Contents

F# Domain Modeling – Data type declarations and recursive functions 1

5.1 – Pattern matching, union type 1

5.2 – Tuples vs records 1

F# Domain Modeling – Data type declarations and recursive functions

In this exercise, we are going to represent a binary tree of integers and perform operations on it.

5.1 – Pattern matching, union type

Redefine Declare a type IntegerTree representing a tree of integers and define a recursive function sumIntegerTree that sums all the values in the tree. Test your solution with a couple of inputs.

5.2 – Tuples vs records

We can use tuples as well as records to return a pair of numbers. For instance, we can count and return the number of words and letters in a given string in a tuple as follows:

```
let countWordnLetter (str:string) =
        let wordCount = str.Split [|' '|]
        let letterCount = wordCount |> Array.sumBy (fun w -> w.Length)
        (wordCount.Length, letterCount)

 // test it
 countWordnLetter "functional programming is fun and rewarding"
```

Convert ***countWordnLetter*** function above to use records instead of tuples.