

Client-Side Options (Part 1)

Copyright © 2026 by
Robert M. Dondero, Ph.D.
Princeton University

Objectives

- We will cover:
 - Some client-side options
 - CLI clients
 - GUI clients

Agenda

- **Aside: Multithreaded programming**
- Common server
- CLI client
- GUI client

Multithreaded Programming

- Recall **spawning.py**

```
$ python spawning.py
blue
blue
blue
blue
blue
blue thread terminated
red
red
red
red
red
red thread terminated
main thread terminated
main thread terminated
$
```

```
$ python spawning.py
blue
blue
blue
blue
red
red
red
red
red thread terminated
blue
main thread terminated
blue thread terminated
$
```

Multithreaded Programming

- See **Spawning1.java**

```
$ java Spawning1
main thread terminated
blue
blue
blue
blue
blue
blue
red
red
red
red
red
red thread terminated
blue thread terminated
$
```

```
$ java Spawning1
blue
blue
blue
blue
blue
main thread terminated
red
red
red
red
red
red thread terminated
blue thread terminated
$
```

Multithreaded Programming

- **Problem:**
 - Java does not allow multiple inheritance
 - What if class `PrinterThread` already inherits from some class other than `Thread`?
- **Solution: ...**

Multithreaded Programming

- See **Spawning2.java**

```
$ java Spawning2
blue
blue
blue
blue
blue
main thread terminated
red
red
red
red
red
red
red
blue thread terminated
red thread terminated
$
```

```
$ java Spawning2
blue
blue
main thread terminated
red
red
red
red
red
blue
blue
blue
blue thread terminated
red thread terminated
$
```

Multithreaded Programming

- **Note:**
 - `bluePrinterRunnable` is an object of a class that implements an interface that defines one method
 - `bluePrinterRunnable` can be replaced with a lambda expression
 - Same for `redPrinterRunnable`
- **And so...**

Multithreaded Programming

- See **Spawning3.java**

```
$ java Spawning3
main thread terminated
blue
blue
blue
blue
blue
blue thread terminated
red
red
red
red
red
red thread terminated
$
```

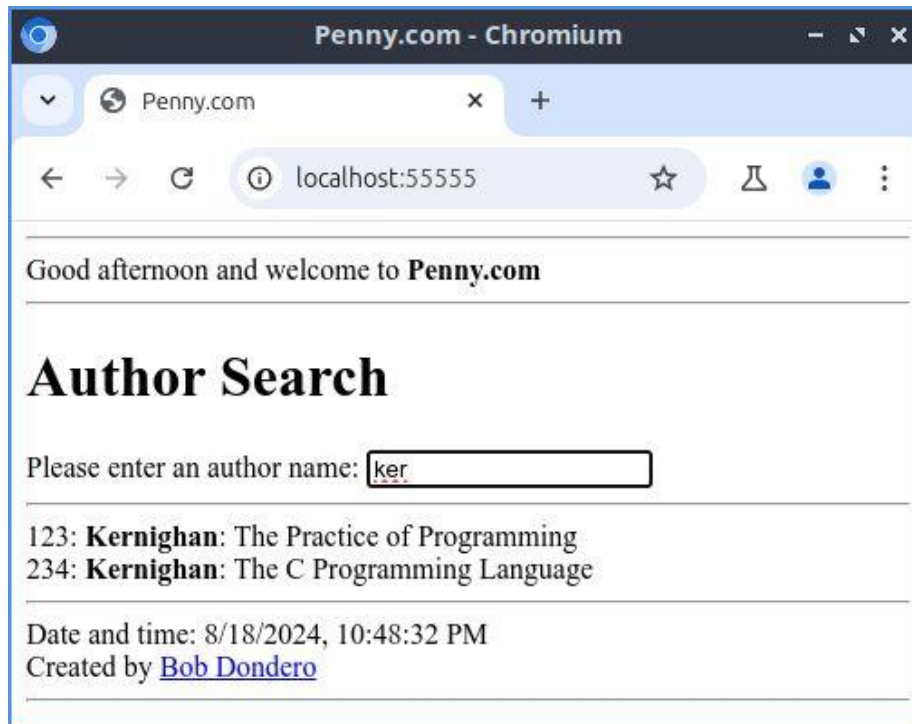
```
$ java Spawning3
main thread terminated
blue
blue
blue
red
red
red
red
red thread terminated
blue
blue
blue thread terminated
$
```

Agenda

- Aside: Multithreaded programming
- **Common server**
- CLI client
- GUI client

Common Server

- See **PennyJson** app
 - (Almost) same as **PennyFlaskSpa** app from previous lecture



Common Server

- See **PennyJson** app (cont.)
 - runserver.py
 - penny.sql, penny.sqllite
 - database.py
 - index.html
 - Irrelevant for this lecture
 - **penny.py**
 - Added IO_DELAY via an environment variable
 - Simulates slow DB
 - Shows how client performs with I/O bound server

Agenda

- Aside: Multithreaded programming
- Common server
- **CLI client**
- GUI client

CLI Client

- See **PennyCommandLine** client
 - **penny.py**

```
$ export IO_DELAY=1
$ cd PennyJson
$ python runserver.py 55555
```

```
$ cd PennyCommandLine
$ python penny.py http://localhost:55555 ker
123: Kernighan: The Practice of Programming
234: Kernighan: The C Programming Language
$
```

Agenda

- Aside: Multithreaded programming
- Common server
- CLI client
- **GUI client**

GUI Client

- GUI client
 - Runs on a desktop/laptop computer
 - Displays a window
 - Window contains widgets/components
 - Buttons, text fields, list boxes, ...

