions but if it is overriden late binding happens

Scanned with CamScanner

```
class Greeting
  {
    fun ( )
  }

class india

  {

    Greeting g = new Greeting (.) {
        pub void fun (.)   (@ override)
        {
          s
        }
```

an instance is created

child, inner class.

Scanned with CamScanner

59

Generic Methods / Template

class

String c[] = new String[] {"Ind", "Ind"}

pub class

{

P void printarr (String[])

{

}

overloading

P void printarr (integer [] n)

fo

e1.printarr (countries)

e1.printarray1 (numbers)

change in arguments

Scanned with CamScanner

Gen - Methods
eric

so

public void printarray (E [] s)

string
int

{

~~fla~~

public <E> void print (E [] s)

in c++

public <class x> x print (x j).

also

placeholder
accepts
string

for (Ex : s)

print (x)

Scanned with CamScanner

E. can only int, string not float (primitive) objects.

Generic class

only objects, wrapper type

```
class my <T>   no primitive
{
    T i;
    pub void add (T i1)
```

so,

```
mydata <Integer> m1 = new mydata <Integer>();
```

Scanned with CamScanner

collections

Frame work

Class

Interfaces

Algorithms

Interface

↓

Class

What is use!

array []

we want to

add

remove

} these functions and array is in collect framework

and size dynamic

group of values as single entity

Scanned with CamScanner

Homo | hetero.

classes are (Abstract data types)
Methods + (variables
data structure)

<u>collection or Array</u>

Array better if size known.

collection gives memory flexibility

<u>Interface in collection framework</u>

(1). collection (Parent interface).
for group of objects

contains fonda methods (common needs like add delete remove

Scanned with CamScanner

collection

↓

List (interface)

- duplicate allowed
- insertion order preserved
- Array list, Linked list, vector, stack implemented

(classes)

Array list | linked List | ~~Stack~~ vector | Stack

Collection

↓

set

- no duplicate elements
- no insert order
- Hash set | linked Hashset

(classes)

Scanned with CamScanner

set

|

sortedset (child interface)

↓

— no duplicate

— some order

→ Navigable set (child of sorted)

→ for navigate purpose

treeset is implement class.

tree set

Queue (child of collection)

interface

Pque Blocking que

priority

P.Block Q linked Block Q.

(classes)

Scanned with CamScanner

Map (Solo interface)

- Hashtable → properties
- Hashmap
- weakHashmap
- IdentityHashmap..
- sortedMap (inter) (key value pair)
  - |
  - navigable map (inter) both are objects

Key should be unique

value can Repeat

m1 (S1) (k1)

m2 (S2) (k2)

Scanned with CamScanner

Sorting (interface)

→ comparable

default sorting

→ comparator

custom sorting

Classes

Collections

Arrays

collections class has sorting method

Sort()

vector
↓ } legacy (old ones)
Stack classes

Array list

0	1	2	3	4	5
A	B	C	D		

d.add(A) | duplicate allo.

Scanned with CamScanner

methods of collection

boolean add (object

boolean addAll (collection c)

boolean remove (object o)

boolean removeall (collection c)

int size()

boolean contains (object o).

clear()

isEmpty()

iterator<E> iterator()

Scanned with CamScanner

E set (int index, E element)

E get (int index).

Add (int index, E element)

0	1	2
A	B	C

add D

0	1	2	3
A	D	B	C

set →

0	1	2	3
~~A~~ A	P	B	C

int indexof (Object)

lastindexof (Object)

E remove (int index.

Null insert available

Except (Treeset, Treemap)
no Homo
Heterogenous allowed

Array list

default capacity (10).

&
new capacity = (current * 3/2) + 1

$10 \times 3/2 + 1 = 16.$

$4 \times 3/2 + 1$ $6 + 1 = 7$

collection c)

Arraylist (c1)

capacity will be

Scanned with CamScanner

Iterator it = li.iterator();

class

while (it.hasnext())

function

{

cosystem (it.next())

}

for in c++ only class.

