Loops

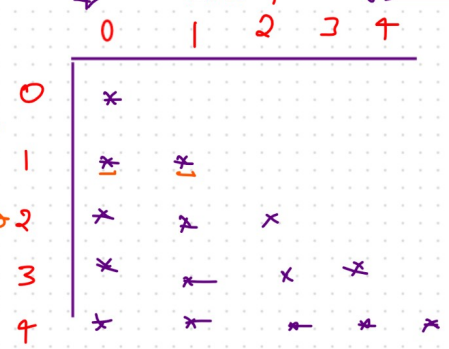
Single, double, three

① Right angled Triangle

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

i = outer loop, j = column / inner loop

j <= i



i = 2 j = 0

0 <= 1
1 <= 1
2 <= 1

```
int n =10;
for(int i=0;i<n;i++) {
  for(int j=0;j<=i;j++) {
    System.out.print("*");
  }
  System.out.println("");
}
```

② Inverted Right angle triangle

i	j
0	5
1	4
2	3
3	2
4	1

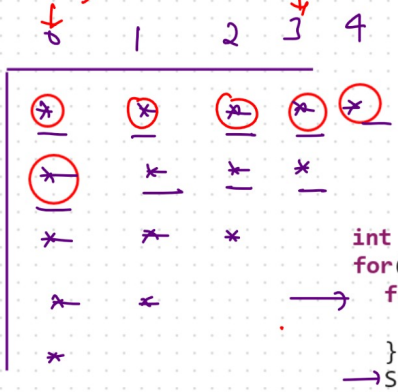
n = 5

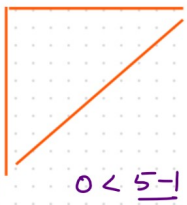
n - i

i =

n - i = 4
n - i = 3
n - i = 2
n - i = 1

j < n - i





i = 0 j = 0

0 < 5 - 1
0 < 4
1 < 4
2 < 4
3 < 4
4 < 4

```
int n = 5;
for(int i = 0;i<n;i++) {
  for(int j=0;j<n-i;j++) {
    System.out.print("*");
  }
  System.out.println();
}
```

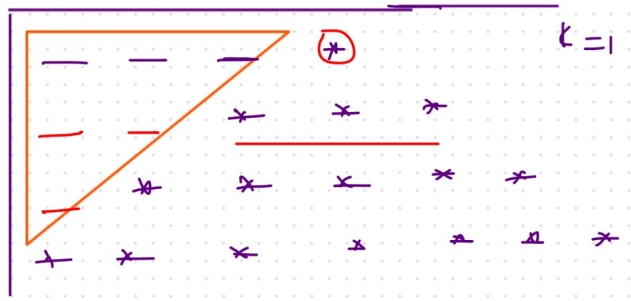
③ Pyramid (3 loops)

n = 4

rows = 4

columns = 7

i	j	k
1	3	1
2	2	3
3	1	5
4	0	7



1 <= 3
2 <= 5
3 <= 7

k = 1 ; k <= (2 x i - 1) ; k++

* * *

n - i = (3)

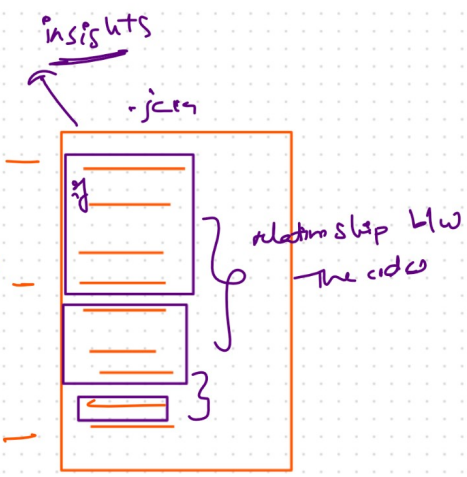
(j = n - i (3) ; j > 0 ; j--)

j = 2 ; j > 0 ; j--

2 1 0

(initial ; cond ; updation)

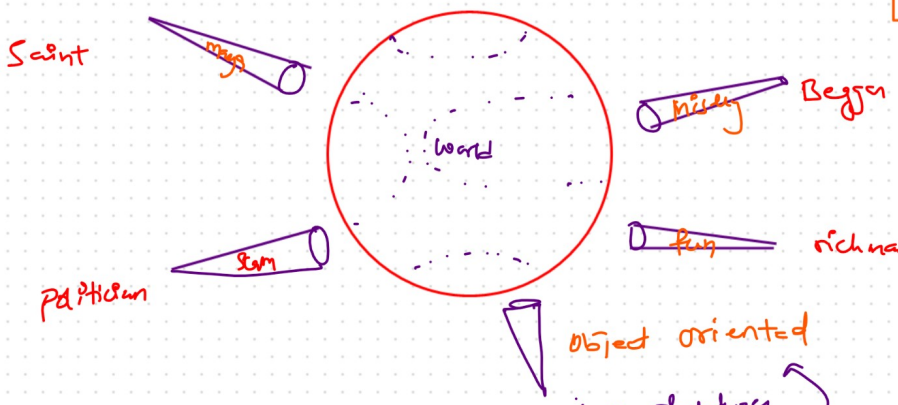
k=1 ; k<=(2*i+1) ; k++



Object Orientation

Functional programming

Orientation : perspective (the way of looking)



