CS1007: Object Oriented Design and Programming in Java

Lecture #20

April 4

Shlomo Hershkop
shlomo@cs.columbia.edu

Outline

- Some brief background to help with homework
- Java beans
- Some interesting programming challenges
- Next time:
  - Applet programming
  - Multi process/multi threaded environments
- Reading for next time: 9-9.2

Announcement

- Next week Thursday, is PASSOVER
- Choice:
  - Review/overview class by TA
  - Day off to the work on assignment
  - Bring your laptop to class day and TAs will help with specific programming ideas
  - Sleep
- Due to the holiday, I will have makeup office hours TBD
- The Ta's will be around more often to help with the hw2

Homework help

- Idea: hopefully you've started on the homework already
- Will try to help you out to understand some fundamental ideas
- Please come see me if you are getting stuck

Pitfalls

- Be careful to understand how graphics and graphic objects are created and treated in JAVA
- Problem: everything works great … except

redraw

- Problems with redraw
  - Update java 1.0.5.0.4 -> 1.0.5.0.6
- Understand how things hook to each other
- Lets look at a simple tictacgame

logic

- Everything works great, except when you play a certain move after 11am on the third Tuesday of the month

Jar files

- Jar files are the equivalent of a zip file with some added information for allowing groups of classes to be bundled together
- Can be used to extract and compress files
- Use of Manifest file to give instructions to the system
  - Which class is main
  - Specify a path of the outside classes
  - metadata

Manifest file example

```
Manifest-Version: 1.0
Main-Class: metdemo.winGui
Class-Path: . lib/derby.jar lib/mail.jar
```

Jar compression

- Jar cvfm some.jar Manifest.txt path/*.class

Jar decompression

- Jar –xvf some.jar
- Can also only list the contents without extracting

Advantages of Jar files

- One file to send to someone
- Can be treated almost like executable if done right
- Can sign itself to guarantee its authenticity

Question of the day

- How would you program a maze game ?

Maze

- How do you represent the game ?
- How would you solve the maze?
- What is the fastest/shortest way to solve the maze?

From object oriented view

- MazeGame mzg = new MazeGame(40,40);
- //assume start is 0,0 end is max,max
- msg.getOptions(x,y);
- Write a solution …

Beyond Objects

- Object represent a single concept (usually)
- Sometimes hard to reuse in complex behavior
- Would like an idea of a Object, a few object, which we can add some behavior necessary to accomplish a specific task

Take away lesson

- Question:
- If I don’t plan on lots of programming, why should I care about this?

- Programming is a way of expressing an idea of an algorithm
- Allows you to solve much larger sets of problems
- More interesting sets of problems (I’ve been introducing them slowly)

- Will allow you to cut through marketing jargon
- Read other peoples code
- Understand or misunderstand CPP templates

OO

- That is not a pair of glasses up there
- We keep stressing the main idea of Object oriented programming approach
- How do you translate these ideas to non programmers

- Anyone hear of visual basic??
- Successful model: Visual Basic controls
  - calendar
  - graph
  - database
  - link to robot or instrument

Bottom line

- Anyone can now write a virus ☺

Component model

- Anyone have an ipod ?
- How do you plug in headphones?
- How do you plug it into your home entertainment system?
- Any of those steps need a hammer and screwdriver ?

Idea of Components

- More functionality than a single class
- Reuse and customize in multiple contexts
- "Plug components together" to form applications

- Components composed into program inside builder environment
- Target all users, not just programmers

Java beans

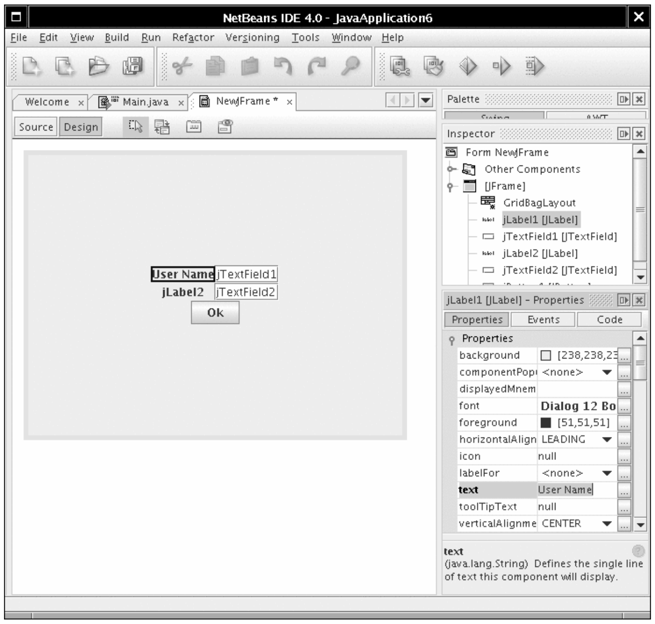
- A reusable piece of code which satisfies the requirements of the JavaBeans framework that can be manipulated by an IDE designed to work with JavaBeans

