Using and Improving GladLibs

HasMap for Flexible Design

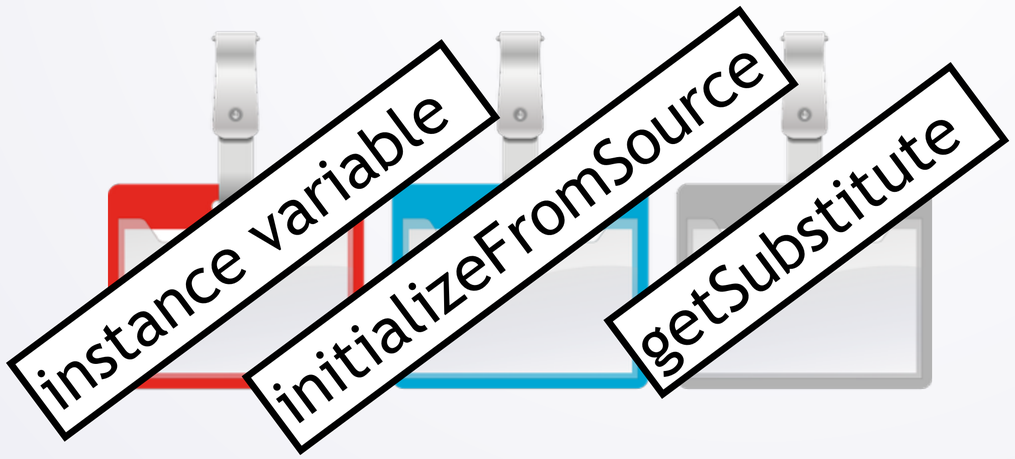
Duke
UNIVERSITY

Summary of Extending GladLib.java

- To add <verb> label must modify code in several parts of the program

Summary of Extending GladLib.java

- To add <verb> label must modify code in several parts of the program
  - Three different parts of the program must be modified
  - Follow conventions: verbList for verb



Summary of Extending GladLib.java

- To add <verb> label must modify code in several parts of the program
  - Three different parts of the program must be modified
  - Follow conventions: verbList for verb
- Difficult to use different .txt files or URLs, names bound to ArrayLists and labels
  - nounList : noun.txt
  - colorList : color.txt

New Structures for Data and Classes

- Each label is associated with an ArrayList
  - <noun> and nounList, <color> and colorList, …
- Named instance variables: poor design
  - Define by name, initialize and use by name
- HashMap class helps with flexible design
  - Map or align <label> to ArrayList, no names!
  - Look up the ArrayList given the <label>
  - Like indexOf, but returns ArrayList for <label>

Flexibility in Using HashMap

- Replace seven or more instance variables by one instance variable

```
private ArrayList<String> adjectiveList;
private ArrayList<String> nounList;
private ArrayList<String> colorList;
private ArrayList<String> countryList;
private ArrayList<String> nameList;
private ArrayList<String> animalList;
private ArrayList<String> timeList;
```

Flexibility in Using HashMap

- Replace seven or more instance variables by one instance variable
  - No new fields to add verbList, or more!

```
private HashMap<String, ArrayList<String>> myMap;
```

```
private ArrayList<String> adjectiveList;
private ArrayList<String> nounList;
private ArrayList<String> colorList;
private ArrayList<String> countryList;
private ArrayList<String> nameList;
private ArrayList<String> animalList;
private ArrayList<String> timeList;
```

Flexibility in Using HashMap

- Replace seven or more instance variables by one instance variable
  - No new fields to add verbList, or more!

```
private HashMap<String, ArrayList<String>> myMap;
```

```
private ArrayList<String> adjectiveList;
private ArrayList<String> nounList;
private ArrayList<String> colorList;
private ArrayList<String> countryList;
private ArrayList<String> nameList;
private ArrayList<String> animalList;
private ArrayList<String> timeList;
```

