Tipos e Semântica

Conceitos de Linguagens de Programação

Tipos e Semântica

Stefano Azevedo da Silva

Tipificação

Vantagens, Desvantagens, Exemplos

Tipificação

- Uma linguagem Tipificada ou Tipada é aquela que permite a declaração de variáveis de tipos específicos;
- As linguagens podem ser classificadas em Fortemente Tipadas, Fracamente Tipadas e Não tipadas;

Vantagens

- Alocação de espaço necessário de memória para a variável;
- Funções com tipos de dados definidos;
- Possuem maior Aprendibilidade por possuir códigos mais legíveis;
- Possibilidade de criação de tipos em algumas linguagens, Ex: Java;

Desvantagens

- Códigos maiores;
- Conversões de Tipos podem gerar erros;
- Limitação no tamanho das variáveis

Linguagens Fortemente Tipadas

- São aquelas que possuem declaração de tipos obrigatórios e explícitos;
- Existem tipos específicos: int, float, boolean*, byte*, etc.
- Exemplos: Java, C, C++, C#, Fortran, Cobol;
- Exemplos em Java.

Exemplos

```
public class Exemplo{
    public static void main(String[] args){
        int numero;
        char caracter;
        String nome;
        boolean verdadeiro;
    }
```

Exemplo (2)

- boolean Cadastra(String nome, int idade, Date nascimento, String Endereco, char letra);
- public final class String;
- int Vetor[TAM];
- struct Pessoa{

  String Nome;

  int idade;

  Date nascimento;

  };
- float real = (int) inteiro;*

Exemplo (3)

- Tamanho de alocação dos tipos:
  - int : 4 bytes;
  - float : 4 bytes;
  - char :2 bytes;
  - byte :1 byte;
  - long : 8 bytes;
  - Vetor: Tipo x Tamanho : int Vet[40] = 40x4 bytes;
  - String : Vetor de char (basicamente)

Linguagens Fracamente Tipadas

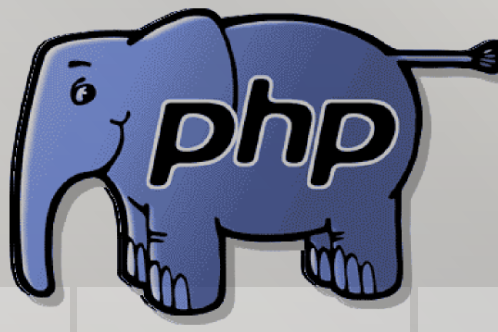
- Conhecidas como Dinamicamente Tipadas;
- Possuem os tipos durante a execução do programa;
- O programa pode alterar o tipo de dado contido na variável;
- Não possui cast (conversão);
- Exemplos: PHP, JavaScript, Ruby, Python;
- Exemplos em PHP.

Exemplos

- **<?php**

```
$numero = 0;
$nome = “John”;
$cpf = 016.578.462-01;
$salario = 205,50;
```

?>



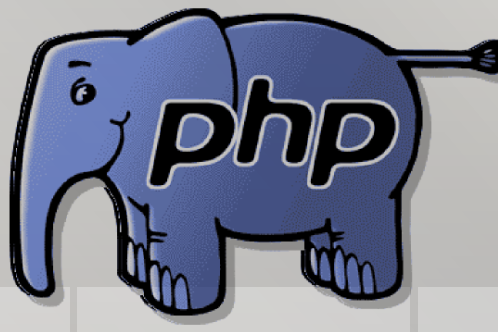
Exemplos (2)

- **<?php**

```
$numero = 0;
$nome = “John”;
$cpf = 016.578.462-01;
$salario = 205,50;

$numero = ‘Maria’
$nome = 2500;
```

?>



Exemplos (3)

- **<?php**

  $a = 50;

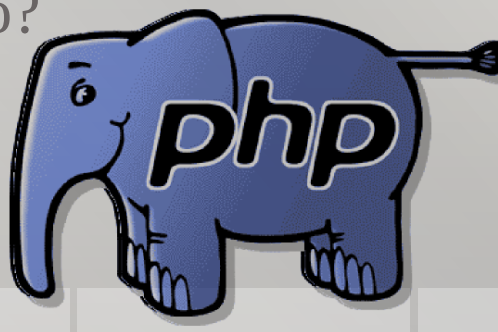
  $b = 60;

  ...

  $retorno = function($a, $b);

  ?>

- E se $a ou $b forem alterados durante a execução?