GUI Client

- **Question:** Why study GUI app programming?
- **Answers:**
 - Mainstream “advanced programming”
 - Illustrates event-driven programming
 - Illustrates concurrent programming
 - Important!

GUI Client

Cross-platform GUI app libraries:

Desktop App Library	Language	Platform
GTK	C, C++	Linux, Windows, macOS
Qt	C, C++	Linux, Windows, macOS
wxWidgets	C, C++	Linux, Windows, macOS
Flutter	C, C++, Dart	Linux, Windows, macOS, Android, iOS
Tkinter	Python	Linux, Windows, macOS
AWT	Java	Linux, Windows, macOS
Swing	Java	Linux, Windows, macOS
JavaFX	Java	Linux, Windows, macOS
...

GUI Client

- Why study Java Swing
 - (Instead of some other GUI library)
 - I know it!
 - Popular
 - Bridges to Android programming

GUI Client

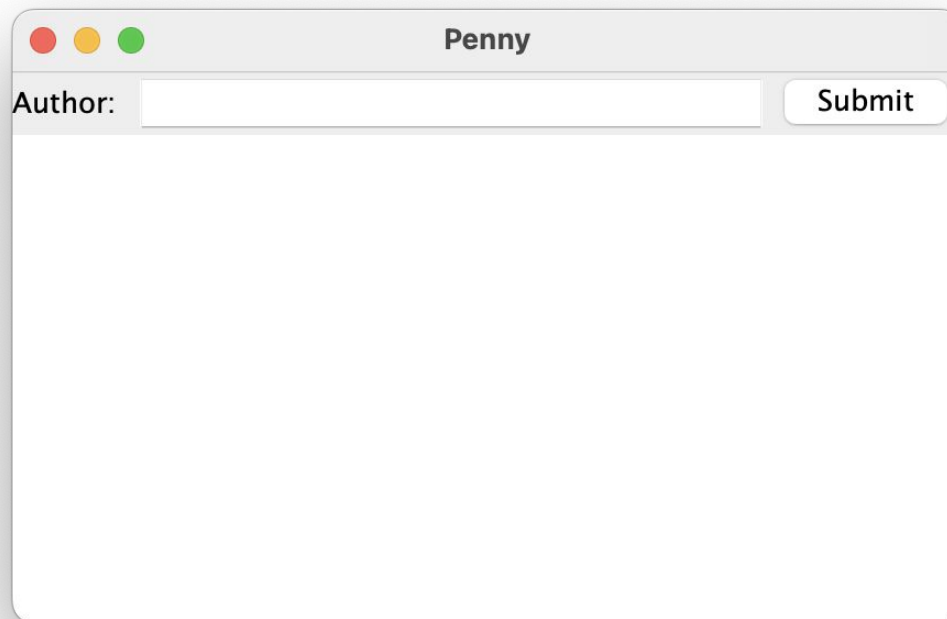
- See **PennySwing1** client

```
$ export IO_DELAY=1
$ cd PennyJson
$ python runserver.py 55555
```

```
$ cd PennySwing1
$ mvn clean
$ mvn compile
$ ./runclient http://localhost:55555
```

