Lec-36

10/24/24 7:18 PM

Inheritance

OOPS

(i) Inheritance
(ii) Stack use dynamic
(iii) String combat Address
(iv) Concat+o

Polygon

Inheritance | Membership

parent
class A E

Child
class B E
fun fun

Classname var name = new classname ();

(i) P obj	new P ()
P obj	new C ()
C obj	new P () (iii)
C obj	new C ()

List ll = new ArrayList

(2k) → Runtime

P obj = new C();

d1 d10
d = 2
d2 = 20
P
C
2k

d d1
fun fun1
P
d d2
fun funC
C

```
// Case-2
        P obj = new C();
        System.out.println(obj.d);
        System.out.println(obj.d1);
```

d1 d10
d = 20
d2 = 2
2k

d d1
fun fun1
P
d d2
fun funC
C

```
P obj = new C();
        System.out.println(obj.d);
        System.out.println(obj.d1);
        System.out.println(((C) obj).d2);
```

(2k)

```
P obj = new C();
System.out.println(obj.d);
System.out.println(obj.d1);
System.out.println(((C) obj).d2);
System.out.println(((C) obj).d);

obj.fun();//P
obj.fun1();
((C) obj).fun2();
((C) obj).fun();//C
```

d = 1
d1 = 10
d = 2
d2 = 20
2k

d d1
fun fun1
P
d d2
fun funC
C

List ll = new ArrayList<>()
P obj = new C();

method overriding
2 different class ही होतात
जर त्या ही एका child
class पुरतीच होत
parent class ला method
name सारखे तरी त्या
दोघ्ज्या scope होतात
दोघ्ज्या implementation
वेगळा होतो

obj = new A()
obj.fun

A → B → C → D → E
D E
C D E
B C D E
A B C D E

B
A
C

method Overriding

A → B → C

import Lec34.Stack;

```
public class DynamicSatck extends Stack {

    @Override
    public void push(int item) {
    }

    public static void main(String[] args) {
        DynamicSatck ds = new DynamicSatck();
    }
}
```

→ Stack (S)

idx = -1

10	20	30	40	50
0	1	2	3	4

arr

2x

ds.push(10)

10	20	30	40	50	

P.push

idx++
arr[idx] = item

arr

10	20	30	40	50
0	1	2	3	4

```
        if (isFull()) {
            int[] newarr = new int[2 * arr.length];
            for (int i = 0; i < arr.length; i++) {
                newarr[i] = arr[i];
            }
            arr = newarr;
        }
```

10	20	30	40	50	60				

arr

```
@Override
    public void push(int item) {
        if (isFull()) {
            int[] newarr = new int[2 * arr.length];
            for (int i = 0; i < arr.length; i++) {
                newarr[i] = arr[i];
            }
            arr = newarr;
            super.push(item);
    }

    public static void main(String[] args) {
        DynamicSatck ds = new DynamicSatck();
        ds.push(10);
        ds.push(20);
        ds.push(30);
        ds.push(40);
        ds.push(50);
        ds.push(60);
    }
```

idx = n → full

10	20	30	40	50
0	1	2	3	4

idx = 0

arr

10	20	30	40	50	60					

O(1)

```
    public void push(int item) throws Exception {
        if (isFull()) {
            throw new Exception("bklol stack full h");
        }
        this.arr[this.idx] = item;
        this.idx++;// idx++;(
```

(i, j)

f: i = (j+1) % 5 = 2
f: i = (2+1) % 5 = 3
f: i = (3+1) % 5 = 4
f: i = (4+1) % 5 = 0
f: i = (0+1) % 5 = 1

60	70	30	40	50

30	40	50	60	70				

idx = (idx+1) % arr.length

memory[i] = arr[idx]