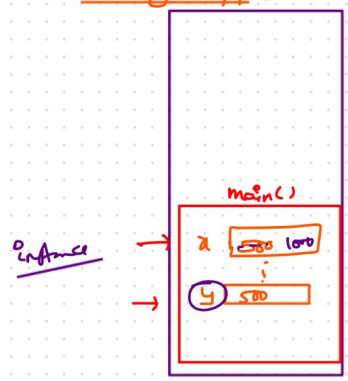
Value type Assignment & Reference type Assignment

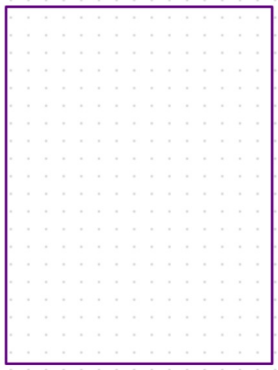
✓ Value type Assignments

```
public static void main(String[] args) {
    int x = 500;
    int y;
    y=x;
    System.out.println(x);
    System.out.println(y);
    x = 1000;
    System.out.println(x);
    System.out.println(y);
}
```

Memory mapping



Stack



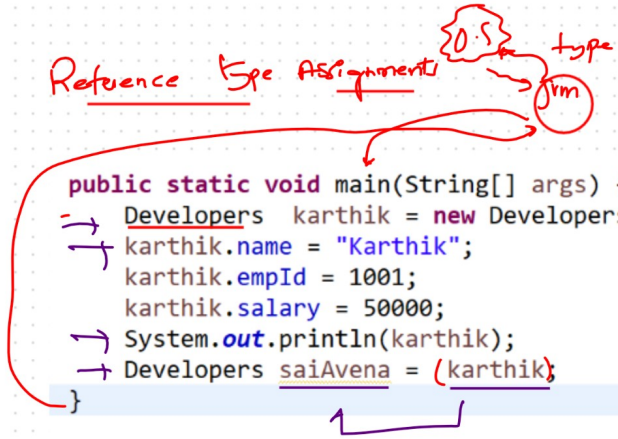
Heap

Objects
String
Array

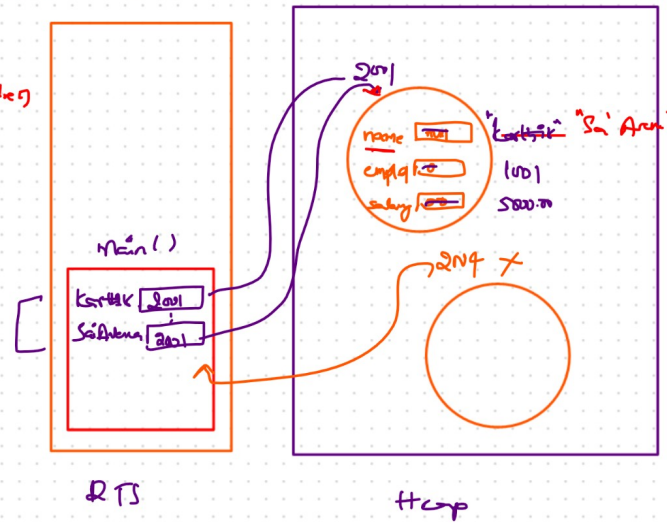
Notes

* In value type assignment, variables always the new set of value / copy of variable value altogether

Reference type Assignments



(Create a copy of value)



RTS

Heap

Note:

→ The value assignment will be address of the reference type i.e address of the reference variable is stored as a value to it

Types of Variables

There are 3 variables

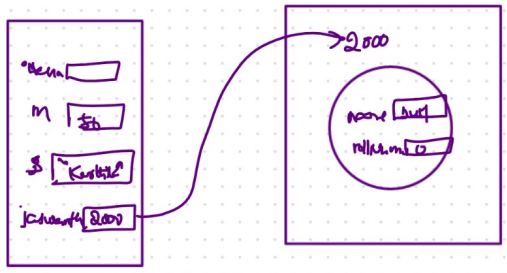
1) local — a.k.a [automatic variables]
2) Instance — [fields]
3) Static — [class variables]