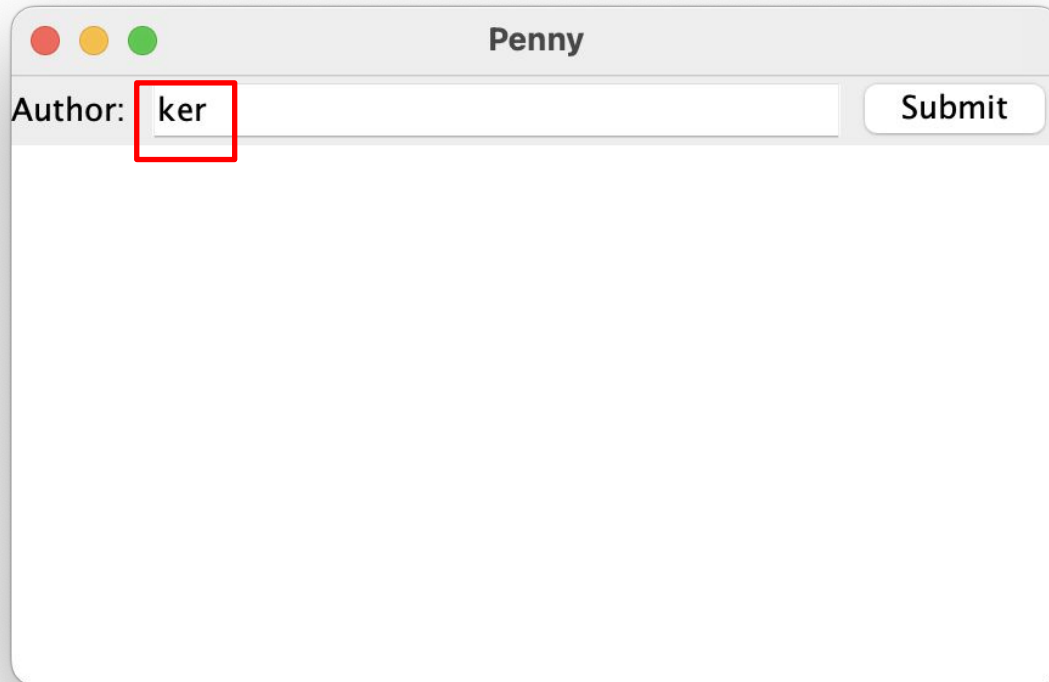
GUI Client

- See **PennySwing1** client (cont.)



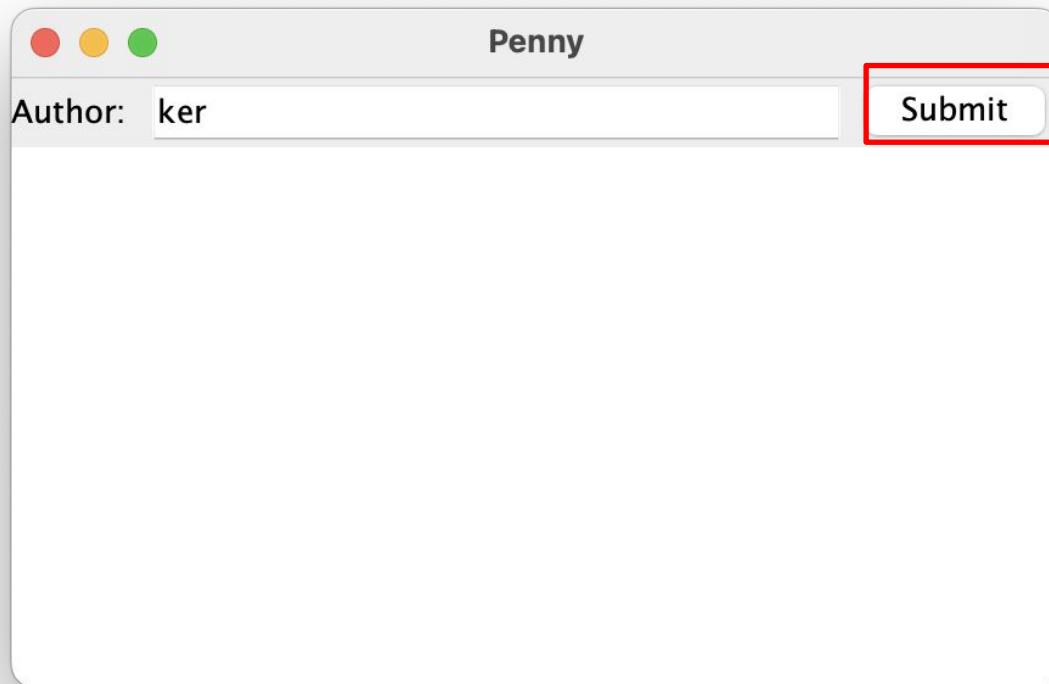
GUI Client

- See **PennySwing1** client (cont.)



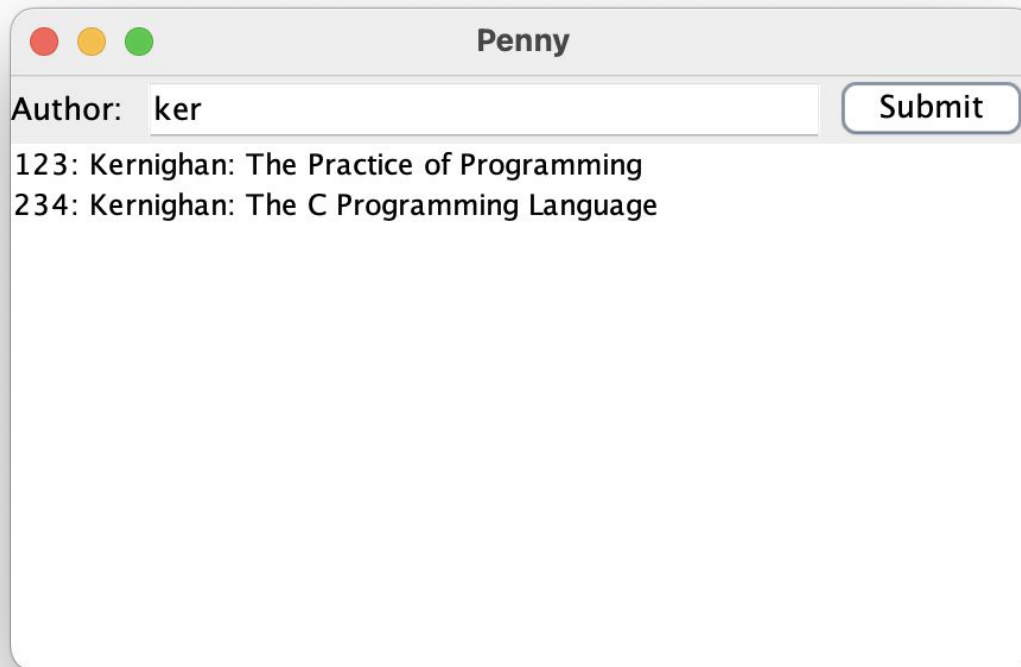
GUI Client

- See **PennySwing1** client (cont.)



