CSSE 220 Day 13

Multithreading
Recursion

Checkout *Multithreading* and *Recursion* project from SVN

Questions

The World is Concurrent

- Joe Armstrong,
  Programming in Erlang

Q1

Multithreading

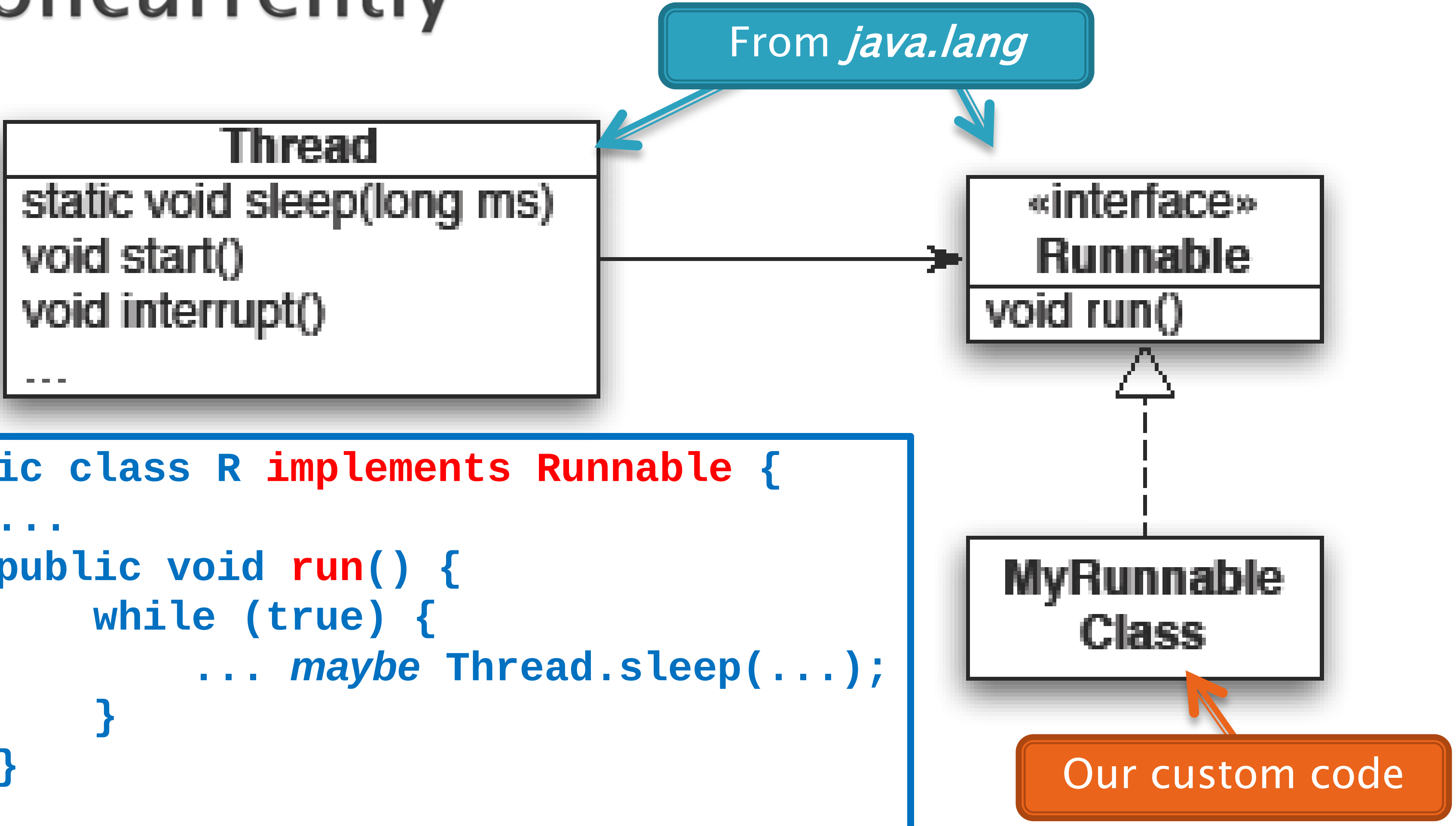
- A technique to:
  - Run multiple pieces of code “simultaneously” on a single machine

