Java Class Definitions, Part 1: Creating & Using Methods

Sept 30, 2019

Announcements

- HW 3 due now
- Lab 3 has been returned; you'll get HW 2 back in lab today/tomorrow (HW 2 was very hard to grade!)
- Please come see us (during office hours) if your HW or lab was graded incorrectly or if you are confused!
- Tom will be back tomorrow aft, normal office hours resume; Jeannie will be at a conference Wed-Fri (no office hours)
- October Mtn Day craziness: HW 4 will be posted *next* week (you get a break this week; solutions to HW 4 will be posted before the midterm since you may not get it back beforehand)

Lab 3 Notes

- Make sure you send QUIT and close connection on invalid login
- connection = new NetConnection... should be done in response to clicking login
- Clear text areas using text.setText(""")
- When submitting code, make it clear which one we should grade. Rename any “old” files Lab.java.bak. We should only see one java file (right now)
- Very nice job overall!!

(Last) Last Time

- String methods
  String s = "www.amherst.edu";
  String newS = s.substring(0,s.indexOf(\"\")+1) + "williams" + s.substring(s.indexOf(\"\",s.indexOf(\"\")+1));
String s = “www.amherst.edu”;  
012345678901234  
11111

s.substring(0,s.indexOf(".")+1) + "williams"  
+ s.substring(s.indexOf(".",s.indexOf("."))+1))

String s = “www.amherst.edu”;  
012345678901234  
11111

s.substring(0,s.indexOf(".")+1) + "williams"  
+ s.substring(s.indexOf(".",4)+1))

String s = “www.amherst.edu”;  
012345678901234  
11111

s.substring(0,4) + “williams”  
+ s.substring(s.indexOf(".",4)+1))

“starting index”  
This is asking for an index of “.” that is >=4
String s = "www.amherst.edu";
   01234567890123
     11111
s.substring(0,4) + "williams"
+ s.substring(11)

String s = "www.amherst.edu";
   01234567890123
     11111
s.substring(0,4) + "williams"
+ s.substring(11)

Find the substring from
index 11 to the end of s
(substring starting at index 11)

String s = "www.amherst.edu";
   01234567890123
     11111
 "www." + "williams" + ".edu"

"www." + "williams" + ".edu"

Last Time

 adolescente
More Strings and loops!
Learned about JComboBoxes
Examples involving the dictionary protocol
Today's Plan

- Finish up ChoiceReplacer
- Begin learning about defining and using multiple Java classes in our programs

ChoiceReplacer

- Combine the functionality of previous examples
- String processing - [www.williams.edu](http://www.williams.edu) example (see buttonClicked())
- DictSentenceClient - find synonyms (see menuItemSelected())
- SentenceMenu - parse String, construct JComboBox (see textEntered())

Avoiding Duplicate Code

if (lineFromServer.startsWith("+OK")) {
   message.append(toServer.in.nextPOPResponse() + "\n");
   log.append(lineFromServer + "\n");
} else {
   message.append("Error in receiving message");
   log.append(lineFromServer + "\n");
} log.append(lineFromServer + "\n");
if ( lineFromServer.startsWith( "+OK" ) ) {
    message.append( toServer.in.nextPOPResponse() + "\n" );
    log.append( lineFromServer );
} else {
    message.append( "Error in receiving message" );
    log.append( lineFromServer );
}
How can we eliminate duplicate code?

- With methods!
- Check out MethodicalTap
- setButtons Appropriately( … )
- addLabeledTextField ( … )

```java
private void setButtons( boolean connected ) {
    connect.setEnabled( ! connected );
    disconnect.setEnabled( connected );
    send.setEnabled( connected );
}
this.setButtons( true );
Result:
connect.setEnabled( false );
disconnect.setEnabled( true );
send.setEnabled( true );
```

What is a method?

The name of an **action** some type of object knows how to perform.

- this.createWindow( … )
- log.setText( … )
- contentPane.add( … )
- message.indexOf( … )
What is a method?

A name associated with a **collection of instructions** that should be executed when some special event(s) occurs.

- public void buttonClicked( ... ) { ...
- public void textEntered( ... ) { ...
- public void menuItemSelected( ... ) { ...

Mutator Methods

- Method structure
  
  public/private void method-name( arguments ) {
    lines_of_code;
  }

- Example
  
  private void enableButtonsToConnect() {
    connect.setEnabled( true );
    disconnect.setEnabled( false );
    send.setEnabled( false );
  }

Accessor Methods

- Method structure
  
  public/private return-type method-name( arguments ) {
    lines_of_code;
    return some value having return-type;
  }

- Example
  
  public String getPrefix() {
    return lineFromServer.substring(0,4);
  }
Next up:

Defining our own Java classes!