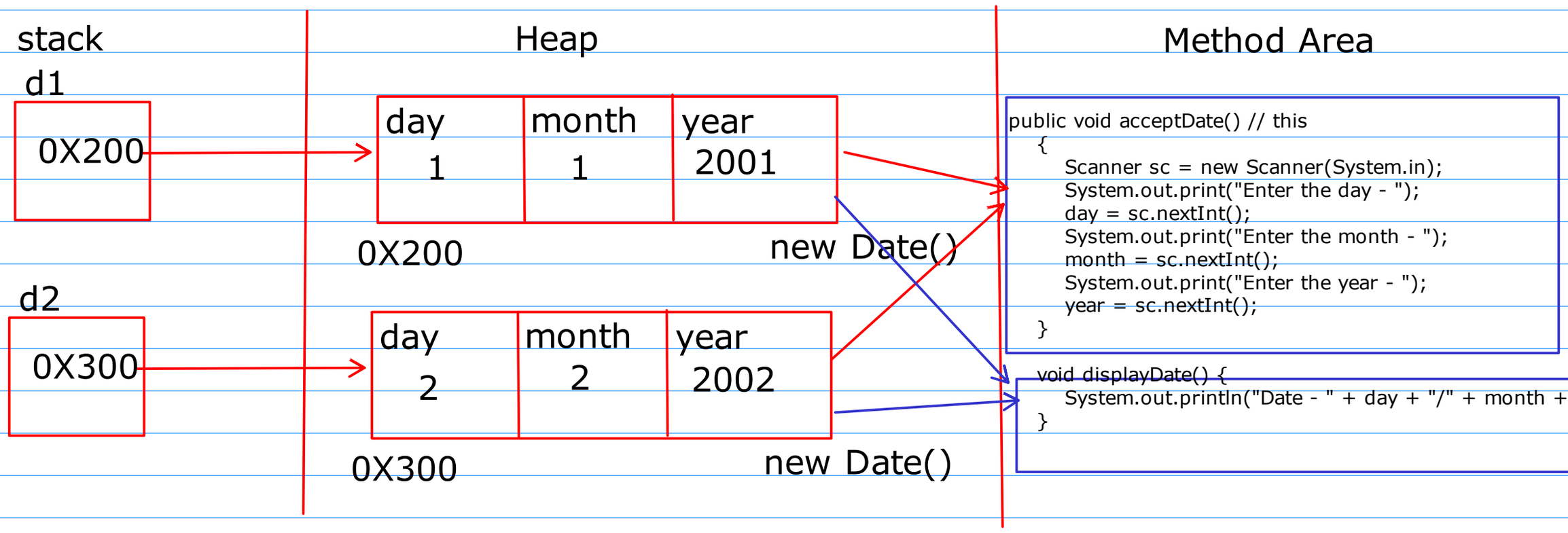
this

all the non static methods in the class gets this reference internally
this reference stores the address of the current calling object

```
Date d1 = new Date();

Date d2 = new Date();
```



d1.day
d1.month
d1.year

d2.day
d2.month
d2.year;

