

<b>Departamento de Matemática da Universidade de Coimbra</b>		
<b>2010/2011</b>	<b>Programação Orientadas para os Objectos</b>	<b>Projecto 1</b>

## Cálculo de Expressões Aritméticas

A leitura e posterior cálculo de uma expressão aritmética na notação usual (a notação infixa) é algo um pouco complicado de implementar, isto dado ser necessário considerar algo que não está explícito na própria expressão, as prioridades das operações. Por exemplo: a expressão  $x + y \times z$ , não é exactamente aquilo que pode parecer à primeira vista, a sua leitura não se pode fazer da forma usual, da esquerda para a direita. Por convenção a multiplicação tem precedência sobre a adição e como tal deve ser feita em primeiro lugar, isto é a expressão deve ser lida como  $x + (y \times z)$ , com a convenção de que tudo o que está dentro de um par de parêntesis é calculado primeiro.

Fica então evidente que a notação infixa é uma notação ambígua em si mesmo, necessitando de informação externa para poder ser lida de forma correcta. Este facto dificulta o seu tratamento automático.

Há alternativas, as expressões aritméticas podem ser escritas em notação pré-fixa (também designada notação Polaca<sup>1</sup>), ou em notação pós-fixa:

**em notação pré-fixa**  $+ \times yzx$

**em notação pós-fixa**  $xyz \times +$

Estas duas formas de escrever as expressões aritméticas são não ambíguas: à esquerda (respectivamente, à direita) de um dado operador estão sempre os seus dois operandos, ou no caso de um operador unário, o seu único operando.

Pretende-se então construir um programa que leia e calcule o valor de expressões aritméticas em notação Polaca inversa (pós-fixa). O procedimento de leitura e cálculo é muito fácil de programar através da utilização de uma pilha auxiliar.

### Projecto de Programação Orientada aos Objectos 2010/2011

1. Implemente, em C++ a classe Pilha de Inteiros:

Pilhas Inteiros = (Pilhas, {push, pop, top, vazia?})

construa uma implementação diferente daquela que foi apresentada na aula. Tenha o cuidado de considerar as situações de erro.

2. Construa um programa que receba (leia) uma expressão em notação polaca inversa (notação pós-fixa) e, com auxílio da pilha de inteiros, calcule e escreva o valor final da expressão.

Documente o seu programa. Tanto em termos de documentação interna, como de documentação externa na forma de um pequeno manual de utilização.

v.s.f.f.

---

<sup>1</sup>Devido a ter sido proposta pelo matemático Polaco Jan Łukasiewicz

A álgebra “Pilhas” pode ser caracterizada da seguinte forma:

### Elementos

$$\text{Pilhas} = \begin{cases} \text{pilhaVazia}, & \text{pilha vazia} \\ \text{elemento} : \text{Pilha}, & \text{pilhas não vazias} \end{cases}$$

com o símbolo “:” a significar a concatenação de elementos;

### Funções internas

$$\begin{array}{llll} \text{push} : & \text{Elementos} \times \text{Pilhas} & \longrightarrow & \text{Pilhas} \\ & (e, p) & \longmapsto & e : p \\ \\ \text{pop} : & \text{Pilhas} & \longrightarrow & \text{Pilhas} \cup \{\text{erro}\} \\ & p & \longmapsto & \begin{cases} p', & \text{para } p = e : p' \\ \text{erro}, & \text{para } p = \text{pilhaVazia} \end{cases} \\ \\ \text{top} : & \text{Pilhas} & \longrightarrow & \text{Elementos} \cup \{\text{erro}\} \\ & p & \longmapsto & \begin{cases} e, & \text{para } p = e : p' \\ \text{erro}, & \text{para } p = \text{pilhaVazia} \end{cases} \\ \\ \text{vazia?} : & \text{Pilhas} & \longrightarrow & \text{Bool} \\ & p & \longmapsto & \begin{cases} \mathcal{F}, & \text{para } p = e : p' \\ \mathcal{V}, & \text{para } p = \text{pilhaVazia} \end{cases} \end{array}$$