GUI Client

- See PennySwing1 client (cont.)



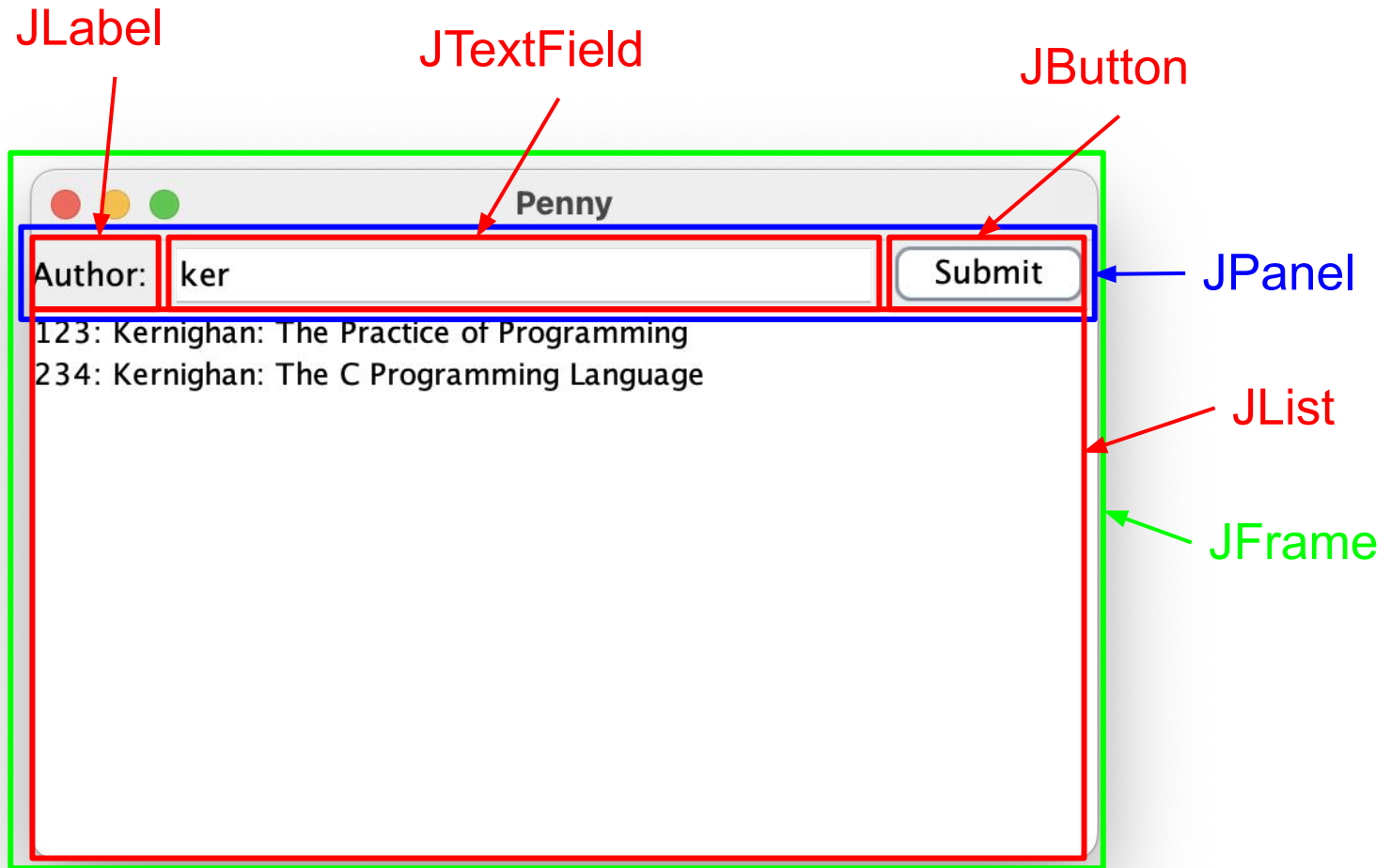
After a
1 second
delay

GUI Client

- See **PennySwing1** client (cont.)

