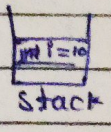
sorting - बाद में

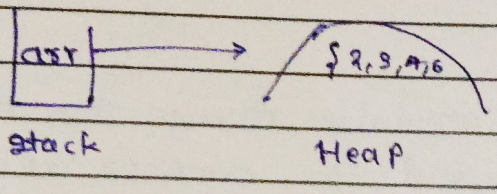
Shrinath
Date 25-7-23
Page No.

Strings

int i = 10; primitives are stored in stack memory



int[] arr = {2, 3, 5, 7, 6}; non-primitives



String name = "Kunal Kushwah";

String ⇒ collection of charectors

Internal working of String

String is most comenly used class in java's library

every string that you creat is actually an object of type String.

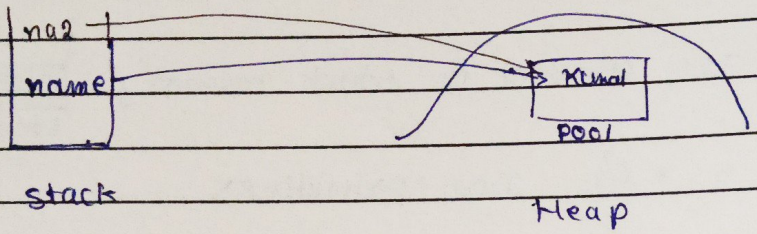
String name = "Nidhu";

data type — reference variable — object

Concepts -
- String pool
- immutability

Shrinath
Date
Page No.

String name = "Kunal";

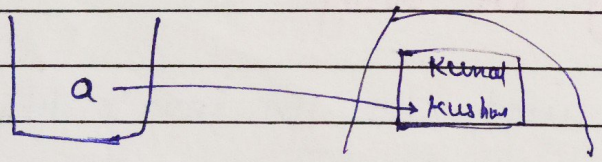


if one reference variable change the value of object, it will be changed for another reference too.

you can not change/modify string object

object can not be changed but ref can reffer another object

String a = "Kunal"
a = "Kushwah"



Comparition of String -

== ~~method~~ Comparitor

a → Kunal
b →

→ cheak if ref. variables are pointing to same object

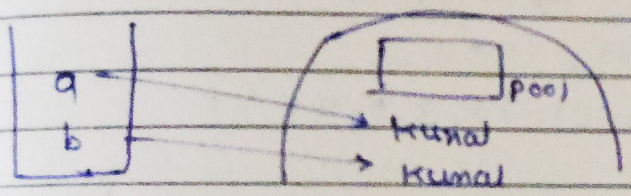
true

a == b

Shrinath
Date
Page No.

// How to creat diff. object of same class.

```
string a = new String("kunal");
String b = new String("kunal")
```



a == b false

- when you only need to cheak value, use .equals method.

a.equals(b); ← give true

String is a class

internally - it is also like a ~~bite~~ char array, acting like a charector array, not mimicing like a charector array.

we can not do ✗ name[0];

for us it is an object

have to use method → name.charAt(0);

PrintStream class

out is a ref variable of type PrintStream

function overloading

System.out.println(56); → o/p 56
System.out.println("Kunal"); →
System.out.println(new int[] {2, 3, 8, 5});

(1) S.O.p(56); → internally println is calling valueOf & valueOf is calling toString

String.valueOF(x);
it is calling toString

p. void println(int x)
{
String.valueOf(x);
}

P.S. String valueOf(int i)
{ return Integer.toString(i); }

o/p
print String 56

(2) S.O.p("Kunal"); → o/p print string Kunal

Internally →

p. void println(String x)
{
(String.valueOF(x));
}

P.S. String valueOf(String x)
{
return
(obj == null) ? "null" : obj.toString();

(3) s.o.p (new int[] {2, 3, 4, 5}); of

s.o.p (object); → [I@5acf9800

println —calls→ valueOf —calls→ toString.

p. s String valueOf (Object obj)
{
return (obj == null) ? "null" : obj.toString();
}

p. String toString()
{
return getClass().getName() + "@" + Integer.toString
(hashCode());
}

we can override toString method

Arrays.toString(ar);

function has been overriden by Arrays class.