Linguagens Não Tipadas

- Não possuem tipos de dados;
- Podem possuir tipos de dados genéricos;
- Exemplos: Assembly, BCPL, Perl;

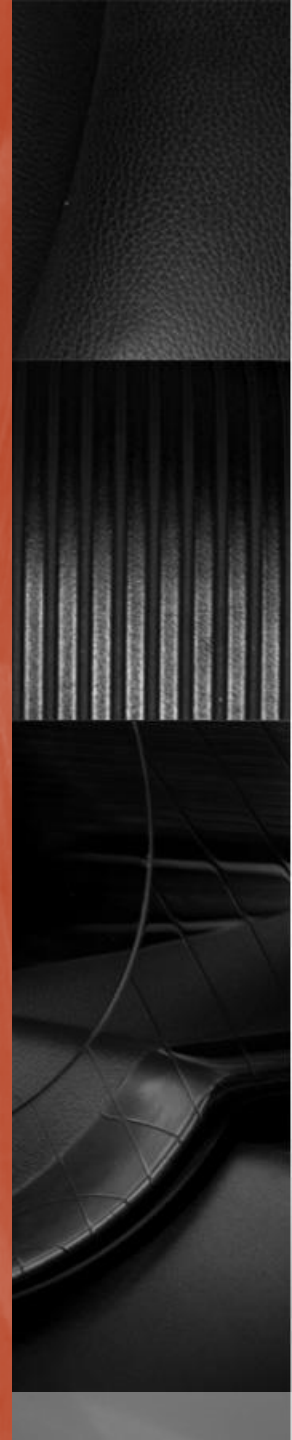
- Exemplos em Assembly.

Exemplos

- **MOV AL**, 1h ;
- **MOV CL**, 2h ;
- **MOV DL**, 3h ;

Linguagens com tipo Estático

Haskell



Haskell

- É uma linguagem puramente Funcional;
- Desenvolvida em 1990 nomeada em homenagem ao lógico Haskell Curry;
- Utiliza o conceito de Avaliação Preguiçosa;
- Criada para ser de fácil ensino, utiliza sintaxe e semântica formal, e de open source;

Avaliação Preguiçosa

- Técnica de Programação para atrasar a computação até um ponto em que o resultado da computação é considerado suficiente, necessário;
- Aumenta o desempenho evitando cálculos desnecessários;
- Evita erros na avaliação das expressões;
- Utiliza funções de repetição com funções regulares, no lugar de comandos internos.

Avaliação Preguiçosa

- Suponha a função pow, que acha a potência do valor: x = 5;
- pow(pow(pow(pow(x))));
- Na avaliação preguiçosa, o valor do primeiro pow (mais externo) só será encontrado quando o valor for realmente pedido pelo usuário;

Haskell

- Em Haskell existem apenas funções, e todas são unárias (não existe F(a,b) e sempre F(a));
- Exemplo:
  - soma x y
- Parece uma função binária, mas na verdade é unária;
- A função soma x dado y, retorna x+y e a função soma que dado x, retorna x+;

Haskell

- A linguagem haskell é estaticamente tipada pois quando o código for compilado, o compilador saberá os tipos das variáveis baseado nos valores atribuídos;
- Portanto, se tentar adicionar um numero a uma string haverá um erro, durante a compilação;

Explicação

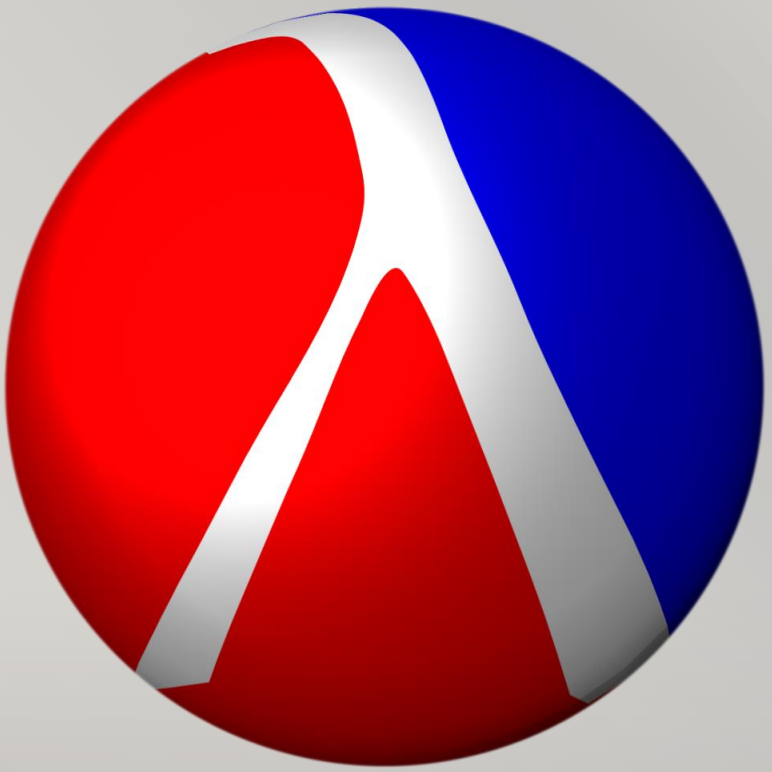
- (soma x) y -> soma x a y
- (soma) x -> soma x

