Reading from Files

2-28-2011

Opening Discussion

- Midterm answers.
- Minute essay comments
  - Similarity between do-while and if-else?
  - Details on Vector. It is basically an immutable Array.
  - E-mail response hours.

Motivation

- Programs are more useful when they can interact with files.
- Everything that isn't in a file is lost when the program stops running.

I/O Redirection

- Using I/O redirection gives you some very basic ability to read from and write to files.
- It has big limitations though because there is only one file each way.
- More over, that one file blocks the ability to use either standard input or output.

Packages and Imports

- To read from a file we will be using the scala.io.Source type. To understand what that means, we need to talk about packages.
- Packages provide a way to organize code and group things of like functionality.
- Import statements let you use things without typing in their fully specified names.

The API

- To get a sense of the different package in Scala, it is helpful to look at the API.
- There are still lots of things in the API you won't fully understand. That isn't a problem as you aren't expected to get too much from it right now.

scala.io.Source

- Call Source.fromFile(fileName:String) to get a Source object that reads from a file.
- There are other methods in the main Source object that we will learn about later.
- The fromFile method technically gives you BufferedSource. This is for efficiency.

Iterators

- Both Source and BufferedSource are of the type Iterator[Char].
- An Iterator has most of the methods you are used to from List and Array. However, you can only go through it once.
- Fundamentally uses hasNext and next methods.

getLines

- This will give you an Iterator[String] that will go through the file one line at a time instead of a character at a time.
- You will often find this more useful.

Minute Essay

- What questions do you have?
- IcP #5 on Wednesday.