Some requirements

1. Consistency
   rules governing the consistency of name conventions (methods and types of methods)
2. Event Handling Model
   SWT and Swing were implemented in this framework
3. Persistence
   can extend values beyond single session
4. Introspection
   ability to find which method/args are taken
5. Builder Support
   IDE

Builder support

- JavaBeans should be able to be created and programmed by anyone who can use a mouse
- Think of video game programming framework


NetBeans IDE 4.0 - JavaApplication6
File Edit View Build Run Refactor Versioning Tools Window Help
Welcome
Main.java
NewJFrame *
Source
Design
User Name
jTextField1
jLabel2
jTextField2
Ok
Palette
Inspector

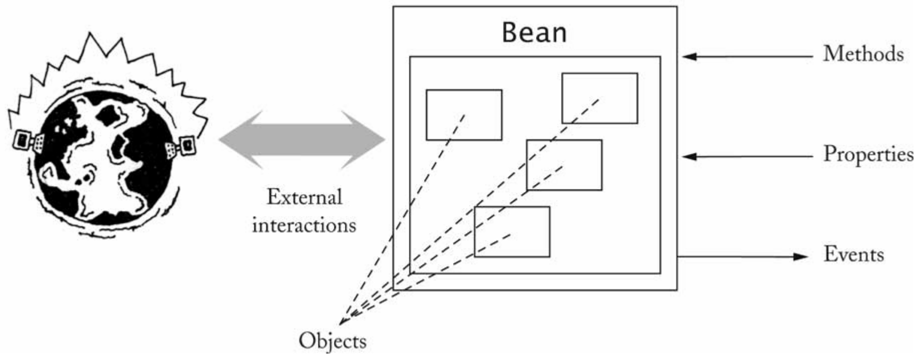
Form NewJFrame
- Other Components
- [JFrame]
  - GridBagLayout
  - jLabel1 [JLabel]
  - jTextField1 [JTextField]
  - jLabel2 [JLabel]
  - jTextField2 [JTextField]

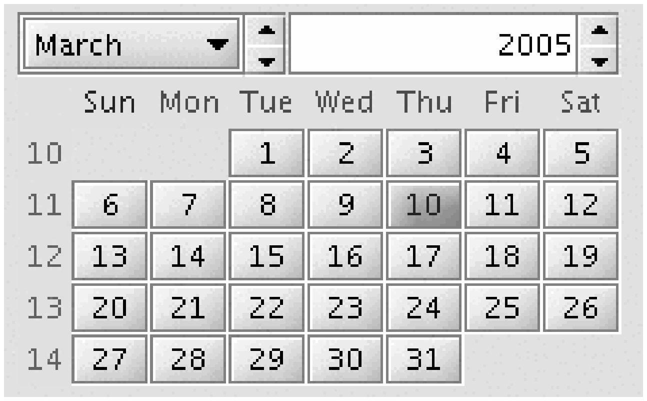
jLabel1 [JLabel] - Properties

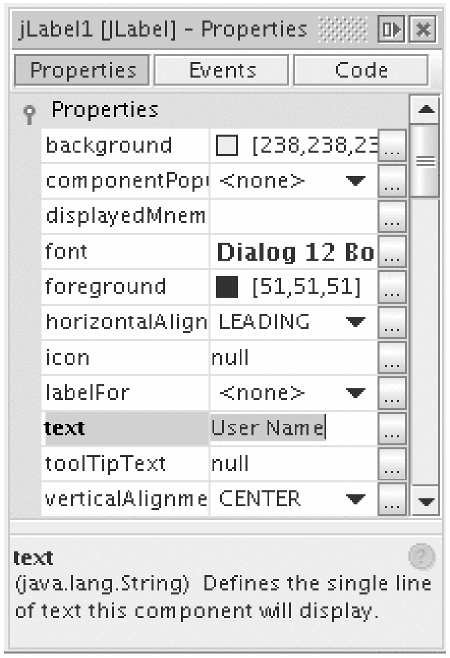
Properties | Events | Code

Properties	
background	[238,238,23
componentPopu	<none>
displayedMnem	
font	Dialog 12 Bo
foreground	[51,51,51]
horizontalAlign	LEADING
icon	null
labelFor	<none>
text	User Name
toolTipText	null
verticalAlignme	CENTER

text
(java.lang.String) Defines the single line of text this component will display.







jLabel1 [JLabel] - Properties

Properties | Events | Code

Properties

background	☐ [238,238,23
componentPop	<none>
displayedMnem	
font	**Dialog 12 Bo**
foreground	■ [51,51,51]
horizontalAlign	LEADING
icon	null
labelFor	<none>
text	User Name
toolTipText	null
verticalAlignme	CENTER

text
(java.lang.String) Defines the single line of text this component will display.

