STACK

and

Queues

→ usage

question

Implementation

- Normal
- Dynamic

Queues

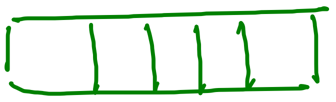
usage

questions

implementation

- Normal
- Dynamic

Array
ArrayList] →



- ✓ get → O(1)
- ✓ set → O(1)
- ✓ remove → O(n)
- ✓ add At → O(n)
  - last → O(1)

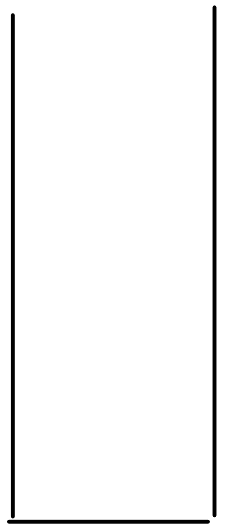
push( )

pop( )

peek( )

size

isEmpty



get → peek( )

set] → peek can be changed

pop ~~remove~~] → peek remove

push ~~add~~] → addition on peek

OOPS:

- → classes and objects.
- → data member + member function
- → constructor.
- → this operation.
- → Access modifier
- → Static / Non. static
- → String args.

user defined data type.

Human

- → Name] → String.
- → age] → integer
- → ht] → double
- → display] → print detail

Name, age, ht: data member

display: member function

public static.

Function] having no return type] → Automatically →

object]

Initialise] → default constructor, parameterised constructor, copy constructor.

