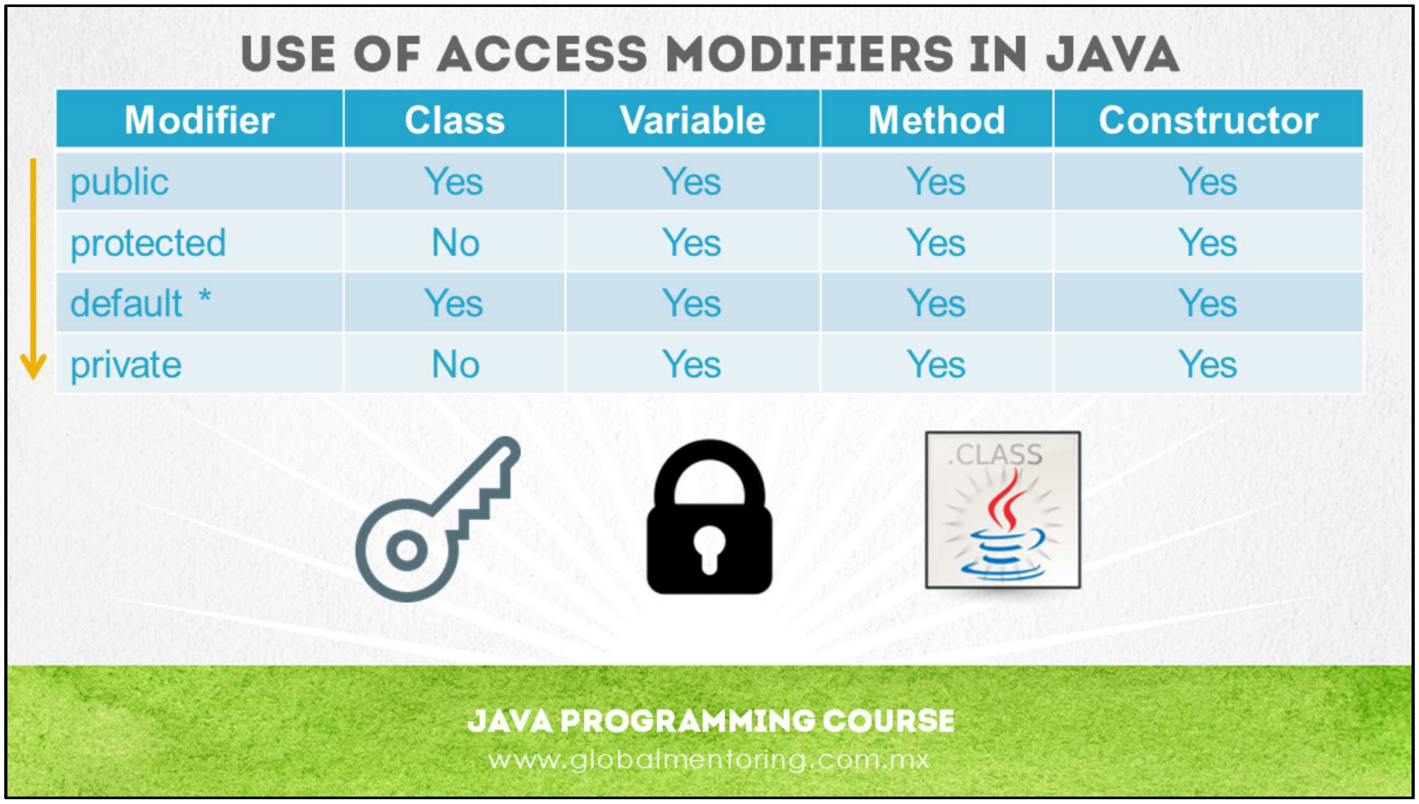
Hello, Ubaldo Acosta greets you. Welcome again. I hope you're ready to start with this lesson.

We are going to study the topic of access modifiers in Java.

Are you ready? OK let's go!



