

# Aula Prática 1

## Resumo:

- Revisões.
- Utilização da biblioteca *standard* do Java para entrada/saída.
- Programas com argumentos na linha de comando.

## Exercício 1.1

Crie uma calculadora simples que leia (do dispositivo de entrada) operações matemáticas como

12.3 + 7.2

e escreva o resultado respectivo (19.5 neste exemplo).

As operações serão sempre do género <número> <operador> <número>, com as três partes separadas por espaços ou em linhas diferentes. Implemente as quatro operações básicas usando os operadores +, -, \* e /. Note que o operador é uma palavra (string) que contém apenas um símbolo. Se for introduzido um operador inválido, deve escrever uma mensagem apropriada para o dispositivo de saída de erros (`System.err`).

## Exercício 1.2

Escreva um programa que determine a nota final  $NF$  de um aluno de Programação 2 na época normal. O programa deverá pedir as notas dos vários momentos de avaliação e calcular e apresentar as notas das componentes e a nota final. A fórmula de cálculo está no guião da unidade curricular.

## Exercício 1.3

No programa fornecido, complete a função para verificar se um número é primo. Um algoritmo simples consiste em testar se o número é divisível por 2, por 3, etc.<sup>1</sup> Qual o maior divisor que é preciso testar?

## Exercício 1.4

Na terra do Alberto Alexandre (localmente conhecido por Aubeto Auexande), o dialecto local é semelhante ao português com duas excepções:

---

<sup>1</sup>Existem algoritmos muito mais eficientes de testar primalidade, mas envolvem conceitos mais avançados.

- Não dizem os Rs
- Trocam os Ls por Us

Implemente um tradutor de português para o dialecto do Alberto. Por exemplo “lar doce lar” deve ser traduzido para “ua doce ua”. A tradução deve ser feita linha a linha, até que surja uma linha vazia.

### Exercício 1.5

Escreva um programa que leia uma lista de números e imprima a sua soma e a sua média. O fim da lista é indicado pela leitura do número zero, que não deve ser considerado parte da lista. (Note que se a lista for vazia, a soma será zero, mas a média não pode ser calculada.)

### Exercício 1.6

Escreva um programa que implemente o jogo “Adivinha o número!”.

Neste jogo, o programa escolhe um número aleatório no intervalo  $[0; 100]^2$  e depois o utilizador tenta descobrir o número escolhido. Após cada tentativa, o programa deve indicar se o número escolhido é maior, menor ou igual à tentativa feita. O jogo termina quando o utilizador acertar no número correcto. A pontuação do jogador é o número de tentativas.

### Exercício 1.7

Crie um programa que copie um ficheiro de texto para outro<sup>3</sup>. Os nomes dos dois ficheiros envolvidos devem ser dados como argumentos na linha de comandos<sup>4</sup>. Assim a execução do programa com os argumentos `Texto1.txt Texto2.txt` deve criar um ficheiro `Texto2.txt` com um conteúdo igual ao do ficheiro `Texto1.txt`.

**Nota:** Tente fazer com que o programa tenha alguma robustez, não só detectando a existência do ficheiro original (apresentando uma mensagem de erro quando este não existe), como também a possível existência do ficheiro destino (neste caso fará sentido perguntar ao utilizador se de facto deseja destruir esse ficheiro). Deve também verificar se sobre os ficheiros se podem realizar as operações de leitura e escrita, e se algum deles é um directório (caso em que deve ser apresentada uma mensagem de erro).

---

<sup>2</sup>`(int)(Math.random()*(100+1))`

<sup>3</sup>Pretende-se que seja uma versão simplificada do comando UNIX `cp`.

<sup>4</sup>No programa Java esses valores farão parte do *array* de *strings* que é passado como argumento da função `main`.