63

64

65

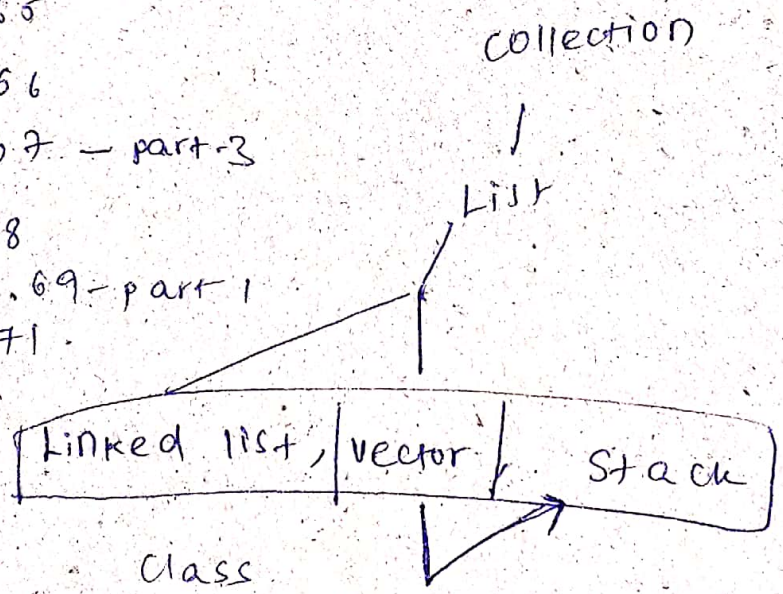
66

67 – part-3

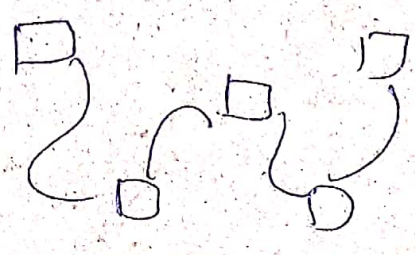
68

69 – part 1

71



link list

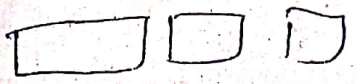


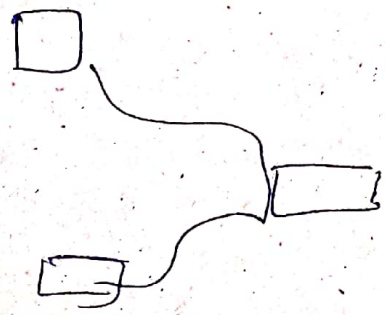
logically sequel, but not physical

Scanned with CamScanner

diff btwn arraylist & linklist

sequence Memory
consume





② l.add(1, E).

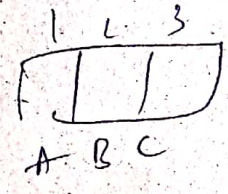
A (E) B → C → D →

add element

but in list

new node

③ indexes

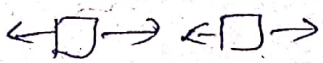


but in array only one shot

(v) no index (if we need ②)

① → ② → ③
then then

doubly link list
structure



Sequence order

duplicate

heterogenous

only non-primitive

also null can keep

Methods

void add first (obj)
void add Last (object)

Object get First ()

Object getLast ()

object remove first ( )
" last ( )

Two constructors

LinkedList()

List (collection c)

Java.util.*

link list = new linkedList( )

Vector:

Dynamic array | also in arraylist.

duplicate
insert order
Null insert
Heterogenous

Methods are synced

Scanned with CamScanner

A List > vector

fast

Stack

push()
pop()
peek()
search()
empty()

duplicate allowed

one constructor

Stack();

CURSORS

to access elem in collection object

3 cursors

1. Enumeration — 3
2. iterator Rank - 2
3. ListIterato Rank - 1

Scanned with CamScanner

Enumeration is interface

only for legacy classes

→ nextelement ().

