Packages

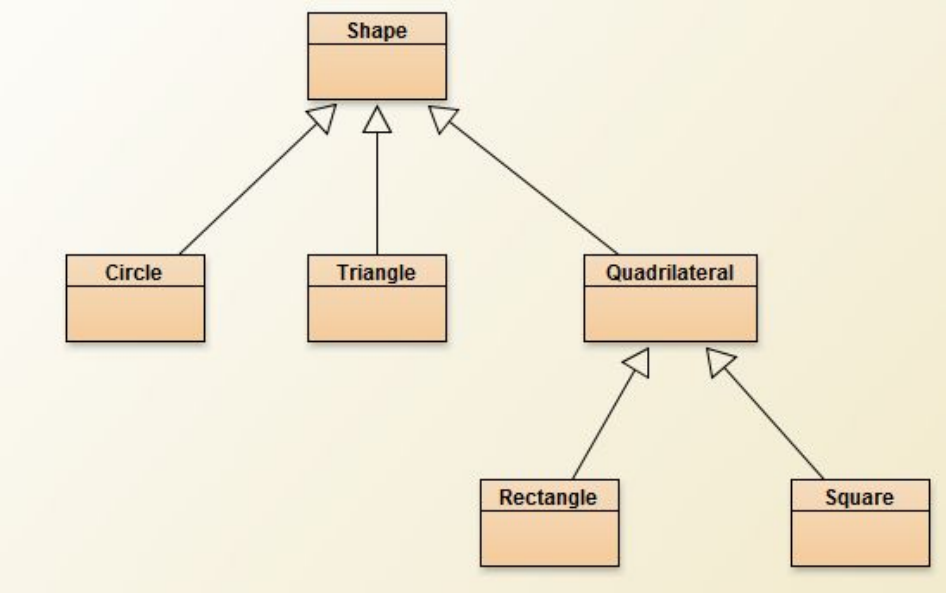
Packages are a useful tool that can help organize related classes together. In order to package a class use the following syntax:

```
package directory.subdirectory;
```

In a file, the package statement must be the first line of code in the program, or else a compiler error will happen. In order to compile this code, accounting for the package statement, use the following command:

```
javac -d . *.java
```

This will create the needed directories and subdirectories, and place the `.class` files in the appropriate directories/subdirectories. Let's take a look at an example with the structure below, where the Shape class is the highest superclass.



A smart way to organize these classes is by hierarchy through packages. Let's make individual packages for each individual class. In order to package the `Shape.class` file into a directory named `shape` you would use the statement:

```
package shape;
```

Since there is no subdirectory, we leave that section out. After compilation, this will create the shape directory, and the `Shape.class` file should be located inside it. However, an example of packaging a class into a directory and a subdirectory can be seen with the `Square class`. In order to package the `Square.class` file into a directory named shape, a subdirectory named quadrilateral, and subdirectory inside `quadrilateral` called `quadtypes` you'd write:

```
package shape.quadrilateral.quadtypes;
```

Upon compiling, `Square.class` should be found in the directory:

```
shape/quadrilateral/quadtypes.
```

Note: Using the default package with an IDE such as NetBeans or Eclipse may lead to unexpected errors.

COLLEGE OF INFORMATION SCIENCE & TECHNOLOGY
COMPUTER SCIENCE LEARNING CENTER

This workforce product was funded by a grant awarded by the U.S. Department of Labor's Employment and Training Administration. The product was created by the grantee and does not necessarily reflect the official position of the U.S. Department of Labor. The U.S. Department of Labor makes no guarantees, warranties, or assurances of any kind, express or implied, with respect to such information, including any information on linked sites and including, but not limited to accuracy of the information or its completeness, timeliness, usefulness, adequacy, continued availability, or ownership.

Packages

The following example code displays the first lines of the Shape, Quadrilateral, and Rectangle classes. It helps show the relationships between importing and packaging.

Shape Class

`package shape;`
 	
`public class Shape {`
`  /*Insert constructor and other methods*/`
`}`
	The Shape.class file will be packaged into the shape directory.

Quadrilateral Class

`package shape.quadrilateral;`
`import shape.Shape;`
 	
`public class Quadrilateral extends Shape {`
`  /*Insert constructor and other methods*/`
`}`
	The Quadrilateral class inherits from the Shape class. Since Shape was packaged, the import statement tells the Quadrilateral class where to find Shape's .class file.