construct

✗ → JVM → ~~default~~ call con

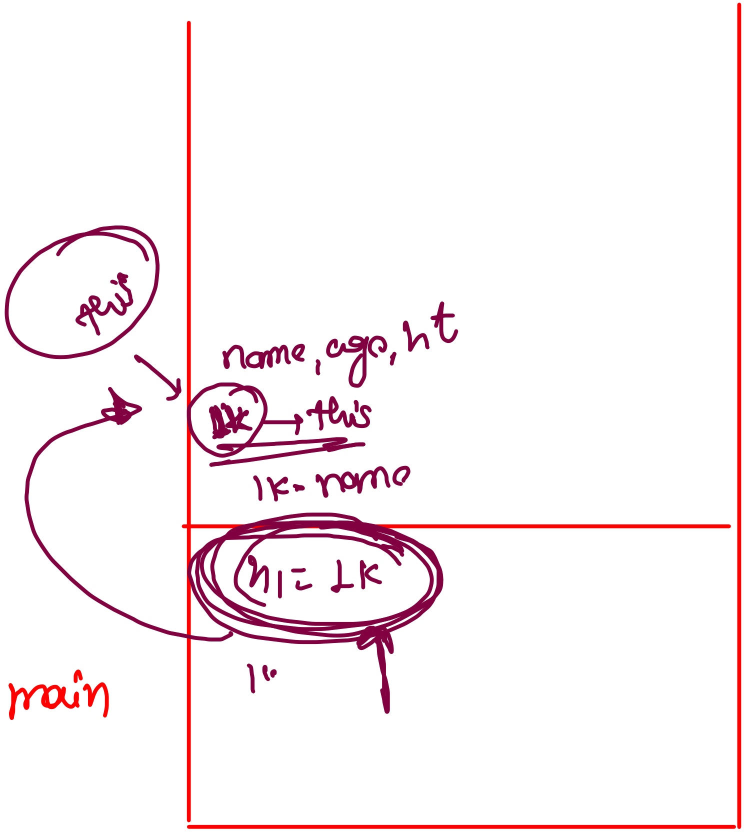
✓ → default, parameterised

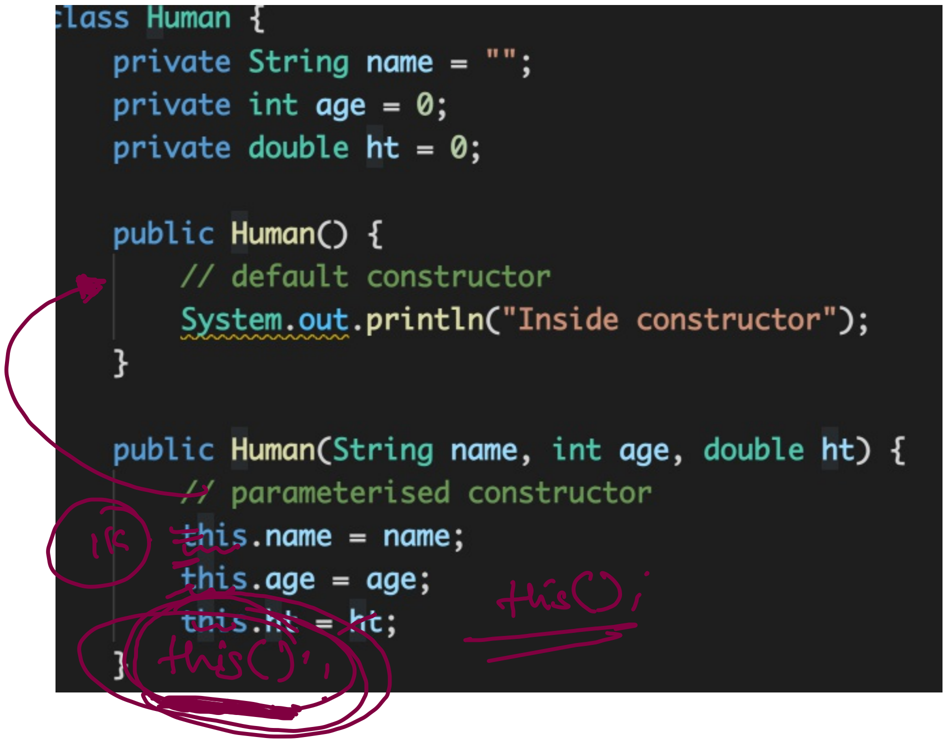
→ destructor. in C++

this - self referenced pointer.

- access modifier
- this
- copy construction

→ Package





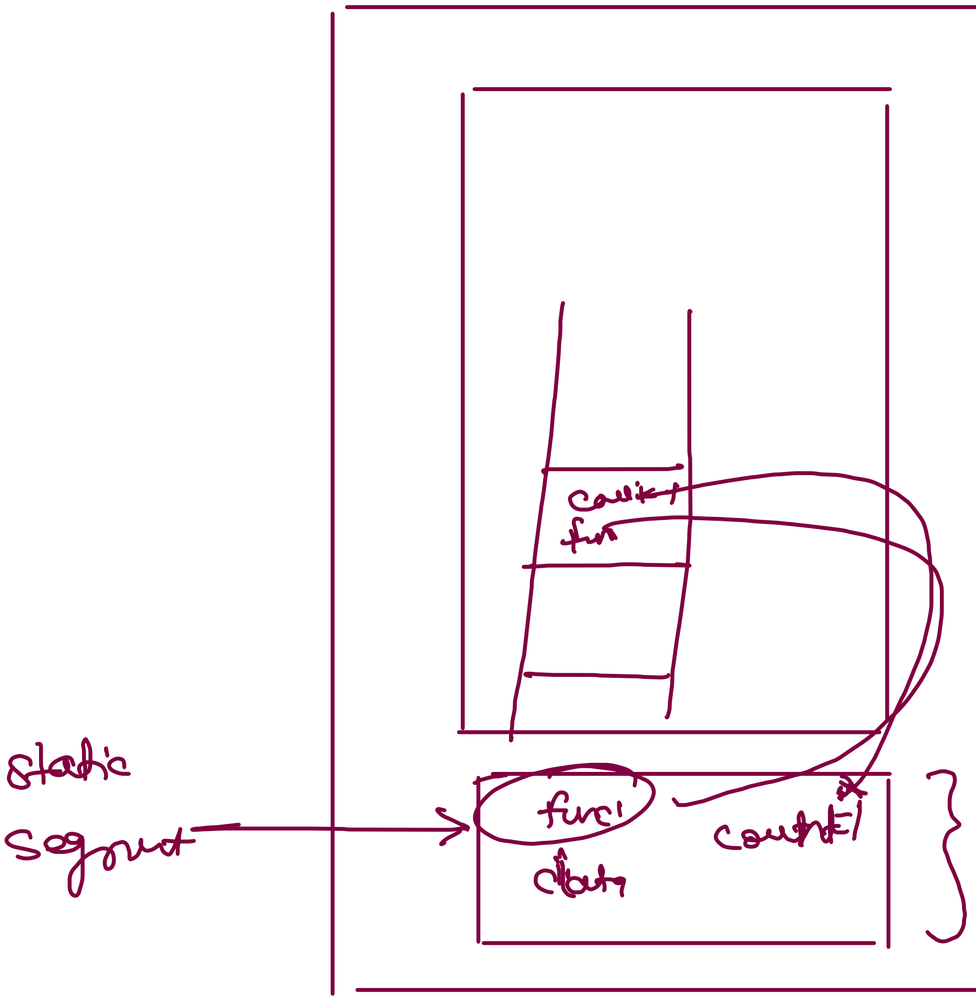
```
class Human {
    private String name = "";
    private int age = 0;
    private double ht = 0;

    public Human() {
        // default constructor
        System.out.println("Inside constructor");
    }

    public Human(String name, int age, double ht) {
        // parameterised constructor
        this.name = name;
        this.age = age;
        this.ht = ht;
    }
```

first

Static memory allocation



constructor - limit of stack → size > limit → stack overflow.

size < 0 → stack underflow

✓push → addlast

✓pop → remove last

✓peek → get last

✓size → overall size?

display → stack printing.