F# example(http://www.soshnikov.com/fsharp/contents.html)

Introduction

- Why to study Functional programming
- What is this book about and for whom
- How to install and start using F#

FUNDAMENTALS OF FUNCTIONAL PROGRAMMING

- Application of vs functions. assignment
- Ordered tuples, lists, and type inference
- Functional types and function descriptions
- Currying
- Conditional operator and optional type
- Data types, marked-up association, and pattern matching
- Recursion, parameter functions, and the for loop
- Constructions >>, |>
- An example is the construction of the Mandelbrot set
- Interoperability with .NET

RECURSIVE DATA STRUCTURES

- Lists and list constructors
- Pattern matching
- The simplest list processing functions
- Higher-order object-orientation functions
- Display
- Filtering
- Convolution
- Other higher-order functions
- List Generators
- Tail recursion
- Complexity features of working with lists
- Arrays
- Multidimensional arrays and matrices
- Lists of lists, or non-rectangular arrays (Jugged Arrays)
- Multidimensional arrays .NET
- Specialized types for matrices and vectors
- Sparse matrices
- Using third-party math packages
- General view trees
- Binary trees
- Definition

- Traversing binary trees
- Search Trees
- Expression trees and Abstract Syntax Trees (AST)
- Other data structures
- Sets (Set)
- Mapping (Map)
- Hash Tables

TYPICAL TECHNIQUES OF FUNCTIONAL PROGRAMMING

- Closures
- Dynamic binding and mutable variables
- Generators and reference variables ref
- Lazy sequences (seq)
- Building a frequency dictionary of a text file
- Calculation of Pi by the Monte Carlo method
- Lazy and energetic calculations
- Memoization
- Continuations

IMPERATIVE AND OBJECT-ORIENTED FEATURES OF F#

- Multiparadigmality of the F language#
- Elements of imperative programming in F#
- Using mutable variables and references
- A loop with a precondition
- Conditional operator
- Null values
- Handling exceptional situations
- Object-oriented programming in F#
- Recordings
- Modeling object orientation through records and closures
- Methods
- Interfaces
- Creating classes using delegation
- Creating a class hierarchy
- Extending the functionality of existing classes
- Modules

METAPROGRAMMING

- Language-oriented programming
- Active Templates
- Quotas

- Expression construction, partial function application and supercompilation
- Monads
- I/O Monad
- Monadic properties
- The monad of nondeterministic computations
- Monadic expressions

PARALLEL AND ASYNCHRONOUS PROGRAMMING

- Asynchronous expressions and parallel programming
- Asynchronous programming
- Asynchronous parallel file processing
- Agent-based design pattern
- Using MPI

SOLVING TYPICAL TASKS

- Computational tasks
- Calculations with high accuracy
- Complex type
- of Unit of Measurement
- Using third-party math packages
- Data Access
- Relational Database Access (SQL Server)
- Access to weakly structured XML data
- Working with data in Microsoft Excel
- Web programming
- Access to web services, XML data, RSS feeds
- Access to the text content of web pages
- Using web-oriented programming interfaces using the Bing Search API as an example
- Implementing web applications in F# for ASP.NET Web Forms
- Implementing web applications in F# for ASP.NET MVC
- Implementation of web applications in F# using the WebSharper system
- F# Cloud Programming for Windows Azure
- Visualization and working with graphics
- Two-dimensional graphics based on the Windows Forms API
- Using the Chart element
- 3D visualization using DirectX and/or XNA
- Text analysis and compiler construction
- Implementation of syntactic parsing by recursive descent

- Using fslex and fsyacc