Recitation 2

Beans and object editor

What is a Bean?

- Object
  - No main method
  - Non-public global variables
  - Has public getters and setters for properties
    - MUST FOLLOW SPECIFIC NAMING CONVENTIONS

Naming conventions for a property

- Given I need a property P of type T:

  - Getter must be called:
    public T getP(){…}

  - Setters must be of the form:
    public void setP(T newValue){…}

Read only vs editable properties

- A Read-only property is one that you cannot directly change, however you should still be able to view its contents
- An editable property is one that should be able to directly pass a value to

Example of Bean Properties:

- Editable property name = Score
- Property type = int
- Read-only property name = Passed
- Property type = boolean

Example Bean code:

```
public class PassFailBean {
  int score;

  public void setScore(int newScore) {
    score = newScore;
  }
  public int getScore() {
    return score;
  }
  public boolean getPassing() {
    if(score<69) {
      return false;
    }
    return true;
  }
}
```

What is object editor?

- Prof. Dewan has created a special UI that allows us to edit objects on the fly to ensure that our projects are working properly.
- Make sure that you have object editor added to EACH of your projects that use it (similar to local checks)
  - Get oeall22.jar from the course site's downloads section and add External JARs by right clicking your java project → properties → java build path → add external JARs

Coding assignment for today

- Create a new java project and package (don't care about the names) and add the external JAR file for the oeall22.jar
- Create a class called "AnUppercaseFilter"
  - Must have two properties:
    - We want to be able to get and set an "inputString" String property. Thus we need an editable property called InputString of type String
    - We want to get, but NOT set an "uppercaseLetters" String property. Thus we need a read-only property called UppercaseLetters of type String

Coding Assignment cont.

- Our class must have two non-public String global variables called “inputString” and “uppercaseLetters”
- In each of your methods (excluding the fix-up method mentioned later) add the line:
  - System.out.println(“METHOD_NAME is called”);

Coding Assignment cont.

- Add the following method to your code:

```
void scanString(){
  uppercaseLetters = "";
    for(int i = 0; i < inputString.length(); i++){
      if (inputString.charAt(i) >= 'A' && inputString.charAt(i) <= 'Z') {
        uppercaseLetters = uppercaseLetters + inputString.charAt(i);
      }
    }
}
```

- Call this method in your setter for inputString

Coding Assignment cont.

- Add the following lines to the top of your bean:

```
import util.annotations.StructurePattern;
import util.annotations.EditablePropertyNames;
import util.annotations.PropertyNames;

@StructurePattern("Bean Pattern")
@PropertyNames({"InputString", "UppercaseLetters"})
@EditablePropertyNames({"InputString"})
```

Coding Assignment final.

- Create a new class in the same package called Driver
- The code for driver is below:

```
import bus.uigen.ObjectEditor;

public class Driver {
  public static void main(String[] args) {
    AnUppercaseFilter filter = new AnUppercaseFilter();
    filter.setInputString("Initial Input");
    ObjectEditor.edit(filter);
  }
}
```

- Please fiddle around with the object editor and then take the quiz on sakai
  - We are here to help and you can chat with your neighbors as well
  - Once you are finished you may leave and have a nice weekend, or stay and get help with your assignment or other quiz