Java packages

Lewis Chapter 11
Savitch Chapter 6.7

Creating a Java Package

public class Rectangle extends Shape
{
    double width, height;
    public
    Rectangle(int x, int y, double h, double w){
        super(x, y);
        width = w;
        height = h;
    }
}

Rectangle.java

// a shape stores its position
// on the screen
public abstract class Shape {
    int x, y;
    public Shape(int x, int y){
        this.x = x;
        this.y = y;
    }
}

Shape.java

package shapes;
public abstract class Shape {
    int x, y;
    public
    Shape(int x, int y){
        this.x = x;
        this.y = y;
    }
}

Shape.java

package shapes;
public class Circle extends Shape {
    double radius;
    public
    Circle(int x, int y, double r){
        super(x, y);
        radius = r;
    }
}

Circle.java

package shapes;
public interface Draggable {
    ...
}

Draggable.java

Creating a Java Package

package shapes;
public abstract class Shape {
    int x, y;
    public
    Shape(int x, int y){
        this.x = x;
        this.y = y;
    }
}

Shape.java

package shapes;
public class Circle extends Shape {
    double radius;
    public
    Circle(int x, int y, double r){
        super(x, y);
        radius = r;
    }
}

Circle.java

package shapes;
public class Rectangle extends Shape {
    double width, height;
    public
    Rectangle(int x, int y, double h, double w){
        super(x, y);
        width = w;
        height = h;
    }
}

Rectangle.java

A package defines a namespace
If you do not use a package statement, your class or interface ends up in the default package, which is a package that has no name.
Using packages

- Only public package members are accessible outside the package in which they are defined. To use a public package member (class, interface) from outside its package, you must either:
  - Refer to the member by its long (disambiguated) name.
    - `java.awt.Rectangle`
  - Import the member's entire package (not recommended).
    - `import java.awt.*;
      Rectangle rectangle = new Rectangle();`
  - Import the package member (recommended).
    - `import java.awt.Rectangle;
      Rectangle rectangle = new Rectangle();`

Package naming

- Package naming convention
  - The name is lower case so it isn't confused with a type or interface
  - Often companies use the reverse of their domain name, e.g., `com.example.orion`
  - All provided packages start with `java` or `javax`.

Summary

- Packages:
  - Group together related Java types
  - Help avoid name conflicts
  - Provide access control

- For more information:

Exceptions revisited

- Until now you only used predefined Java exceptions.
- You can write your own!
- Why would you want to do that?

Example

```java
public class DivideByZeroException extends Exception {
    public DivideByZeroException() {
        super("Divide by zero");
    }
    public DivideByZeroException(String message) {
        super(message);
    }
}
```