Exemplos

- http://tryhaskell.org

Linguagem Dinamicamente Tipada

Scheme



Scheme

- É uma Linguagem de Programação multiparadigma (funcional, procedural);
- Criada a partir da linguagem Lisp;
- Possui uma filosofia minimalista, possuindo pouca sintaxe comparada a outras linguagens;
- Utiliza notação Prefixa (Notação Polonesa)
- Não possui precedência de operadores;
- Notação Parentizada;

Exemplo

Em Linguagem C/Java	Scheme
a+b	(+ a b)
(a+b)*c	(* (+ a b) c)
(a*b)+(c*d)	(+ (* a b) (*c d)
If(a==b) { a+b };	(if (= a b) (+ a b))

Exemplo (2)

- (define (fatorial n)

  (cond ((= n 0) 1)

  (else (* n (fatorial (- n 1))))))

  (fatorial 5) ;; => *120*

Scheme

- Nomes em Scheme podem consistir em letras, em digitos(não no começo), caracteres especiais(exceto parenteses)
- Scheme não é case Sensitive
- O interpretador Scheme é um laço infinito de leitura-avaliação-escrita. Lê repetidamente uma expressão digitada, interpreta-a e exibe o resultado
- Os parâmetros em Scheme são passados por valor, assim, indepedente se as funções alteram o valor real dos parâmetros não são afetados.

Exemplos (3)

(if (> 3 2) 'yes 'no) --> yes

(if (> 2 3) 'yes 'no) --> no

(if (> 3 2) (- 3 2))--> 1

Exemplo

- (let loop ((n 1))
  (if (> n 10) '()
  (cons n
  (loop (+ n 1)))))

Referências

- http://tryhaskell.org
- http://pt.wikibooks.org/wiki/Haskell
- http://haskell.tailorfontela.com.br/chapters
- http://pt.wikipedia.org/wiki/Scheme
- http://pt.wikipedia.org/wiki/Nota%C3%A7%C3%A3o_polonesa
- Slides Passados

Semântica

Conceitos de Linguagens de Programação

Introdução a Sintaxe

- Sintaxe é a forma como a linguagem de programação é escrita;
- Muitas vezes a sintaxe é escrita mas não se sabe seu significado;
- Diferentes linguagens podem usar sintaxes parecidas;
- Mas também podem usar sintaxes totalmente diferentes com a mesma semântica

Semântica

- É o complementar da Sintaxe, descrevendo o significado das instruções;
- A semântica é a mesma em qualquer linguagem;
- É preciso saber usar a sintaxe, mas é muito mais importante saber a semântica, em programação;

Exemplo

- Linguagens com sintaxes parecidas:
- C/C++ e Java
  - int a;
  - float b;
  - a = b;

Exemplo (2)

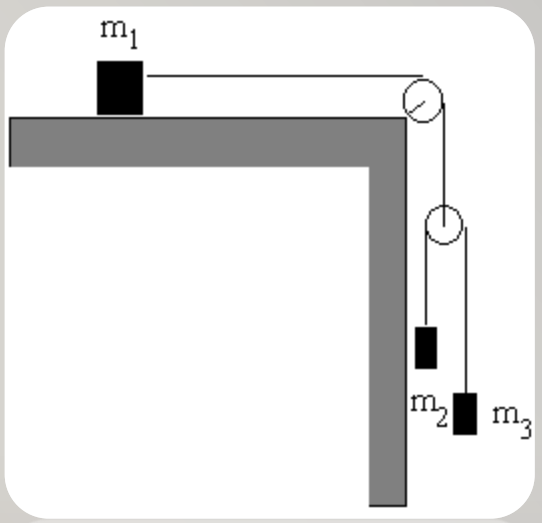
- Linguagens com Sintaxes diferentes, mas a mesma semântica;
- C, Php, Java:
  - C: printf(“Hello World”);
  - Java: System.out.printf(“Hello World”);
  - Php: echo ‘Hello World’;

Semântica

- Em Programação, a Semântica é dividida em Dinâmica e Estática;
- Na Dinâmica existem divisões de métodos:
  - Operacional
  - Denotacional
  - Axiomática
- Na Estática existem as:
  - De Ações
  - Algébricas

Semântica Dinâmica

Operacional, Denotacional e Axiomática



Conceito

- Representa os significados das expressões, instruções e operações do código;
- Como o programa é executado?
- Que operações são realizadas?
- O que o programa significa?
- Que objetos matemáticos ele denota?
- Quais proposições lógicas são válidas para um programa?