d1
FAR->acceptDate()

```
Date this = d1

this.day
//d1.day
```

d1
FAR->displayDate()

```
Date this = d1
```

d2
FAR->acceptDate()

```
Date this = d2

this.day
//d2.day
```

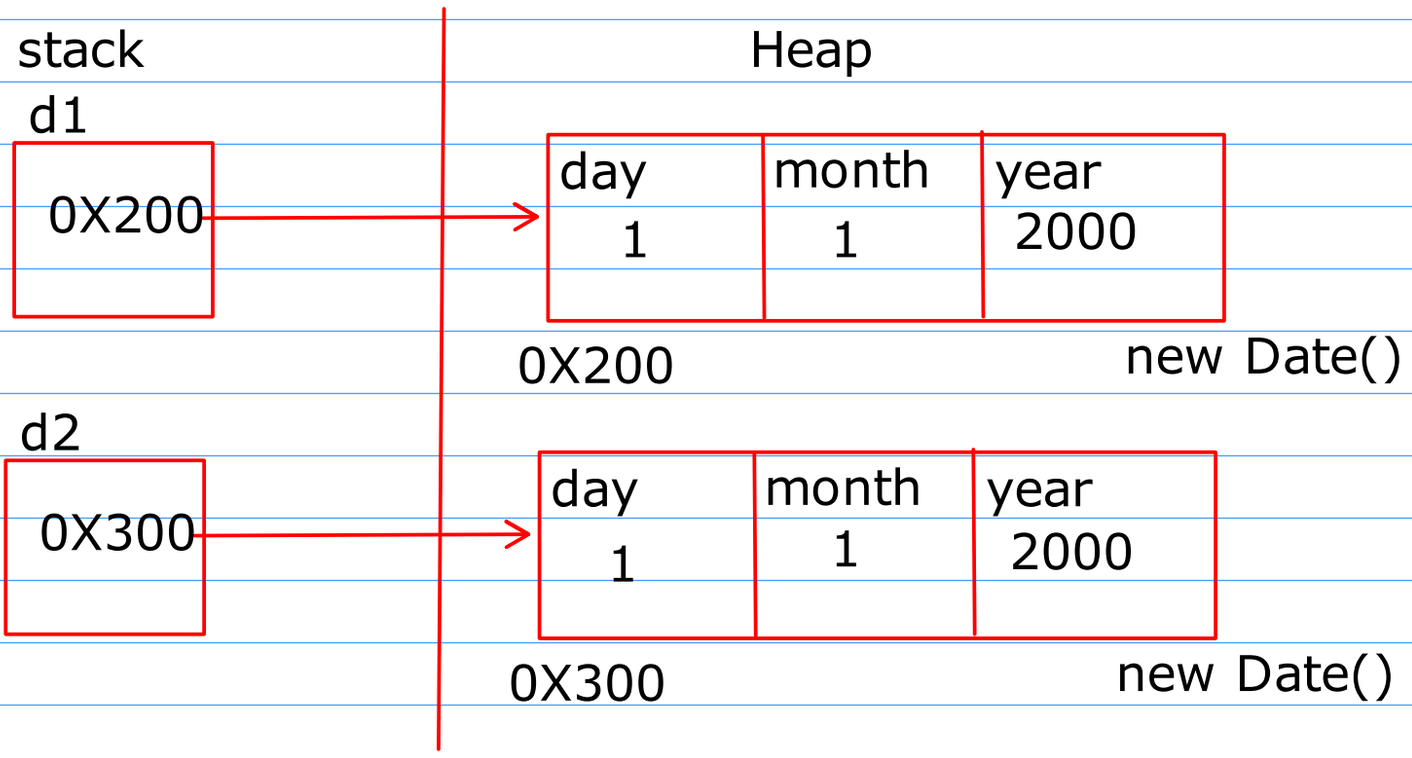
d2
FAR->displayDate()

```
Date this = d2
```

this reference

- It is a reference that is internally passed to the non static method of the class
- It points to the current calling object
- this referece is of the same type as that of the class
- using this reference is optional.

#Types of methods
1. Constructor
2. Setter
3. Getter
4. Facilitator



Constructor
- It is a special method of the class
- Why it is so special
1. The name of the ctor(method) is same as that of class name
2. The ctor does not have any return type
3. It gets called automatically when the object is created.

