01

10 December 2024 09:23

Basic elements of Java

1) Variables
   1) Local variables
   2) Instance variables
   3) class variables (static)

```
class Employee
{
    private int empno;
    private String name;
    private static int noofEmps;

    public void setEmpno(int eno)
    {
        int x;
        empno = eno;
    }
}
```

Employee x = new Employee();
Employee y = new Employee()

Object

empno | name | empid | name

noofEmps

2) Constants final

final float PI = 3.142f;

PI 3.142

3) Identifiers
- → A-Z, a-z, 0-9, _
- → Start by A-Z, a-z, _, $
- → No spl. char
- → No space allowed.
- → No reserve words allowed

a s

if

for

public

4) Data types
- → Primitive data types
- → Reference data types (class types)

int x;

x = 25

x 25

class Employee
{
}

Employee x;

x = new Employee(); Reference

Stack | Heap

x F000 → F000 empno name

pointer

5) Primitive data types → 8

I) Integeral
1) byte → 1 byte -128 to +127
2) short → 2 bytes -32768 to +32767
3) int → 4 "
4) long → 8 "

-2147483648 to +2147483647

-9223372036854775808 to + 9,223,372,036,854,775,807

byte b;

b = 25;

0	0	0	1	1	0	0	1
7	6	5	4	3	2	1	0

Sign bit — Data bits

+ve → 0

-ve → 1

$2^7 - 1 = 127$

2 | 25 1
2 | 12 0
2 | 6 0
2 | 3 1
1

11001

1 1 1 1

2^3 2^2 2^1 2^0

8 + 4 + 2 + 1 = 7 → 3

= 15

1 1 1 → 7 (2) (10)

1 1 1 → 7 $2^3 - 1 = 7$

1 1 1 1 → 15 $2^4 - 1 = 15$

II) Floating-point
1) float → 4 bytes → 6-7 digits of precision
2) double → 8 bytes → 15-16 " " "

float x;

x = 5.0f/3;

↳ 1.666666

double x;

x = 5.0/3;

↳ 1.6666666666666666

3) 5 (1.6666666
3
20
18
20
18
2

III Character
1) char → 2 bytes

char ch;

ch = 'A';

ch 65 → ASCII (7 bits) → -128 to +127

char ch;

ch = 'A';

ch 65 → Unicode 0 to 65535

IV Boolean
1) boolean

boolean x;

x = true

x = false

x = 5 > 3;

x = 5 > 15

0 1

while(true)
{
}

6) Reference types

String

class String → java.lang
{

}

new

1) String s;

Stack | Heap

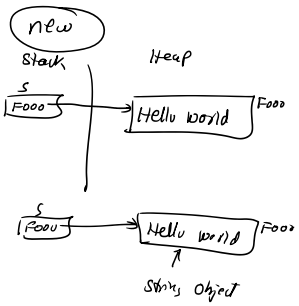
s

① String s;
s = new String("Hello world");

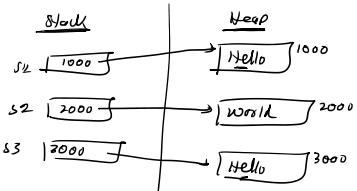
② String s = new String("Hello world");

③ String s;
s = "Hello world";

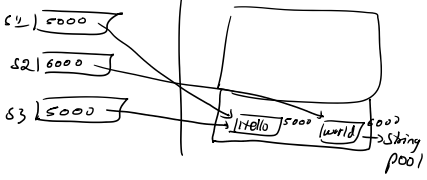
④ String s = "Hello world";



String s1, s2, s3;
s1 = new String("Hello");
s2 = new String("world");
s3 = new String("Hello");



String s1, s2, s3;
s1 = "Hello";
s2 = "world";
s3 = "Hello";



+

String s1, s2, s3;
s1 = "Hello";
s2 = "world";
s3 = s1 + s2;
↳ "Helloworld"

String s;
s = "Hello" + "world";
↳ "Helloworld"

