

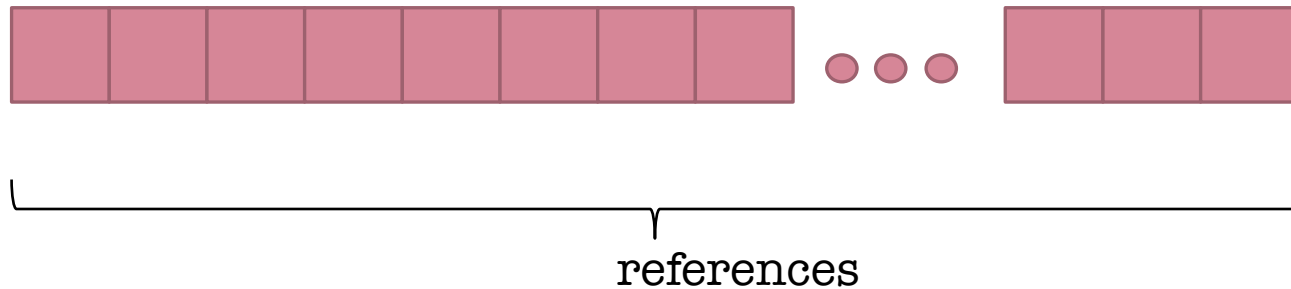


Collections I: Arrays of Objects

WEEK 3 PRESENTATION 2

When you create an array of objects, each element isn't an object it's just a place to store a reference. (note: this is unlike arrays of base types such as int or float etc).

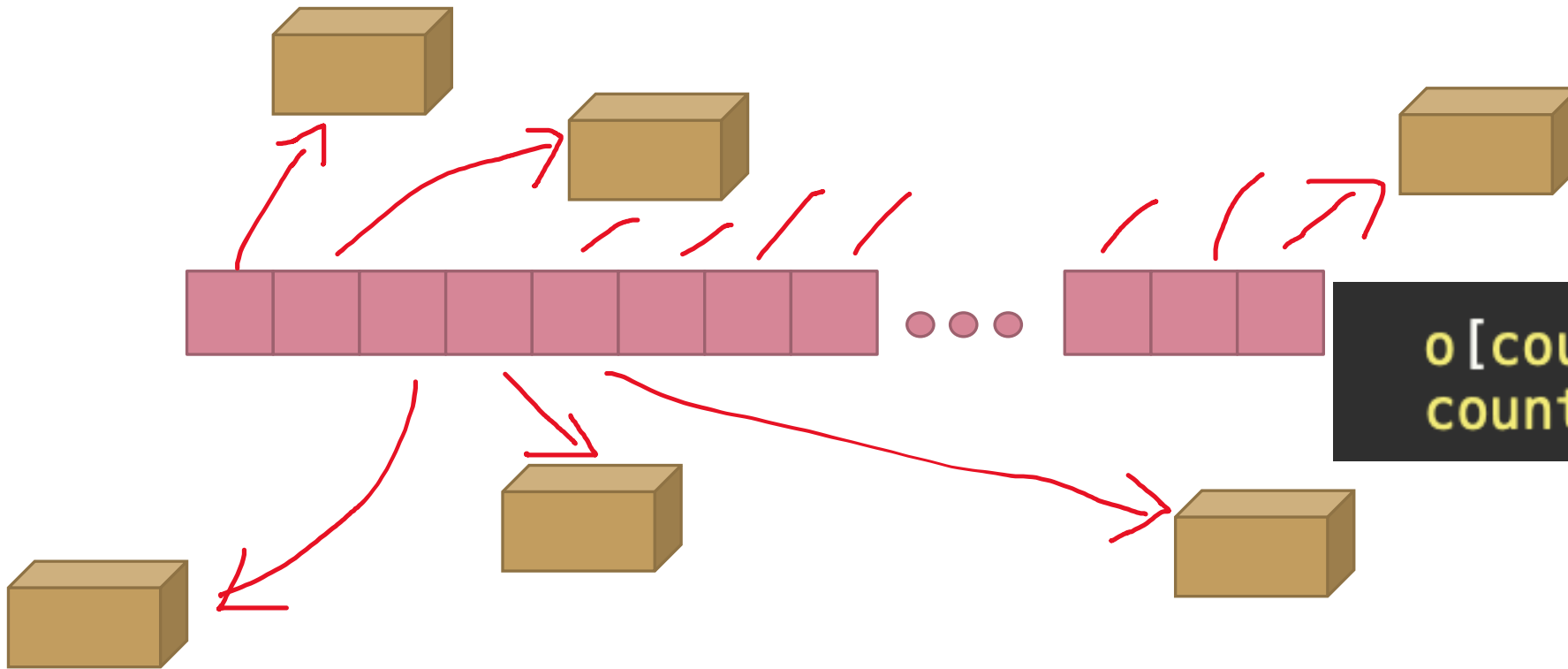
```
int size = 20;  
Object[] o = new Object[size];
```



```
Object[] o = new Object[size];  
int count = 0;
```