USE OF ACCESS MODIFIERS IN JAVA

Modifier	Class	Variable	Method	Constructor
public	Yes	Yes	Yes	Yes
protected	No	Yes	Yes	Yes
default *	Yes	Yes	Yes	Yes
private	No	Yes	Yes	Yes

.CLASS
JAVA PROGRAMMING COURSE
www.globalmentoring.com.mx

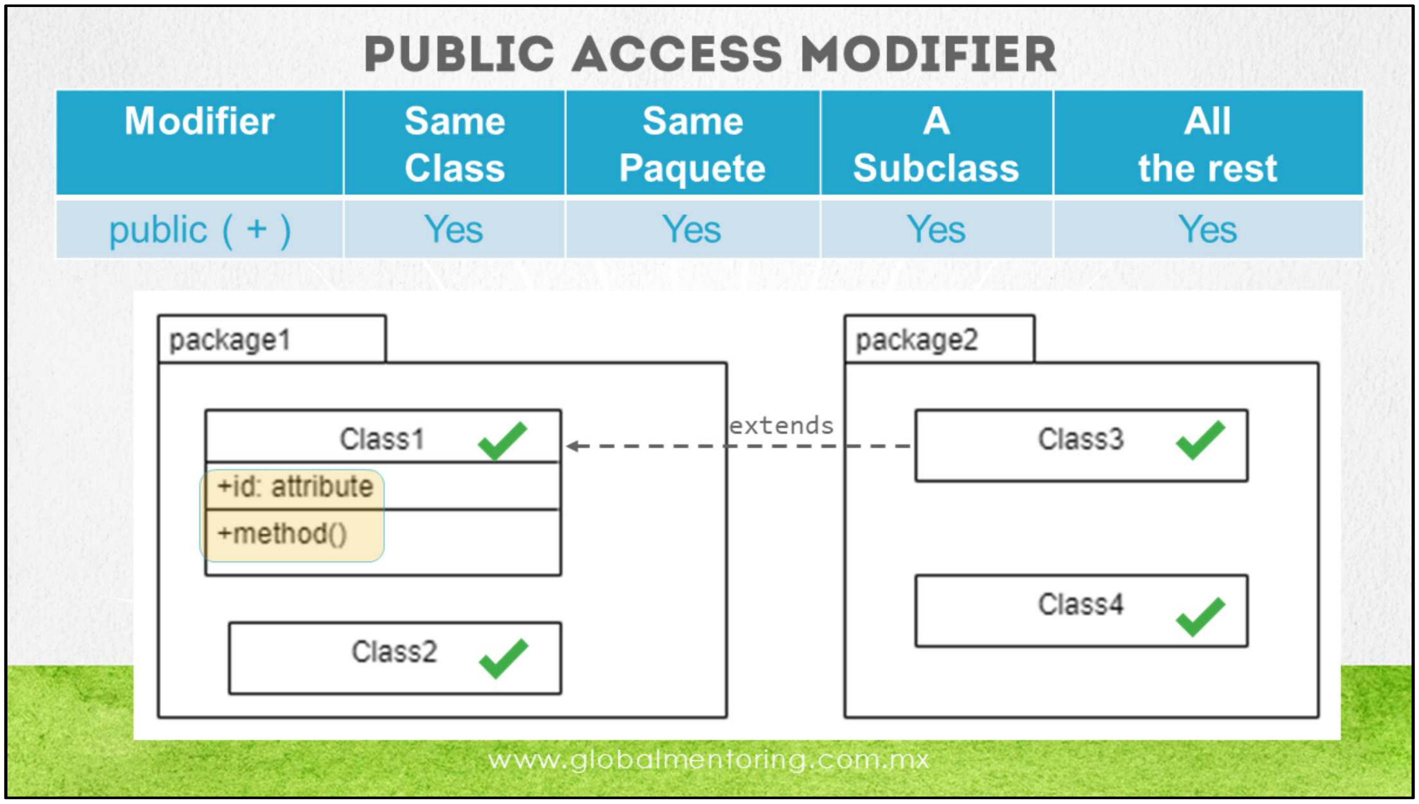
In this lesson we will see the topic of access modifiers, which control who can use each of the characteristics of the class that we are defining.

