XML and Patterns

11-12-2010

Opening Discussion

- Questions about the assignment?
- Minute essay comments:
  - How painful would it be to redo the map file in XML?
  - If you have problems with getting into classes, remember that CS is here for you and we care about it.
  - XML in files, case classes in code.
  - How does one write a website?

XML in Scala

- The Scala language supports XML at the language level.
- Go to the REPL and enter some XML.
- There is a scala.xml package that contains the libraries for XML.
  - The NodeSeq, Node, and Elem types are particularly useful. I'll typically just use the word Node to describe something from the XML.
  - So is the XML object.

The XML Object

- The loadFile method can be passed a file name and it will read in the file and return a NodeSeq that allows you to get to the contents.
- There is also a save method that takes a file name and an XML node and writes it to file.

Using \ and \\

- Use the \ operation on a node to search for the occurrences of something at the top level.
- The second argument is a string.
  - Normal string searches for tags with that label.
  - If the string starts with @ it searches for attributes.
- Use \\ to search deeply.

Generating XML

- We saw that Scala will let you enter XML literals directly into the code.
- Inside XML literals you can put expressions inside curly braces and they will be expanded out.
  - <quizzes>{stu.quizzes.mkString(“ “)}</quizzes>
- The thing needs to expand to a String or an XML node.
- If it is for an attribute is must be a String and you don't give the double quotes.

Pattern Matching

- We have used three types of patterns previously:
  - Value literals
  - Tuples
  - Type matches
- The last two start to show the power of pattern matching. In particular, they show that part of a pattern can be a variable name that binds to part of the pattern.

Variable Binding

- When a pattern is matched, any words that start lower case are assumed to be variable names you want bound.
- Use an underscore for anything you want to match stuff, but ignore the value.
- Use @ to bind a name to a match you are also further specifying.
- To match the value of an outside variable put the variable name in backticks.

XML Patterns

- You can use patterns to pull out parts of XML or match on different types of nodes.
- Simply put the variable names you want inside of curly braces.
  - val <a>{s} = node

Case Class Patterns

- The real power of case classes in Scala comes from the fact they can be used in matches.
  - stu match {
  - case Student(n,q,t,a) => ...
  - }
- You can do this type of matching on events to pull out the fields you care about if you don't want the full event.
  - case MouseMoved(source,point,mod) => ...

Patterns Everywhere

- Patterns are used in a lot of places in Scala, not just cases and matches.
- The initial declaration of variables is a pattern match. That is why we could assign from tuples.
- The “variable name” in a for loop is actually a pattern. If the pattern isn't matched by an element, that element is skipped.

Minute Essay

- What questions do you have about XML or patterns?
- Interclass Problem:
  - Make a program that can read from an XML file into a case class and write from that case class back into an XML file.