

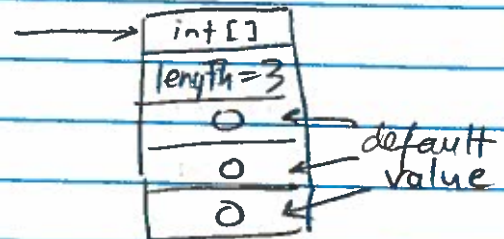
Recitation 1 : Java pitfalls

Arrays

Arrays in Java are boxed values (like objects).

`int[]` — array of `int`.

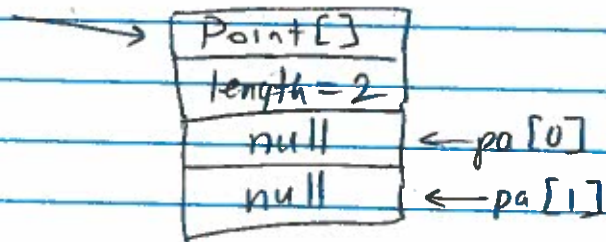
`new int[3]`



`Point[]` array of `Point`

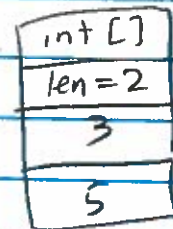
`pa = new Point[2]`

`pa.length // = 2`



initializing:

`new int[] {3, 5}`

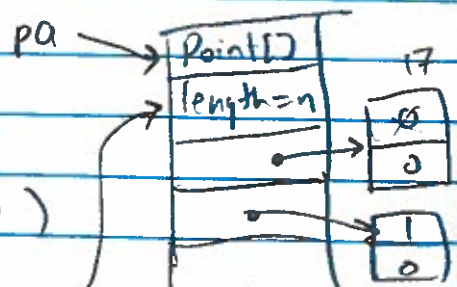


Can use loops to access arrays.

`Point[] pa = new Point[n];`
`for (int i=0; i<n; i++) {`
 `pa[i] = new Point(i, 0);`
`}`

Initializer (only) guard end of each iteration

`pa[i] = new Point(i, 0);`



Aliasing

`Point[] qa = pa;`

`qa[0].x = 17`

not a copy

`pa[0].x = ?`

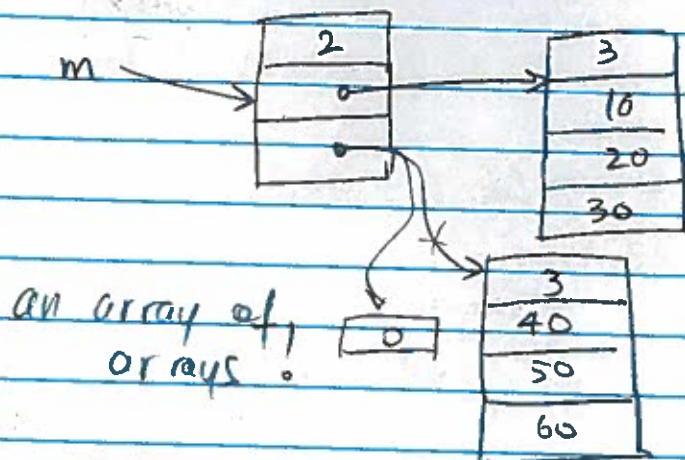
`qa`

`qa` is an alias

6

Multidimensional arrays

```
int[][] m = new int[2][3] {{10, 20, 30}, {40, 50, 60}}
```



```
m[1] = new int[0]
```

Ex.

```
class Point {
```

```
    int[] coords;
```

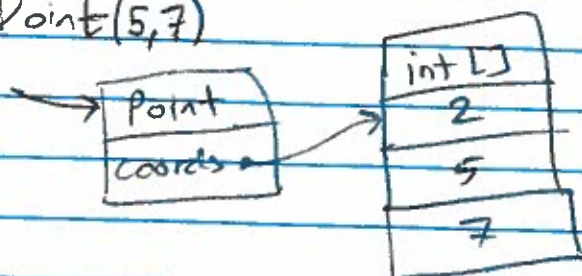
```
    Point(int x, int y) {
```

```
        coords = new int[] {x, y};
```

```
    }
```

```
}
```

```
new Point(5, 7)
```



true =
false =
str
true if
get!
preboxed.

Autoboxing

9, 63}

class Integer is a boxed int.

Int

new Integer(5)



float — Float

double — Double

char — Char

boolean — Boolean

Java converts between boxed / primitive automatically.

Integer i = 200;

int j = i; ;

Integer k = j;

i → 200

j → 200

k → 200

true

false

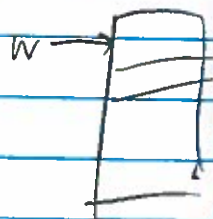
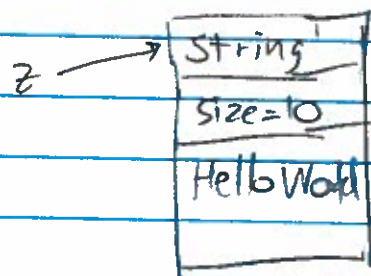
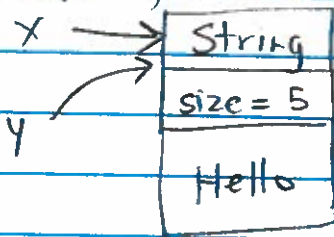
i == j

i == k

(use .equals()) String constant

not true! get! boxed.

Expression "Hello" creates object, class String.



String x = "Hello";

String y = x;

String z = y + "World"; w = y + "World"

(else)

concatenation

What can we do with strings?

x.charAt(1) < 'e'

x.equals(y) < true

z.equals(w)

x == y < true

same object?

true

z == w

false

== tests object identity.

equals() != whether objects interchangeable.

Autoboxing (prev pg)

Scope

names refer to:

variables (local to method
incl. parameters)
fields
methods
types.
classes
packages

rule: inside-out

- walk outward, bind to first name (of right kind)
local vars > fields

```
Point(int x, int y) {
```

"x" \nearrow

this.x = x

this.y = y

```
}
```