- ctor's job is to initialize the state of an object

Types of Ctor
- Two types of Ctor
1. Default/Paramaterless
2. Parameterized

Constructor Chaining

```
public Date() {  2 usages   new *
    this( day: 1,  month: 1,  year: 2001); // this statement
}

public Date(int day, int month, int year) {  3 usages   new *
    this.day = day;
    this.month = month;
    this.year = year;
}
```

Constructor Chaining

Calling another ctor from the current ctor is called as ctor chaining
this() should be the first statement inside the ctor body

Setter

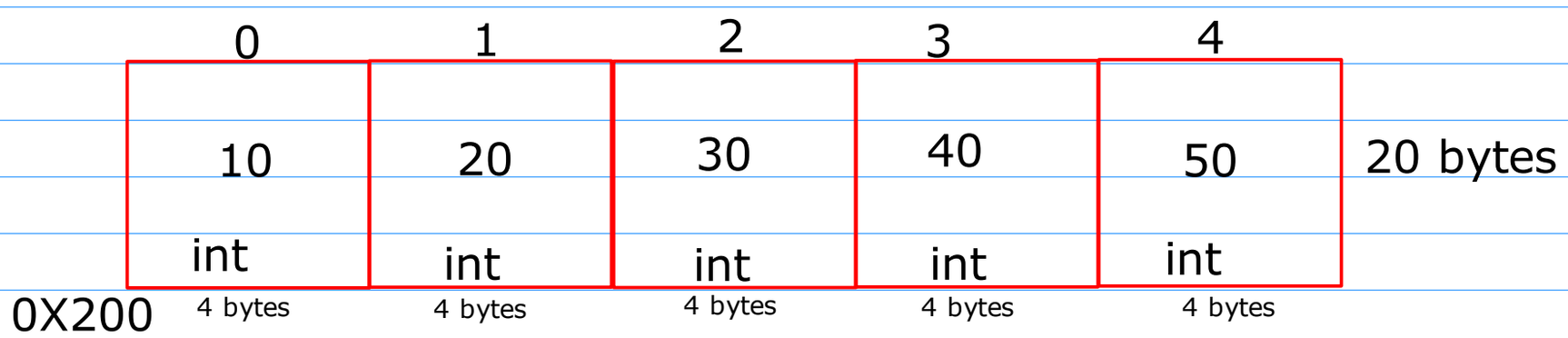
- A method used to change/assign the value to the single field of the class
- Setter should compulsary accept the value through its parameter
- Setter does not return any value
- It is a convention to start the name of the settter with set followed by the name of the Field

Getter

- A method used to get the value of the single field of the class
- Getter should compulsary return the value of the single field
- It is a convention to start the name of the gettter with get followed by the name of the Field

Array

- It is a data structure that is ued to store similar type of elements in contigious memory location
- int arr[5]

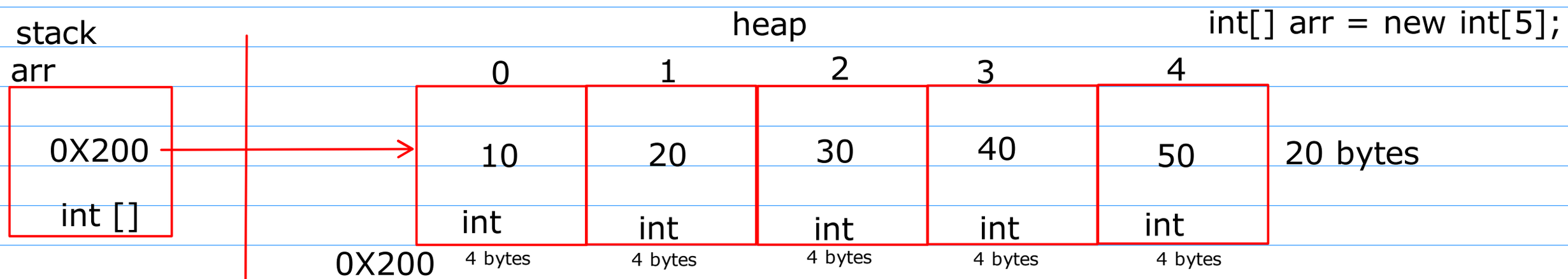


0X200[0] = 10;
arr[0] = 10

0X200[1]=20
arr[1]=20

```
print(arr[0])
print(arr[1])
print(arr[2])
print(arr[3])
print(arr[4])
```

```
for(int i=0;i<5;i++){
    print(arr[i])
}
```