String s;
s = "sum:" + 5+8;
"sum:" + "5"
"sum:5" + "8"
"sum:58"

s = "sum:" + (5+8);
"sum:" + "13"
"sum:13"

s = 2+5+"sum";
"7" + "sum"
"7sum"

s = "Product" * 3;
"Product".repeat(3)

s = "Product:" + 5*3;
"Product:" + "15"
"Product:15"

⑦ Operators

1) Arithmetic + − * / %
2) Relational > < >= <= != ==
3) Logical && || !
4) Increment/Dec. ++ −− x++ ++x

a = 10
b = 20;
c = a++ + b++;
10 20
a 11 b 21 c 30

a = 10
b = 20
c = ++a + ++b
11 21
a 11 b 21 c 32

5) Conditional op ?:

exp1 ? exp2 : exp3
↓ ↓ ↓
Condition True False

c = a>b ? a : b;
a>b ? c=a : c=b

```
a=25        a>b ? S.o.p("Largest :" +a) : S.o.p("Largest :" +b);
b=13         ↓
            25>13
```

6) Assignment op

```
=        x=25         x=x+5 ;
+=                    x+=5
-=                    x-=5
*=                    x*=5
/=
etc
```

7) Bitwise op

```
&        byte a,b,c;
|        a=25          a -> 00011001
^        b=10          b -> 00001010
>>       c=a&b         c -> 00001000
<<       c=a|b         c -> 00011011
```

		&	|	^
0	0	0	0	0
0	1	0	1	1
1	0	0	1	1
1	1	1	1	0

```
b = 2
b = b<<1

a = 00011001
a = a<<2
    01100100

a = a>>2
    00000110
```

⑧

```
class System -> java.lang
{
    public static PrintStream out = new PrintStream();      -> java.io
    public static InputStream in = new InputStream();       -> java.io
}

System.out.println      System.in
System.in

class PrintStream
{
    public void print(String s)
    {
    }
    public void println(String s)
    {
    }
}

class InputStream  -> java.io
{
    public int read()
    {
    }
    public int read(byte[] b)
    {
    }
}

char ch = (char)System.in.read()

System.out.println("Hello world")
```

Scanner -> java.util

```
Scanner scan = new Scanner(System.in);
```

```
byte    nextByte()
short   nextShort()
int     nextInt()
long    nextLong()
float   nextFloat()
double  nextDouble()
String  next()
String  nextLine()

String s = scan.next();
    -> Anil
String s = scan.nextLine();

Anil Kumar ↵
```

Input Buffer

```
| Anil Kumar \n Suresh Sharma \n |

String s1 = scan.next();
String s2 = scan.next();
```

00000010 → 2

00000100 → 4

00001000 → 8

A

↓

65

String s1 = scan.nextLine();
String s2 = scan.nextLine();

Anil kumar \r 25 \n Suresh sharma \r 26 \r

String s1 = scan.nextLine(); ⟵ s1 = "Anil kumar"
int a1 = scan.nextInt(); ⟵ a1 = 25
scan.nextLine();
String s2 = scan.nextLine(); ⟵ s2 = ""
int a2 = scan.nextInt();

02

10 December 2024 09:23

1) Conditional Statements

1) if
2) if..else
3) if..else..if
4) switch

```
if
if(condition)
{
  ≡
}
≡
```

```
if..else
if(condition)
{
  ≡
}
else
{
  ≡
}
```

```
Nested if
if(condition1)
{
  if(condition2)
  {
    ≡ s1        1->T   2->T
  }
  else
  {
    s2          1->T   2->F
  }
}
else
{
  if(condition3)
  {
    s3 -        1=F    3->T
  }
  else
  {
    s4 ->       1->F   3->F
  }
}
```

```
if..else..if
if(c1)                 F   F   T
{
  ≡ s1                         s1
}
else if(c2)            F   T
{
  s2                       s2
}
else if(c3)            T
{                      ↓
  (s3)                 s3  s
}
```

```
op = scan.nextInt();
switch(op)
{
  case 1: c=a+b;
          break;
  case 2: c=a-b;
          break;
  case 3: c=a*b;
          break;
}
```

```
if(op==1)
{
  c=a+b;
}
else if(op==2)
{
  c=a-b;
}
else if(op==3)
{
  c=a*b;
}
```

while
do..while
for

```
while(condition)
{
  ≡
}
```

```
do
{
  ≡
}while(condition);
```

```
          init     cond      inc/dec
for(exp1 ; exp2 ; exp3)
{
}
```

