

Aula Prática 5

Resumo:

- Robustez e Excepções.

Exercício 5.1

Altere o programa 4.2 da aula anterior por forma a torná-lo totalmente robusto. Em particular, o programa deve garantir a sanidade dos argumentos e dos valores introduzidos pelo utilizador.

Exercício 5.2

Crie um programa que copie um ficheiro de texto. Os nomes dos dois ficheiros envolvidos devem ser dados como argumentos na linha de comandos.¹ Por exemplo, `java -ea p52 Texto1.txt Texto2.txt` deve criar um ficheiro `Texto2.txt` com um conteúdo igual ao do ficheiro `Texto1.txt`. Se `Texto2.txt` já existir, deve perguntar ao utilizador se deseja reescrever o seu conteúdo.

O programa deve ser robusto. Deve detetar falhas e apresentar mensagens de erro apropriadas. Várias condições têm de estar reunidas para que o programa funcione. O ficheiro original tem de existir, tem de ser um ficheiro normal e o utilizador tem de ter permissão para o ler. Se o ficheiro de destino existir, tem de ter permissão de escrita. Se não existir, então tem de ter permissão de escrita no directório pai. Mesmo assim, podem ocorrer falhas imprevisíveis, como erros no disco ou de comunicação, que devem ser reportadas.

Exercício 5.3

Crie um programa que apresente alguma informação sobre os ficheiros existentes num determinado directório. O nome do directório é passado como argumento do programa. Se for chamado sem argumentos, o programa deve considerar o directório actual (.). Para cada ficheiro, deve ser indicado se o mesmo corresponde a um directório (DIR) e se tem permissão de leitura (READ) ou de escrita (WRITE), como se vê no exemplo abaixo.

¹Pretende-se que seja uma versão simplificada do comando UNIX `cp`.

dir1	(DIR,READ,WRITE)
dir2	(DIR,READ)
dir3	(DIR)
fich1	(READ,WRITE)
ficheiro2	()

Exercício 5.4

Altere o programa 4.4 da aula anterior por forma a torná-lo totalmente robusto. Em particular, o programa deve garantir a sanidade dos argumentos e dos valores introduzidos pelo utilizador.

Exercício 5.5

Construa um programa que dado um ficheiro de texto de entrada escreva somente as palavras existentes numa determinada coluna em cada linha do texto (se não existir essa coluna deve escrever uma linha em branco). Considere que as palavras são separadas por espaços em branco ou por caracteres de tabulação (leia com o método `next()`).

Quer o nome do ficheiro quer o número da coluna devem ser dados como argumentos do programa.

Por exemplo, no caso de termos o seguinte texto de entrada:

```
1 2 3 4 5 6
    boa    tarde
um dois tres quatro cinco seis
```

A saída do programa, se for escolhida a coluna número 1, deverá ser:

```
1
boa
um
```

No caso da coluna número 4, a saída passará a ser:

```
4

quatro
```