

PRÁTICA LABORATORIAL 06

Objetivos:

- Funções/Métodos/Procedimentos
- Consultar `.length()` – Para determinar o tamanho de um vetor
- Consultar `equalsIgnoreCase()` – Para avaliar se uma String é igual a outra ignorando maiúsculas e minúsculas.

EXERCÍCIOS

Parte 1

1. Implemente uma função void **fazerBarulho()** que receba um animal (String) como argumento. De seguida, de acordo com o animal deve fazer o barulho característico (Ex.: Cão – “Au au au” | Gato – “Miau” | Peixe – “Glub Glub” | Vaca – “Muuuu” | Porco – “Oinc oinc”).

Exemplo de execução:

```
> Introduza um animal: Gato
> Miau
```

2. Escreva uma função que leia um valor inteiro positivo (deve solicitar números até que seja inteiro e positivo) e crie uma função que imprima numa linha um número de asteriscos igual ao valor inserido pelo utilizador.

Exemplos de execução:

```
> Introduza um número: 7
> *****
```

```
> Introduza um número: -5
> Introduza um número: 0
> Introduza um número: 4
> ****
```

Parte 2

3. O exercício deve ser resolvido na mesma classe (por exemplo: Ex_03), deve acrescentar as funções conforme solicitado. No início não conseguirá testar se as funções funcionam ou não (uma vez que não é pedido para desenvolver nada no main), considere fazer o exercício 4 em concorrente para conseguir perceber se está tudo como esperado.

a) Implemente uma função que determina se um número (passado por argumento) é par ou ímpar, a função deve retornar true se par ou false se ímpar.

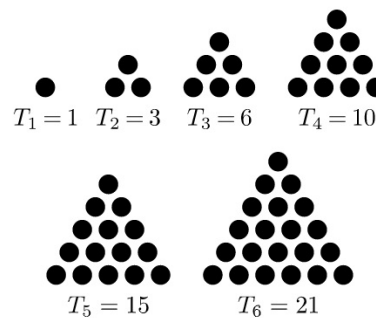
Exemplo de Código:

```
static boolean par(int num){  
    ...  
}  
  
public static void main(String[] args) {  
}
```

- b) Implemente uma função que determina se um número (passado por argumento) é positivo ou negativo, a função deve retornar true se zero ou positivo, ou false se negativo.

```
static boolean par(int num){  
    ...  
}  
  
static boolean positivo(int num){  
    ...  
}  
  
public static void main(String[] args) {  
}
```

- c) Implemente uma função que determina se um número (passado por argumento) é ou não primo, a função deve retornar true se primo ou false se não primo.
- d) Implemente uma função que determina se um número (passado por argumento) é perfeito, a função deve retornar true se perfeito ou false se não perfeito (Os números perfeitos são iguais à soma dos seus divisores: 6 pode ser dividido por 1, 2 e 3 e, quando soma esses números, o resultado é 6).
- e) Implemente uma função que determina se um número (passado por argumento) é triangular, a função deve retornar true se triangular ou false se não triangular (Um número triangular é um número que pode ser representado pela soma de números inteiros consecutivos. Exemplo: $6 = 1+2+3$ ou $15=1+2+3+4+5$).



- 4. Implemente o programa “**Análise de um Número**” que pergunte um número ao utilizador e, de seguida, apresente um menu de opções ao utilizador, que permitem analisar o número inserido. Esse menu de opções deve ter o seguinte aspeto:
 - 1. Par ou Ímpar
 - 2. Positivo ou Negativo
 - 3. Primo ou Não Primo
 - 4. Perfeito ou Não Perfeito
 - 5. Triangular ou Não Triangular
 - 6. Trocar de Número

Exemplo de Código:

```
import static Ficha06.Ex_03.*;

public class Ex_04 {
    public static void main(String[] args) {

        int numero, opcao;

        System.out.print("Introduza um numero: ");
        numero = input.nextInt();

        //Menu de opções
        ...
        switch(opcao){

            case 1: // Par ou Impar

                if(par(num)){
                    System.out.print("Par");

                }else{
                    System.out.print("Impar");
                }
                break;

            case 2: // Positivo ou Negativo

                if(positivo(num)){
                    System.out.print("Positivo");

                }else{
                    System.out.print("Negativo");
                }
                break;

            ...

            case 6:
                System.out.print("Introduza um novo numero: ");
                numero = input.nextInt();
                break;

        }
    }
}
```

Parte 3

5. O exercício deve ser resolvido na mesma classe (por exemplo: Ex_05), deve acrescentar as funções conforme solicitado. No início não conseguirá testar se as funções funcionam ou não (uma vez que não é pedido para desenvolver nada no main), considere fazer o exercício 6 em concorrente para conseguir perceber se está tudo como esperado.

- a) Escreva uma função que recebe um array de inteiros como parâmetro e retorna o maior valor do array.

Exemplo de Código:

```
static int maior(int[ ] vetor){  
    ...  
}  
  
public static void main(String[] args) {  
  
}
```

- b) Escreva uma função que recebe um array de inteiros como parâmetro e retorna o menor valor do array.
c) Escreva uma função que recebe um array de inteiros como parâmetro e retorna se o array está crescente.

```
static int maior(int[ ] vetor){  
    ...  
}  
  
static int menor(int[ ] vetor){  
    ...  
}  
  
static boolean crescente(int[ ] vetor){  
    ...  
}  
  
public static void main(String[] args) {  
  
}
```

6. Implemente o programa “**Análise de um Vetor**” que pergunte um o tamanho do vetor que o utilizador deseja inserir. De seguida, permita ao utilizador preencher o vetor. Posteriormente apresente um menu de opções ao utilizador, que permitem analisar o vetor inserido. Esse menu de opções deve ter o seguinte aspeto:
7. Maior Elemento
 8. Menor Elemento
 9. Crescente ou Não Crescente

Parte 4

7. Escreva um programa para desenhar um quadrado no ecrã. Esse quadrado deverá ser desenhado por uma função para a qual são passados três argumentos: caracter a utilizar, número de linhas e número de colunas. Segue-se um exemplo do algoritmo a ser executado, ilustrando o pretendido:

```
> Introduza um caracter: z
> Introduza número de linhas: 4
> Introduza número de colunas: 6

ZZZZZZ
Z      Z
Z      Z
ZZZZZZ
```

8. Implemente funções que, dadas 2 matrizes de inteiros com a mesma dimensão, apresente (numa função) em forma de matriz o resultado da soma das 2 matrizes e (noutra função) o somatório dos seus elementos.

Exemplo de Código:

```
static void somarMatrizes(int[ ][ ] matriz1, int[ ][ ] matriz2){
    ...
}

static int somatorioMatrizes(int[ ][ ] matriz1, int[ ][ ] matriz2){
    ...
}

public static void main(String[] args) {

    int[ ][ ] matriz1 = new int[3][3];
    int[ ][ ] matriz2 = new int[3][3];
    int somatorio;

    // Inicializar matriz

    somarMatrizes(matriz1, matriz2);
    somatorio = somatorioMatrizes(matriz1, matriz2);

}
```

Bom trabalho! 😊