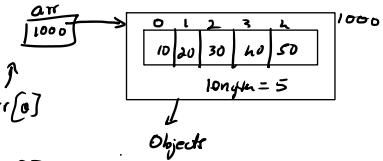
n1	n2
1	10

9%2==0
9%3==0

```
for(n=n1; n<=n2; n++)
{
  boolean prime = true;
  for(int i=2; i<n; i++)
  {
    if(n%i==0)
    {
      prime=false;
      break;
    }
  }
  if(prime)
    s.o.p(n);
}
```

A

Arrays



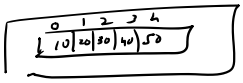
(1)
```
int[] arr;
arr = new int[5]
```

(2) `int[] arr = new int[5];`

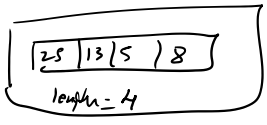
arr[0]
:
arr[4]

↓ ArrayIndexOutOfBoundException.

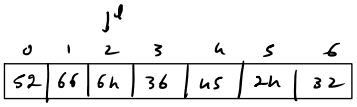
```
Class Main
{
    public static void main(String[] args)
    {
        Scanner scan = new Scanner(System.in);
        int[] arr = new int[5];

        for(int i=0; i<arr.length; i++)
        {
            arr[i] = scan.nextInt();
        }
        for(int i=0; i<arr.length; i++)
            s.o.p(arr[i]);

        for(int data : arr)
        {
            s.o.p(data);
        }
    }
}
```




```
int n = scan.nextInt();
int[] arr = new int[n];
for(int i=0; i<arr.length; i++)
    arr[i] = scan.nextInt();
int sm = 0;
for(int data : arr)
{
    sm += data;
}
s.o.p(sm);
```



sm = 0 + 25 data = 25
= 25 + 13 13
= 38 + 5 5
= 43 + 8 8
= 51




```
boolean leader(int[] arr, int i)
{
    for(int j=i+1; j<arr.length; j++)
    {
        if(arr[i] <= arr[j])
        {
            return false;
        }
    }
    return true;
}
```

leader(arr, 2)

main

```
int sm=0;
for(int i=0; i<arr.length; i++)
{
    if(leader(arr, i))
    {
        sm = sm + arr[i];
    }
}
s.o.p(sm);
```

```
boolean unique(int[] arr, int data)
{
    int count = 0;
    for(int i=0; i< arr.length; i++)
    {
        if(arr[i] == data)
        {
            count++;
            if(count > 1)
                return false;
        }
    }
    return true;
}
```

0	1	2	3	4	5	6
13	45	26	13	83	26	83

count = 0 1 data = 45

main

```
int sm = 0;
for(int i = 0; i < arr.length; i++)
{
    if(unique(arr, arr[i]))
    {
        sm = sm + arr[i];
    }
}
S.O.P(sm);
```

0	1	2	3	4	5	6
25	13	43	5	56	34	12

```
L = arr[0];
for(int data: arr)
{
    if(data > L)
        L = data;
}
return L;
```

data = 25 L = ~~25~~ ~~43~~ 56

0	1	2	3	4	5	6	7	8
90	80	70	60	50	40	30	20	10

i	j
0	8
1	7
2	6
3	5
4	4

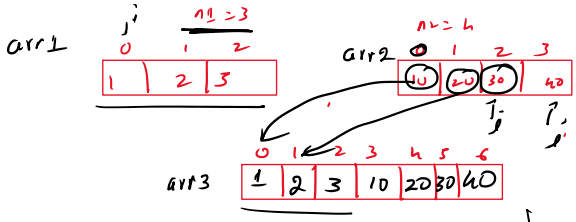
```
for( i=0, j= arr.length-1; i < j ; i++, j--)
{
    tmp = arr[i];
    arr[i] = arr[j];
    arr[j] = tmp;
}
```