The access modifiers can be applied to the definition of a class, an attribute, a method of a class or the constructor of a class.

The table shows the possible combinations of each of the modifiers and where they can be applied. There are four types of access modifiers, and the arrow indicates the order from least to most restrictive, for example public is the least restrictive since all external classes can make use of the element that has been defined as such, and the most restrictive is private, which can only be accessed by the same class where the element has been defined.

The word "default" or package is not a reserved word, unlike public, protected and private, this means that to define a variable of type default or of type package, simply omit the access modifier, that is, if no access modifier is specified by default the defined element is of type default or package.

Next we will see what each one of the modifiers means.



PUBLIC ACCESS MODIFIER

Modifier	Same Class	Same Paquete	A Subclass	All the rest
public (+)	Yes	Yes	Yes	Yes

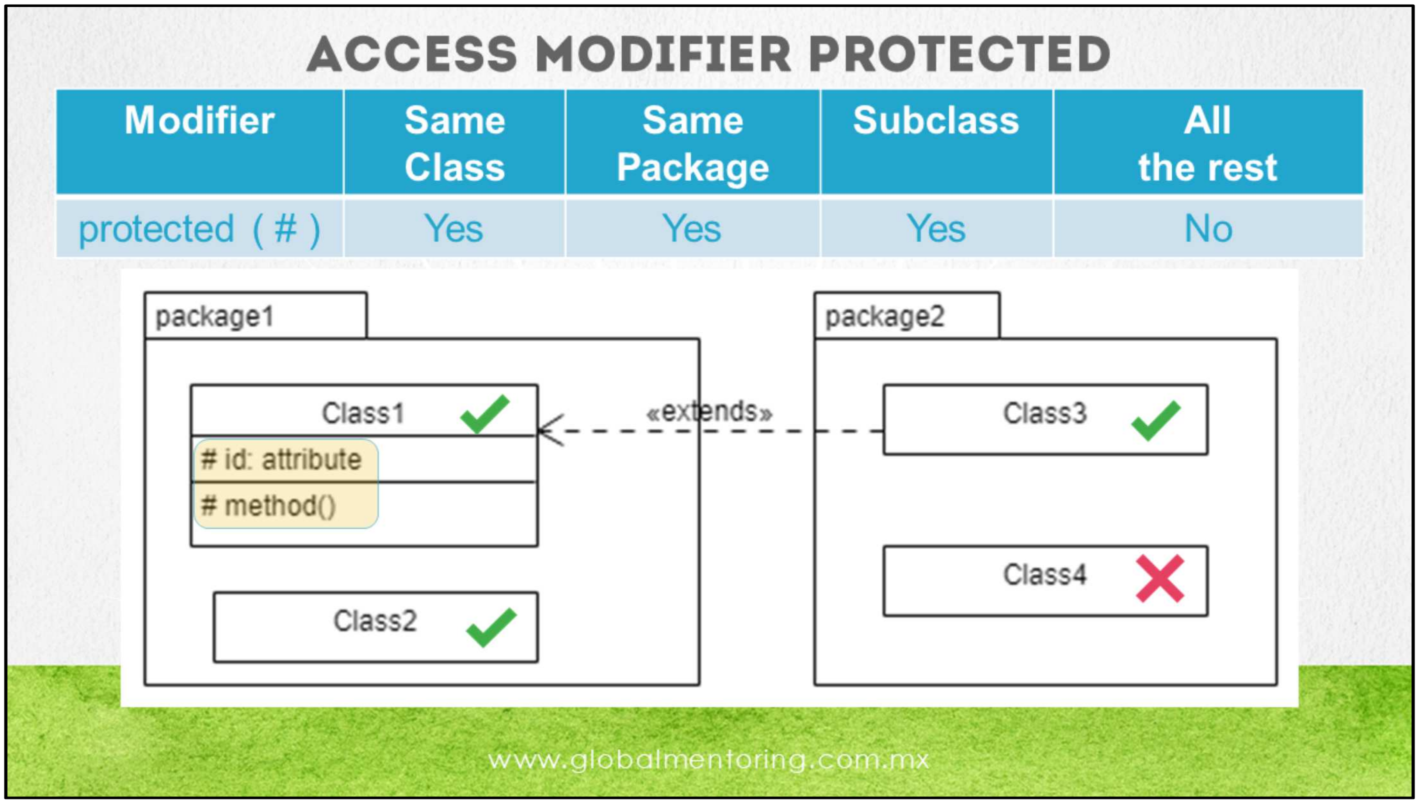
In this slide we will see the use of public modifier. As we saw in the first slide, the public modifier applies to all kinds of elements in Java, from a class, an attribute, a method or a constructor.