Time → Slices	1	2	3	4	5	6	7	8	9	10	11	12	13	14
running thread 1	■	■	□	■	□	□	□	■	□	■	□	□	■	■
running thread 2	□	□	■	□	■	■	■	□	■	□	■	■	□	□

  - Run different parts of a program on different processor cores

Q2

Running Our Own Code Concurrently



```
public class R implements Runnable {
    ...
    public void run() {
        while (true) {
            ... maybe Thread.sleep(...);
        }
    }
}
```

Wherever you want to start the Thread:

`new Thread(`*object of type R*`).start();`

Q3

Animation with Threads

- Example 1: A single object
  - "Animate" it with button clicks
  - Animate it with a Timer

```
Timer timer = new Timer(50, animatorButton);
timer.start();
```

  - Animate it by using a thread

```
public class R implements Runnable {
    ...
    public void run() {
        while (true) {
            ... maybe Thread.sleep(...);
        }
    }
}
```

Wherever you want to start the Thread:

```
new Thread(object of type R).start();
```

Animation with Threads

- Example 2: Multiple objects
  - Use separate thread for each object's "brain"
  - Another thread asks Java to update the GUI

http://www.roadsideamerica.com/story/8543

Other Uses for Threads

- Web servers: many users connecting
- Desktop applications:
  - layout, spellchecking, auto-save, …
- Scientific computing
- Weather forecasting
- …

Caution!

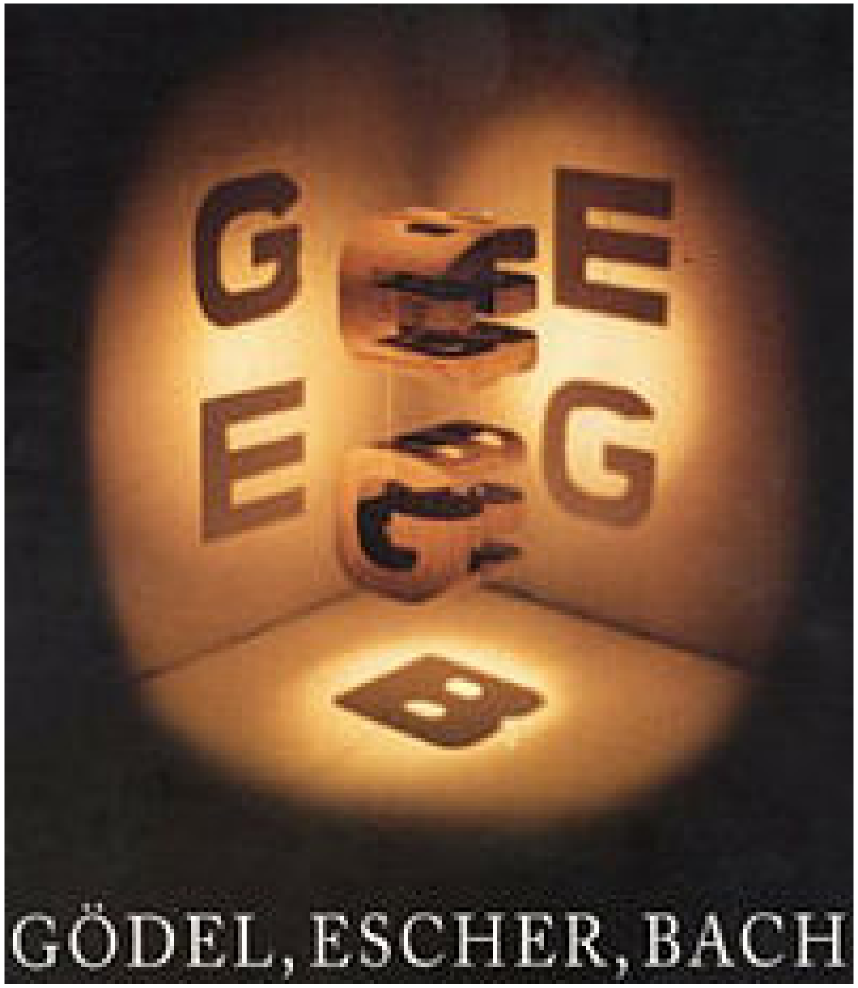
- What if one thread is in the middle of performing an action when its time slice ends?
- What if a second thread's action interferes with the first's action?
- See bank example in today's project

