

# Desenvolvimento de uma linguagem com tipos estruturais

## Sumário

1	Informações	1
2	Tema	1
3	Resumo	2
4	Objetivo	2
5	Atividades já realizadas	3
6	Cronograma de atividades	3
7	Estrutura da monografia	4

## 1 Informações

- **Orientador:** Prof. Francisco Reverbel
- **Aluno:** Henrique Stagni

## 2 Tema

O trabalho proposto consiste na especificação de uma linguagem de programação com *tipos estruturais* e no desenvolvimento de um compilador para

essa linguagem, capaz de produzir programas executáveis pela JVM<sup>1</sup>.

### 3 Resumo

Linguagens com *tipagem dinâmica* têm sido alvo de grande atenção nos últimos anos. Por não se basearem em sistemas estáticos de tipagem – que geralmente são muito rígidos – esse tipo de linguagem facilita a produção de um código mais flexível, reusável e, portanto, de sistemas capazes de se adaptar melhor à mudanças de especificações. Por outro lado, a não checagem de tipos em tempo de compilação também traz algumas desvantagens. A principal delas é que, evidentemente, muitos erros que seriam detectados durante a compilação em linguagens estáticas só são detectados em tempo de execução em linguagens dinâmicas. Em muitos casos, estes erros só são acusados em um lugar do código que é executado bem depois do que o trecho onde a inconsistência de tipos é realmente feita, dificultando o processo de debug. Além disso, linguagens dinâmicas podem apresentar um desempenho inferior ao alcançado por linguagens estáticas, que muitas vezes possuem compiladores capazes de gerar códigos otimizados baseados nas informações providas pela tipagem estática.

Linguagens com *tipos estruturais* podem ser consideradas, de certa forma, como um meio termo entre linguagens com tipagem estática e dinâmica. Essas linguagens possuem um sistema de tipos checado em tempo de compilação. Mas ao contrário de linguagens estáticas nominativas (como C, Pascal ou Java) – onde tipos são definidos por nomes explicitamente declarados – os tipos são dados implicitamente pelo conjunto de operações suportado por cada objeto da linguagem. Esse sistema de tipos permite uma flexibilidade quase tão grande quanto aquela fornecida por linguagens dinâmicas, mantendo, ainda assim, a possibilidade de detecção de muitos erros em tempo de compilação.

### 4 Objetivo

O objetivo principal do trabalho proposto é especificar e desenvolver uma linguagem com tipos estruturais para a Java Virtual Machine (JVM). A JVM

---

<sup>1</sup>Java Virtual Machine: <http://java.sun.com/docs/books/jvms/>

foi escolhida como alvo por ser um ambiente disponível em diversas plataformas e arquiteturas, por possuir uma grande quantidade de bibliotecas, e por possuir, em suas implementações mais usuais, um sistema de otimização adaptativa, capaz de aumentar significativamente o desempenho dos programas que rodam sobre ela. O desenvolvimento de uma linguagem desse tipo para a JVM não é uma tarefa trivial, uma vez que essa máquina virtual foi originalmente projetada para fazer checagens estáticas de tipos e só recentemente vem sofrendo mudanças voltadas para linguagens com um caráter mais dinâmico.

O desenvolvimento do trabalho proposto consistirá na especificação da linguagem para a qual um compilador será gerado e pelo desenvolvimento do compilador propriamente dito. Este último deve ser capaz de inferir os tipos estruturais dos programas de entrada – detectando possíveis erros – além de gerar os *bytecodes* correspondentes – de modo a permitir que esses programas sejam executados pela JVM.

## 5 Atividades já realizadas

Até o momento, foi estudada a estrutura e a especificação da JVM, dando especial ênfase às recentes melhorias que estão sendo adicionadas para facilitar a implementação de linguagens dinâmicas para a plataforma. Foram produzidos alguns protótipos que produzem *bytecodes* executáveis para a JVM e que fazem uso de algumas dessas novas funcionalidades. Nesses protótipos, uma biblioteca (**ASM**)<sup>2</sup> foi escolhida para facilitar a geração dos *bytecodes*.

Geradores de analisadores sintáticos e construtores de árvores sintáticas também foram estudados e foram feitos protótipos capazes de analisar sintaticamente algumas linguagens simples, usando a ferramenta **JavaCC**<sup>3</sup>. Por fim, também foram estudados alguns sistemas de tipos e algoritmos de inferências de tipos que são usualmente empregados em linguagens com tipos estruturais.

## 6 Cronograma de atividades

---

<sup>2</sup>ASM framework: <http://asm.ow2.org/>

<sup>3</sup>Java Compiler Compiler: <http://javacc.java.net/>

Atividade/Mês	Fev	Mar	Abr	Mai	Jun	Jul	Ago	Set	Out	Nov
Estudo sobre a geração de analisadores sintáticos	•	•								
Estudo sobre a geração de <i>bytecodes</i> para JVM		•	•	•						
Estudo de sistemas de tipos e inferência de tipos				•	•	•	•			
Especificação da linguagem					•	•	•			
Implementação da linguagem						•	•	•	•	•
Elaboração da Monografia							•	•	•	•

## 7 Estrutura da monografia

- **Parte Objetiva:**

- **Introdução e Objetivos**
- **Definições e conceitos/terminologia:** Explicação e fixação de terminologia sobre linguagens com tipos dinâmicos, estáticos. Explicação sobre o porquê da escolha da JVM como alvo.
- **Definição do sistema de tipos da linguagem:** Definição do sistema de tipos da linguagem que será desenvolvida, assim como o algoritmo de inferência de tipos usado.
- **Especificação da linguagem** Especificação da linguagem que será desenvolvida.
- **Detalhes técnicos sobre a implementação do compilador:** Explicação de alguns detalhes do compilador, em especial sobre a geração dos *bytecodes* correspondentes às diferentes construções aceitas pela linguagem.
- **Conclusão:** Resultados e considerações finais sobre o trabalho.
- **Referências Bibliográficas**

- **Parte Subjetiva:**

- **Dificuldades e desafios encontrados**
- **Disciplinas relevantes para o trabalho**
- **Trabalhos futuros**