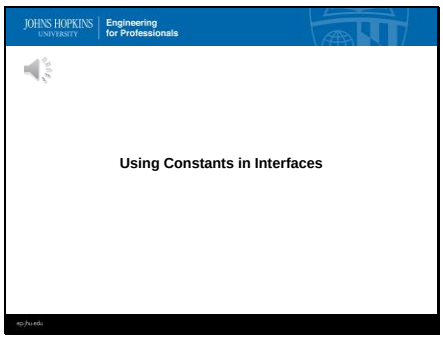
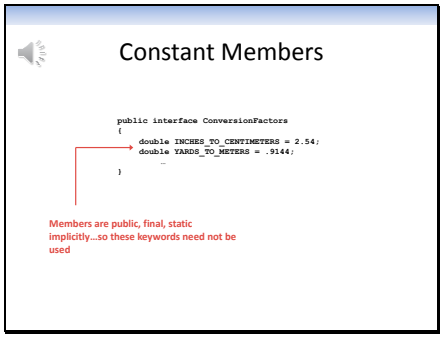
1



In this lecture you will learn how to use constants in interfaces.

2

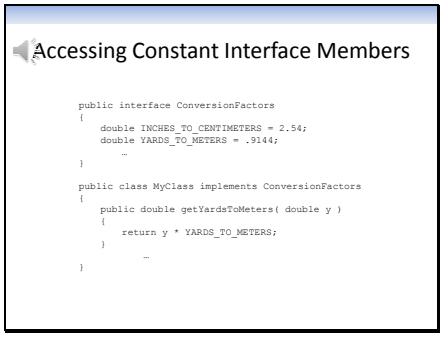


The most commonly used members of interfaces are abstract methods…but interfaces can also contain constants.

This interface contains a number of conversion factors.

Constant interface members are implicitly final, static, and public…so these keywords need not be used in the member definitions. In fact, it is standard practice not to use them.

3

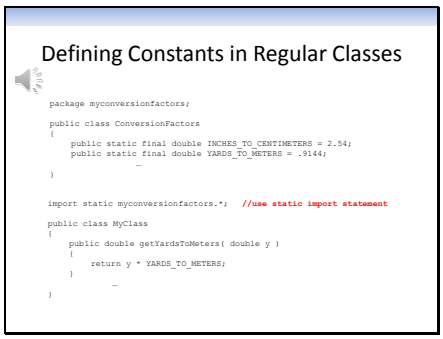


To get access to the constant members of an interface…you simply implement the interface in any class that needs to use the constants.

The class then has access to the constants via their unqualified names, as indicated in this example.

In this example, any subclasses of MyClass will also have access to the constants as well.

4



The technique of defining constants in interfaces and then implementing the interfaces in classes that need the constants has been used since the inception of the Java language in the mid-1990s.

However, with the more recent versions of Java, there is a more effective approach.

The approach defines the constants in a regular class…and makes use of a static import statement to make the class available to other classes.

In this example, the constants are defined in a regular class in a package called myconversionfactors.

Note that they had to explicitly be defined as public static final variables.

Code components that need to use the conversion factors can use a static import statement to access them by their unqualified names.

They could also be accessed by using the fully-qualified names of the variables.