

Utilização de tipos definidos por classes

1. Escreva um programa `Leituras` que pede ao utilizador um valor inteiro, um valor decimal e um nome. A leitura deve ser feita do teclado recorrendo à classe `java.util.Scanner`. O output deve ser como se mostra abaixo (onde o input está apresentado a negrito).

```
$java Leituras
Insere o teu nome: Manuel Joao
Insere um número inteiro: 12
Insere um número decimal: 33.44
Caro Manuel Joao, somar 12 com 33.44 da 45.44
```

Note que na leitura do decimal, o símbolo que deverá ser usado como separador (, ou .) depende da configuração do sistema operativo.

2. Recorrendo ainda à classe `java.util.Scanner` para fazer leituras:
 - a. Programe um método `static int lerValorNoIntervalo (int n, int m, Scanner canal)` que pede ao utilizador um valor inteiro no intervalo $[n,m]$, lê-o através o canal dado e devolve-o. No caso de o utilizador inserir um valor fora do intervalo, o método deve dar uma mensagem de erro para o ecrã e voltar a pedir o valor.
 - b. Escreva um programa que pede ao utilizador as notas de um aluno de IP nos diferentes componentes da avaliação e o informa se aprovou ou não. No caso de aprovação, informa-o da sua nota final, arredondada a um valor inteiro entre 10 e 20. Para fazer o arredondamento deve utilizar o método `round` fornecido pela classe `java.lang.Math`.
3. Recorrendo às classes `java.lang.String` e `java.util.Scanner`:
 - a. Escreva uma função que, dadas uma *string* e uma letra, calcula e devolva o número total de vezes que a letra aparece na *string*, ignorando se aparece em maiúscula ou minúscula.
 - b. Escreva um programa que lê um texto do teclado e imprime no ecrã o número de ocorrências da letra *A* na frase dada (deve utilizar a função definida anteriormente).
4. Recorrendo às classes `java.lang.String` e `java.lang.StringBuilder`, escreva um programa `ObfuscarTextos` que lê uma linha de texto do teclado e imprime o texto mas sem espaços e com grupos de 5 letras alternadamente maiúsculas e minúsculas, como se mostra no exemplo abaixo:

```
$java ObfuscarTextos
Insere linha de texto: As armas e os barões assinalados
Linha de texto obfuscada: ASARMaseosBARÕEsassiNALADos
```

5. Recorrendo à classe `java.util.Random`, escreva um programa `GerarMatriculas` que gera aleatoriamente uma matrícula de automóvel portuguesa. Considere apenas as matrículas que têm letras no meio. Sugestão: use a *string* "ABCDEFGHILMNOPQRSTUVWXYZ" e o método `charAt`.

```
$java GerarMatriculas
23-AC-43
```

6. Recorrendo à classe `java.util.Random`, escreva um programa `PasseadorAleatorio` que simule o caminho percorrido por um personagem que só sabe dar passos em frente mas com a possibilidade de se desviar para a direita ou esquerda em cada passo. A tomada de cada decisão deve ser aleatória e o número de passos executados pelo personagem assim como a posição inicial devem ser dadas como argumentos de execução do programa.

```
$java PasseadorAleatorio 10 25
```

```
-----
                        *
                        *
                        *
                        *
                        *
                        *
                        *
                        *
                        *
                        *
-----
```

7. Considere que tem disponível uma classe `Dado` cujos objetos representam dados de 6 faces. Em cada momento, o dado encontra-se com uma das suas faces para cima. A classe disponibiliza:

- um construtor sem argumentos, que cria o dado e o coloca arbitrariamente uma das suas faces viradas para cima;
- o método `void lancar()` que lança o dado ao ar;
- o método `int obterValor()` que retorna o valor da face que se encontra atualmente virada para cima (um valor entre 1 e 6).

- a. Assumindo que a classe `Dado` está implementada corretamente, o que vai acontecer quando é executado o seguinte pedaço de código?

```
Dado dado1 = new Dado();
int v1 = dado1.obterValor();
Dado dado2 = dado1;
dado2.lancar();
int v2 = dado1.obterValor();
int v3 = dado2.obterValor();
System.out.println(v1==v2);
System.out.println(v2==v3);
```

- b. Escreva um programa que implementa um jogo de sorte muito básico que consiste no lançamento de dois dados. O jogador ganha se o resultado dos dados somar 7 e neste caso recebe o dobro do que apostou na jogada. O jogador perde no caso contrário, perdendo tudo o que apostou na jogada. O programa deve executar as seguintes tarefas:

- ler do teclado o nome do jogador;
- ler do teclado o valor que ele pagou inicialmente para jogar;
- enquanto o jogador quiser continuar a jogar,
 - i. lê o valor da aposta que a) não pode ser superior ao valor disponível e b) indica que o jogador não quer continuar a jogar se for zero;
 - ii. se o valor da aposta não for zero
 - a. manda lançar os dados;
 - b. informa de resultado e atualiza valor disponível.