METHODS IN JAVA & Fundamental of Stack and Heap area

Date 21/10/2022

⇒ **Methods:-**

Wherever there is a ~~task~~ task/activity/some group of work which is related to each others if you are performing then the method comes in the picture.

⇒ In other language method is called as function also and there have many ways/purpose to write a function. But in Java only one way/purpose we can write method.

Methods ⇒ any Task/activity

What must have? ⇒ ① name, ② Input (Parameter), ③ Body, ④ return type

```
ex:- return type name (parameter){

        Activity/Body

     }
```

⊛ 04 different ways we can write a method in java.

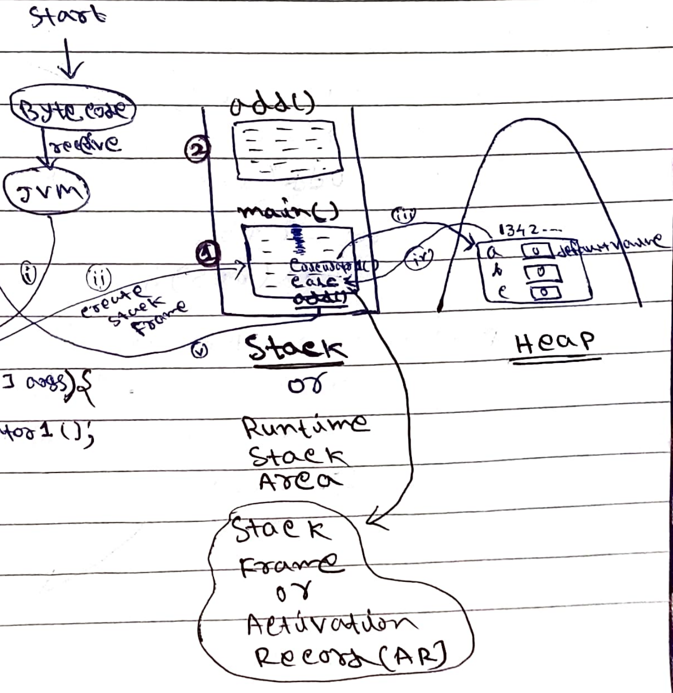
⊛ Memory Management for methods:-

→ Whenever we write a method, when we execute then JVM take the method and put it into Stack Area.

→ To execute the task/activity which is there present inside your method on your stack area one record will be created which record is called Activation Record of any method or Stack Frame.

⊛ ⇒ e.g #1 (type 1)

```
② Class Calculator1{
       int a,b,c;
       Void add(){
          a=10;
          b=20;
          c=a+b;
          S.o.Pln(c);
       }
① Public class LaunchCalc1{
    Public static void main(S[] args){
    Calculator1 calc = new Calculator1();
    calc.add();
    }
}
```

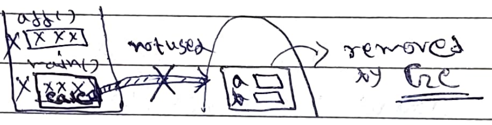


→ After finish of task of any method the Stack frame or AR will also remove from stack area.

→ When Stack frame removed the reference variable (calc) will also remove. So, in Heap area there is nothing which indicates those instance variables. So, those also removed by Garbage collection process automatically by JVM.

⇒ Garbage Collection-

Java application obtain objects in memory as needed. It is the task of garbage collection (GC) in the JVM to automatically determine what memory is no longer being used by a Java application and to recycle this memory for other uses. So, continuously JVM searching for which memory not used. find it and remove it.



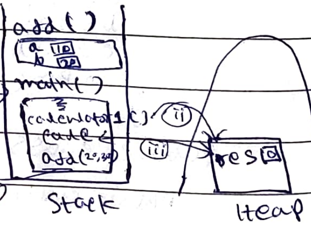
e. g#2: We can also take input from Parameter of any method.

(i) If there is a need in java you can write such a method which is accepting in parameters.

(ii) as a caller who call the method compulsorily whatever the input is accepting you have to pass it.

