

Well, now I want to introduce something called a static attribute.

And a static attribute is a shared attribute across all objects based on a single class. It is not dependent on instantiation of each object. This means we can access a static attribute without ever having to create an object based on that class.

So why would we want to use static attributes?

Well, they're useful for when we want to keep track of things like object counters.

Let's say we create five objects from a specific class, but we can use a static attribute to keep track. Then we can find out how many we created.

Now, I've just mentioned that we don't need to create an object to be able to access static attributes. But when we do create an object or create many objects based on the same class with a static attribute, all the objects have access to that same static attribute.

They all reference the same memory area.

So, let's have a look and we declare them well, it's very similar to a normal data statement, but instead they're using data.

What we do is use the keyword class-data.

REPORT ZYNY\_CLASS.  
  
CLASS STUDENT DEFINITION.  
  PUBLIC SECTION.  
  DATA : NAME TYPE C LENGTH 40,  
        AGE TYPE I,  
        gender TYPE c LENGTH 1 READ-ONLY.  
\*        GENDER TYPE C LENGTH 1 READ-ONLY VALUE 'U'.  
    
  CLASS-DATA: count TYPE i.  
  
  PRIVATE SECTION.  
  DATA :  LOGINID TYPE C LENGTH 20,  
        PWD TYPE C LENGTH 15.  
  ENDCLASS.

So, as you can see in the example here, we have class my class definition, we have the public section. And then instead of just collecting data, we're saying class data, we declare the name and then we declare the type and then we close the class with end class.

Now, keep in mind that changing the value of a static attribute will make the change to value visible in any object that is instantiated from that class.