Collection Methods

2-20-2012

Opening Discussion

- Minute essay comments:
  - Exciting tech for 2012.
  - Calling collection methods.
  - Do I like Star Trek, Star Wars, …?
  - What are lists and arrays best used for?
  - Good ultrabook for Ubuntu?
  - Splitting lists into 3+ parts.
  - Ignoring string case.
  - Methods we are using work on both arrays and lists.

More

- Meaning of an empty list.
- Which do you use more often, Arrays or Lists?
- What languages will a CS major know when he/she leaves Trinity?
- Preferred beach.
- Meaning of recursion with 2+func(x+1).
- Building your own surveillance drone.
- Name and limitations for Array(7,8,9).
- Of course you should consider CS as a major, especially if fixing a bug makes you want to dance.

Standard Methods

- There are lots of methods on collections. The API can help us see all of them.
- Part of collections:
  - drop, init, last, slice, splitAt, take, takeRight
- Boolean tests:
  - contains, endsWith, isEmpty, nonEmpty, startsWith
- Searching:
  - indexOf, lastIndexOf
- Other:
  - mkString, reverse, zip, zipWithIndex

Other Methods

- If the elements in a list support addition or multiplication, you can use the sum and product methods.
- If they are ordered you can do min and max.
- Having sum and length makes averages really easy.
- With min you can even drop a grade easily.

Higher Order Methods

- The most powerful methods are ones you can pass functions into.
  - exists, forall – Boolean checks like for math.
  - filter, partition – separate collection based on Boolean.
  - map – apply function to all the elements.
  - reduceLeft – apply function moving through collection
  - foldLeft – apply function moving through, but allows initial value so it can return a different type. This is curried.

Let's Put These Into Action

- I want to spend some class time playing with these methods and seeing what we can do with them.
- A String is a collection so you can do these things with a String as well.
- String also has a method called split.
- BLS data
  - ftp://ftp.bls.gov/pub/time.series/la/

Variable Length Argument Lists

- You can make functions that don't specify exactly how many arguments they take.
- These are often called var-args.
- To do this, but a * after the type. It can only be the last argument in a list.

Calling Var-Args with Collections

- It is often helpful to call a var-args method passing a collection for the variable length arguments.
- You can do this, but you have to tell Scala what you are doing.
- Follow the collection with :_* to do this.
- The : is like specifying a type.
- The _ says you don't care about the exact type.
- The * is like the * in var-args declarations.

Aliasing and Mutability

- I argue that immutable collections like Lists can be safer than mutable ones like Arrays.
- One of the big reasons for this is aliasing.
- An alias in programming is just like in normal life. It is a second name for something.
- Variables are really references to objects.
- If a second variable is assigned the same value as the first, they are aliases to that object.
- Let's play with this and draw on the board.

Aliasing for Argument Passing

- When you pass arguments, you are really passing references.
- So arguments in functions are aliases to the objects outside the function
- If the object is mutable, the function can change it.

Pass-by-Name

- There is another way to pass things in Scala called pass-by-name.
- When you pass something by name, it isn't evaluated at the time it is passed. Instead it is turned into a function and that function is evaluated every time the variable is used.
- The syntax is to put an => before a type, but not have an argument list before the arrow.

Fill and Tabulate

- There are two other ways of creating collections: fill and tabulate. Both are curried. Second argument to fill is by name, second argument to tabulate is a function.
- The fill method on Array or List takes a first argument of how many elements. After that is a by-name parameter that gives back the type you want in the array or list.
- Tabulate also takes a size first. After that is a function that takes the index.

More BLS Games

- Yearly averages.
- Monthly averages.
- City differences.

Minute Essay

- What questions do you have?
- Getting your head around the higher-order methods can take time. Practice is your best friend.
- Midterm is a week from today.