

Aula Prática 11

Resumo:

- Aplicações de dicionários;
- Dicionários implementados como listas de pares chave-valor.

Exercício 11.1

Considere o problema de autenticar utilizadores que pretendam aceder a um determinado programa ou sistema informático. A autenticação baseia-se no nome e senha do utilizador. Desenvolva um programa **CheckPasswd** que leia os nomes e senhas contidos num ficheiro dado na linha de comando e em seguida simule o processo de autenticação de utilizadores. Use a classe `p2utils.KeyValueList` para guardar as associações entre nomes de utilizadores e senhas. O programa deve terminar quando for detectado “fim de ficheiro” (EOF), o que numa consola Unix se faz introduzindo **Ctrl+D** no início de uma linha.

```
Username: carlos
Password: minhasenha
Authentication successful

Username: ines.m
Password: ines#m
Authentication failed

Username: jose.antunes
Password: senutna
Authentication successful

Username: <Ctrl-d>
```

Exercício 11.2

O programa **CountWords** determina e apresenta a tabela de frequências de ocorrência das palavras contidas num ou mais ficheiros de texto cujos nomes são dados na linha de comando. O programa deve usar uma `KeyValueList` para resolver o problema. Cada palavra será uma chave de acesso e o valor associado será simplesmente o contador de ocorrências dessa palavra.

- Complete a função principal para atualizar o contador correspondente a cada palavra processada. Para tornar o processo insensível a diferenças entre maiúsculas e minúsculas, pode usar a função `toLowerCase()` da classe **String** para converter as palavras para minúsculas antes de as introduzir na lista.

- b. Complete o método `toString(left, sep, right)` da classe `KeyValueList` que deve devolver uma representação da lista como uma sequência de pares (*chave, valor*) separados entre si pela string `sep` e delimitada pelas strings `left` e `right`. Verifique o resultado das invocações no programa.
- c. Complete a função `mostFrequent` para descobrir a palavra mais frequente e indicar a sua frequência relativa.

Exercício 11.3

O ficheiro `numbers.txt` contém uma lista de números e respetivas traduções por extenso em língua inglesa. Fazendo uso de um dicionário, escreva um programa `TranslateNumbers` que traduza num texto, todas as ocorrências de números por extenso pelo respectivo valor numérico, mantendo as restantes palavras. Exemplo de utilização:

```
$ echo "A list of numbers: two hundred thousand five hundred twenty four" | java -ea TranslateNumbers
A list of numbers: 2 100 1000 5 100 20 4
```

Exercício 11.4

Utilizando o dicionário do exercício anterior, crie um programa que converta um número escrito por extenso em língua inglesa para o respectivo valor numérico. Por exemplo:

```
$ echo "two thousand thirty three" | java -ea NumberValue
two thousand thirty three -> 2033
```

```
$ echo "eight million two hundred thousand five hundred twenty four" | java -ea NumberValue
eight million two hundred thousand five hundred twenty four -> 8200524
```

Tenha em consideração as seguintes regras na construção do algoritmo:

- Os números são sempre descritos partindo das maiores ordens de grandeza para as mais pequenas (*million, thousand, ...*);
- Sempre que os números se sucedem por ordem crescente (*eight million* ou *two hundred thousand*), o valor vai sendo acumulado por multiplicações sucessivas ($8 * 1000000$ ou $2 * 100 * 1000$);
- Caso contrário, o valor acumulado é somado ao total.
- Não é preciso detetar números mal formados como: *one one million, eleven one* ou outros.

Exercício 11.5

O acesso a cada elemento numa `KeyValueList` tem complexidade linear no número de elementos. Assim, quanto maior o número de elementos, mais crítico se torna otimizar o acesso. Uma optimização possível consiste em manter a lista ordenada por chave. Crie uma classe `SortedKeyValueList`, com as seguintes melhorias em relação à classe `KeyValueList`:

- O método `set` deve garantir que a lista está sempre ordenada por chave.
- Modifique também o método `contains` para tirar partido da ordenação dos elementos. Assim, só precisa de percorrer a lista até encontrar a chave procurada (caso em que devolve `true`) ou até encontrar uma chave maior (caso em que devolve `false`).

Modifique uma cópia do programa `CountWords` para usar e testar a nova classe.