CS1110 27 Jan 2009 Customizing a class

Summary of lectures: On course home page, click on “Lectures” and then on “Outline of lectures held so far”.

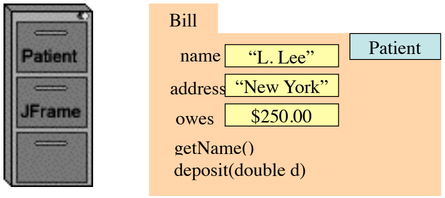
Reading for this lecture: Sections 1.4, (p. 41); 13.3.1 (p. 376).

Read all “style notes” and referenced PLive lectures (activities).

Quote for the day:
I have traveled the length and breadth of this country and talked with the best people, and I can assure you that data processing is a fad that won't last out the year.
—Editor in charge of business books for Prentice Hall, 1957

Reading for next lecture:
- Fields; getter & setter methods. Secs 1.4.1 (p. 45) & 3.1 (pp. 105–110 only)
- Constructors. Sec. 3.1.3 (p. 111–112)
- Testing. App. I.2.4 (p. 486)

A class is a file-drawer. Contents: manila folders.



Bill

name “L. Lee”

address “New York”

owes $250.00

Patient

getName()
deposit(double d)

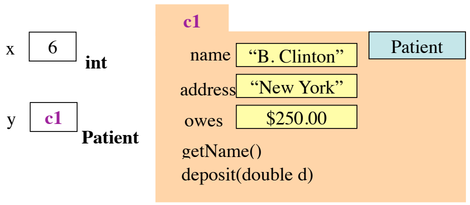
(1) unique name on tab of manila folder.

(2) manila folder, instance, object of the class

(3) fields (variables)

(4) methods (procedures and functions): instructions to do tasks and to produce values.

This reviews what we did last time.



x 6 **int**

y c1 **Patient**

c1

name “B. Clinton”

address “New York”

owes $250.00

Patient

getName()
deposit(double d)

x has value 6 y has value c1

y.getName() has the value “B. Clinton”

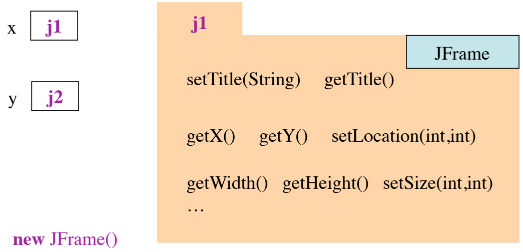
y.deposit(250) ; will change the value of field owes to 0.

This reviews what we did last time.

procedure call

function call

Class javax.swing.JFrame: an object is a window on your monitor.



x j1

y j2

j1

JFrame

setTitle(String) getTitle()

getX() getY() setLocation(int,int)

getWidth() getHeight() setSize(int,int)

…

new JFrame()

Expression: create a new object of class JFrame and yield its name

This reviews what we did last time.

Class definition: The java construct that describes the format of a folder (instance, object) of the class.

```
/** description of what the class is for
    */
public class <class-name> {
      declarations of methods (in any order)
}
```

This is a comment

A class definition goes in its own file named

<class-name> . java

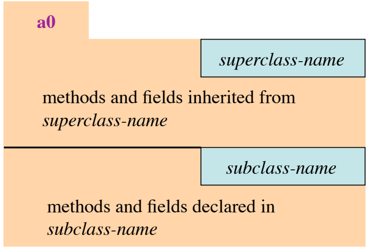
On your hard drive, have a separate directory for each Java program that you write; put all the class definitions for the program in that directory.

Class definition: The java construct that describes the format of a folder (instance, object) of the class.

```
/** description of what the class is for
    */
public class C  extends <superclass-name>   {
      declarations of methods (in any order)

}
```

Class C has all the fields and methods that <*superclass-name*> does, in addition to those declared in C. Class C **inherits** the fields and methods of <superclass-name>.

```
/** description of what the class is for */
public class subclass-name extends superclass-name {
          declarations of methods
}
```


a0
superclass-name
methods and fields inherited from superclass-name
subclass-name
methods and fields declared in subclass-name

folder (object) belongs in file drawer for class *subclass-name*

First example of a procedure and of a function

```
/** description of what the class is for */
public class subclass-name extends superclass-name {
          /** Set the height of the window to the width */
          public void setHeightToWidth() {
                 setSize(getWidth(), getWidth());
          }

          /** = the area of the window */
          public int area() {
                 return getWidth() * getHeight();
          }
}
```

```
import javax.swing.*;
/** An instance is a JFrame with methods to square it and
     to provide the area of the JFrame */
public class SquareJFrame extends JFrame {
          declarations of methods
}
```

folder (object) belongs in file drawer for class SquareJFrame

To the left, draw a manila folder of class SquareJFrame. When we define methods, put them in the proper place

Javadoc

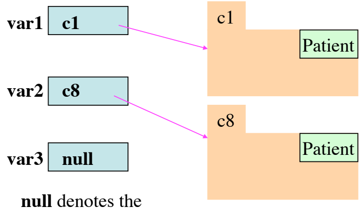
```
import javax.swing.*;
/** An instance is a JFrame with methods to square it and
     to provide the area of the JFrame */
public class SquareJFrame extends JFrame {
          /** = the area of the window */
          public int area() { … }

          /** Set the height equal to the width */
          public void setHeightToWidth() {…}
}
```

The class and every method in it has a comment of the form

/ specification */**

It is a Javadoc comment. Click on javadoc icon in DrJava to extract class specification. DO THIS AT LEAST ONCE IN LAB.

About null



null denotes the absence of a name.

var3.getName() is a mistake! You get a NullPointerException