Optional: For a way to fix this, see Big Java Section 20.4

Q4

Gödel, Escher, Bach

- By Douglas Hofstadter
- Argues that a major component of intelligence is **our ability to think about thinking**



Recursion

- A solution technique where the same computation **occurs repeatedly** as the problem is solved

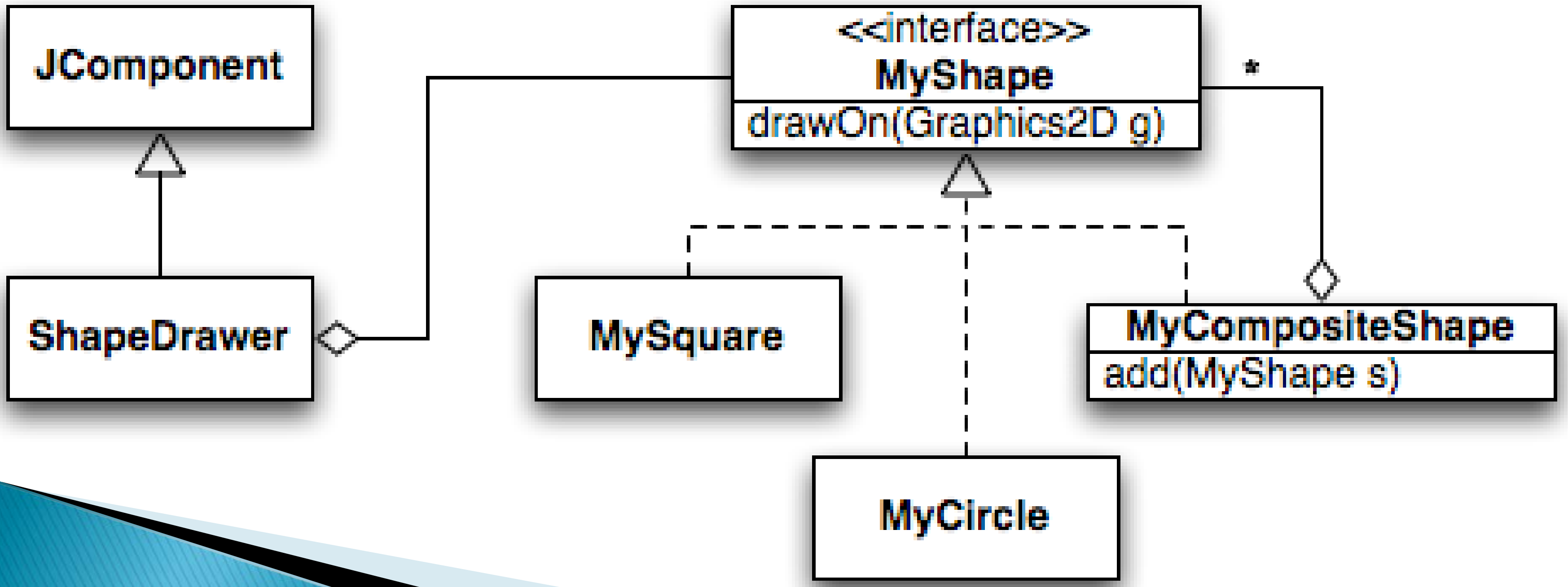
recurs

- Examples:
  - Sierpinski Triangle: tonight's HW
  - Towers of Hanoi: http://www.mathsisfun.com/games/towerofhanoi.html or search for Towers of Hanoi

