

# Paradigmas de Programação

## Ordem Lexicográfica

### "ordem do dicionário"!

Visando ilustrar o tópico, considerem a linguagem de programação em Java (já conhecida de vocês). A "ordem do dicionário" utilizada em Java é um pouco diferente de um dicionário normal. Java faz distinção entre letras maiúsculas e minúsculas (ou seja, trata-se de uma linguagem *case sensitive*), e classifica os caracteres colocando primeiro os números, depois os caracteres com letras maiúsculas, e então os caracteres com letras minúsculas.

Ou seja, 1 vem antes de B, que vem antes de a.

Vale ressaltar que o caractere de espaço em branco vem antes de todos os outros caracteres. Na comparação de *strings* em Java, temos que letras correspondentes são comparadas até que uma das *strings* termine, ou mesmo até que a primeira diferença seja encontrada. Se uma das *strings* terminar, a mais longa é considerada maior. Se for encontrada uma não correspondência de caracteres, os caracteres são comparados para determinar qual *string* será a maior na sequência do dicionário.

Esse processo se chama **comparação lexicográfica** ou **ordenação lexicográfica**!

Por exemplo, consideremos "car" e "cargo". As três primeiras letras são iguais, e chegamos ao final da primeira string. Portanto, "car" vem antes de "cargo" no ordenamento lexicográfico em Java.

Consideremos agora "cathode" e "cargo". As duas primeiras letras são iguais. Na terceira posição de caractere, t vem depois r. Portanto, a string "cathode" vem depois de "cargo" no ordenamento lexicográfico em Java.

Desejo a Todos Ótimos Estudos!

:)

Atenciosamente,

Profa. Milene.