Estudo sobre a Linguagem de Programação Groovy

Felipe Melo e Pedro Aleph

[1]Departamento de Ciência da Computação – Universidade Federal de Roraima (UFRR)
Boa Vista – RR – Brazil

felipesmelo196@gmail.com, Pedro.aleph.allen@gmail.com

Resumo. O projeto em questão, desenvolvido na disciplina de linguagem de programação do professor Herbert Rocha, seu objetivo era de ter um código que gerasse um programa que faria sugestão e gerenciamentos de músicas com informações retiradas de sites de rádios, mas como não foi possível devido a impotência dos membros, nós optamos ao menos apresentar esta linguagem por acha-la interessante.

1. História do surgimento da linguagem

No dia 29 de agosto de 2003 James Strachan publicou em seu blog o primeiro artigo sobre aquilo que viria a ser o Groovy (veja em Links o endereço do post). Ele deixava bem claro as suas intenções na época: “minha idéia inicial é fazer uma pequena linguagem dinâmica, que seja compilada diretamente em classes Java e que tenha toda a produtividade elegante encontrada em Ruby e Python, mas que permita reusar, estender, implementar e testar código Java já existente”.

James uniu-se com Bob McWhirter,e juntos, em 2003 fundaram oficialmente oprojeto Groovy. Em 2004, com a fundação do Groovyone, mais desenvolvedores se uniram ao projeto e a coisa decolou. Foi criada a Groovy Language Specification (GLS) e o kit para testes de compatibilidade (o TCK), além do parser básico da linguagem. O embrião do projeto estava pronto e a partir daí teria mais como voltar atrás.

Groovy evoluiu desconhecido por algum tempo, e até dezembro de 2007 várias versões foram lançadas sob o número 1.1.x. Em 7 de dezembro de 2007 a versão final da família 1.1 foi lançada, e então nomeada Groovy 1.5 devido às diversas modificações realizadas na mesma.

http://www.devmedia.com.br/java-magazine-69-um-pouco-de-groovy-uma-linguagem-de-scripts-100-compativel-com-java/12542 (terça-feira, 19 de julho de 2017)

2. Domínios de aplicação

Não foi encontrado resultados específicos que utilizam de aplicações em groovy, somente que por ser uma simplicação de java pode substitui-la em aplicações em java, e que é usada no desenvolvimento de Web com framework Grails como sites google.

3. Paradigmas suportados pela linguagem

Orientada a Objetos

De modo semelhante a java:

```
class Aluno {
  // definimos as propriedades
  String nomeAluno
  Integer matriculaAluno

  // variavel para a matricula
  static int matricula = 0

  // Construtor
  Aluno(String nome) {
    matricula = matricula + 1
    this.setNomeAluno(nome)
    this.setMatriculaAluno(matricula)
  }

  public String toString() {
    return "Matricula: " + this.getMatriculaAluno().toString() +
      " Nome: " + this.getNomeAluno()
  }
}
```

(também suporta paradigma funcional)

4. Variáveis e tipos de dados

Não existem tipos primitivos em Groovy, ou seja, todas as variáveis são objetos, até mesmo valores literais escritos diretamente no código como, por exemplo, true, false, 10.

Valores literais

Classe	Valor literal de exemplo
java.lang.Long	1000L
java.lang.Integer	100
java.lang.Float	1.0F
java.lang.Double	100D
java.math.BigDeciamal	1000.50
java.math.BigInteger	123G

Char recebe um caractere ex.: ´z´.

String recebe um conjunto de caracteres ex.: groovy.

4. Comandos de controle

if / else, exemplo:

```
        int idade = 19
        if(idade > 18){
                println("você já é adulto");
        }else{
                println("você não é adulto");
         }
```

Switch, exemplo:

```
switch(variavel_a_ser_avaliada){

    case [valor_1]:
      [bloco_de_codigo]
      break

    case [valor_2]:
      [bloco_de_codigo]
      break
    case [valor_3]:
      [bloco_de_codigo]
      break
    default:
     [bloco_de_codigo]

  }
```

5. Escopo (regras de visibilidade)

Endentação:

Na linguagem existe endentação tanto por tab quanto por espaço;

Função:

Para executar parâmetros em uma função, não é preciso declarar tipos de variáveis, somente fazer com que as variáveis recebam alguma informação para que não retorne resultados do tipo null, ex: a = 2, b = 3, return a+b ;

6. Exemplo prático de uso da linguagem de programação

A partir desse ponto serei mais ousado no que digitar, espero que isso não tire o mínimo que posso extrair de pontos do pouco conteúdo contido neste relatório. Como dito no resumo não foi possível a criação do código previsto no início do projeto a tempo para os limites da conclusão, mas ainda assim prendendo fazer algo mesmo que não tenha valor nenhum, para mostrar a funcionalidade da linguagem.

7. Conclusões

Groovy é uma linguagem que merece destaque, e por isso estou fazendo o que está aqui, mesmo que não tenha valor necessário para passar na disciplina, pretendemos apresentar Groovy, e de uma forma bem explicada, não é certo que vou fazer exatamente isso já que tenho compromisso em outras disciplinas, mas ainda assim tentaremos algo.

8. Referências

http://www.devmedia.com.br/java-magazine-69-um-pouco-de-groovy-uma-linguagem-de-scripts-100-compativel-com-java/12542

http://www.devmedia.com.br/tipos-de-dados-em-groovy/34187

http://www.devmedia.com.br/linguagem-de-programacao-groovy-introducao/34099