Wrapper class – if you want to use all the object oriented programming principle with premives (stored stack in) you can use wrapper classes.

Shrinath
Date 27/6/23
Page No.

Preety printing

you want to print till two digit decimal numbers

```
float a = 53.1234f;
System.out.printf("Formatted number is %.2f", a);
```

o/p → 53.12

// if a = 53.1274f ⇒ o/p ⇒ 53.13 (round off)

if you want to print the value of pi

```
System.out.println( Math.PI );
```

o/p → 3.141592653589793

if you do not want entire thing

```
System.out.printf("pie : %.3f", Math.PI);
```

o/p → 3.142

system.out.printf("Hello my name is %s and I am %s", "kunal", "cool");

Google → placeholders in java string.

Shrinath
Date
Page No.

place holders list / Formate specifier

%c - charactor
%d - Decimal number (base 10)
%e - Exponential floating-point number
%f - floating-point number
%i = Integer (base 10)
%o = Octal number (base 8)
%s = String
%u = unsigned decimal (integer) number
%x = Hexadecimal number (base 16)
%t = Date / time
%n = Newline # explore it

String Concatenation Operator →

```
S.O.pln ( 'a' + 'b' );          105
S.O.pln ( "a" + "b" );          ab
S.O.pln ( 'a' + 3 );            100
S.O.pln ( (char) ('a'+3) );     d
```

when you are doing addition with charectors - it converts it into ASCII value

```
S.O.pln ( "a" + 1 );            a1
```

when an integer is concatinated with a String, it is converted to its wrapper class. that will call toString() this is same as after a few stapes ("a" + "1");

S.o.pln ("kunal" + new Arraylist<>());

o/p kunal[]

S.o.pln ("kunal" + new Integer (value:56));

kunal56

S.o.pln (new Integer(56) + new Arraylist<>());

error - The operator + can not be implemente applied on integer and Arraylist

in java + can be used with primitive & you can only use this with all the complex objects but the only condition is atleast one of this object should be of type string.

valed →

Soplin (new Integer(56) + "" + new Arraylist<>());

o/p 56[]

(-) operator - can not be applied on string

+ is overloaded for strings only. otherwise no operator overloading is possible in java

new String( )

Shrinath
Date
Page No.

+ is the only operator that is intentionally overloaded in Java to support string concatination or string joing.

print ("a" + "b") ;

"a" + "b"

↓

"ab"

after printing, garbage collector will take it bcz we have not saved it.

primitives are stored in the stack it requires less space.

objects are stored in heap.
string that are not explicitly created new are stored inside the heap in a pool (string pool)

== & . equals difference

String performance →

95

```
psvm main ( )
{   String series = "";
    for ( int i = 0; i < 26; i++ )
    {   char ch = (char) ('a' + i);
        s.o.pln(ch);
    }
```

o/p in new line a b c d e f g
h i j k l m n
o p q r s t u . . .

Shrinath
Date
Page No.

want to add in series

```
string series = "";

for ( int i=0; i<26; i++)
{
    char ch = (char) ('a' + i)
    series = series + ch;      // series += ch
}
System.out.println ( series );
```

o/p abcdefghijklmnopqrstuvwxyz

series will add all the characters together.

same as S.O.P ("a" + 'b');
string and a character added
it will become "ab"

series ⇒

"" + 'a' = "a"
"a" + 'b' = "ab"
"ab" + 'c' = "abc"
"abc" + 'd' = "abcd"

memory wastage

new object is being created every time (bcz strings are immutable). (coping the old one and then appending the new changes.