Flexibility in Using HashMap

- Replace seven or more instance variables by one instance variable
  - No new fields to add verbList, or more!

```
private HashMap<String, ArrayList<String>> myMap;
```

```
private ArrayList<String> adjectiveList;
private ArrayList<String> nounList;
private ArrayList<String> colorList;
private ArrayList<String> countryList;
private ArrayList<String> nameList;
private ArrayList<String> animalList;
private ArrayList<String> timeList;
```

Flexibility in Using HashMap

- Replace seven or more instance variables by one instance variable
  - No new fields to add verbList, or more!

```
private HashMap<String, ArrayList<String>> myMap;
```

```
private ArrayList<String> adjectiveList;
private ArrayList<String> nounList;
private ArrayList<String> colorList;
private ArrayList<String> countryList;
private ArrayList<String> nameList;
private ArrayList<String> animalList;
private ArrayList<String> timeList;
```

Looking up Values in HashMap

- Sequence of if-statements in getSubstitute replaced by one statement with HashMap!

```
private String getSubstitute(String label) {
    if (label.equals("country")) {
      return randomFrom(countryList);
    }
    if (label.equals("color")){
      return randomFrom(colorList);
    }
    // more code here for labels …

    if (label.equals("number")){
      return ""+myRandom.nextInt(50)+5;
    }
```

Looking up Values in HashMap

- Sequence of if-statements in getSubstitute replaced by one statement with HashMap!

```
private String getSubstitute(String label) {
    if (label.equals("country")) {
      return randomFrom(countryList);
    }
    if (label.equals("color")){
      return randomFrom(colorList);
    }
    // more code here for labels …

    if (label.equals("number")){
      return ""+myRandom.nextInt(50)+5;
    }
```

Looking up Values in HashMap

- Sequence of if-statements in getSubstitute replaced by one statement with HashMap!
  - No modifications need to add verbList!
  - **HashMap<String,ArrayList<String>>**

```
private String getSubstitute(String label) {
   if (label.equals("number")){
     return ""+myRandom.nextInt(50)+5;
   }
   return randomFrom(myMap.get(label));
 }
```

Looking up Values in HashMap

- Sequence of if-statements in getSubstitute replaced by one statement with HashMap!
  - No modifications need to add verbList!
  - **HashMap<String,ArrayList<String>>**

```
private String getSubstitute(String label) {
   if (label.equals("number")){
     return ""+myRandom.nextInt(50)+5;
   }
   return randomFrom(myMap.get(label));
 }
```

Looking up Values in HashMap

- Sequence of if-statements in getSubstitute replaced by one statement with HashMap!
  - No modifications need to add verbList!
  - **HashMap<String,ArrayList<String>>**

```
private String getSubstitute(String label) {
   if (label.equals("number")){
     return ""+myRandom.nextInt(50)+5;
   }
   return randomFrom(myMap.get(label));
 }
```

Improvement, but Room for More!

- In GladLib.java must assign values to each named instance variable

```
private void initializeFromSource(String source) {
   adjectiveList= readIt(source+"/adjective.txt");
   nounList     = readIt(source+"/noun.txt");
   colorList    = readIt(source+"/color.txt");
   countryList  = readIt(source+"/country.txt");
   nameList     = readIt(source+"/name.txt");
   animalList   = readIt(source+"/animal.txt");
   timeList     = readIt(source+"/timeframe.txt");
}
```

Improvement, but Room for More!

- In GladLib.java must assign values to each named instance variable
  - HashMap is better, filename and label linked

```
private void initializeFromSource(String source) {
   String[] labels = {"country","noun", "animal",
                      "adjective","name","color",
                      "timeframe"};
   for(String s : labels){
     ArrayList<String> list = readIt(source+"/"+s+".txt");
     myMap.put(s, list);
}
  }
```

Improvement, but Room for More!