This is the least restrictive access modifier, therefore, if we want any other class to access some defined element, then we must use the public access modifier. In this way any other class, either within the same package or in a different one can access the element defined as public.

The table indicates that if we use this access modifier, the notation inside the shown class diagram marks the element with the + symbol, indicating that it is a public access element, and as its name says when marking it as public then all other classes can access this element, which can be a class, an attribute, a method or a constructor as we have already commented.

So, regardless of where the other class that wants to access the public item is located, and regardless of whether it descends or not from the class that defines the public item, these other classes can access the element defined as public. This modifier is the simplest to apply and in summary we will use it when we want the other classes to access without problems our element marked as public.

It is therefore common that the methods of type get and / or set of the attributes of a class are defined as public, since these methods the objective is that they are accessed by any other class, and thus they can either read and / or modify the state of an object through these methods.



ACCESS MODIFIER PROTECTED

Modifier	Same Class	Same Package	Subclass	All the rest
protected (#)	Yes	Yes	Yes	No

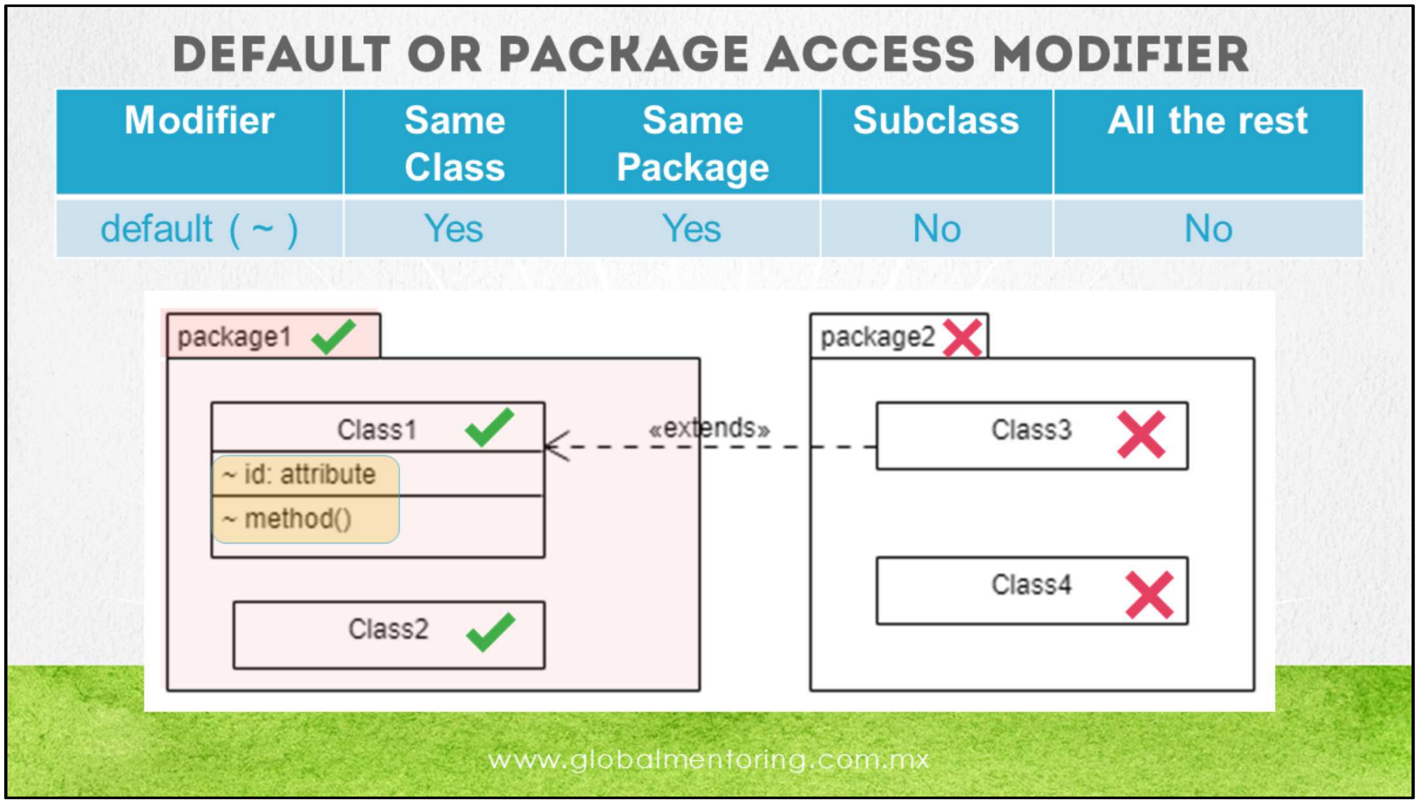
The next modifier that we will review is the protected modifier.

We will use this modifier when we want to protect from classes that do not descend directly from the class that defines the protected element and that can not access that element.

Therefore, it will be common to find the protected modifier in attributes or methods of a class, with the objective that the child classes can access these elements, either in the same package or in a different package. Hence its name, which protects from the access of classes, they express that they have nothing to do with the hierarchy of subclasses.

In the diagram you can see the symbol #, which is used to denote an element marked with the protected access modifier, either an attribute or a method.

It should be noted that this modifier does not apply to the definition of a class, that is, it is not possible to define a class of type protected.