Rules

1) The world is collection of objects
2) Every object is useful object, and no object is useless
3) every object is in constant interaction with other object
4) every object belongs to a type
5) Type doesn't exists in reality, only the object of that type exists

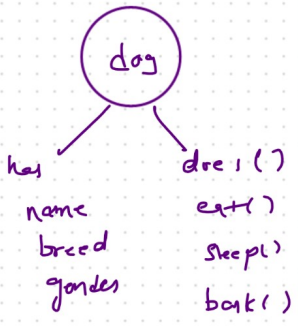
In reality [type technically class]

type | class | blueprint | template

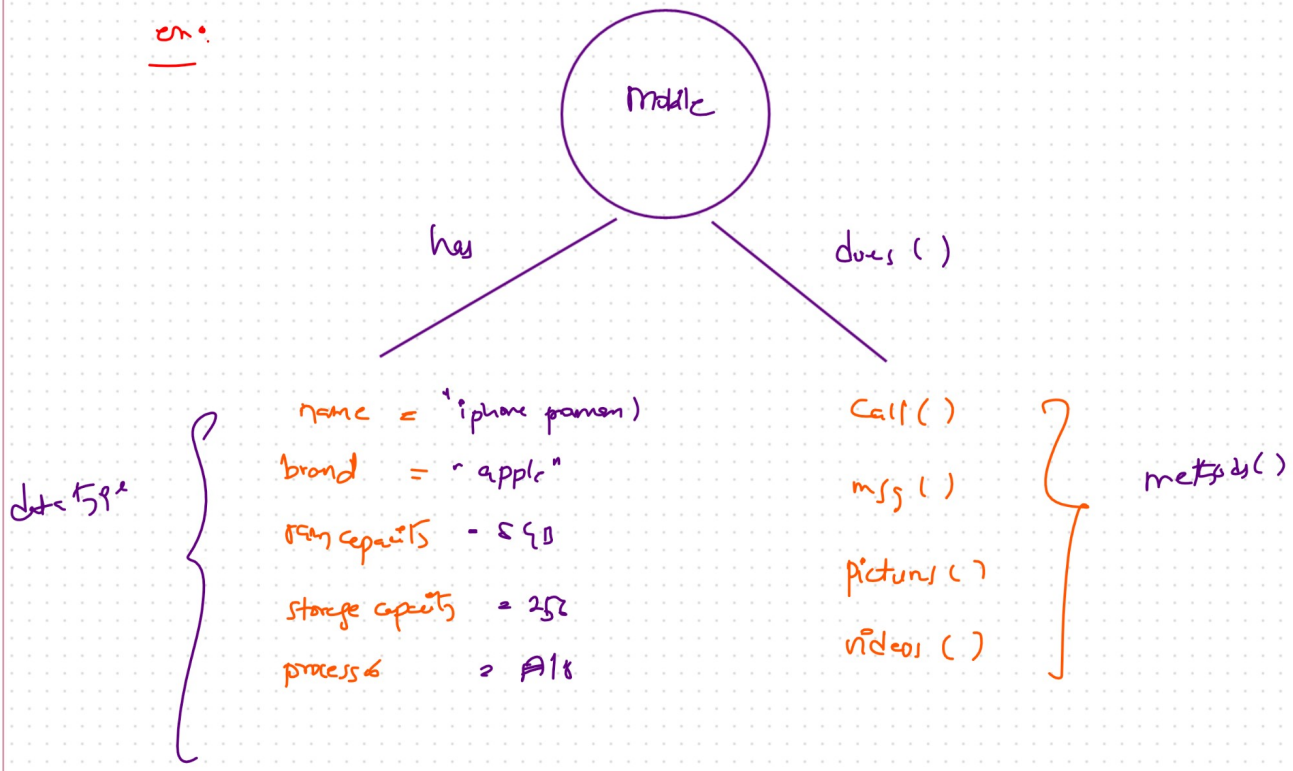
6) Every object has 2 parts i.e has part and does part

Has part : It represents the properties of the object
[by representing data type]

Does part : It represents the behaviour of the object
[represents using methods]



ex:


Mobile
has
does ( )
name = "iphone pamen)
brand = "apple"
ram capacity - 8 GB
storage capacity = 256
process = A18
data type
Call ( )
msg ( )
pictures ( )
videos ( )
methods ( )