Property sheet

Façade pattern

- We have a bunch of classes working together
- Want to allow easy use
- Want to change things in the background without changing everything

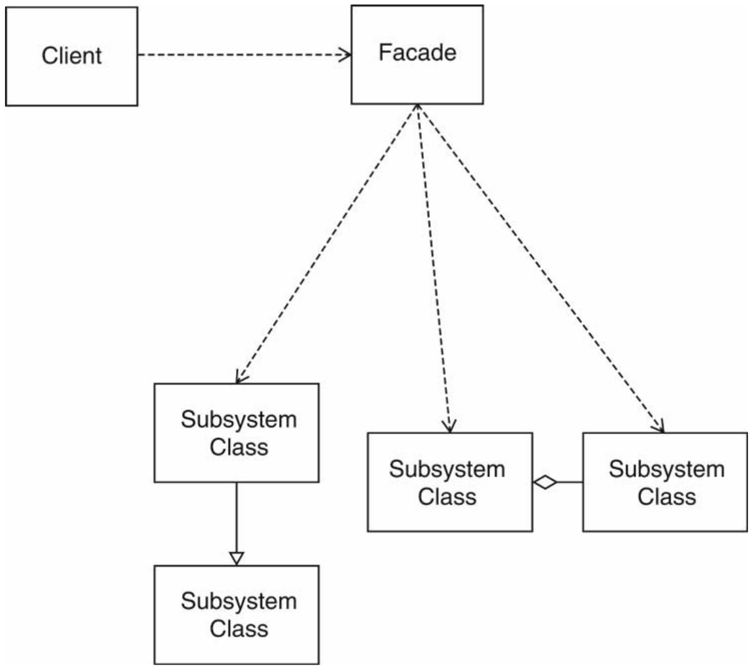
- Sounds familiar ??

Façade class

- Bean usually composed of multiple classes
- One class nominated as facade class
- Clients use only facade class methods

How JAVABEAN does it

- Define a facade class that exposes all capabilities of the subsystem as methods
- The facade methods delegate requests to the subsystem classes
- The subsystem classes do not know about the facade class



Bean Properties

- Property = value that you can get and/or set
- Most properties are get-and-set
- Can also have get-only and set-only
- Property not the same as instance field
- Setter can set fields, then call repaint
- Getter can query database

Syntax

- Not Java :-(
- C#, JavaScript, Visual Basic
- b.propertyName = value
  – calls setter in background
- variable = b.propertyName
  – calls property getter

Conventions

- property = pair of methods

```
public X getPropertyName()
public void setPropertyName(X newValue)
```

- Replace propertyName with actual name

```
(e.g. getColor/setColor)
```

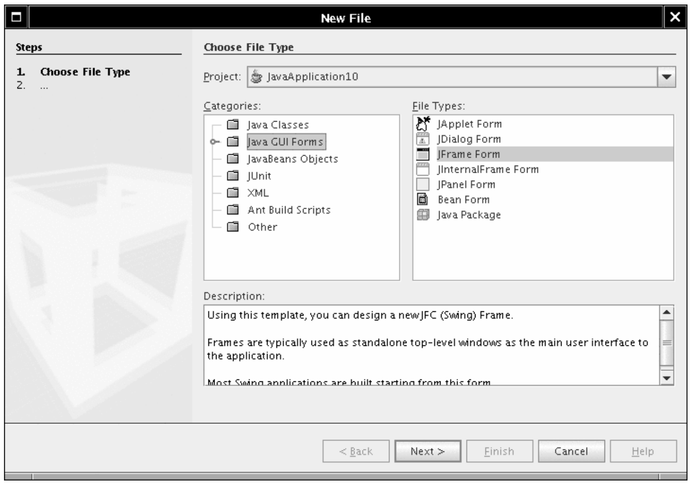
- Exception for boolean properties:

```
public boolean isPropertyName()
```

