

40. [Sebesta, 2000] Algumas linguagens, notavelmente o C e o Java, fazem distinção entre maiúsculas e minúsculas nos identificadores. Quais são os prós e os contras nessa decisão de projeto?

A principal vantagem de uma linguagem de programação ser *case-sensitive*, ou seja, fazer distinção entre letras maiúsculas e minúsculas, é o incremento na velocidade para compilar e/ou interpretar o código-fonte do usuário, uma vez que não exista a necessidade de se padronizar os identificadores e/ou palavras reservadas fornecidos pelo usuário. Por exemplo, caso o código-fonte contenha os nomes `soma`, `Soma` e `SOMA`, o compilador e/ou interpretador terá que padronizar o nome para `soma`, por exemplo, o que tornará o processo mais lento do que o necessário.

A segunda vantagem seria o aumento do conjunto de nomes disponíveis ao usuário, pois `soma`, `Soma` e `SOMA` seriam considerados nomes distintos pelo compilador e/ou interpretador.

Em relação a desvantagem, ela é mais subjetiva do que objetiva. Por exemplo, os três nomes seguintes são distintos em C: `soma`, `Soma` e `SOMA`. Para alguns, esse é um sério prejuízo à legibilidade, porque nomes muito semelhantes denotam, de fato, entidades diferentes. Nesse sentido, a distinção entre maiúsculas e minúsculas viola o princípio de projeto segundo o qual construções de linguagem de aparência semelhante devem ter o mesmo significado (Sebesta, 2000).

Evidentemente, nem todo mundo concorda que a distinção entre maiúsculas e minúsculas é ruim para os nomes. No C, este problema pode ser evitado por meio do uso exclusivo de minúsculas para nomes. Em Java, entretanto, não é possível evitá-lo porque muito dos nomes predefinidos incluem as duas formas. Por exemplo, o método Java para converter uma `string` em um valor inteiro é `parseInt`, e grafias como `ParseInt` e `parseint` não são reconhecidas. Esse é um problema de capacidade de escrita, não de legibilidade, porque a necessidade de lembrar grafias estranhas dificulta a escrita correta de programas. É um tipo de intolerância da parte do projetista da linguagem imposta pelo compilador (Sebesta, 2000).