Recursion

- A solution technique where the same computation **occurs repeatedly** as the problem is solved

recurs

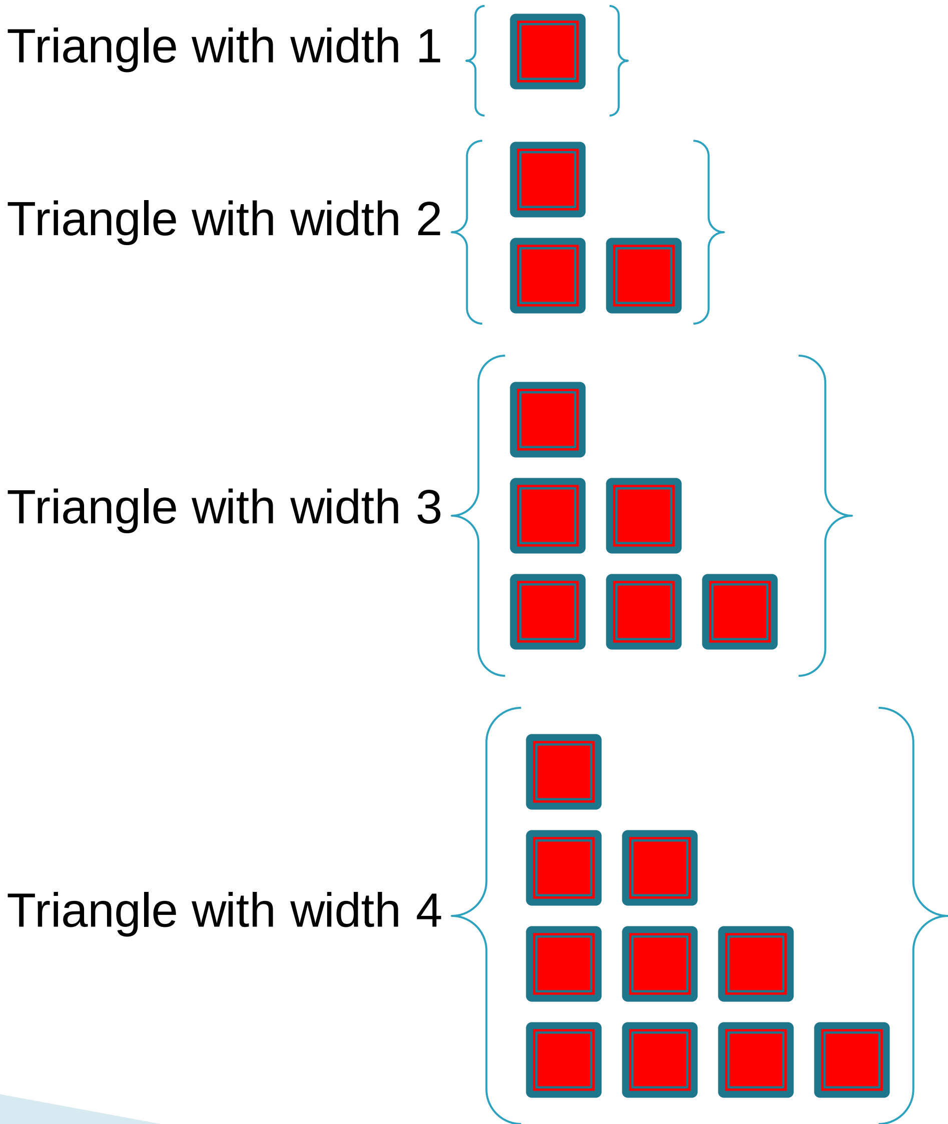


An example – Triangle Numbers

- If each red block has area 1, what is the ***area*** **A(n)** of the Triangle whose *width* is n?
  - Answer:

    A(n) = n + A(n–1)
- The above holds for which *n* ? What is the answer for other *n* ?
  - Answer: The recursive equation holds for n >= 1.

    For n = 0, the area is 0.