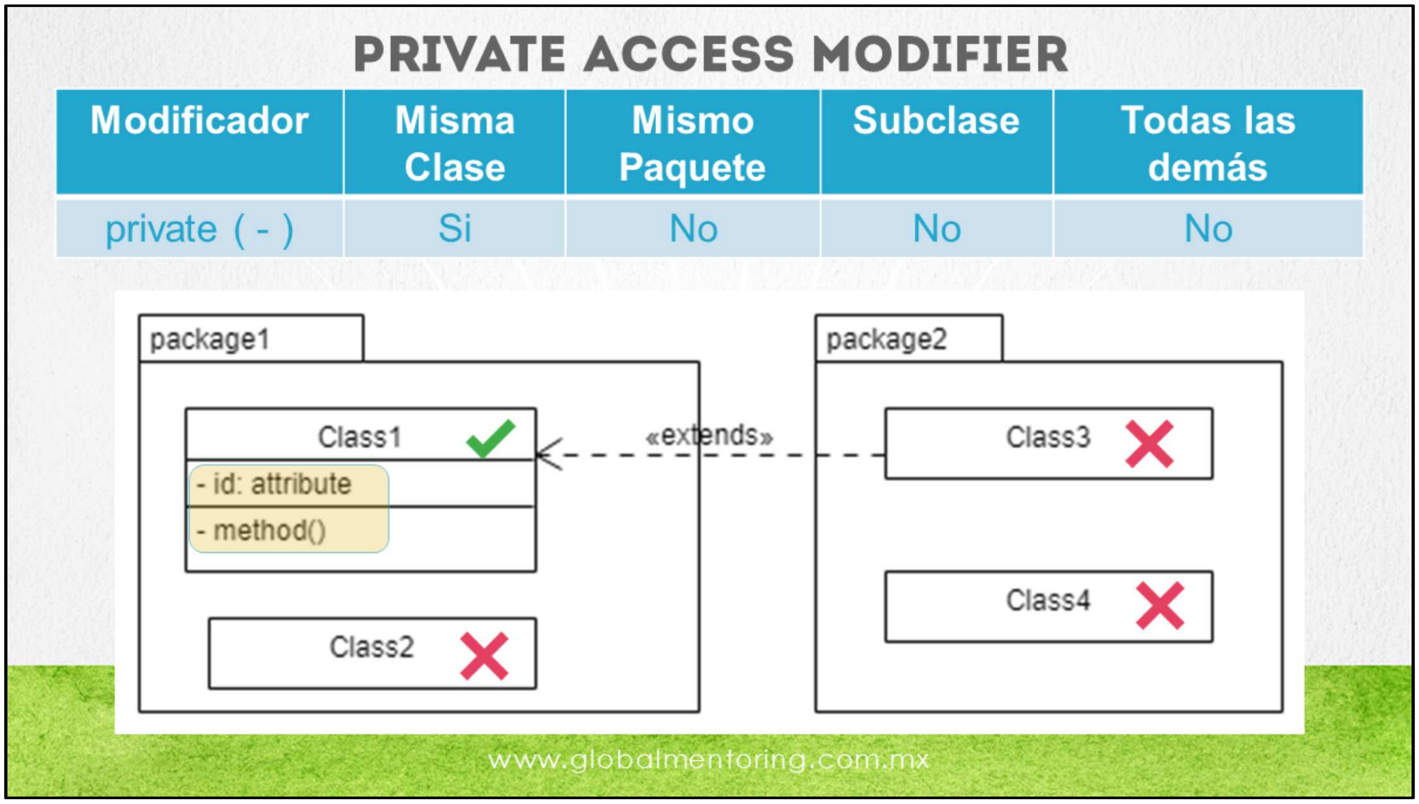
DEFAULT OR PACKAGE ACCESS MODIFIER

Modifier	Same Class	Same Package	Subclass	All the rest
default (~)	Yes	Yes	No	No

Now we will see the default access modifier or package. As its name implies, this modifier limits access to the classes defined within the same package, so any other class outside the package where the item has been defined will not be able to access those elements.

This modifier does not have a reserved word, but the omission of any modifier, the compiler understands that it will be treated as default or package. This modifier is not common to find, because it is unsafe, since it allows any class of the package to access the element in question, however, if it is what we want we can apply precisely this access modifier to the desired element.

The notation in the diagram to recognize an attribute of type default is the symbol ~.



PRIVATE ACCESS MODIFIER

Modificador	Misma Clase	Mismo Paquete	Subclase	Todas las demás
private (-)	Si	No	No	No

Finally we will see the private access modifier. This modifier basically prevents any other class, even subclasses from accessing the element marked as private.

This modifier can not be applied to the definition of a class, since we can not define a private class.

The symbol we use to define a private element is the symbol -.

It is common to apply this type of modifier to the attributes of a class, since the idea is that it is through the get / set methods that the object is accessed, but not directly to the attributes, therefore the attributes are defined as private.

We can also create private methods, which can only be accessed from the class where they are defined and from no other class. Finally, it is possible to define private constructors, with a similar objective, to be used only by the class that is defining them, and thus prevent their access from other classes.

We are going to create an exercise where we put these access modifiers into practice.

ONLINE COURSE

JAVA PROGRAMMING

By: Eng. Ubaldo Acosta

Global Mentoring

JAVA PROGRAMMING COURSE

www.globalmentoring.com.mx