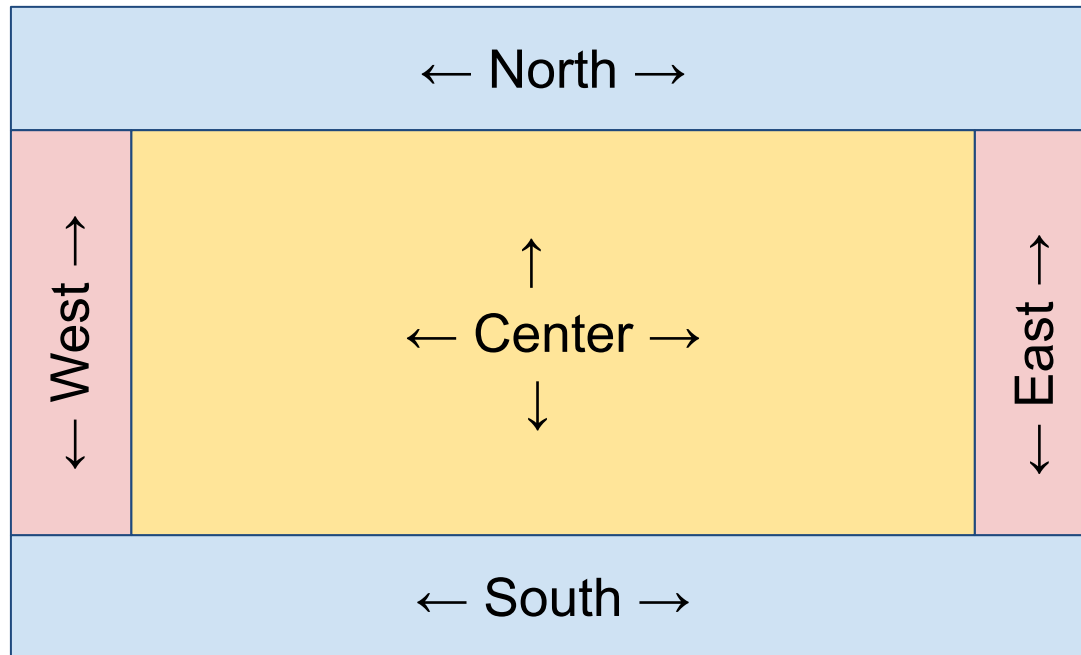
```
PennySwing1/  
  runclient  
  runclient.bat  
  pom.xml  
  src/  
    main/  
      java/  
        edu/  
          princeton/  
            penny/  
              PennyDesktop.java  
              PennyDesktopRunnable.java
```

GUI Client



GUI Client

The Java Swing `BorderLayout` layout manager

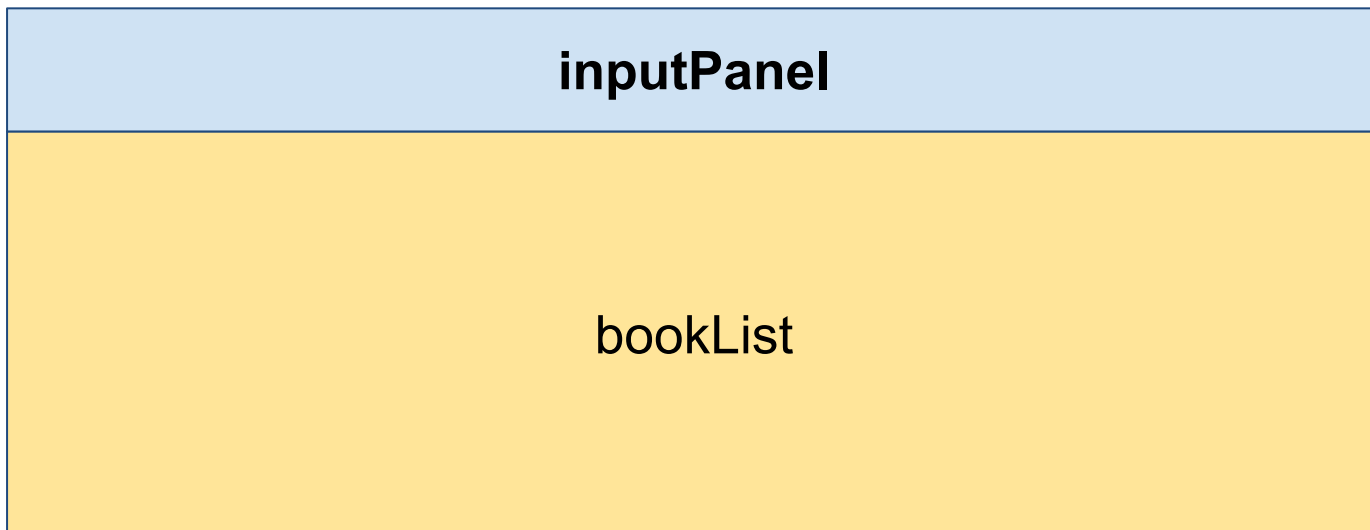


GUI Client

inputPanel



frame



GUI Client

- See **PennySwing1** client (cont.)
 - **(maybe) Problem:**
 - Inconsistent window state after typing author and before clicking Submit button
 - **Solution: redesign...**
 - Eliminate Submit button
 - GUI refreshes with each keystroke

