Recursion and Repetition

2-9-2011

Opening Discussion

- Minute essay comments:
  - Copy and paste commands in vi.
- Let's look at solutions to the interclass problem.
- Finishing the intersecting squares.

The match Expression

- There is a second conditional expression in Scala called match.
  - *expr* match {
    - case *pattern* => *expr*
    - case *pattern* => *expr*
    - ...
  - }
- There are lots of options for the pattern, but the simplest one is literal values.
- Lowercase names will be bound as val declarations.

Motivation

- We have the ability to do things once and to control whether or not certain things happen that once.
- Computers are really great for doing things multiple times.
- Reading a whole file or doing something until the user tells us to stop.

Mathematical Recursion

- The idea of recursion comes from mathematics.
- A function is recursive if it is defined in terms of itself.
- All recursive functions will have at least two cases.
  - One where the function refers to itself.
  - A base case where it doesn't refer to itself.
- Let's look at some examples of this.

Programmatic Recursion

- Now I want us to write some Scala functions that are recursive.
- They will look much like the math functions.
- We have to provide a return type.
- One argument changes to tell us when to stop.

Scripts and Redirection

- One way to process data from a file is to write a script and use redirection.
  - scala script.scala < input.txt
- This way you don't have to enter the input over and over. Also handy if the input is really large.
- We'll learn other ways to deal with files later.

Minute Essay

- Make sure you are reading.
- What questions do you have about this topic?

Minute Essay

- Make sure you are reading.
- What questions do you have about this topic?