- In GladLib.java must assign values to each named instance variable
  - HashMap is better, filename and label linked
  - What changes for <verb> if stored in verb.txt?

```
private void initializeFromSource(String source) {
   String[] labels = {"country","noun", "animal",
                      "adjective","name","color",
                      "timeframe"};
   for(String s : labels){
     ArrayList<String> list = readIt(source+"/"+s+".txt");
     myMap.put(s, list);
}
  }
```

Improvement, but Room for More!

- In GladLib.java must assign values to each named instance variable
  - HashMap is better, filename and label linked
  - What changes for <verb> if stored in verb.txt?

```
private void initializeFromSource(String source) {
   String[] labels = {"country","noun", "animal",
                      "adjective","name","color",
                      "timeframe"};
   for(String s : labels){
     ArrayList<String> list = readIt(source+"/"+s+".txt");
     myMap.put(s, list);
}
  }
```

Improvement, but Room for More!

- In GladLib.java must assign values to each named instance variable
  - HashMap is better, filename and label linked
  - What changes for <verb> if stored in verb.txt?

```
private void initializeFromSource(String source) {
   String[] labels = {"country","noun", "animal",
                      "adjective","name","color",
                      "timeframe"};
   for(String s : labels){
     ArrayList<String> list = readIt(source+"/"+s+".txt");
     myMap.put(s, list);
}
  }
```

Finding Nouns and Verbs Anywhere

- File associates labels and replacement info
  - .properties or property file

```
verb:http://gladlibs.com/verbs.txt
noun:http://gladlibs.com/nouns-funny.txt
color:http://gladlibs.com/colors.txt
...
```

Finding Nouns and Verbs Anywhere

- File associates labels and replacement info
  - .properties or property file

```
verb:http://gladlibs.com/verbs.txt
noun:http://gladlibs.com/nouns-funny.txt
color:http://gladlibs.com/colors.txt
...
```

Finding Nouns and Verbs Anywhere

- File associates labels and replacement info
  - .properties or property file
  - Often used in Java applications (and others)

```
verb:http://gladlibs.com/verbs.txt
noun:http://gladlibs.com/nouns-funny.txt
color:http://gladlibs.com/colors.txt
...
```

Finding Nouns and Verbs Anywhere

- File associates labels and replacement info
  - .properties or property file
  - Often used in Java applications (and others)
  - Read file, store info in HashMap!

```
verb:http://gladlibs.com/verbs.txt
noun:http://gladlibs.com/nouns-funny.txt
color:http://gladlibs.com/colors.txt
...
```

Finding Nouns and Verbs Anywhere

- File associates labels and replacement info
  - .properties or property file
  - Often used in Java applications (and others)
  - Read file, store info in HashMap!
- HashMap<String,String> myLabelSource

```
private void initializeFromSource() {

   for(String s : myLabelSource.keySet()){
     ArrayList<String> list = readIt(myLabelSource.get(s));
     myMap.put(s, list);
   }
```

Finding Nouns and Verbs Anywhere

- File associates labels and replacement info
  - .properties or property file
  - Often used in Java applications (and others)
  - Read file, store info in HashMap!
- HashMap<String,String> myLabelSource

```
private void initializeFromSource() {

    for(String s : myLabelSource.keySet()){
      ArrayList<String> list = readIt(myLabelSource.get(s));
      myMap.put(s, list);
    }
```

Finding Nouns and Verbs Anywhere

- File associates labels and replacement info
  - .properties or property file
  - Often used in Java applications (and others)
  - Read file, store info in HashMap!
- HashMap<String,String> myLabelSource

```
private void initializeFromSource() {

   for(String s : myLabelSource.keySet()){
     ArrayList<String> list = readIt(myLabelSource.get(s));
     myMap.put(s, list);
   }
```

Finding Nouns and Verbs Anywhere

- File associates labels and replacement info
  - .properties or property file
  - Often used in Java applications (and others)
  - Read file, store info in HashMap!
- HashMap<String,String> myLabelSource

```
private void initializeFromSource() {

   for(String s : myLabelSource.keySet()){
     ArrayList<String> list = readIt(myLabelSource.get(s));
     myMap.put(s, list);
   }
```