

JAVA PROGRAMMING COURSE

INTERFACES IN JAVA



By the expert: Eng. Ubaldo Acosta



JAVA PROGRAMMING COURSE

www.globalmentoring.com.mx

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

We are going to study the topic of interfaces in Java.

Are you ready? Come on!

INTERFACES IN REAL LIFE



JAVA PROGRAMMING COURSE

www.globalmentoring.com.mx

In this lesson we are going to study the topic of interfaces. An interface is a common way of communicating between different devices or systems. It is a contract between a provider and a customer of a service.

The concept of Interface in Java replaces in some way the need for the use of multiple inheritance, since in Java we can implement several interfaces, but we can only inherit a class.

An interface in Java is used to define behavior and then the class must implement the behavior defined by the interface, so it is common for the English verb (able) to be used in some interfaces of the Java API, such as Serializable, Cloneable, etc, since this verb in English indicates capacity to do something.

USE OF INTERFACES IN JAVA

Defining an interface in Java :

```
<modifiers> interface <interface_name> [extends <interface parent>]
{
    <attributes>
    <methods>
}
```

Use of an interface in Java :

```
<modifiers> class <class_name> [extends <superclass>] [implements
<interface1, interface2, etc>]
{
    <implement_methods_of_the_interfaces>
}
```

JAVA PROGRAMMING COURSE

www.globalmentoring.com.mx

In Java, an interface is a formal declaration of a contract, in which the methods do not contain any implementation, very similar to the abstract methods discussed above.

The declared attributes in an interface are by default public, static and final, that is to say they are constants that we can access simply by putting the name of the interface, and later the name of the constant, for that reason if we define attributes in an interface we will write them with letters capital letters, and if it contains several words we will separate them by an underscore, for example: MY_CONSTANT.

The methods declared in a default interface are public, abstract, and do not contain any implementation (they end with a semicolon), this is similar to the abstract methods of an abstract class.

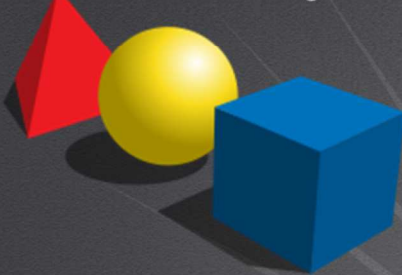
Similar to inheritance in classes, an interface can inherit from another interface, but not from a specific class. Then an interface that inherits from another interface will add to its method definition both those of the parent interface, and the methods that this interface defines.

An interface follows the same nomenclatures of names as a class, so we will find several similarities between the definition of a class and an interface.

Many classes can implement the same interface. A class can implement many interfaces even though they have no relation. As in an abstract class, we can not create objects from an interface, but we can create variables of type Interface that point to objects that implemented this Interface, thus also applying polymorphism type concepts as we have studied previously.

ABSTRACT CLASSES VS INTERFACES

Geometric Figures



VS



JAVA PROGRAMMING COURSE

www.globalmentoring.com.mx

We will comment on some of the differences between abstract classes and interfaces, as well as when to use one or the other concept.

An abstract class is used to encapsulate common functionality between classes (characteristics). To make use of an abstract class we must extend the class.

Instead, an interface defines related behavior that can belong to any class or structure. To use an interface we must implement the interface.

So the recommendation is that abstract classes should be used primarily for closely related objects, while interfaces are more suited to provide functionality common to unrelated classes.

We are going to create an example of the use of interfaces below.

ONLINE COURSE

JAVA PROGRAMMING

By: Eng. Ubaldo Acosta



JAVA PROGRAMMING COURSE

www.globalmentoring.com.mx