Classes and Packages

Learning Objectives

Import a class from a package

Classes and packages

All classes in Java are organized in packages. Some of the packages, like java.lang, contain classes necessary for all Java programs to run, and are thus included for use by default. For other classes in other packages, you will need to *import* them. It is important to draw a distinction between the *Java language* and the library of classes that come with it. Classes, even those that are always available (such as System and String) are part of the library Sun provides with Java. Most modern languages have extensive libraries of classes that add the power necessary to easily solve problems.

You can view the documentation for the classes in Java (http://docs.oracle.com/javase/7/docs/api/) to learn about how to use them, or look for the documentation (http://www.oracle.com/technetwork/java/javase/documentation/index.html) associated with your installed version of Java. There are probably many classes we won't cover in this course that you may want to use one day.

learn by doing

Visit the documentation for java.awt.Color What else can you do with Color?
Visit the documentation for Java's API's What are some other classes in the java.awt package?
What are some other packages besides the java.awt package?

Our answer

Visit the documentation for java.awt.Color What else can you do with Color?

You can get the transparency of a pixel using getTransparency() or convert from RGB to HSB (a different way of representing colors). You can also print a color using toString().

Visit the documentation for Java's API's What are some other classes in the java.awt package?

Click on "java.awt" in the menu on the right hand side of the page. Packages are listed in the smaller box at top right, while classes are listed in the longer box at bottom right. You will find that there are lots of classes in java.awt. Some examples include: Font, Image, and Button.

What are some other packages besides the java.awt package?

Packages are listed in the smaller box at top right, while classes are listed in the longer box at bottom right. You will find lots of packages from there. Some examples include: java.applet, java.io, java.sql, java.text, and so on.

package

(definition) A package is a group of related classes. Java comes with many packages that are very useful. These are in Java's *library*. Packages have names (*e.g.*, java.awt). The objects in a package can be accessed using the syntax *packagename.ClassName* (*e.g.*, java.awt.Color).

You can import more than one class at a time by writing: import *packagename.*;* For example, import java.awt.* imports all classes in the java.awt package. If you want to use specific class from the package, then you can import that class by specifing the class name after the package. For example, in the previous module we told you to import java.awt.Color, which imports only the Color class from the java.awt package. You have to put these codes for importing classes at the beginning of each java file (first lines of the code for each java file). Don't forget to include semi-colon at the end of each code line.

did I get this

Suppose you want to import the java.awt.Color class. Which of the following import statements would work? What is the difference between the two?
import java.awt.Color;
import java.awt.*;

Hint: The '*' keyword means 'all'. If you put the .* keyword at the end when you import the package, that means you are importing all classes from a certain class.

Our answer

Both ways would work. The first import statement would only import the Color class, while the second import statement would import all classes in the java.awt package.

Alternatively, if for some reason you do not want to import a package, you can use a fully qualified class name to refer to it. Thus instead of using Color, you would use java.awt.Color in every instance where you would have just referred to Color, ensuring tha Java is using the correct class.

If you forget to import a class, but try to use its name in your program, it will not compile and you will get an undefined class error.

What does an undefined class error mean? It means that you have tried to make use of a class Java knows nothing about. Some common reasons this could occur are that you forgot to import the class, or you misspelled a class name, or you used the wrong case for the class name (remember that Java is case sensitive!).

Summary

We have talked about libraries, classes, and packages before. Here we show how to import a class or package so that you can use it to define new objects. If a class (or its package) has not been imported correctly, you will get an "Undefined class" error.

To be specific, If you want to use a specific library class (like Color), you have a couple of choices for how to access it.

- You can type import java.awt.Color; or import java.awt.*;

  If you type that in the interactions pane, you should be able to simply refer to Color from then on in the interactions pane (until it is reset). If you are using Color in a java file, you should put the import statement right at the top of the file, and from then on you will be able to simply use Color in your code.

- You can use java.awt.Color each time you need an to refer to the type Color. For example you might write java.awt.Color red = java.awt.Color.RED;
- If you forget to import a class, but try to use its name in your program, it will not compile and you will get an undefined class error.