0X200[0] = 10
arr[0] = 10

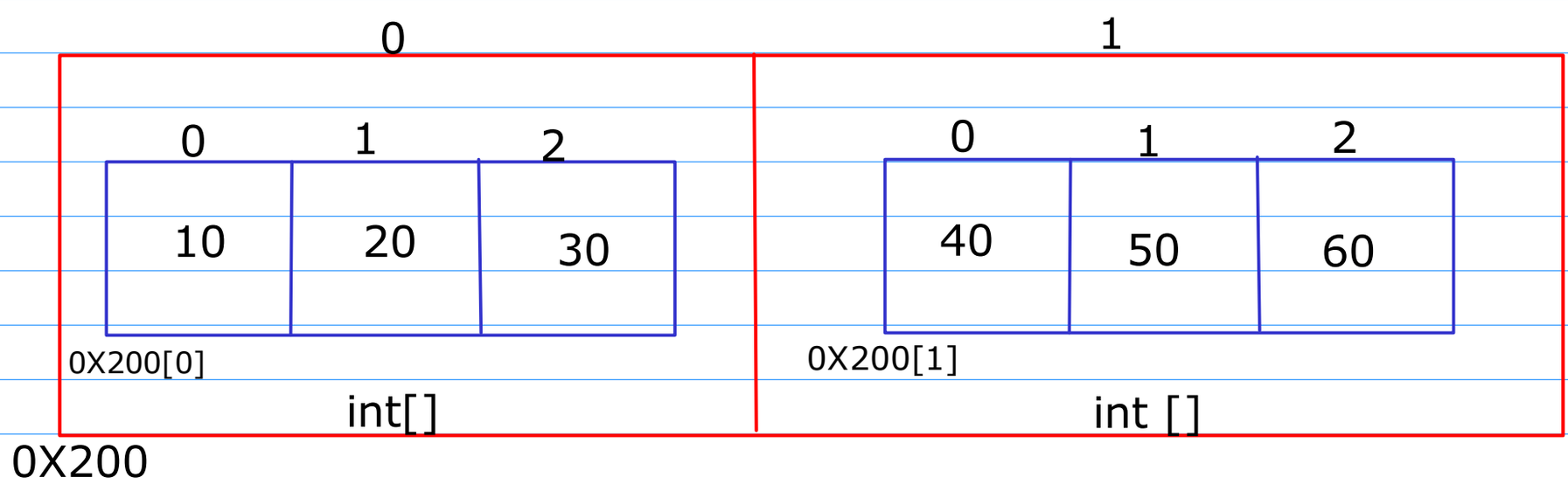
0X200[1]=20
arr[1] = 20;

int value = arr[0]

Types of Arrays
1. Single Dimensional Array
2. MultiDimensional Array
3. Ragged Array

int arr[][][][][]....

int arr[2][3]



