



Universidade Federal de Pernambuco

Centro de Informática
Pós-graduação em Ciência da Computação

2014.1

A large, faint watermark of the UFPE coat of arms is centered in the background. It features three red flames above a shield containing a figure, with a red banner at the bottom.

Documento de escopo do projeto

Paradigma de Linguagens de Programação

Equipe: Bruno Cesar Ferreira Silva (bcfs@cin.ufpe.br)
Bruno Medeiros (bmo@cin.ufpe.br)
Leandro Santana de Melo (lsm3@cin.ufpe.br)

Recife, Maio de 2014

1. Introdução

Projeto da disciplina Paradigmas de Linguagens de Programação, ministrada pelo professor Augusto Sampaio, lecionada no programa de pós-graduação do Centro de Informática da Universidade Federal de Pernambuco (www.cin.ufpe.br).

O presente trabalho tem como objetivo criar uma linguagem com características do paradigma Orientado a Objetos e com características do paradigma funcional. Para a construção das características funcionais será usada como base a linguagem LF3 (apresentada na referida disciplina) e para as características OO serão utilizadas como base as linguagens Java e Scala.

Com o intuito que a linguagem estendida contenha características de ambas as linguagens; entretanto, mantenha os conceitos fundamentais de cada paradigma, isto é, prover a integração destas linguagens, de maneira ortogonal, formando uma linguagem multiparadigma.

2. Características de Orientação a Objetos

Dos conceitos de orientação a objetos no projeto utilizará os seguintes:

1. **Classe**, representa um conjunto de objetos com características afins. Uma classe define o comportamento dos objetos através de seus métodos, e quais estados ele é capaz de manter através de seus atributos.
2. **Objeto / instância de uma classe**, Um objeto é capaz de armazenar estados através de seus atributos e reagir a mensagens enviadas a ele, assim como se relacionar e enviar mensagens a outros objetos.
3. **Atributo** são características de um objeto.
4. **Método** definem as habilidades dos objetos.
5. **Mensagem** é uma chamada a um objeto para invocar um de seus métodos, ativando um comportamento descrito por sua classe.
6. **Herança** (ou generalização) é o mecanismo pelo qual uma classe (sub-classe) pode estender outra classe (super-classe), aproveitando seus comportamentos (métodos) e variáveis possíveis (atributos).

3. Paradigma Funcional

Dos conceitos de orientação a objetos no projeto utilizará os seguintes:

1. Avaliação preguiçosa (Lazy Avaluation)
2. Aplicação parcial
3. Overloading
4. Listas e verificação de tipos de listas
5. Casamento de padrão com equações condicionais
6. Funções como valor

4. Recursos adicionais

Como escolha de projeto será implementado os seguintes conceitos:

1. Tipagem estática

2. Tipos primitivos (boolean, char, int, float, long, double, String, Character , funcValue)
3. Comandos de controle (if,while,for)
4. Expressão lógica (Expression "?" Expression ":" Expression)

5. Motivação

A proposta para a nova linguagem proporciona ao usuário os benefícios de uma linguagem funcional, por exemplo, lazy evaluation e valor função, ao mesmo tempo que agrega toda a facilidade de controle, manipulação e reuso do paradigma Orientado a Objetos.

4. Referências

<http://docs.oracle.com/javase/6/docs/api/java/util/Collection.html>
<http://www.stanford.edu/group/coursework/docsTech/jql/user/Maps.html>
<http://www.caelum.com.br/apostila-java-orientacao-objetos/collections-framework/>