

**Arthur Branco Costa - 7278156**  
**Daniel Paulino Alves - 7156894**  
**Felipe Yamaguti - 7295336**  
**Pedro Paulo Vezz  Campos - 7538743**  
**Thiago Tatsuo Nagaoka - 7289197**

## ***Conceitos da Linguagem Modula-3***

S o Paulo - SP, Brasil

2 de abril de 2012

**Arthur Branco Costa - 7278156**  
**Daniel Paulino Alves - 7156894**  
**Felipe Yamaguti - 7295336**  
**Pedro Paulo Vezz  Campos - 7538743**  
**Thiago Tatsuo Nagaoka - 7289197**

## ***Conceitos da Linguagem Modula-3***

Primeiro exerc cio-programa apresentado para avalia  o na disciplina MAC0316, do curso de Bacharelado em Ci ncia da Computa  o, turma 45, da Universidade de S o Paulo, ministrada pela professora Ana Cristina Vieira de Melo.

DEPARTAMENTO DE CI NCIA DA COMPUTA  O  
INSTITUTO DE MATEM TICA E ESTAT STICA  
UNIVERSIDADE DE S O PAULO

S o Paulo - SP, Brasil

2 de abril de 2012

# *Sumário*

<b>1</b>	<b>Introdução</b>	p. 4
<b>2</b>	<b>Tipos primitivos e compostos</b>	p. 5
2.1	Tipos primitivos . . . . .	p. 5
2.1.1	INTEGER . . . . .	p. 5
2.1.2	Cardinal . . . . .	p. 5
2.1.3	Boolean . . . . .	p. 6
2.1.4	Char . . . . .	p. 6
2.1.5	Enumerated . . . . .	p. 6
2.1.6	Reference . . . . .	p. 6
2.2	Tipos Compostos . . . . .	p. 6
2.2.1	Array . . . . .	p. 6
2.2.2	Record . . . . .	p. 6
2.2.3	Set . . . . .	p. 6
2.2.4	Object . . . . .	p. 6
<b>3</b>	<b>Variáveis simples e compostas</b>	p. 7
3.1	Variáveis simples . . . . .	p. 7
3.2	Variáveis compostas . . . . .	p. 7
<b>4</b>	<b>Variáveis quanto a sua existência</b>	p. 8
4.1	Variáveis globais . . . . .	p. 8
4.2	Variáveis locais . . . . .	p. 8

4.3	Variáveis <i>heap</i> . . . . .	p. 8
4.4	Variáveis persistentes . . . . .	p. 8
<b>5</b>	<b>Forma e tempo de vinculação de tipos às variáveis</b>	<b>p. 9</b>
<b>6</b>	<b>Sistemas e verificação de tipos utilizados na linguagem</b>	<b>p. 10</b>
<b>7</b>	<b>Abstrações</b>	<b>p. 11</b>
	<b>Referências</b>	<b>p. 12</b>

# *1 Introdução*

Programas são feitos com o intuito de, através de um conjunto de pré-condições, manipular os dados de entrada ao longo de processos, de forma a produzir um resultado útil e eficaz, desde que elas sejam respeitadas.

Eles podem diferir em diversas características: paradigma, lógica computacional, eficiência, legibilidade, simplicidade, entre outros. No entanto, um fator poderia ser classificado como principal: a linguagem de programação na qual eles são escritos. Com muitos de seus conceitos já pré-definidos pelos seus projetistas, e a possibilidade de que o programador desenvolva outros, as linguagens de programação são altamente relacionadas aos fatores mencionados.

Assim, a escolha adequada da linguagem determina a expressividade e facilita a criação de um bom programa. Desta forma, é necessário que o programador tenha um bom conhecimento da linguagem escolhida, para que possa obter seu maior aproveitamento.

Este trabalho visa analisar as propriedades presentes na linguagem Modula-3 e os conceitos a serem apresentados decorrem de tal necessidade. [1] [2]

## **2    *Tipos primitivos e compostos***

Para alcançar o seu objetivo, um programa deve lidar com valores. Para isso, é preciso que operações que os relacionem sejam definidas. A fim de evitar a repetição excessiva dessas definições, os valores são agrupados em conjuntos de mesma espécie com propriedades próprias: os tipos.

Divididos em tipos primitivos e compostos.

### **2.1   Tipos primitivos**

Ocorrem caso seus valores não possam ser subdivididos em elementos pertencentes ao mesmo conjunto. Ou seja, as partes podem existir, mas não são semanticamente relacionadas.

Eles dividem-se em Integer, Cardinal, Boolean, Char, Enumerated e Reference.

#### **2.1.1   INTEGER**

Lorem ipsum

#### **2.1.2   Cardinal**

Lorem ipsum

### **2.1.3 Boolean**

### **2.1.4 Char**

### **2.1.5 Enumerated**

### **2.1.6 Reference**

## **2.2 Tipos Compostos**

Este tipo diz respeito a um conjunto de valores cujos elementos podem ser separados em partes menores e mais simples, que ainda pertencem ao todo.

Dividem-se em Array, Record, Set e Object.

### **2.2.1 Array**

### **2.2.2 Record**

### **2.2.3 Set**

### **2.2.4 Object**

## **3     *Variáveis simples e compostas***

Programas manipulam e armazenam valores através de variáveis. Elas são caracterizadas por um identificador, um endereço de memória e um valor associado.

Do ponto de vista de seus armazenamento e acesso são classificadas em simples e compostas.

### **3.1   Variáveis simples**

Armazenam valores que são acessados de maneira pontual, ou seja, a modificação e o acesso de seus valores são feitos como um todo.

a: INTEGER

### **3.2   Variáveis compostas**

Variáveis que armazenam valores que podem ser acessados e modificados individualmente. Isto é, o tratamento de seus elementos é seletivo.



## **4    *Variáveis quanto a sua existência***

Variáveis também podem ser caracterizadas segundo sua existência. Tal classificação abrange variáveis globais, locais, *heap* e persistentes.

### **4.1   Variáveis globais**

São variáveis que existem antes do programa ser executado e permanecem ativas até o seu término.

### **4.2   Variáveis locais**

Variáveis cuja existência é delimitada pelo escopo no qual está inserida. Logo, mostram-se ativas desde o momento de sua declaração até o fim do escopo.

### **4.3   Variáveis *heap***

Ao contrário das anteriores, a existência destas é condicionada à alocação e desalocação dinâmicas de memória. Portanto, elas são criadas durante o tempo de execução do programa e encerradas após a desalocação de sua memória ocupada, ou no final, prática não recomendada por gerar vazamento de memória, pois cabe ao sistema operacional forçar a desalocação desse espaço.

### **4.4   Variáveis persistentes**

Estas variáveis destacam-se das demais por serem as únicas que, com o encerramento do programa, não são destruídas. Desta maneira, suas informações são armazenadas para uso posterior.

## ***5    Forma e tempo de vinculação de tipos às variáveis***

## ***6    Sistemas e verificação de tipos      utilizados na linguagem***

## ***7 Abstrações***

## *Referências*

- [1] MELO, A. C. V. de; SILVA, F. S. C. da. *Princípios de Linguagens de Programação*. São Paulo, SP: Edgard Blucher, 2003.
- [2] CARDELLI, L. et al. Modula-3 language definition. *SIGPLAN Notices*, v. 27, n. 8, p. 15–42, 1992.