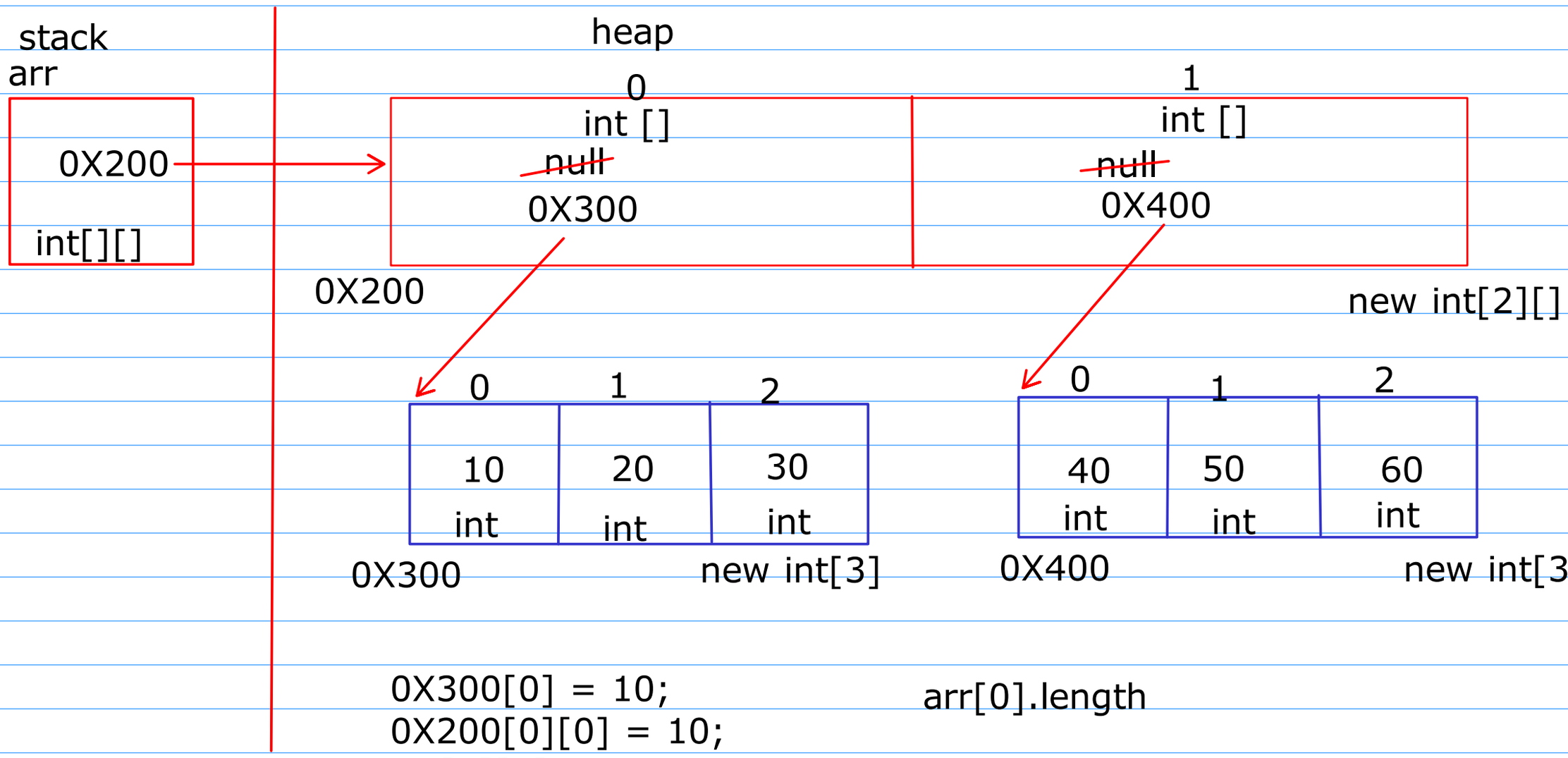
arr

arr -> 0X200

0X200[0][0] = 10
arr[0][0] = 10
0X200[0][1] = 20
arr[0][1] = 20

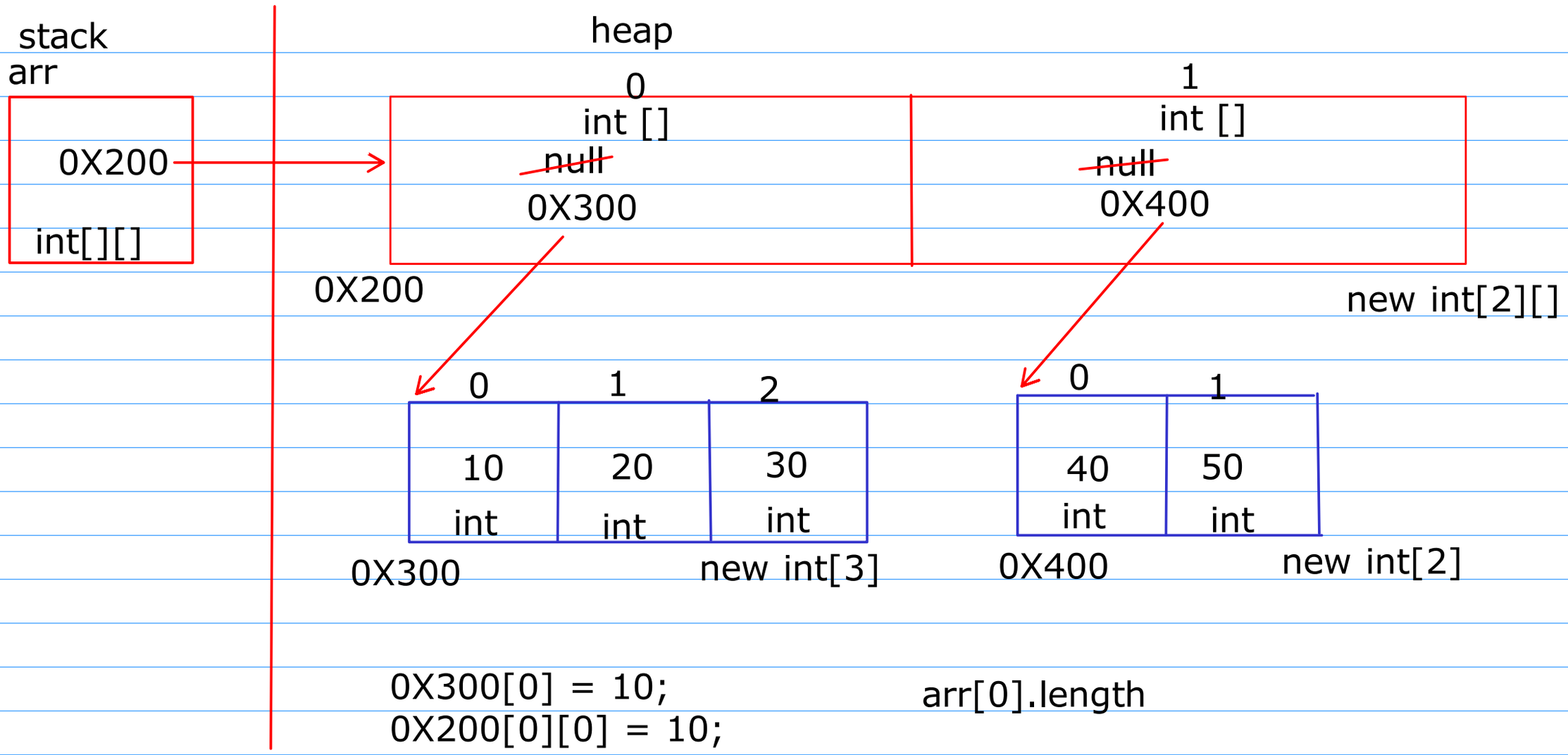
0X200[1][0] = 40
arr[1][0] = 40;

```
for(int i=0;i<2;i++){
    for(int j=0;j<3;j++){
        int ele = arr[i][j];
        print(ele);
}
}
```



0X300[0] = 10;
0X200[0][0] = 10;
arr[0][0] = 10;

arr[0].length




```
0X300[0] = 10;
0X200[0][0] = 10;
arr[0][0] = 10;
```

arr[0].length

```
int **ptr = new int*[2];
ptr[0] = new int[3];
ptr[1]= new int[2];
```

Multi Dimensional Array

Ragged Array