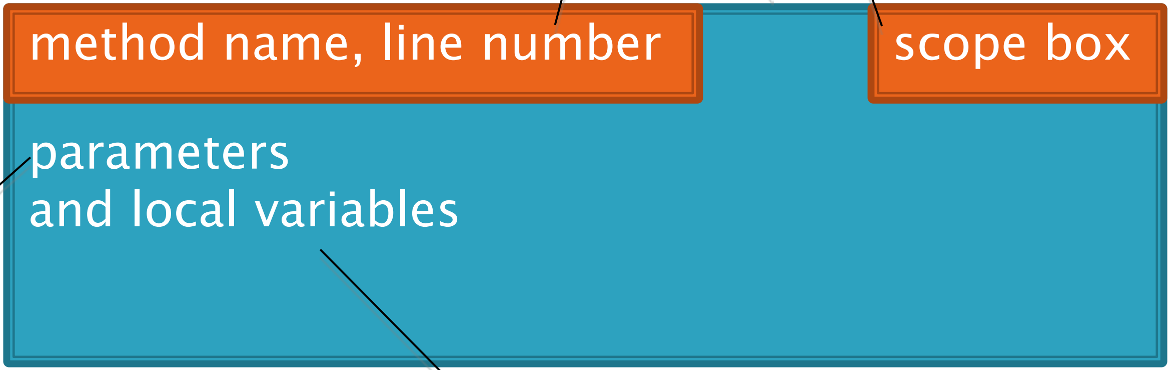


Frames for Tracing Recursive Code

1. Draw box when method starts

2. Fill in name and first line no.

3. Write class name (for static method) or draw reference to object (for non-static method)



4. List every parameter and its argument value.

5. List every local variable declared in the method, **but no values yet**

6. Step through the method, update the line number and variable values, draw new frame for new calls

7. “Erase” the frame when the method is done.

Thanks to David Gries for this technique

Q5–Q6

Optional Practice

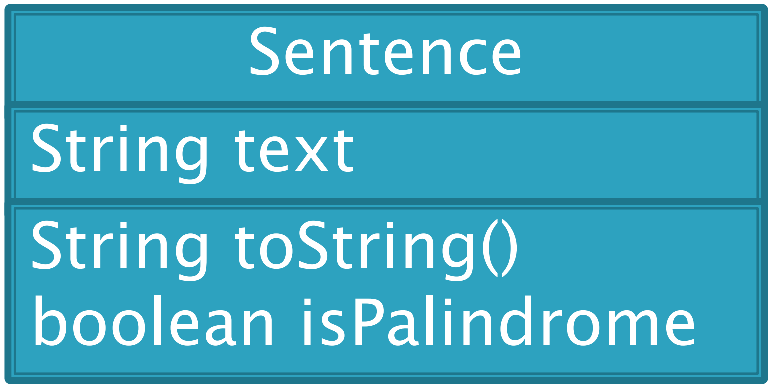
- Trace the **buildShape(MAX_DEPTH)** method call in **shapes.Main**'s **main** method

Key Rules to Using Recursion

- Always have a **base case** that **doesn't recurse**
- Make sure recursive case always **makes progress**, by **solving a smaller problem**
- **You gotta believe**
  - Trust in the recursive solution
  - Just consider one step at a time

Programming Problem

- Add a recursive method to Sentence for computing whether Sentence is a palindrome



Sentence

String text

String toString()
boolean isPalindrome

Recursive Helpers

- Our isPalindrome() makes lots of new Sentence objects
- We can make it better with a “recursive helper method”
  - Many recursive problems require a helper method

```
public boolean isPalindrome() {
    return isPalindrome(0,  this.text.length() – 1);
}
```

Position of first letter of the remaining String to check

Position of last letter of the remaining String to check

Homework part 1

- Reverse a string…recursively!
- A recursive helper can make this really short!

Another Definition of Recursion

- “If you already know what recursion is, just remember the answer. Otherwise, find someone who is standing closer to Douglas Hofstadter than you are; then ask him or her what recursion is.”

  —Andrew Plotkin

Recursive Functions

- Factorial:

$$n! = \begin{cases} 1 & \text{if } n \leq 1 \\ n * (n-1)! & \text{otherwise} \end{cases}$$

Base Case

Recursive step

- Ackermann function:

$$A(m,n) = \begin{cases} n+1 & \text{if } m = 0 \\ A(m-1, 1) & \text{if } m > 0 \text{ and } n = 0 \\ A(m-1, A(m, n-1)) & \text{otherwise} \end{cases}$$

Team Project

- Work time

  Be sure everyone is getting a chance to drive.

Q7–8