Local Variable

1) Local variables are declared inside the method
2) memory is allocated inside the stack
3) No default values will be assigned by the JVM
4) We cannot use local variables without initializing the value
5) Memory will be deallocated while the control comes out of the method

Instance Variable

1) Instance variables are declared outside the method & inside the class
2) memory is allocated inside the Heap Area
3) Default values will be assigned by the JVM (while allocating memory)

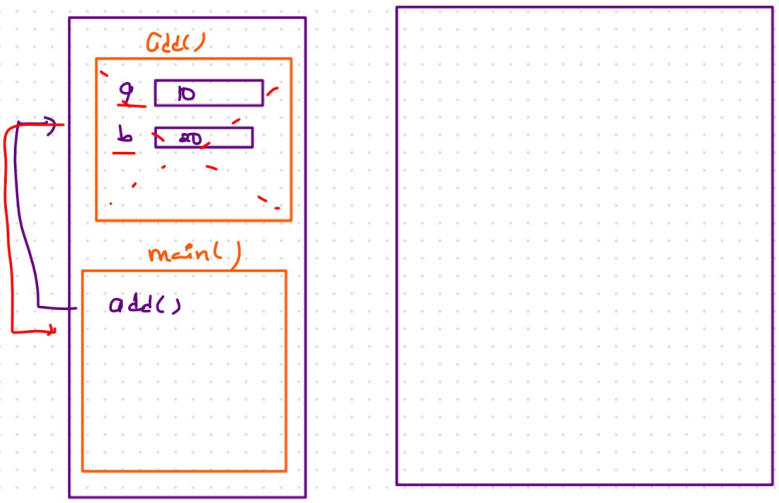


4) We can use instance variables without initializing any value
4) Default values for all data types
5) Memory will be deallocated by the garbage collector, when there is no reference to the object

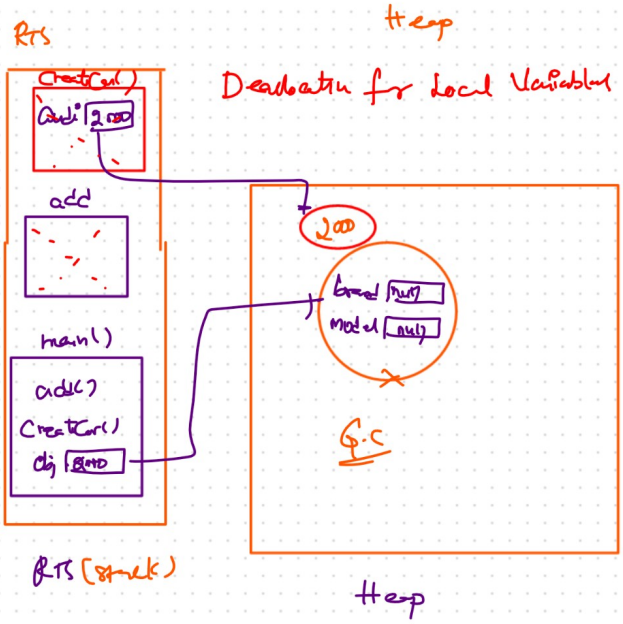
```java
static void add() {
    int a =10;
    int b = 20;
    System.out.println(a+b);
}

public static void main(String[] args) {
    add();
}
```

Local Variable Case



GC



Deallocation for local Variables

```java
static void createCar() {
    Car audi = new Car();
}

public static void main(String[] args) {
    add();
    createCar();
}
```

Instance Variable Case

Methods

Structure (Signature)

modifier return-type name(parameters) {exception list}

body

}

void — Nothing return

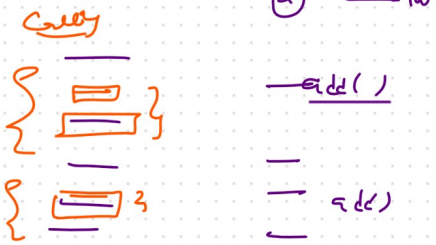
type — int

Example

```
int add(int a, int b) {
    int result = a+b;
    return result;
}
```

Common

do-to



Note:

1) method is a set of instructions, used to execute multiple time as when required

2) methods are used when we have common logic to call multiple times

3) methods help in code re-usability & also easy to maintain