has more elements ()

V.elements () - inbuilt

returns object of class which implemented Enumeration

String (e.nextElement ());

iterator) for all

iterator iterator ();

hasnext ()

next ()

remove () : not in enumeration

Iterator it = Obj, iterator ()

it.hasnext ()

Scanned with CamScanner

only in forward direction

Listiterator is child of iterator

bidirecton.

listiterator ()

Scanned with CamScanner

Set Hashset

No. Duplicates

no order only in Linked Hashset

use Hash Table

Hetero

NULL allowed

Searching

Constructors

Hashset()

def cap (16)

After 75% fill, dynamic increase start

0.75 load factor

Hashset(cap)

(cap, loadfactor)

(collection c)

Scanned with CamScanner

Linked Hash set

child of hashset

order ✓

Hash Table, double list used

Same load factor 0.75

Sorted set / Navigable set

---

interfaces

(tree set) class

no dupes

sorted

No insertion order in set & sorted set

6 new methods

not sequence order, but real sorting

A C B

↓

A B C

Scanned with CamScanner

S-S

|

M-S

|

Treeset

A B C D

first() sorted order 1st elem

last()

headset (E to element)

A B ← C D

headset (C)

Less than

A B

tailset more than C

A B C ⇒ D

C D

subset (B, D) ~~comparator~~ ()

B C D

✓ ✓ ✗

Scanned with CamScanner

Comparator ().

Navigable set

A B C D E F

to navigate

+ ceiling ("C")

gives more or equal value

A C D

ceil (B)

↓

C

floor (B)

A

higher (B)

C

lower (B)

A

Scanned with CamScanner

pollFirst()

removes & shows

pollLast()

remove & shows

Set

class

Hashset

No order

Linked Hashset

(preserve insertion order)

SortedSet No order

NO

Navigable set NO

Tree set

But for an list insert order preserve

Scanned with CamScanner

Tree set (class)

Sorted

→ Tree set & Tree Map are only Homogenous allowed

→ Null possible but only if nothing is present values.

(enpired)

Tree set ( ) class object.

" (comparator c)

" (collection c)

(Sorted set S)

After v 6.0 no null can be put

Scanned with CamScanner

So, it only compare by default integer & string but what if

StringBuffer ("A")

so to compare & sort we require

Compare To () → function in string & integer

because

Integer, String } has implemented Comparable interface

B B.A

↑ ↘

This calls compare To () function B

B comp with A

↓ →

← ⊖ ⊕ →

B A A B

Scanned with CamScanner

when both 10 10 insert only 1 insert

interface comparator

{

int compare (obj1, obj2)
equals ();
}

comparable for default

comparator for custom

Object has equals ()
class
So, we can ~~cat~~ def as our wish

'-' obj1 < obj2

A ← B

(+)

Scanned with CamScanner

Like

Book b1 } implement comparator
b2 } new objects
b3 }

Treeset arranges in a sorted way.

so

Treeset = new Treeset ();

compare (obj1, obj2)

b2 b1

(−) → +

b2 — b1 b1 — b2

Book Book a2 (Book) o1;

if (book1.getP() < book2.getPrice())

ret (−1);

else

ret (1);

Scanned with CamScanner

iterator it = t. iterator ()

object func

returns obj. which points to first one

while (it.hasnext ()).

{

b = it. next ()

b. get ~~name~~ name ();

Queue (interface)

priority que

Null not allowed

— duplicate allowed

— Heterogenous allowed

orders in FIFO order

Methods

add

E poll ()

rem E peek ()

Boolean offer (E e)

view

} returns ~~true~~ or false

Any

Scanned with CamScanner

bool add (E e)

E remove (E e)
E element () } Throws exception on fail

priority queue is $\boxed{\text{child of}}$ abstract queue

ab. q (class) implem que
↓
Prio. que

Not thread safe

if thread safe → Priority Blocking Queue

<u>Constructors</u>

Priority queue () initial capacity = 11

can dynamically grow

→ (collection c)
(int initial capacity
initial capacity, comparator c)
(sorted set)
(priority que p)

Scanned with CamScanner

poll()

A B C D

↑

poll()

A B C D

↑

it may not store in order

but deleti in order

A B D C

but

poll()

A → B → C → D.

