Object Oriented Programming

Week 6 Part 2
Using Packages

Lecture

- Defining Packages
- Using Packages
- Using classes from packages
- Using packages when there is a name collisions

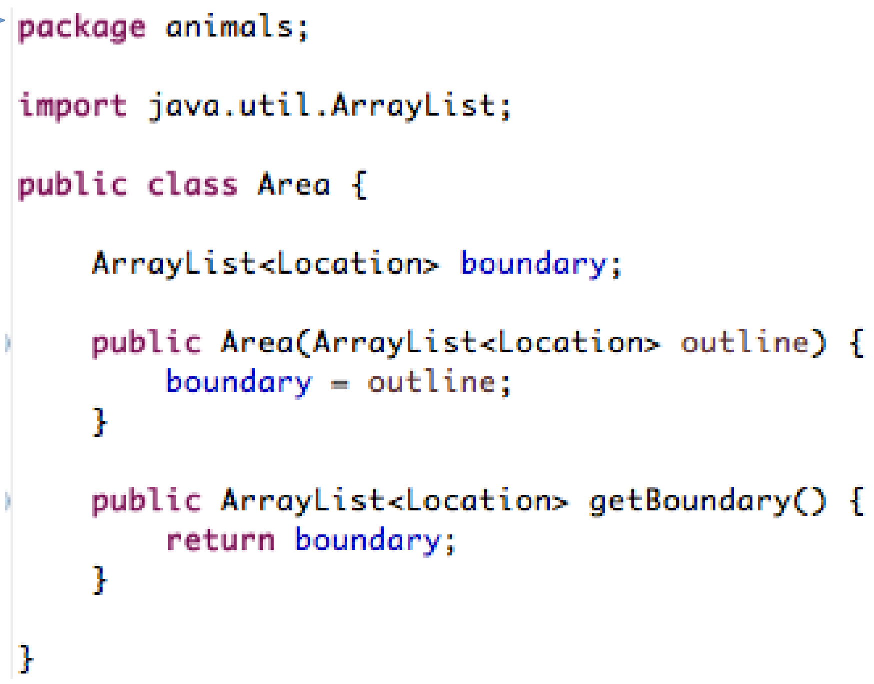
Defining Packages

- By putting “package <package name>” at the top of a file that defines a class, we put that class in the package
- Example: Region.java

Example: Defining Package

Everything in file is in package animals

Class Area is in class animals

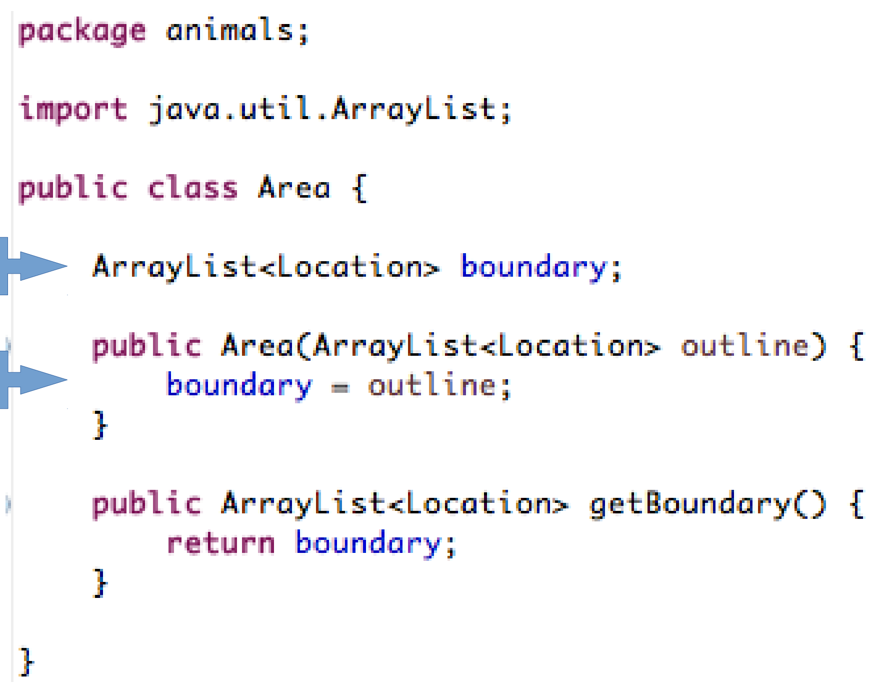


```java
package animals;

import java.util.ArrayList;

public class Area {

    ArrayList<Location> boundary;

    public Area(ArrayList<Location> outline) {
        boundary = outline;
    }

    public ArrayList<Location> getBoundary() {
        return boundary;
    }

}
```

Using Packages

- To use a package we use the keyword import
  - E.g. “import animals.*”: import all of the classes in animals
  - E.g. “import animals.Region”: import the Region class from the package animals

Example: Using Packages



Import ArrayList from java.util

Use ArrayList as if defined in animals

ArrayList methods also available

```java
package animals;

import java.util.ArrayList;

public class Area {

    ArrayList<Location> boundary;

    public Area(ArrayList<Location> outline) {
        boundary = outline;
    }

    public ArrayList<Location> getBoundary() {
        return boundary;
    }

}
```

