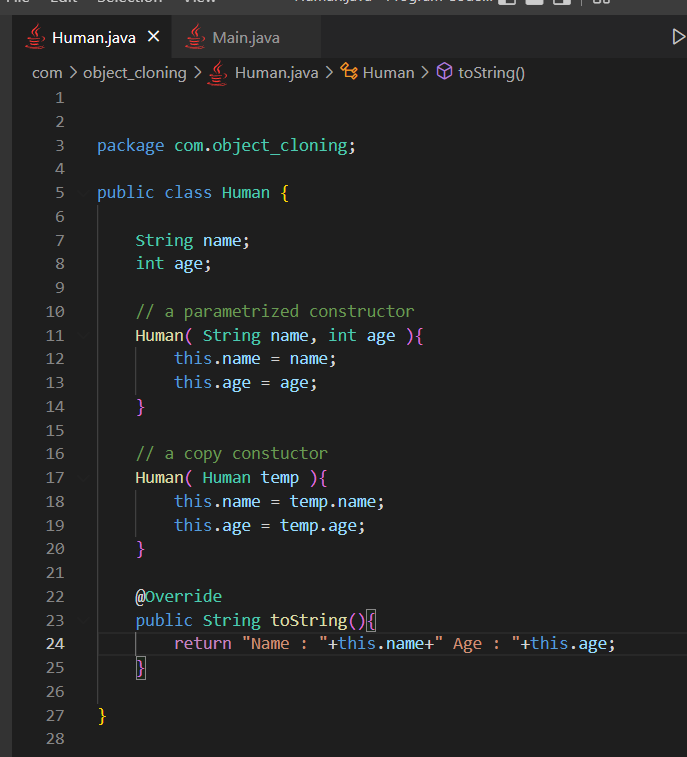
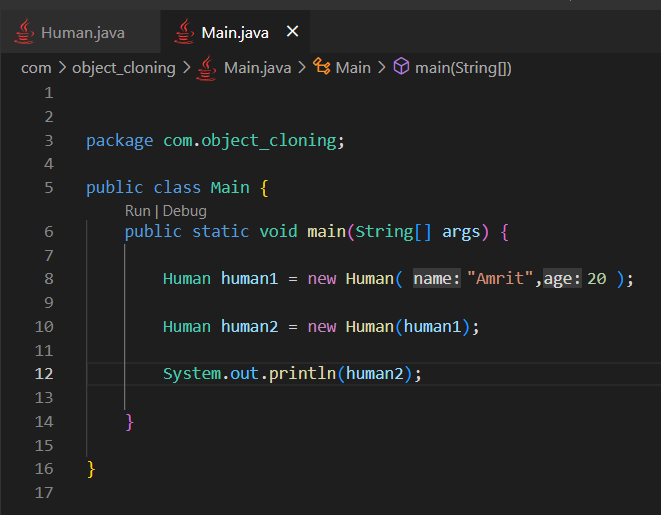
Cloning an object means just to make a copy of the object

We can make a copy of the object using the Copy constructor in the class of that object



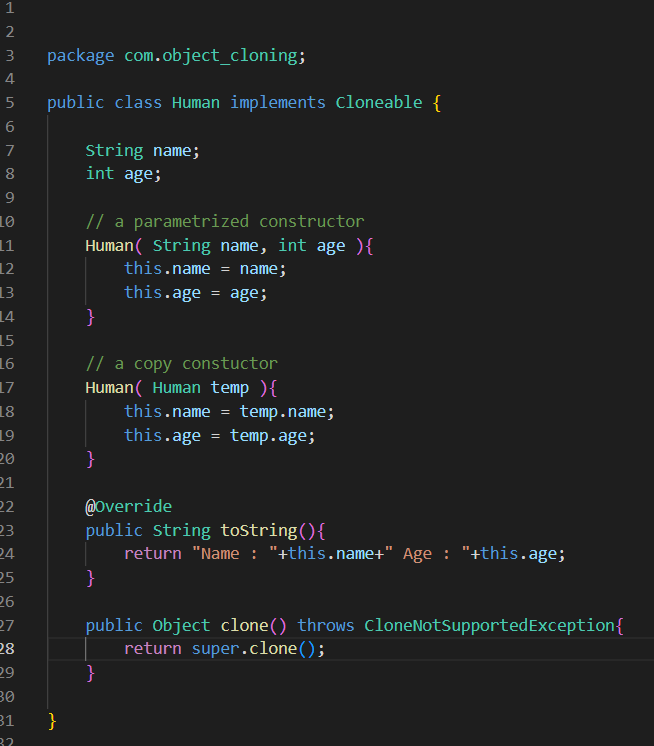


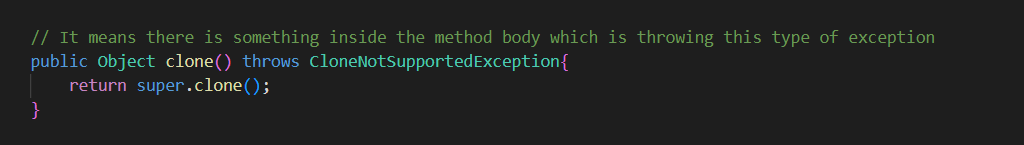
**Drawback**

This is consuming too much memory

In java in the Lang package we have something called Clonable interface, so we must implement that interface by the class whose object we want to clone