GUI Client

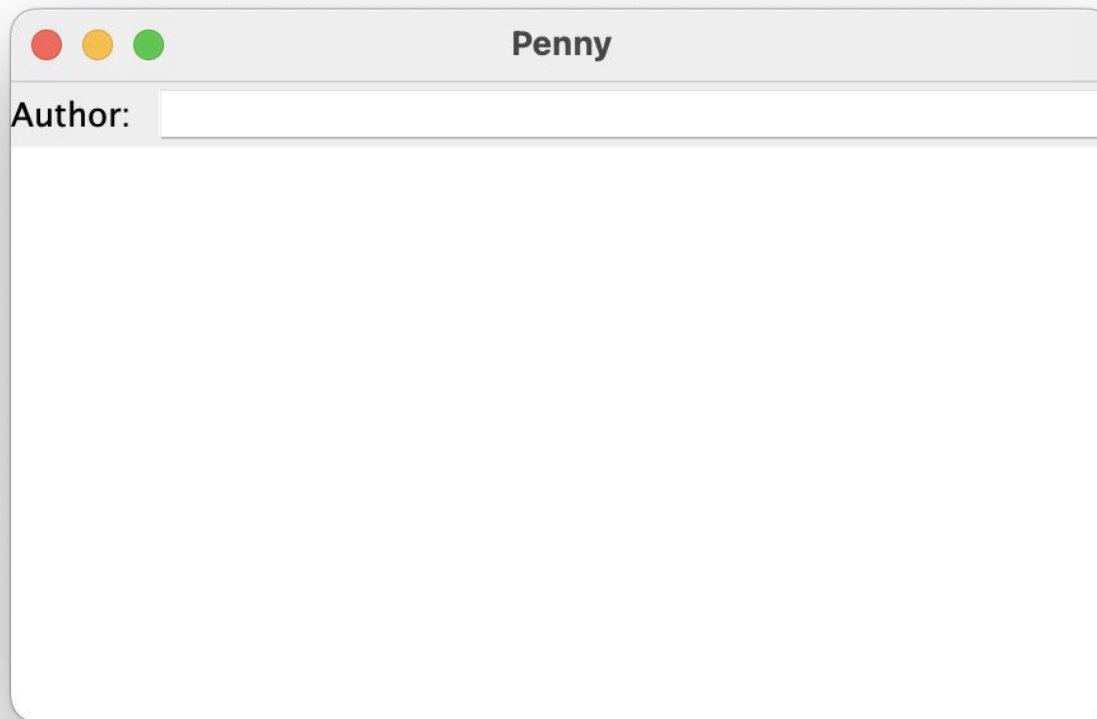
- See PennySwing2 client (cont.)

```
$ export IO_DELAY=1  
$ cd PennyJson  
$ python runserver.py 55555
```

```
$ cd PennySwing2  
$ mvn clean  
$ mvn compile  
$ ./runclient http://localhost:55555
```

GUI Client

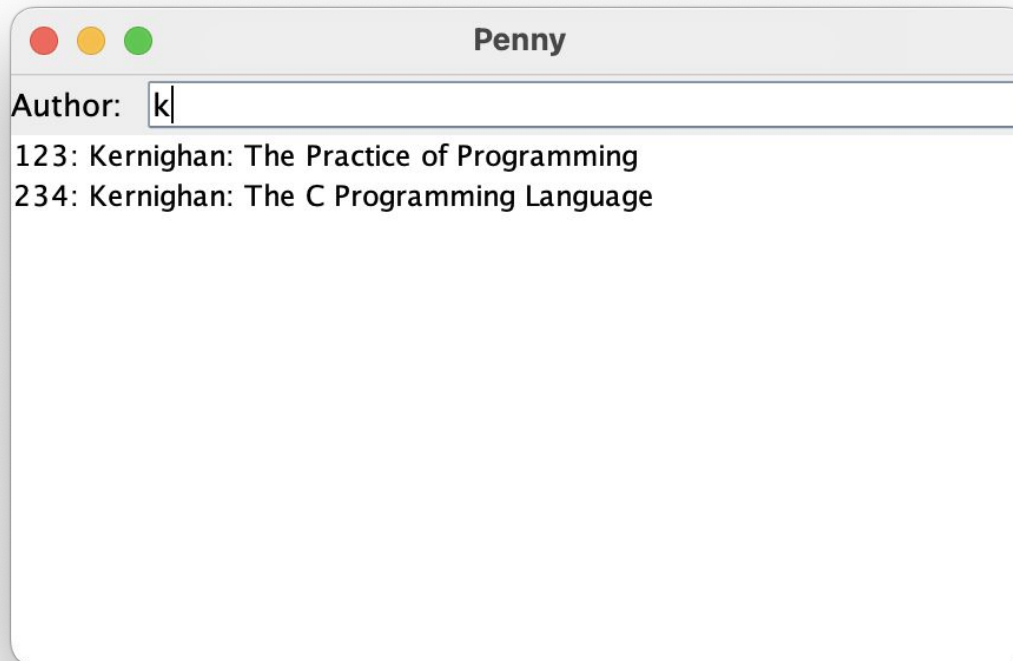
- See PennySwing2 client (cont.)



