Customizing Your UI
Inner Classes

• Consider a DataStructure outer class with methods to:
  – 1) add an integer
  – 2) print values of even indices

• Consider an inner class InnerEvenIterator with methods to:
  – 1) test for last element
  – 2) retrieve the current element and move to the next element

http://java.sun.com/docs/books/tutorial/java/javaOO/innerclasses.html
Examine DataStructure.java

- Import 09.Code\InnerClasses
Java enum

• We are familiar with C/C++ enums
• At first glance Java enums seem the same

```java
public class Enums {
    public enum Day {MON, TUE, WED, THUR, FRI, SAT, SUN}

    /**
     * @param args
     */
    public static void main (String[] args) {
        Day day = Day.MON;
        System.out.println (day); //MON is printed
    }
}
```
Java enum continued

• Java enums are MUCH more powerful (confusing) than other languages
• The enum defines a class (therefore each enum value is an object!!!!)
• The enum class body can include other fields and methods
• The compiler automagically creates an array containing ALL enum values in the order declared for easy iteration through the enum list
Examine Planet.java

• Import 09.Code\Planets

• Run the program using your weight

• Debug the program setting a break point at the first line of the constructor AND the first line of main

• Step through the program
Leveraging Inner Classes

• Import 09.Code\LayoutUtils

• Each row is to dissect the class LayoutUtils

• Using LayoutUtils, create the following Android UI
Creating Custom Widgets

• Custom Widgets can easily be created by:

  1. Extending an existing View or View subclass

  2. Overriding superclass methods of interest (e.g. onDraw, onKeyDown, ...)

Let’s Modify onDraw

• Import 09.Code\CustomView

• Modify the onDraw method to
  1. Turn the canvas blue before outputting
  2. Change all text in the View to uppercase