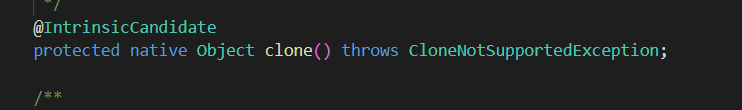
And there is a method called clone( ) which we define inside the class whose object we want to clone

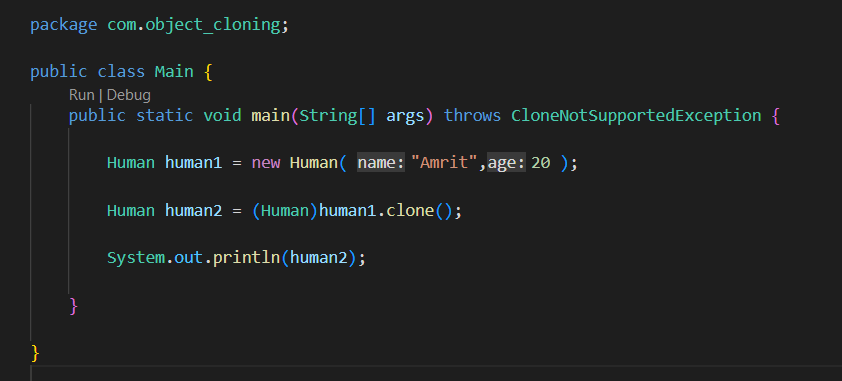


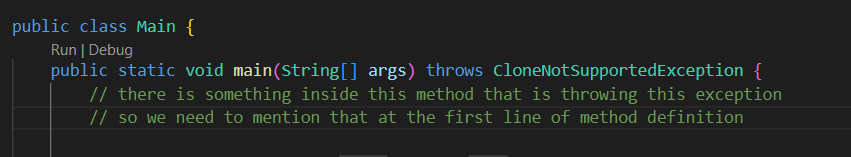


And how do we know that this clone( ) method throws that exception?

We see the original definition of this method

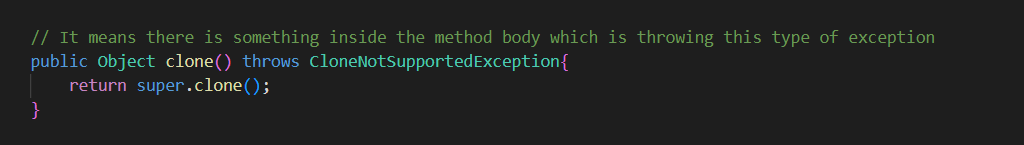






Also  type conversion is really important

Because

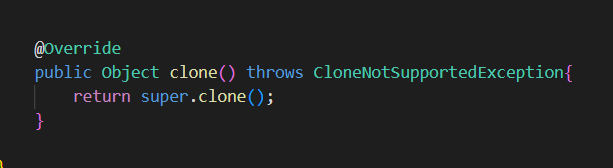


We see that the .clone( ) method returns the Object type, but here we are trying to get Human type

So, we must do the type conversion from the Object type to Human type

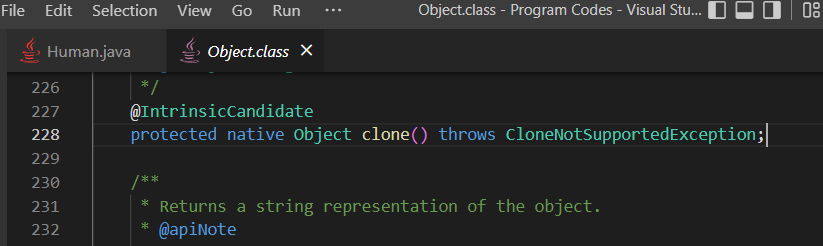
Also, we can do this type conversion because the Object class is the top most class of every class, in a way every class is inheriting from the Object class internally, so we can do this type conversion

But we can never do type conversion from the Human type to the Object type

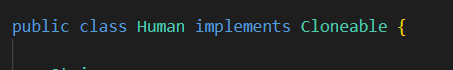


Do not get the wrong idea, this clone( ) method is not present inside the interface Clonable

This method is actually present inside the Object class, and we are overriding this method in our class



See clone ( )method is present inside the Object class



So here all this is doing is, it is saying the JVM that we can to make clone the objects of Human class

So JVM knows that okay it is allowed to make the clone of objects made from Human class