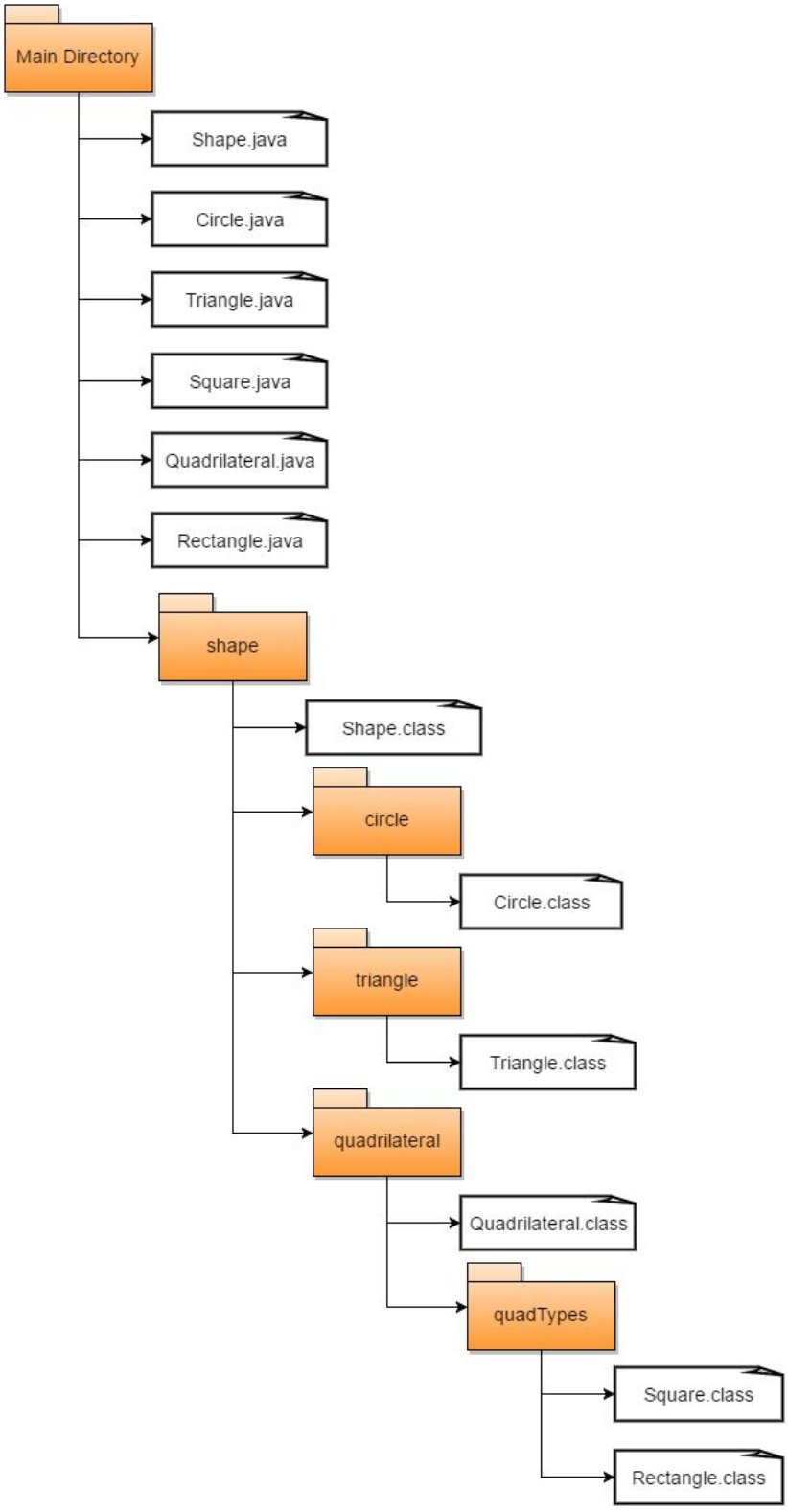
Rectangle Class

`package shape.quadrilateral.quadtypes;`
`import shape.quadrilateral.Quadrilateral;`
 	
`public class Rectangle extends Quadrilateral {`
`  /*Insert constructor and other methods*/`
`}`
	The Rectangle class inherits from Quadrilateral, which was packaged. Therefore the import statement links is used to find the Quadrilateral.class inside the directory it was packaged into.

As stated before, after compilation the necessary directories and subdirectories are created, and the .class files are placed into their appropriate directories/subdirectories. Using the previous example, the next page shows visually what the file structure would look like after compiling.

This workforce product was funded by a grant awarded by the U.S. Department of Labor's Employment and Training Administration. The product was created by the grantee and does not necessarily reflect the official position of the U.S. Department of Labor. The U.S. Department of Labor makes no guarantees, warranties, or assurances of any kind, express or implied, with respect to such information, including any information on linked sites and including, but not limited to accuracy of the information or its completeness, timeliness, usefulness, adequacy, continued availability, or ownership.

Packages



COLLEGE OF INFORMATION SCIENCE & TECHNOLOGY
COMPUTER SCIENCE LEARNING CENTER

This workforce product was funded by a grant awarded by the U.S. Department of Labor's Employment and Training Administration. The product was created by the grantee and does not necessarily reflect the official position of the U.S. Department of Labor. The U.S. Department of Labor makes no guarantees, warranties, or assurances of any kind, express or implied, with respect to such information, including any information on linked sites and including, but not limited to accuracy of the information or its completeness, timeliness, usefulness, adequacy, continued availability, or ownership.