⇒ type2 ⇒

```
Class Calculator2{                       → Parameter
    int res;
  void add(int a, int b){
      res = a+b;   // int res = a+b;
      S.O.Pl(res);
  }
}
Public class Approch2 {
Public static void main(String[] args){
   Calculator2 calc = new Calculator();
   calc.add(20, 30);
                  → arguments
 }
}
```



```
e.g#3 >  class  Calculator3 {
type3 =>    int  a, b, res;
            int add() {
               a = 10;
               b = 20;
               res = a+b;
               return res;  // give only the integer value
            }
         }
         public class Approach3 {
            P.S.V.m (S[] args) {
            Calculator3  calc = new  Calculator3();
            //calc . add();
            int receiver = calc.add();   } receiving the integer value
            S.o.pln (receiver);          } of res into a variable (receiver)
            }                            ⊛ we can do any other
         }                               operation ~~instead~~ of printing

e.g#4 >  class Calculator4 {
type4 =>   int  res;
           int  add (int a, int b) {
              res = a+b;
              return res;
           }
         }
          Public class  Approach 4 {
           P.S. v.m (S[] args) {
           Calculator 4  calc = new Calculator4();
           int   receiver = calc. add (10, 20);
           S.o.pln (receiver);
         }  }
```

Code Snippets

Q1)
```
public class Test{
    PSvm(S[] args){
    int x=?;
    switch(x){
    default: S.o.pln("default")
    case 0: S.o.Pln("0");
        break;
    case 1= S.o.Pln("1");
    case 2 = S.o.Pln("2");
}}}
```

option
x=0 => output = 0
x=1 => 1, 2,
x=2 => 2
x=3 => default , 0
(Ans)

NOTE: replace x with 0,1,2,3 and predict the output.

Q)
```
Boolean b1 = true;   // wrapper class
boolean b2 = false;  // primitive.
boolean b3 = true;   // primitive
if ((b1 & b2) | (b2 & b3) & b3)  // false (if(false)|(false) & true),
                                     if (false & true),
                                     if (false)
    S.o.Pln ("alpha");
if (b1 = false) | (b1 & b3) | (b1 | b2))  // if (false |(false & true)
                                            |(false | false))
    S.o.Pln ("beta");
                                     ∴ if(false | false | false)
                                       if (false)
```
=> what is the result?

a) beta, b) alpha, c) alpha beta, d) CE, [✓ e)] No output

F) An exception is thrown at runtime.

Ans: No output.

Q)
```
Class Maybe{
  PS vm(S[] a){
    boolean b1 = true;
    boolean b2 = false;
    S.o.Pl(!false ^ false);  // true ^ false => true
    S.o.Pl("  " +(!b1 & (b2 = true)));  // false & true => false
    S.o.Pl("  " + (b2 ^ b1));  // true ^ true => false
  }
}
```

Which is true?

A. line 5 = true, B) line 5 = false, c) line 6 = true, d) line 6 = false

e) line 7 = true, f) line 8 = false.

Ans: A. line 5 is true

NOTE: exor => both operands same means false, otherwise true. eg: true ^ true => false, true ^ false => true } | false ^ false = false, false ^ true = true }

Q) class Hexy{
 PSVm(S[] a){
 Integer i=42; // ~~Integer~~ ⇒ wrapperclass of primitive datatype of integer.
 String s = (i<40)?"life":(i>50)?"universe":"everything";
 S.o.Pln(s);
}}

//(42<40)?true:(42>50)?true:
"everything";

→ what is the result?
A) null, B) life, C) universe, D) everything, E) CE, F) An exception is thrown at runtime

Ans ⇒ D) everything.

Q) class Foozit{
 P.S.V.m(S[] args){
 Integer x = 0;
 Integer y = 0;
 for(Short z=0; z<5; z++):
 if ((++x > 2) || (++y > 2))
 x++;
 S.O.Pln(x+" "+y);
}}

(I)
++x
// x = 0 , 1
// y = 0 , 1 ++y
// z = 0, 0<5 (true)
// if (false || false) ↓X

(II)
//∴ x = 1, 2
y = 1, 2
z = 1 1<5 (true)
if (false || false) ↓X

→ What is the result? a) 5 1, b) 5 2, c) 5 3, d) 8 1, e) 8 2,
F) 8 3, G) 10 2, H) 10 3, I) CE, J) some problem created by JVM during execution

(III) // x = 2, 3, $\textcircled{4}$ ↓++x
y = 2, 3
z = 2; 2<5 (true)
if (true) ↓X
print x = $\textcircled{3}$
and x is now = 4

(IV) // x = 4, 5, 6 ↓++x
y = 2
z = 3; 3<5 (true)
∴ if (true) ✓↓
∴ print x = 5
and x = $\textcircled{6}$

(V) x = 6, 7, 8 ↓
y = 2
z = 4; 4<5(t)
∴ if (true
print = 6
x = $\textcircled{8}$

(VI) $\boxed{x = 8 \quad y = 2}$ ✓
z = 5 5<5 (false) ↓$\textcircled{X}$

Print
x = = 8
y = 2 } Ans