- Decapitalization hokus-pokus:

```
getColor -> color
getURL -> URL
```

Builder tool



Project: JavaApplication10

Categories: Java GUI Forms

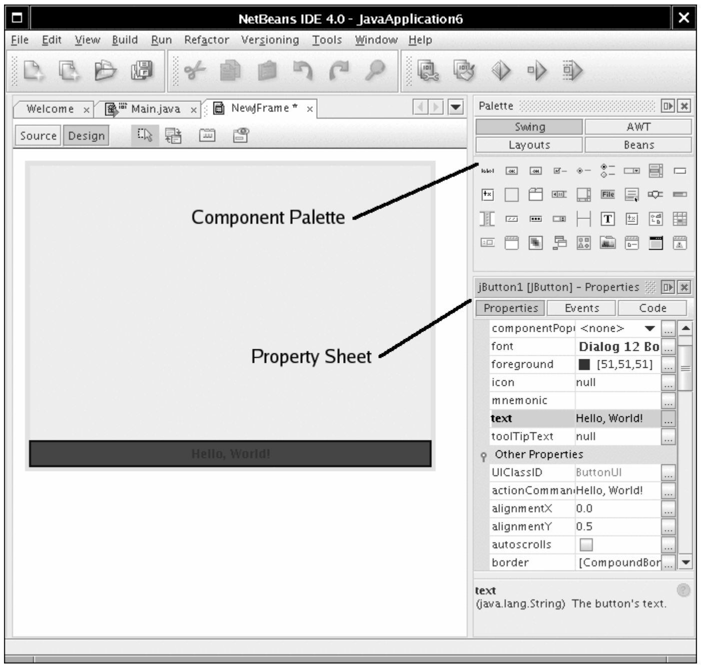
File Types: JFrame Form

Description:

Using this template, you can design a new JFC (Swing) Frame.

Frames are typically used as standalone top-level windows as the main user interface to the application.

Most Swing applications are built starting from this form.



jButton1 [JButton] - Properties

Properties | Events | Code

componentPop	<none>
font	Dialog 12 Bo
foreground	■ [51,51,51]
icon	null
mnemonic	
text	Hello, World!
toolTipText	null
Other Properties	
UIClassID	ButtonUI
actionComman	Hello, World!
alignmentX	0.0
alignmentY	0.5
autoscrolls	☐
border	[CompoundBor

text
(java.lang.String) The button's text.

Packaging

- Compile bean classes

Ch7/carbean/CarBean.java

- Create manifest file

Ch7/carbean/CarBean.mf

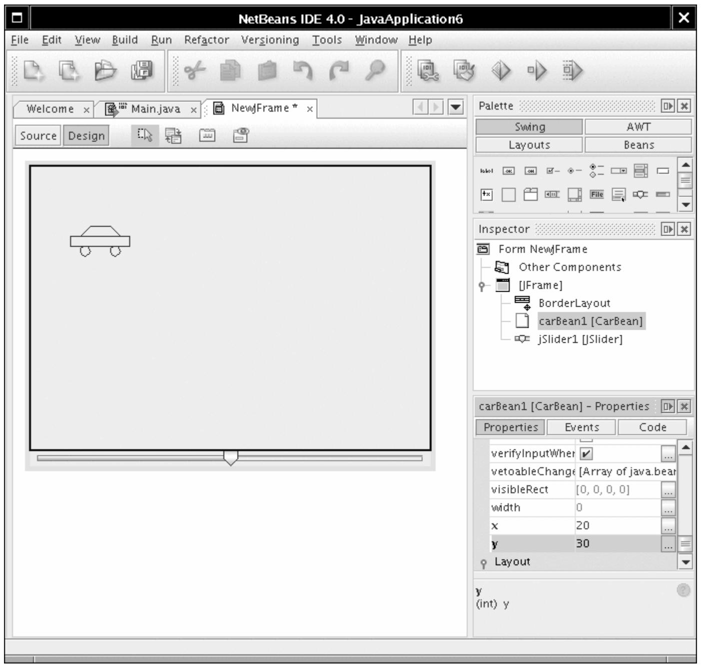
- Run JAR tool:
- jar cvfm CarBean.jar CarBean.mf *.class
- Import JAR file into builder environment

Composing Bean

- Make new frame
- Add car bean, slider to frame
- Edit stateChanged event of slider
- Add handler code

```
carBean1.setX(jSlider1.getValue());
```

- Compile and run
- Move slider: the car moves



NetBeans IDE 4.0 - JavaApplication6
File Edit View Build Run Refactor Versioning Tools Window Help
Welcome
Main.java
NewJFrame *
Source
Design
Palette
Swing
AWT
Layouts
Beans
Inspector
carBean1 [CarBean] - Properties
Properties
Events
Code

Form NewJFrame
- Other Components
- [JFrame]
  - BorderLayout
  - carBean1 [CarBean]
  - jSlider1 [JSlider]

verifyInputWher	✓
vetoableChange	[Array of java.bear
visibleRect	[0, 0, 0, 0]
width	0
x	20
y	30

Layout

y
(int) y

Framework

- Set of cooperating classes
- Structures the essential mechanisms of a problem domain
- Example: Swing is a GUI framework
- Framework != design pattern
- Typical framework uses multiple design patterns

Application framework

- Implements services common to a type of applications
- Programmer forms subclasses of framework classes
- Result is an application
- Inversion of control: framework controls execution flow

Bottom line

- So when would it make sense to work with beans rather than low level code??