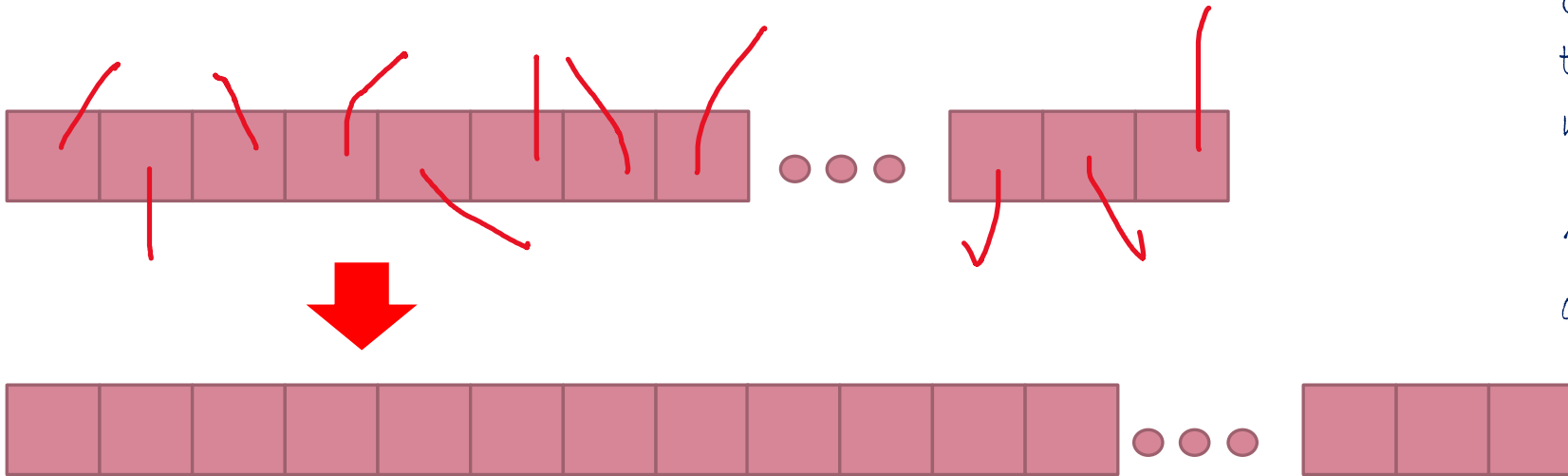
**Each object is
created one by one.**

Each object added to the array must be separately created ("instantiated"), which is very different from arrays of base types.



```
o[count] = new Object();  
count++;
```

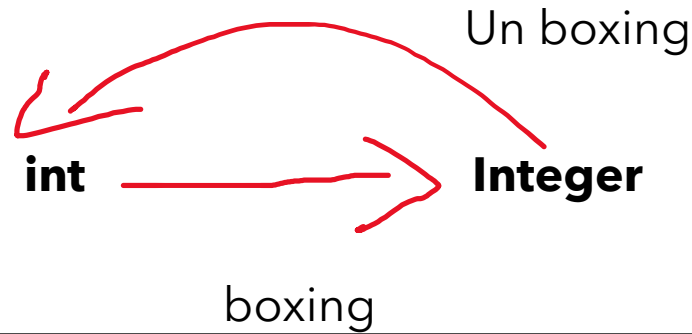
Because there is no limit on the memory required by objects using object references makes it easier to process arrays, for example if you exceed the size of an array you can create a bigger one just by copying the object references...



NB: Making a "Deep Copy" of an array means to the process of copying not just the references (which is the effect of "=") but of copying each object... in a loop

```
Object [] o2;  
o2 = o;  
o = null;
```

"Boxing"



Boxing is between the primitive types and their corresponding object wrapper classes. For example, converting an **int** to an **Integer**, a **double** to a **Double**, and so on. If the conversion goes the other way, this is called *unboxing*.

Autoboxing is the automatic conversion that the Java compiler makes between the primitive types and their corresponding object wrapper classes.

Here is the simplest example of autoboxing:

- `Character ch = 'a';`

See: <https://docs.oracle.com/javase/tutorial/java/data/autoboxing.html>

If I'm calling a method that takes an Integer parameter with an int parameter, everything works fine.

```
1  
2 public class Boxing1 {  
3  
4     public static void displayVal(Integer val) {  
5         System.out.println(val);  
6     }  
7  
8  
9     public static void main(String[] args) {  
10         int n = 10; displayVal(n);  
11     }  
12 }
```

WORKS OK

Problems Javadoc Declaration Console X
<terminated> Boxing1 [Java Application] /Library/Java/JavaVirtualMach
10

Compilation error

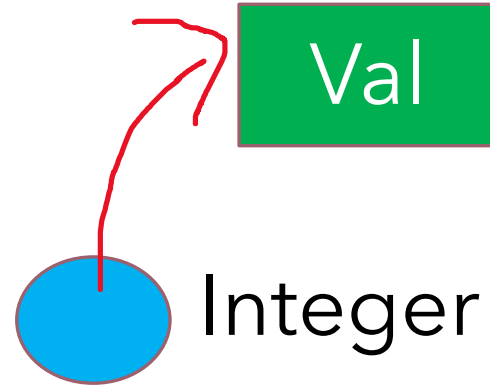
It fails. The reason of the problem is that `int[]` cannot be converted to `Integer[]`. Boxing/unboxing know how to convert between base type and object, but not between object and object ... and an array is an object (it's created by calling `new`). We have "lost" the base type.

```
1
2 public class Boxing2 {
3
4
5     public static void displayVal(Integer[] valarr) {
6         for (int i = 0; i < valarr.length; i++) {
7             System.out.println(valarr[i]);
8         }
9
10
11     public static void main(String [] args) {
12
13         int [] n = new int[3];
14
15         for (int i=0; i < 3; i++) {
16             n[i] = i;
17         }
18
19         displayVal(n);
20     }
21 }
```

Exception in thread "main" java.lang.Error:
Unresolved compilation problem:
The method `displayVal(Integer[])` in the type
`Boxing2` is not applicable for the arguments
`(int[])`



int



Blue indicates a reference.

The relationship between int and Integer is straightforward.