



Introdução à Programação

Licenciatura em Engenharia Informática

Série de Exercícios 4

2021/2022

Utilização de tipos definidos por classes

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

```
$java MyFirstReader
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 dá 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. Escreva um procedimento `int readIntegerInInterval(int n, int m, Scanner canal)` que pede ao utilizador um valor inteiro no intervalo $[n,m]$, lê esse valor através do canal dado e devolve-o. Se o utilizador inserir um valor inteiro fora do intervalo $[n,m]$, deve ser enviada uma mensagem de erro para o ecrã e deve ser pedido e lido um novo 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 devolve 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 `TextLineObsfuscation` 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 TextLineObsfuscation
Insere linha de texto: As armas e os barões assinalados
Linha de texto obfuscada: ASARMaseosBARÕEsassiNALADos
```

5. Recorrendo à classe `java.util.Random`, escreva uma classe `PtCarPlateGenerator` 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 PtCarPlateGenerator
23-AC-43
```

6. Recorrendo à classe `java.util.Random`, escreva um programa `RandomWalker` 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 RandomWalker 15 10
```

Number of nodes in the largest component	Frequency
0	0
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9
10	10
11	9
12	8
13	7
14	6
15	5
16	4
17	3
18	2
19	1
20	0

7. Considere que tem disponível uma classe Dice 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 com uma face arbitrária virada para cima;
- o método `void roll()` que lança o dado ao ar;
- o método `int getValue()` que retorna o valor da face que se encontra atualmente virada para cima (um valor entre 1 e 6).



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

```
Dice dice1 = new Dice();
int v11 = dice1.getValue();
Dice dice2 = dice1;
dice1.roll();
int v12 = dice1.getValue();
int v2 = dice2.getValue();
System.out.println(v11 == v12);
System.out.println(v12 == v2);
```

- 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) se for zero indica que o jogador não quer continuar a jogar;
 - ii. se o valor da aposta não for zero: a) manda lançar os dados; b) informa de resultado e atualiza valor disponível.