Using Classes in Packages

- What if two classes from different packages have the same name
- There is a class javax.swing.plaf.synth.Region
  - It defines an area in a User Interface
  - What if we want to use javax.swing.plaf.synth.Region and animals.Region?
    - We use the fully qualified names

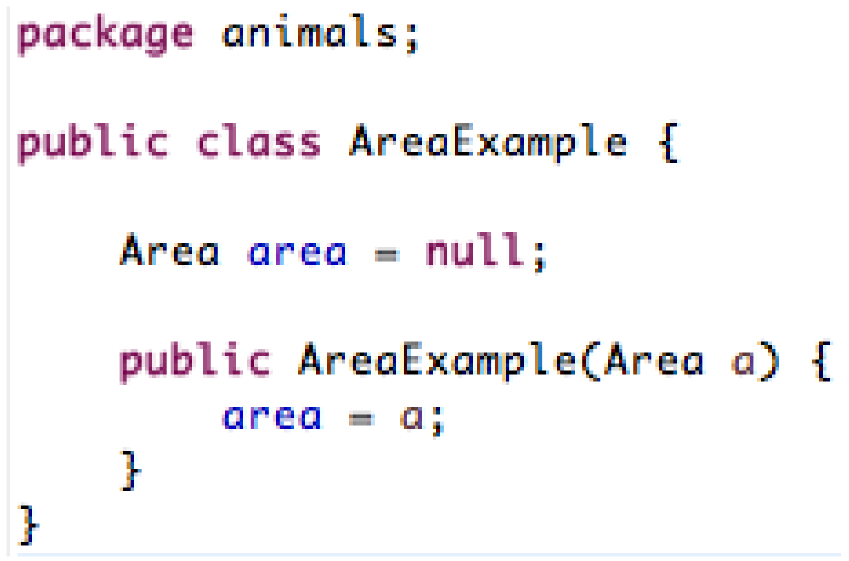
Dealing with Name Conflicts

- There is a class called “Area” in java.awt.geom.Area for an area in the UI framework AWT
- What if we want to show our animals.Area in an java.awt.geom.Area?
  - We need to use fully qualified names.
- If you import java.awt.geom.Area, the imported class will be used.
- If you do not import Area, the class in the animals package will be used.

Using animals Area

In animals package

Area refers to animals.area

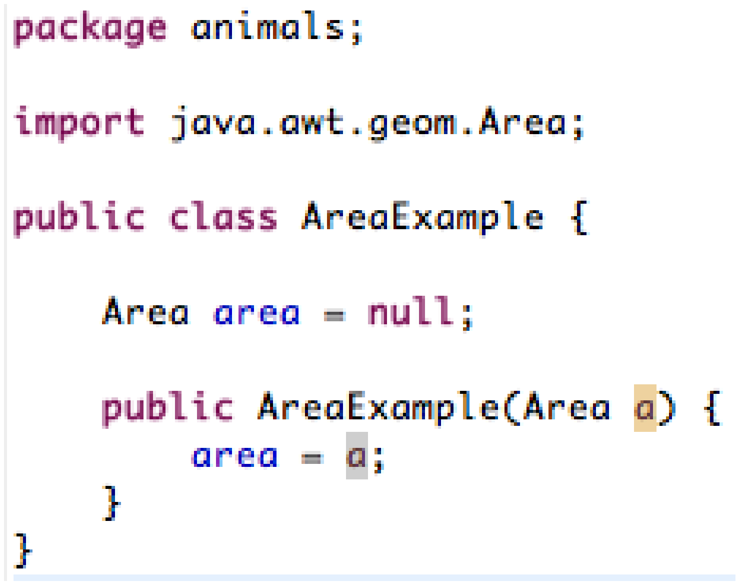


```
package animals;

public class AreaExample {

    Area area = null;

    public AreaExample(Area a) {
        area = a;
    }
}
```