No
Button

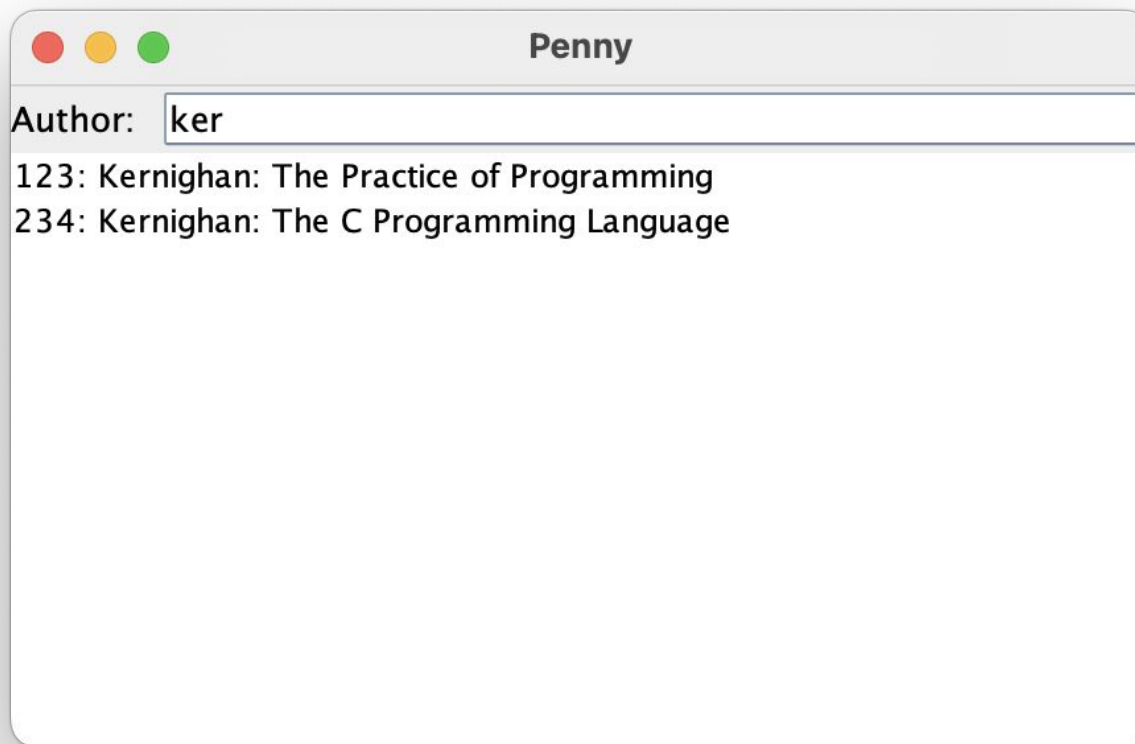
GUI Client

- See PennySwing2 client (cont.)



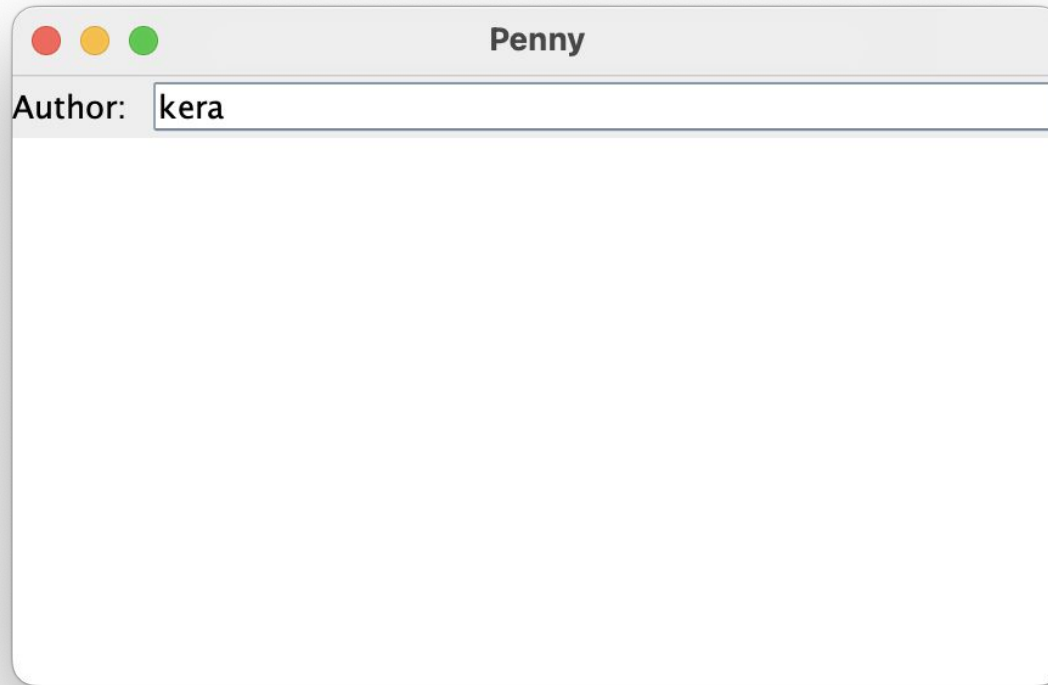
GUI Client

- See **PennySwing2** client (cont.)



GUI Client

- See **PennySwing2** client (cont.)



GUI Client

- See **PennySwing2** client (cont.)

```
PennySwing2/  
  runclient  
  runclient.bat  
  pom.xml  
  src/  
    main/  
      java/  
        edu/  
          princeton/  
            penny/  
              PennyDesktop.java  
              PennyDesktopRunnable.java
```

GUI Client

- See **PennySwing2** client (cont.)
 - **Problem:**
 - Serious GUI lag
 - **Solution: redesign...**
 - Use threads!!!

