

## PRÁTICA LABORATORIAL 06

### Objetivos:

- Funções/Métodos/Procedimentos
- Consultar `.length()`
- Consultar `equalsIgnoreCase()`

## 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. Implemente uma função que determine 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) {  
  
    int numero;  
  
    System.out.print("Introduza um numero: ");  
    numero = input.nextInt();  
  
    if(par(numero)){  
        System.out.print("Par");  
    }else{  
        System.out.print("Impar");  
    }  
}
```

3. Implemente uma função que determine se um número (passado por argumento) é positivo ou negativo, a função deve retornar true se zero ou positivo, ou false se negativo.
4. Implemente uma função que determine se um número (passado por argumento) é ou não primo, a função deve retornar true se primo ou false se não primo.
5. (Opcional) Implemente uma função que determine 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).
6. Implemente um programa que contenha uma função com menu de opções que permitam invocar as funções desenvolvidas nos exercícios 2 a 4 (ou 5), devendo o número ser escolhido antes das opções do menu.

Exemplo de Código:

```
static boolean par(int num){
    ...
}

static boolean positivo(int num){
    ...
}

static boolean primo(int num){
    ...
}

static boolean perfeito(int num){
    ...
}

static void menu(int num){
    ...
    // De acordo com a escolha do utilizador deve invocar a função correta.
}

public static void main(String[] args) {

    int numero;

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

    menu(numero);
}
```

7. Escreva uma função que leia um conjunto de números e os armazene num array. O array e a quantidade de números a ler são passados como argumento.

Escreva uma função que imprima o conteúdo de um vetor passado como argumento, assim como o seu tamanho. Teste a função no âmbito de um programa que pergunte ao utilizador quantos números o vetor deve armazenar, leia tais números e imprima o conteúdo do array. Exemplo de execução:

```
- Quantos números quer introduzir: 4
- Introduza número [0]: 1
- Introduza número [1]: 4
- Introduza número [2]: 7
- Introduza número [3]: 9
- [1,4,7,9]
```

Exemplo de Código:

```
static int[ ] preencherVetor(int[ ] vetor, int tamanho){
    ...
    // Ciclo para preencher vetor com valores do utilizador.
}

static void imprimirVetor(int[ ] vetor, int tamanho){
    ...
    // Ciclo para imprimir vetor.
}

public static void main(String[] args) {

    int tamanho;

    System.out.print("Introduza o tamanho do array: ");
    tamanho = input.nextInt();

    int[ ] vetor = new int[tamanho];

    vetor = preencherVetor(vetor, tamanho);
    imprimirVetor(vetor, tamanho);

}
```

8. 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) {  
  
    int[ ] vetor = new int[10];  
    int maiorElemento;  
  
    // Ler números do utilizador  
  
    maiorElemento = maior(vetor);  
}
```

9. Escreva uma função que recebe um array de inteiros como parâmetro e retorna o menor valor do array.
10. Escreva uma função que recebe um array de inteiros como parâmetro e retorna se o array está crescente.
11. Implemente um programa que contenha:
- Uma função que pergunte o tamanho de um array a ser preenchido e o permita preencher.
  - Uma função com menu de opções que permitam invocar as funções desenvolvidas nos exercícios 8 a 10.

12. 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.
13. 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 o número de linhas: 4

Introduza o número de colunas: 6

zzzzzz

z    z

z    z

zzzzzz

14. 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! ☺**