Arrays and Lists

9-27-2010
Opening Discussion

- Let's look at solutions to the interclass problem.
- Turning in assignments.
- Minute essay comments
  - Recursion is horrifying?
  - Why no if on the second case in math form?
  - When else will we use recursion? How will you know? Paradigms.
  - Is recursion faster than iteration?
  - Can you use recursion with Strings?
Need for Collections

- Computers are good at dealing a lot of data. So far we can only store one value in each variable. This is a significant limitation.
- Collections are types that can store multiple data values.
- Allow us to remember many things to work on.
- The collection libraries in a language are very significant.
- Scala has great collections.
Basic Arrays and Lists

- The two most basic collection types in Scala are arrays and lists.
- We can make either by following the object name with a parenthesized list of elements.
- Can create an “empty” array using new.
- Can build Lists with :: operator. Nil is empty.

Comparison

- Arrays are mutable, but fixed in size.
- Lists are immutable, but it is easy to add an element and get a new list.
You should notice that when we make an array or a list, the type is followed by square brackets.

These types are parametric. So they take type arguments.

In Scala, type parameters are placed in square brackets.
Using Arrays

- We can get to the elements in an array by putting an index in parentheses.
  - arr(5)
- This syntax can be used in expressions to read values.
- It can also be used in assignments to store values in the array. This is what it means to be mutable.
- Let's look at some examples of this.
Using Lists

- You can do direct access on lists, but it is inefficient.
- The better method is to use the head and tail methods.
- The elements in a list can't be changed. However, you can efficiently add new elements at the front of the list.
What are some examples us uses of arrays or lists?

Remember to turn in assignment #2 by midnight.

The third quiz is on Wednesday.

Interclass problem:

- Write recursive functions to count the number of even numbers in a List[Int] and an Array[Int].