[a ab abc abcd abcde abcdef abcdefg abcdefgh . . .]

all of this reference variables
⇒ large strings will have no

memory wastage

Size $[1+2+3+4+5+\cdots+n]$

$\frac{N(N+1)}{2}$ $\quad$ $\frac{N^2+N}{2}$

$O(N^2)$

if you want to add n charectors in a string are having a complexisty of n^2, not good.

String does not allow us to modify the value, once an object is created in string you cannot modify it, you can only creat new ones.

? Can we creat a sort of data type that is actually not creating new objects but actually adding in just the original object

means if I wanna add n items i can run just in n iterations in n times i can just do it.

Even in previous e.g. we were running the for loop n times, but when i was adding strings together that itself was also running it once twice thrice four time 5 times .... n times. that's why n squared, this time it will not do it, it will added in the end like arrays

if you want to add something in array just add, it's going to take n operations

does there exist a data type like this
yes.
it is known as string builder

only one object is made and the changes are done in that object only, the reference is also the same and it is not changed,

StringBuilder is seperate class just like String.

```
Public class SB {
public static void main ( String[] args )
{
  StringBuilder builder = new StringBuilder ( );
  for ( int i=0 ; i<26 ; i++ )
  {
    char ch = (char) ('a' + i );
    builder. append (ch);
  }
  S.o.pln ( builder );   // internally builder.tostring
}
}
```

thisismutable, just like array we can change, we can change StringBuilder

Shrinath
Date
Page No.

methods available

- append( )
- tostring()
- compareTo (stringbuilder another)
- delete (int start, int end stringbuilder—)
- deleteCharAt (int index)
- indexOf (string str)
- reverse ();

try all
↳ learn more

Shrinath
Date
Page No.

Methods of String class

```
String name = "Nidhi";

Sopln ( Arrays.toString(name.toCharArray()));

[ N, i, d, h, i ]

Sopln ( name.toLowerCase()); nidhi, new object created
Sopln ( name ); ⇒ Nidhi ↖ immutable after it too.

sopln ( name.indexOf('d')); 2

Sopln ( "  Nidhi  ".strip()); Nidhi

Sopln ( Arrays.toString(name.split(" ")));

[ Nidhi ]
```

Shrinath
Date
Page No.

Q. if a string is palindrom or not

abccba abcdcba aba abcba

approches 1. reverse the string & cheak whether it is equal to the original string or not

2. creat a StringBuilder then stared adding in that StringBuilder in reverse order, it the value is same that's palindrom

~~Simple~~

* 3

start end

a b c d c b a

while (s <= e) or for (till hulf of array)

```
static boolean isPalindrom ( string str )
{
    if ( str == null || str.length() == 0 )
      return true.

    str = ~~str.charAt~~ str.toLowerCase();

for ( int i=0 ; i<= str.length()/2 ; i++ )
{
    char start = str.charAt(i);
    char end = str.charAt ( str.length() - 1 - i );
if ( star != end ) { return false; } }
return true;
}
```

Shrinath
Date
Page No.

String - questions for Base

1 Defanging an IP adress

1.1.1.1 ——change it——→ 1[.]1[.]1[.]1

ip adress o/p defanged ip adress

पहले तो . character पर पहुँचना है, इसके लिए मैने (charAt(i)) method use की

आर है तो answer वाली string में . की जगह पर [.] add करना
नहीं है तो जो character है वही कर दिया

अब answer add कराने के भी दो तरीके है।

या तो String ले

इसमे add करने के लिए
String s = "";

s = s + "[.]" ; या
s = s + ch ;

या StringBuilder ले

better why? read previous notes

s.append(ch);
s.append("[.]");

Shrinath
Date
Page No.

Shuffle String

given a string & array of indeces

string codeleet
indeces 4 5 6 7 0 2 1 3

o/p l e e t c o d e
0 1 2 3 4 5 6 7

creat a array of ~~strin~~ char type ⇒ ans

return new String (ans);

Goal parser interpretation