Functional languages

Roland Király
EKF TTK
Institute of Mathematics an Informatics

Alkalmazható természettudományok oktatása a tudásalapú társadalomban – TÁMOP-4.1.2.A/1-11/1-2011-0038

Functional Languages

The World of Functional Programing Languages

The world of functional programming languages is not very well known even among programmers. Most of them are skilled in the use of object oriented and imperative languages but they lack any kind of knowledge regarding the ones mentioned above. It is often hard to explain what makes a language functional...

General Characteristics of Functional Programs

Among functional languages there are pure and not purely functional languages. LISP, Erlang and F# are some other well-known functional languages are not purely functional, as their functions contain side effects.

General Characteristics of Functional Programs

In mathematical logic and in computing science Lambda calculus is the formal tool of function definitions, function implementations and recursion. Alonzo Church introduced it as part of his research of the basics of mathematics in the 1930s.

General Characteristics of Functional Programs

Term rewriting system (TRS) is an alternative version of Lambda calculus. In this model the functions of a functional program correspond rules of rewriting, the initial expression corresponds a reducible term. TRS is a more universal model than Lambda calculus.

Basic Input-Output

Initial Steps of Running Erlang Programs

Considering Erlang programs, flexibility is an important aspect both in their development and in their application. Erlang does not have a specific development tool.

Basic Input-Output

Introduction to Clean

Creating Clean programs with the integrated development system is significantly easier than creating programs in those functional languages which do not have IDE.

Basic Input-Output

Writing and running F# programs

A simple option is to use Visual Studio 2010. This is the first Visual Studio that contains F#.

Data Management

Handling of variables in functional languages is a lot different from the way you have got accustomed to in imperative and OO environment. Variables can be assigned only once, namely you can only bind a value to them once.

The lack of destructive assignment rules out constructions which are based on consecutive multiple assignment

(I = I + 1).

Naturally, there is a functional equivalent to every single form used in OO and imperative languages.

Expressions

- Functional languages also possess the same simple, numeric and logical operators as their imperative and OO counterparts.
- The only difference in how they evaluate expressions is that in functional languages you can find both lazy and strict evaluation.
- Despite the differences there is a relevant similarity in writing simple operations and in the structure of expressions.

Complex Data

List data structure is significantly more complicated than tuple, however, in return for its complexity it empowers the programmer with possibilities that no other data structure is capable of doing.

List expression is a construction for creating lists, tied strongly to them. Its base is the Zermelo-Fraenkel set expression.

The list expression is in fact a generator, which, strangely, does not define elements of the list but gives the criteria of being part of it and regulates the number of elements and how they should be created

Functions a Recursion

As we have mentioned earlier, we define functions and an initial expression in the modules of functional programs and start evaluation with the initial expression.

Of course, this is not true to library modules, in which you want to make the call of several, or all of the functions, possible for the outside world.

Functions a Recursion

Naturally, functions can call other functions or themselves. This phenomenon, when a function calls itself, is called recursion. recursion can be found in OO languages as well, but its use can cause several problems, since the number of recursive calls is rather limited.

Industrial Use of Functional Languages

The spreading of functional languages in the industry and in telecommunication is not negligible. Erlang is used for creating communication and highly fault tolerant systems and Clean appears in several industrial projects.

Industrial Use of Functional Languages

Systems carrying out mathematical calculations are programmed in functional languages and people involved in other branches of science, like physics and chemistry are also empowered to work quickly and effectively with them because mathematical formulas can be easily transcribed to the dialect of the particular programming language.

Functional languages

Roland Király

EKF TTK

Institute of Mathematics and Informatics