Semântica Operacional

- Descreve o significado do programa pela execução das instruções
- Utiliza alguns métodos para explicar como cada instrução é feita pelo Hardware (ou máquina virtual, se for o caso)
- Depende de algoritmos, não da matemática

Semântica operacional

- Comparativo:

C	Semântica Operacional

for (i=0; i<100; i++) {
...
}
	i=0;
	loop: if i == 100 goto out
	...
	i++;
	goto loop
	out:...

Semântica Denotacional

- Baseado nas teorias das funções recursivas
- Método mais informal de descrição da semântica
- Na semântica denotacional, as mudanças de estados são definidas por funções matemáticas

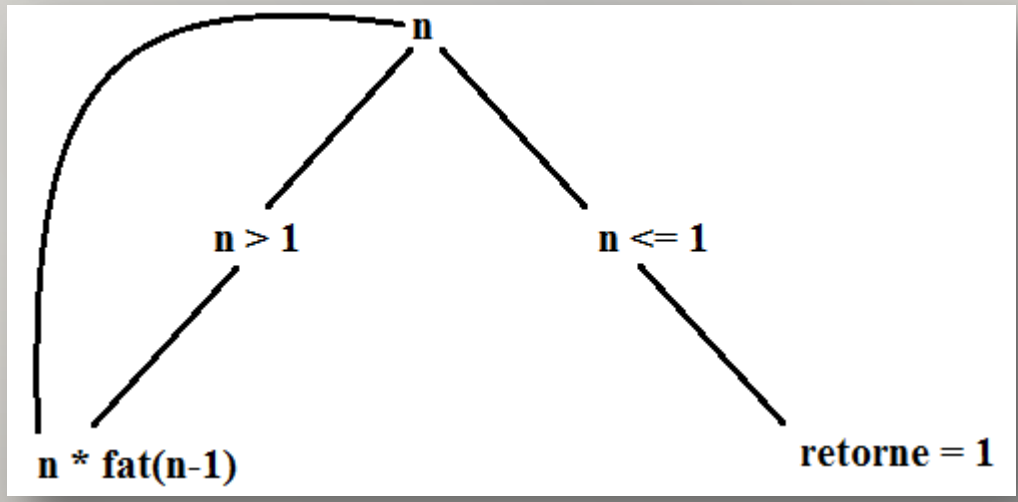
Exemplo

- Função recursiva

```
função fatorial(n) {
        se (n <= 1) retorne 1;
        senão retorne n * fatorial(n-1);
}
```

Semântica Denotacional

F(n) -> n * F(n-1)



Semântica Axiomática

- Baseada em Lógica de Predicados
- Verificação formal do programa
- Cada expressão é um Predicado ou asserção
- Utilizada para provar algoritmos
- Possui pré e pós condições de execução

Exemplo

soma = 5*x + 50 (soma >= 50)

Pré-Condição seria x>=0

Pós Condição

Semântica Estática

Algébrica, de ações

Conceito

- Em resumo pode-se entender Semântica Estática como o conjunto de restrições que determinam se programas sintaticamente corretos são válidos.
- Compreende checagem de tipos, análise de escopo de declarações, checagem de número e tipo de parâmetros de funções/procedimentos, ou seja, checagem em tempo de compilação.
- Exemplo:
  - Todo identificador deve ser declarado antes de ser usado.
  - Os parâmetros utilizados na chamada de uma função têm o tipo correto

Algébrica

- Usa a Álgebra Abstrata para definir as propriedades e operações de objetos da mesma categoria
- Especifica os tipos abstratos de dados
- Visa a consistência dos programas, mas torna o código menos legível

Semântica de Ações

- Formalismo para definição de linguagens de programação.
- Define um mapeamento da sintaxe do programa para o seu significado.
- Significado de programa é dado através da notação de ações.

Referências

- http://sibetim.blogspot.com.br/2009/03/sintaxe-e-semantica-de-linguagens-de.html
- Slides dos Semestres passados