**Scala application using AKKA**

1. Introduction

The hardware has been developing very quickly in the past few years, in fact so quickly that the software has not been able to keep up with it. The most popular programming languages like Java or C# cannot use multicore processor to the fullest of its abilities. That’s why it has become very important to create languages that would be able to cope with the modern hardware.

One of those languages is Scala, created by Martin Odersky, a German computer scientist.

1. Programming paradigms

A paradigm, in science, is defined as a set of concepts or thought patterns in a given discipline. (ref. Odersky – 1st course). In computer science a programming paradigm is a way of creating the structure and components of computer applications. Computer scientists distinguish between several programming paradigms, the main ones are:

* imperative
* functional
* object-oriented
* logic
* symbolic.

Some, like Martin Odersky, the creator of Scala, name only the first 3 as main programming paradigms and claim for the object-oriented to be orthogonal to the 3 paradigms and one that combines the features of the 3 paradigms.

Imperative programming uses statements for computation. Statements change the state of the program. In imperative programming, the mutable variables are modified, assignments are used, and so are control structures such as if-then-else, loops, break, continue, return. Programs written using imperative programming paradigm specify a list of tasks that the computer is to perform. Imperative programming indicates how the program should achieve the end result (in what sequence the tasks should be executed) and not what the program should accomplish, like in declarative programming, which is the opposite to imperative programming. Functional and logic programming are examples of declarative programming.

Functional programming uses functions to perform calculations. It does not use mutable variables, assignments, loops and other imperative control structures. In functional programming functions can act like values and those values can be produced, consumed and composed. Additionally functions can be defined in other functions, they can be passed as parameters and returned as results.

Logic programming uses mathematical logic to create programs. Programs written using programming paradigm consist of sequences of logical statements that present facts and rules of the domain problem and an inference algorithm.

Object-oriented programming uses objects to describe the current world state. Objects are described by data fields and the behaviour of the objects is described by methods. Objects then, as instances of classes, interact with one another to create applications and programs.

1. Functional languages

<http://en.wikipedia.org/wiki/Functional_programming>

1. Object-oriented languages
2. Scala

* Parallelism
* concurrency

1. AKKA
2. Application
3. References:

* <http://www.drdobbs.com/architecture-and-design/interview-with-scalas-martin-odersky/231001802>
* Wikipedia