Scanned with CamScanner

Map() (solo interface).

- Hashmap
- Weakhashmap
- Hashtable
  - properties
- Sortedmap
  - Navigablemap
    - Treemap

key-value pair()

Must be objects.

Hash constructor should have classes which implement maps

void (nothing) - no map

Const (type map)

	value		
bus	b1	b2	b3
driver	d1	d2	d3

key

map.

One entry is pair of objects

no primitive

key not repeat

Scanned with CamScanner

also inner interface

Map.Entry (key-value pair)

methods

get.key ()

get value ()

put (key, object)

put all (map)

remove (ob key)

get (key)

contains key (key)

get keyset() all keys

get entryset() all values

Scanned with CamScanner

HashMap (class)

Hashtable uses

unique elements

one null key & multi null values

put(1, "A")

put(null, null) ✓

but again

(null) ✗

size()

isEmpty()

get(obj key)

Hashtable

→ Thread safe

→ unsorted / unordered

→ Hashtable data structure

= no null value

Scanned with CamScanner

- LinkedHashMap (subclass of HashMap)

— insertion maintained

→ slow insert, delete

→ Doubl linked list & hash table

Sorted Map (interface)

has sorting order

sorting on "keys"

Navigable Map

to navigable

To traverse

↓

Tree Map.

not thread safe

Uses Red-Black tree use

sorted in natural order

sorted

Scanned with CamScanner

69. in util package

Collections & Arrays in JAVA
class class

Collection – interface
but collections is a class

Arrays is different from Array list class
class

Collections
[Used for NON-PRIMITIVE TYPES]

Array list
group of elements.

Arrays
has some utility Methods, like sorting

Collections has Static Methods

sort (list a)
sort (list, comparator)
binarysearch (list, T key)
reverse
shuffle
swap (list, i, j)

Scanned with CamScanner

binary search work only on sorted ~~set~~ list

~~Min (Collection c~~,

Max

interface

Min (Collection c , comparator)

Max (Collection c

generic collections disable type casting

before

List list = new ArrayList ();

list.add (10);
add (100);

Now

List<Integer> li = new ArrayList<Integer> ()

String str = list.get(0) // no need of type cast

but before

= (String)(list.get(0)

Scanned with CamScanner

iterator <String> itr = list.iterator();

sends elements in list

(it.hasNext()) checks

it.next();

Collections & Arrays: part - 2

Arrays [Arrays used for "PRIMITIVE" (ints, char, float)]

① util.

② contains static methods for manipulating

Sort (int [] a)

Sort (int [] a, int fromIndex, toIndex)

Sort (long[] a

int a[] =

Arrays.sort(a)

Can we declare two static methods same name?

Yes possible because we can overload

like
static sum(int a);

static sum();

Scanned with CamScanner

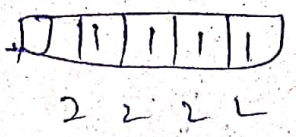
binarysearch (int a[] int key)

(int a[], int from, int to, int key)

equals (int [] a, int b[]).

fill (int [].a , int val) also

too (from , to)



int sum(), cant overload
float sum() - in c++, java

compile time error

can we over load ?

static foo ()

void foo ()

No, we cant overload in c++ & java

Scanned with CamScanner

can we over ride static methods

yes

it is not considered overriding

cause no, runtime polymorphism

but known as method hiding

```
Fine
A ()
{
  func()
}

B()
{ func()
}
```

when call A.fun, only A called

B.fun B only called

Scanned with CamScanner

imp points

1. For static methods

type of reference is checked

Means at compile time

2. For instance methods

type of object is checked

decided at runtime.

instance can override static &

vice versa

static can over instance

Scanned with CamScanner