Using AWT Area

Import java.awt.geom.Area

Refers to java.awt.geom.Area

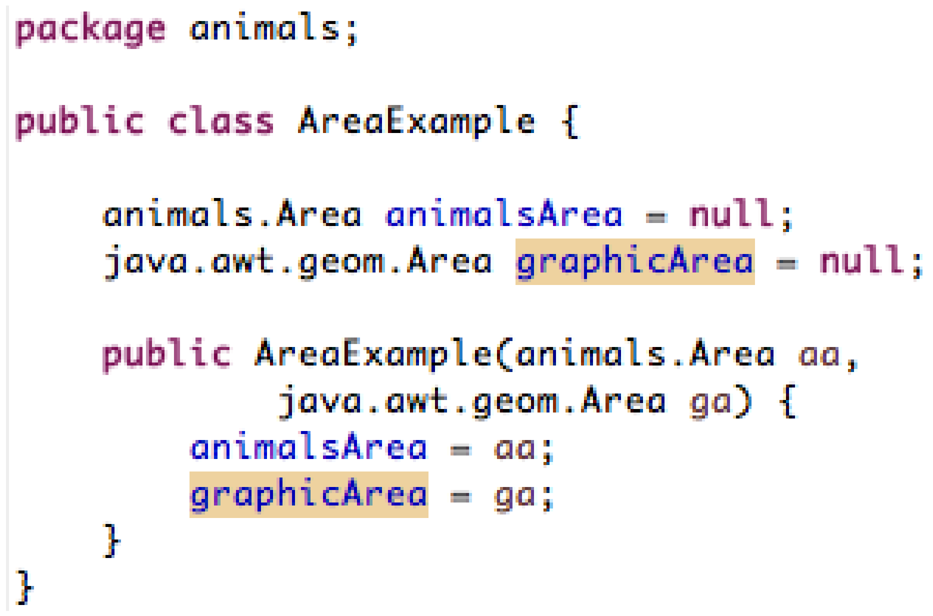


```
package animals;

import java.awt.geom.Area;

public class AreaExample {

    Area area = null;

    public AreaExample(Area a) {
        area = a;
    }
}
```

Correct Way

Use fully qualified names for both

You could use just Area for animals.Area, but this is going to be confusing to the next person trying to figure out what this program does



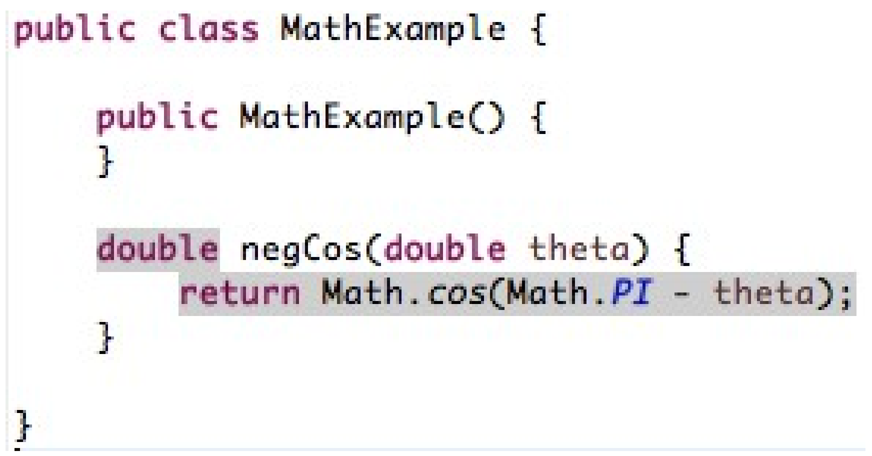
```
package animals;

public class AreaExample {

    animals.Area animalsArea = null;
    java.awt.geom.Area graphicArea = null;

    public AreaExample(animals.Area aa,
            java.awt.geom.Area ga) {
        animalsArea = aa;
        graphicArea = ga;
    }
}
```

Static Import

- When you use static methods and fields (i.e. constants) you need to include the name of the class.
- You may import the static methods and constants from a class using static import
- For example, the Math class has many mathematical constants and function defined as static fields and methods
  - It would be nice to be able to say PI instead of Math.PI

Static Import Example: Math

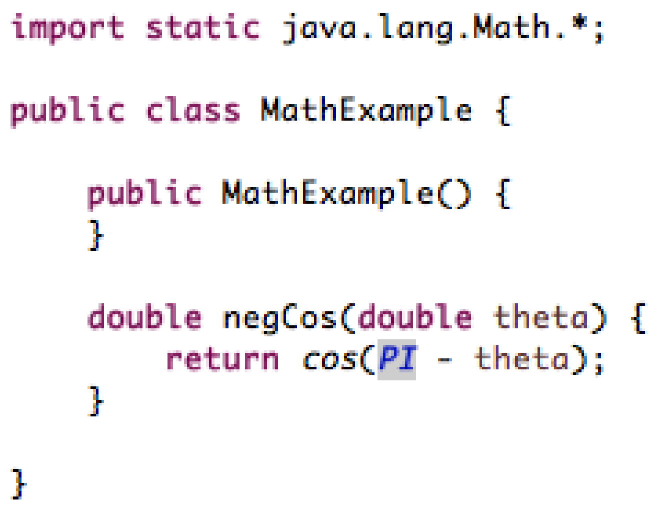
Math.PI and Math.cos



```
public class MathExample {

    public MathExample() {
    }

    double negCos(double theta) {
        return Math.cos(Math.PI - theta);
    }

}
```

Import static java.lang.Math

PI and cos



```
import static java.lang.Math.*;

public class MathExample {

    public MathExample() {
    }

    double negCos(double theta) {
        return cos(PI - theta);
    }

}
```

CLASSPATH

- Java looks for classes in the directories specified in the CLASSPATH environment variable
- To see the classpath
  - Window: C:\> set CLASSPATH
  - Unix: % echo $CLASSPATH
- To add to the classpath <new> to CLASSPATH
  - Windows
    - C:\> set CLASSPATH=CLASSPATH;<new>
  - Unix
    - % CLASSPATH='$CLASSPATH;<new>'; export CLASSPATH