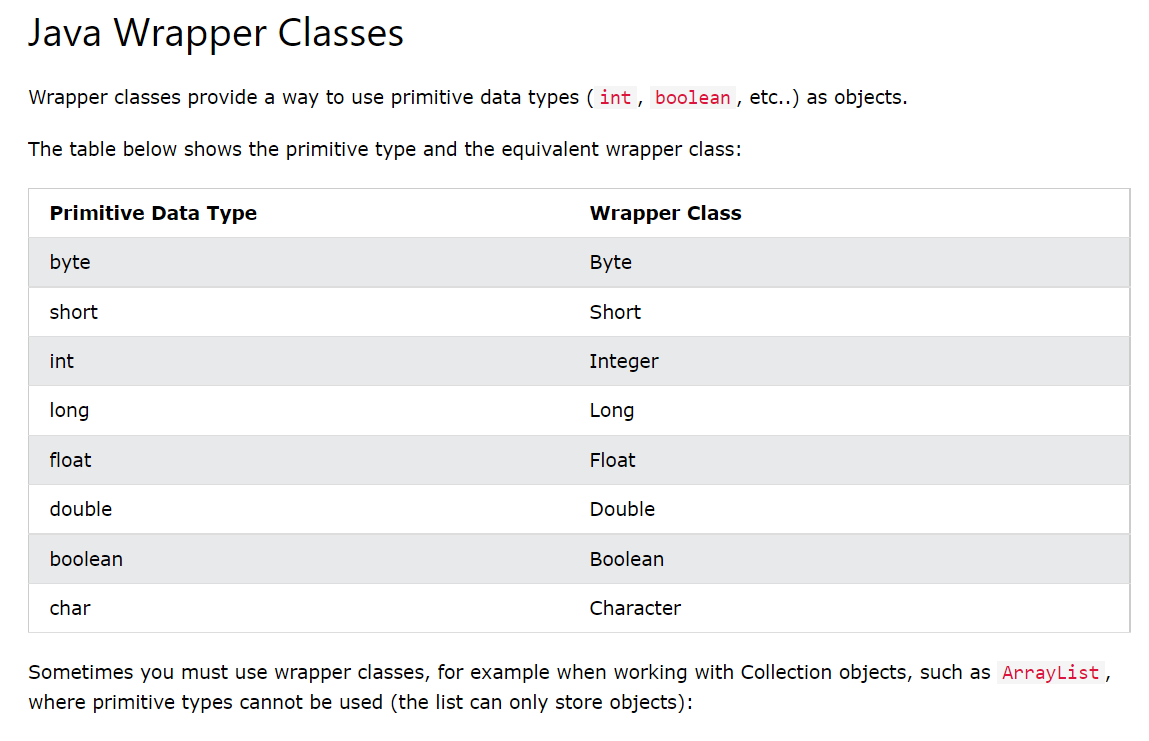
**Wrapper class**

In java we have wrapper classes corresponding to each primitive data types



A Wrapper class is a class whose object wraps or contains primitive data types.

It contains a field and in this field, we can store primitive data types

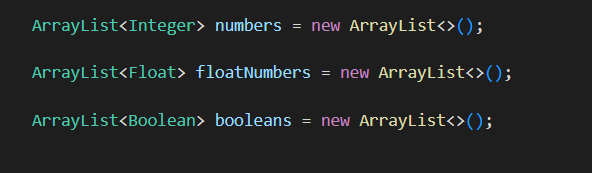
In other words, we can wrap a primitive value into a wrapper class object

**ArrayList in Java**

In arrays we must need to specify the size of the array

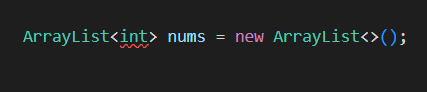
But when we might not know the exact size of array, we can use ArrayList

It’s like <vector> in C++



This is how we declare the ArrayList in java

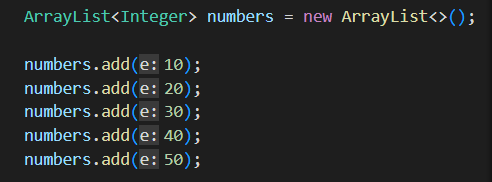
**We do not use the primitive data types, we use wrapper classes**



This is invalid

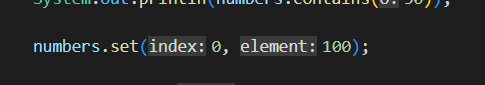
We can not use primitive data types, must use wrapper classes

**ArrayList has various essential methods**



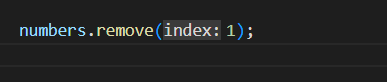
**.add() this is used to add the items in the array**

**.set( index, value )**



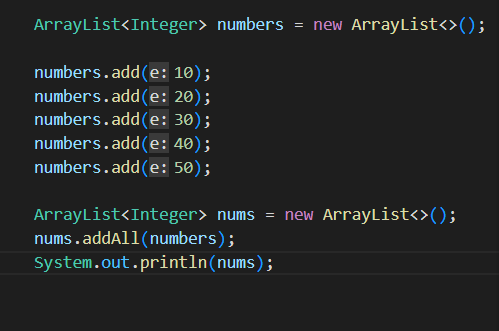
used to set a value at the particular index in the ArrayList object

**.remove( index )**

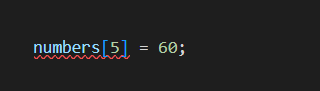


used to remove the item of the particular index number

**.addAll( ) we can add all the items of another ArrayList type of object**

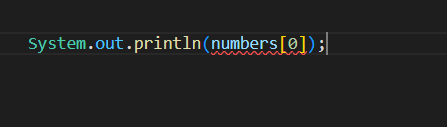


Here neither we can set an item like this we do in array



Nor

We can get an element like this, as we do in array



So

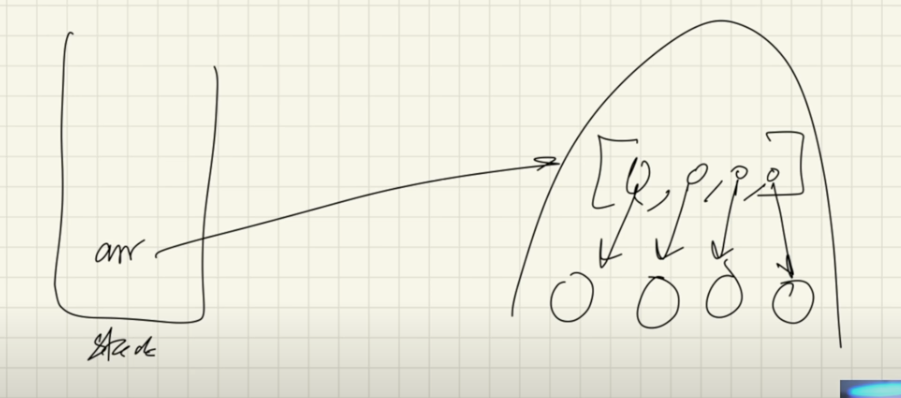
We have

**.add( value ), .set( index, value ), .addAll( ArrayList\_object )** method to add the item

And

**.get( index )** to get the item

Internal working of ArrayList



the reference variable of the ArrayList object lives in stack and points to the object in the heap

Also in ArrayList primitive data types are not allowed, instead we are only allowed wrapper classes

So each index of the array in the heap points to the object of the wrapper class