**Pratica 06**

/\*

\*

\* 1) Em linguagem de programação Java, escreva um algoritmo que leia 10 números inteiros

\* a partir do teclado e os armazene em um vetor. O algoritmo deve imprimir o vetor e

\* apresentar a soma dos números pares.

\*

\* \*/

**import** javax.swing.JOptionPane;

**public** **class** Main {

**public** **static** **void** main(String[] args) {

**int** index, soma = 0;

**int**[] vetor = **new** **int**[10];

String numInt;

// Inicia a leitura do vetor.

**for** (index = 0; index < 10; index++) {

numInt = JOptionPane.*showInputDialog*("Informe um número inteiro:");

vetor[index] = Integer.*parseInt*(numInt);

}

// Mostra na tela o vetor de inteiros.

numInt = "Vetor informado:\n";

**for** (index = 0; index < vetor.length; index++)

numInt += String.*format*(" vetor[%d]=%4d\n", index, vetor[index]);

JOptionPane.*showMessageDialog*(**null**, numInt);

// Somar valores pares do vetor:

**for** (**int** numPar : vetor)

**if** (numPar % 2 == 0)

soma += numPar;

JOptionPane.*showMessageDialog*(**null**, "Soma dos números pares do vetor: " + soma);

}

}

/\*

\* 2) Em linguagem de programação Java, escreva um algoritmo que leia 10 números reais a

\* partir do teclado e os armazene em um vetor. O algoritmo deve imprimir o vetor e as

\* posições do vetor que armazenam números negativos.

\* \*/

**import** javax.swing.JOptionPane;

**public** **class** Main {

**public** **static** **void** main(String[] args) {

**int** index;

**int**[] vetor = **new** **int**[10];

String numInt;

// Inicia a leitura do vetor.

**for** (index = 0; index < 10; index++) {

numInt = JOptionPane.*showInputDialog*("Informe um número inteiro:");

vetor[index] = Integer.*parseInt*(numInt);

}

// Mostra na tela o vetor de inteiros.

numInt = "Vetor informado:\n";

**for** (index = 0; index < vetor.length; index++)

numInt += String.*format*(" vetor[%d]=%4d\n", index, vetor[index]);

JOptionPane.*showMessageDialog*(**null**, numInt);

// Mostra as posições com números negativos.

numInt = "Posições contendo os negativos:\n";

**for** (index = 0; index < vetor.length; index++)

**if** (vetor[index] < 0)

numInt += String.*format*("%d: vetor[%d] = %d\n", index, index, vetor[index]);

JOptionPane.*showMessageDialog*(**null**, numInt);

}

}

/\*

\*3) Em linguagem de programação Java, escreva um algoritmo que leia 10 números reais a

\* partir do teclado e os armazene em um vetor. O algoritmo deve imprimir o vetor, o

\* menor e o maior número informado.

\*/

**import** javax.swing.JOptionPane;

**public** **class** Main {

**public** **static** **void** main(String[] args) {

**int** index, menor, maior;

**int**[] vetor = **new** **int**[10];

String numInt;

// Inicia a leitura do vetor.

**for** (index = 0; index < 10; index++) {

numInt = JOptionPane.*showInputDialog*("Informe um número inteiro:");

vetor[index] = Integer.*parseInt*(numInt);

}

// Mostra na tela o vetor de inteiros.

numInt = "Vetor informado:\n";

**for** (index = 0; index < vetor.length; index++)

numInt += String.*format*(" vetor[%d]=%4d\n", index, vetor[index]);

JOptionPane.*showMessageDialog*(**null**, numInt);

// Mostra o menor e o maior valor informado.

menor = vetor[0];

maior = vetor[0];

**for** (index = 1; index < vetor.length; index++) {

**if** (vetor[index] < menor) {

menor = vetor[index];

} **else** {

**for** (index = 1; index < vetor.length; index++) {

**if** (vetor[index] > maior) {

maior = vetor[index];

}

}

}

}

JOptionPane.*showMessageDialog*(**null**, "Menor valor informado: " + menor);

JOptionPane.*showMessageDialog*(**null**, "Maior valor informado: " + maior);

}

}

**Pratica 06**

**package** atividade071;

/\*1) Em linguagem de programação Java, escreva um algoritmo que leia os elementos de

\* uma matriz 5 x 5 a partir do teclado. O algoritmo deve:

\* A) Imprimir a matriz informada.:

\*

\* B) Imprimir os elementos armazenados nas posições em que o índice da linha é

\* par e o índice da coluna é ímpar. Exemplo:

\*

\* c) Imprimir a matriz transposta a matriz informada. OBS: na matriz transposta,

\* as linhas da matriz tornam-se colunas e as colunas, linhas.

\*

\* D) Trocar os elementos da diagonal principal com os elementos da diagonal

\* secundária. Imprimir a matriz resultante.

\*/

**import** javax.swing.JOptionPane;

**public** **class** Main {

**public** **static** **void** main(String[] args) {

**int** linha, coluna;

**int**[][] matriz = **new** **int**[5][5];

String finalString;

// Recebe a matriz de inteiros

**for** (linha = 0; linha < 5; linha++)

**for** (coluna = 0; coluna < 5; coluna++) {

finalString = JOptionPane.*showInputDialog*("Informe um nº Inteiro: ");

matriz[linha][coluna] = Integer.*parseInt*(finalString);

}

*imprimir*(matriz);

// Mostra os elementos em posições de linhas pares e colunas impares.

finalString = "Elementos nas posições de linhas pares e colunas ímpares:\n";

**for** (linha = 0; linha < 5; linha += 2) {

**for** (coluna = 1; coluna < 5; coluna += 2)

finalString += String.*format*("%4d", matriz[linha][coluna]);

finalString += "\n";

}

JOptionPane.*showMessageDialog*(**null**, finalString);

// Mostra a matriz transposta em tela.

finalString = "Matriz Transposta informada:\n";

**for** (linha = 0; linha < 5; linha++) {

**for** (coluna = 0; coluna < 5; coluna++)

finalString += String.*format*("%4d", matriz[coluna][linha]);

finalString += "\n";

}

JOptionPane.*showMessageDialog*(**null**, finalString);

// Troca os valores da diagonal principal com diagonal secundaria.

**for** (linha = 0; linha < 5; linha++) {

**int** pos = (matriz[linha].length - 1) - linha;

**int** temp = matriz[linha][pos];

matriz[linha][pos] = matriz[linha][linha];

matriz[linha][linha] = temp;

}

*imprimir*(matriz);

}

**public** **static** **void** imprimir(**int** mat[][]) {

// Mostra a matriz em tela

**int** linha, coluna;

String finalString;

finalString = "Matriz informada:\n";

**for** (linha = 0; linha < 5; linha++) {

**for** (coluna = 0; coluna < 5; coluna++)

finalString += String.*format*("%4d", mat[linha][coluna]);

finalString += "\n";

}

JOptionPane.*showMessageDialog*(**null**, finalString);

}

}

**Pratica 06**