arr1

0	1	2
1	2	3

arr2

0	1	2	3
10	20	30	40




```
int[] arr3 = new int[n1 + n2];
for(int i = 0; i<n1; i++)
{
    arr3[i] = arr1[i];
}
```

```
for(i=0; i<n2; i++)
    arr3[n1 + i] = arr2[i];
```

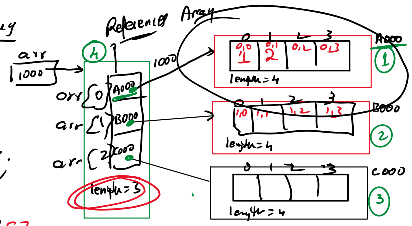
```
import java.util.*;
class Main
{
    p s v main(String[] args)
    {
        Scanner scan = new Scanner(System.in);
        int n1 = scan.nextInt();
        int arr1[] = new int[n1];
        for(int i=0; i<n1; i++)
            arr1[i] = scan.nextInt();

        int n2 = scan.nextInt();
        int[] arr2 = new int[n2];
        for(int i=0; i<n2; i++)
            arr2[i] = scan.nextInt();

        int[] arr3 = concatArray(arr1, arr2);
        for(int data: arr3)
            S.O.print(data + " ");
    }
}
```

```
public static int[] concatArray(int[] a1, int[] a2)
{
    int[] a3 = new int[a1.length + a2.length];

    for(int i=0; i< a1.length; i++)
        a3[i] = a1[i];

    for(int i=0; i<a2.length; i++)
        a3[a1.length+i] = a2[i];

    return a3;
}
}
```

2 Dim Array

(1)
```
int[][] arr;
arr = new int[3][4];
```

(2)
```
int[][] arr = new int[3][4];
```



arr[0][0] = 1
arr[0][1] = 2

```
for(int i=0; i<arr.length; i++)
{
    for(int j=0; j<arr[i].length; j++)
    {
        arr[i][j] = scan.nextInt();
    }
}
```

i	j
0	0
	1
	2
	3
	4
1	0
	1
	2
	3
	4
2	

	0	1	2	3
0	1	2	3	4
1	5	6	7	8

Bubble Sort

j		1st pass	2nd pass	3rd pass	4th pass
0	25	13	13	13	5
1	13	25	19	5	13
2	43	19	5	19	19
3	19	5	25	25	25
4	5	43	43	43	43

```
for(int i=1; i<arr.length; i++)
{
    for(int j=0; j<arr.length-1; j++)
    {
        if(arr[j] > arr[j+1])
        {
            tmp = arr[j];
            arr[j] = arr[j+1];
            arr[j+1] = tmp;
        }
    }
}
```

Arrays.sort(arr);

```
Class Main
{
    public static void main(String[] args)
    {
        int m = scan.nextInt();
        int n = scan.nextInt();
        int[][] arr = new int[m][n];
        for(int i=0; i<m; i++)
        {
            for(int j=0; j<n; j++)
            {
                arr[i][j] = scan.nextInt();
            }
        }
        for(int i=0; i<m; i++)
        {
            bubbleSort(arr[i]);
        }
        for(int i=0; i<m; i++)
        {
            for(int j=0; j<n; j++)
            {
                s.o.print(arr[i][j] + " ");
            }
            s.o.println("");
        }
    }
    public static void bubbleSort(int[] a)
    {
        int tmp;
        for(int i=1; i<a.length; i++)
        {
            for(int j=0; j<a.length-1; j++)
            {
                if(a[j] > a[j+1])
                {
                    tmp = a[j];
                    a[j] = a[j+1];
                    a[j+1] = tmp;
                }
            }
        }
    }
}
```

Arrays.sort(a);