CS1007: Object Oriented Design and Programming in Java

Lecture #16
Nov 22
Shlomo Hershkop
shlomo@cs.columbia.edu

Outline

• Java beans
• Applets

• Reading: finish chapter 7, starting 8

Announcements

• 4 more lectures after today
• 1 more homework after today (no actual code needed).
• Plan
• No class Thursday (or office hours).

Feedback

• Questions on concepts
  – Please stop me and I can give more examples
• Generic example
  – Will review some more
• Homework help
  – Will try to help later in class
Generics

• Generally when you manipulate a group of objects, in order for the compiler to be aware of the manipulations (error checking) you need to explicitly cast the general objects you are working with.

```java
List myints = new LinkedList();
myints.add(new Integer(3));
Integer x = (Integer)myints.iterator().next();
```

Errors runtime

• When manipulating objects and casting, if you make a mistake on what you think is in a collection, will throw errors.

Idea

• Allow you to tell the compiler what you are thinking…

```java
List<Integer> myints = new LinkedList<Integer>();
//adding same
Integer x = myints.iterator().next();
```

Another advantage

• Allows you to program algorithms which don’t work in the dark
• Allows you to setup constraints on the objects you allow to be passed to your methods
wildcarding

- Allows you specify general types and bounded general types in your algorithms

Map <String, ? extends employee>

Confused?

- Please review book
- Please review last lecture
- Email myself/TAs
- Can ask other students
- Use internet (not responsible for anything you happen to dig up there).

Introducing Java Beans

- Java component model
- Bean has
  - methods (just like classes)
  - properties
  - events
Façade class

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

What kind of pattern can we extract?

- A subsystem consists of multiple classes, making it complicated for clients to use
- Implementor may want to change subsystem classes
- Want to give a coherent entry point
**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
- variable = b.propertyName calls 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

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
  
  ```java
  carBean1.setX(jSlider1.getValue());
  ```
- Compile and run
- Move slider: the car moves

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
Applet

- Applet: Java program that runs in a web browser
- Programmer forms subclass of Applet or JApplet
- Overwrites
  - init/destroy
  - start/stop
  - paint

Applets

- Interacts with ambient browser
  - getParameter
  - showDocument
- HTML page contains applet tag and parameters
  ```html
  <applet code="BannerApplet.class" width="300" height="100">
    <param name="message" value="Hello, World!">
    <param name="fontname" value="Serif"/>
    <param name="fontsize" value="64"/>
    <param name="delay" value="10"/>
  </applet>
  ```

Example

- Shows scrolling banner
- init reads parameters
- start/stop start and stop timer
- paint paints the applet surface

Ch8/applet/BannerApplet.java
Next

• Finish homework
  – Please email me if you get stuck/clarifications
• Do reading
  – Chapter 8 – 8.5