GUI Client

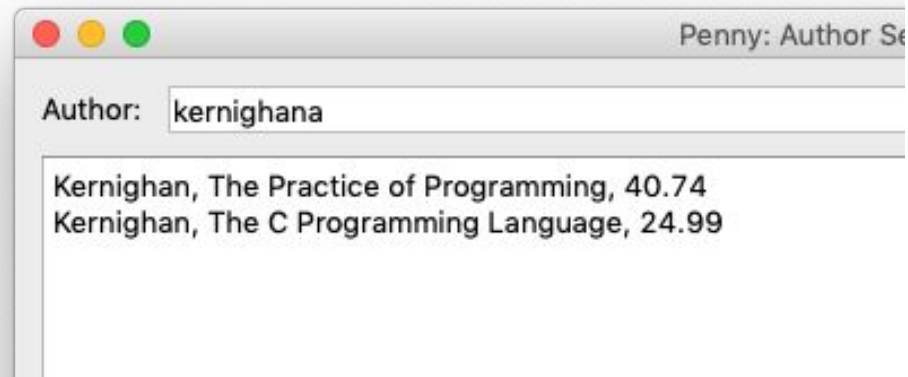
- See **PennySwing3** client

```
PennySwing3/  
  runclient  
  runclient.bat  
  pom.xml  
  src/  
    main/  
      java/  
        edu/  
          princeton/  
            penny/  
              PennyDesktop.java  
              PennyDesktopRunnable.java  
              WorkerThread.java
```

GUI Client

- See **PennySwing3** client (cont.)
 - **Problem:**
 - Server will respond to requests in arbitrary order

old
screen
shot



- **Solution:**
 - Abort previous request

GUI Client

- See **PennySwing4** client

```
PennySwing4/  
  runclient  
  runclient.bat  
  pom.xml  
  src/  
    main/  
      java/  
        edu/  
          princeton/  
            penny/  
              PennyDesktop.java  
              PennyDesktopRunnable.java  
              WorkerThread.java
```

GUI Client

- See **PennySwing4** client (cont.)
 - **Problem:**
 - Server could be overwhelmed with requests
 - **Solution:**
 - *Debounce* the requests
 - **Bonus:**
 - Reduces (but does not eliminate) the need to abort old requests!

GUI Client

- See **PennySwing5** client

```
PennySwing5/  
  runclient  
  runclient.bat  
  pom.xml  
  src/  
    main/  
      java/  
        edu/  
          princeton/  
            penny/  
              PennyDesktop.java  
              PennyDesktopRunnable.java  
              WorkerThread.java
```

Lecture Summary

- In this lecture we covered:
 - Some client-side options
 - CLI clients
 - GUI clients
- See also:
 - **Appendix 1: Java Multithreaded Programming**

Appendix 1: Java Multithreaded Programming

Java Multithreaded Programming

- Recall race.py

```
$ python race.py
1
2
3
4
5
6
7
8
9
10
8
6
4
2
0
Final balance: 0
$
```

```
$ python race.py
1
2
3
4
-1
5
6
-3
-5
-7
-9
7
8
9
10
Final balance: 10
$
```

```
$ python race.py
1
2
3
4
5
6
7
8
9
6
4
10
2
0
-2
Final balance: -2
$
```

Java Multithreaded Programming

- See **Race.java**

```
$ java Race
1
2
3
4
5
6
7
8
0
-2
-4
-6
-8
9
10
Final balance: 10
$
```

```
$ java Race
1
2
3
4
5
6
7
8
0
-2
-4
9
10
-6
-8
Final balance: -10
$
```

Java Multithreaded Programming

- Recall locking.py

```
$ python locking.py
1
2
3
4
5
6
7
8
9
10
8
6
4
2
0
Final balance: 0
$
```

```
$ python locking.py
1
2
3
4
2
0
-2
-4
-6
-5
-4
-3
-2
-1
0
Final balance: 0
$
```

Java Multithreaded Programming

- See **Locking.java**

```
$ java Locking
1
2
3
4
5
6
7
8
9
10
8
6
4
2
0
Final balance: 0
$
```

```
$ java Locking
1
2
3
4
5
6
7
8
9
10
8
6
4
2
0
Final balance: 0
$
```

Java Multithreaded Programming

- Recall conditions.py

```
$ python conditions.py
1
2
3
4
5
6
7
8
9
10
8
6
4
2
0
Final balance: 0
$
```

```
$ python conditions.py
1
2
3
4
5
3
1
2
3
4
5
6
4
2
0
Final balance: 0
$
```

Java Multithreaded Programming

- See **Conditions.java**

```
$ java Conditions
```

```
1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
8  
6  
4  
2  
0  
Final balance: 10  
$
```

```
$ java Conditions
```

```
1  
2  
3  
4  
2  
0  
1  
2  
3  
4  
5  
6  
4  
2  
0  
0  
Final balance: 0  
$
```

Java Multithreaded Programming

- Multithreaded programming across languages

	Language Support	Library Support
Python	no	yes
C	no	yes
Java	yes	yes
JavaScript	no	no