**A - Ler 10 elementos de uma matriz tipo vetor e apresentá-los.**

**package ADO01;**

import javax.swing.JOptionPane;

public class ADO\_A {

public static int[] vetor;

public static int[] CriaVetor(int[] vetor) {

for (int i = 0; i < vetor.length; i++) {

vetor[i] = Integer.parseInt(JOptionPane.showInputDialog("Digite o " + String.format("%dº ", (i + 1)) + "numero do vetor: "));

}

return vetor;

}

public static void ExibeVetor(int[] vetor) {

String saida = "";

for (int i = 0; i < vetor.length; i++) {

saida += vetor[i];

saida += "\n";

}

JOptionPane.showMessageDialog(null, saida);

}

public static void main(String[] args) {

vetor = new int[10];

vetor = CriaVetor(vetor);

ExibeVetor(vetor);

}

}

Tela de computador com texto preto sobre fundo branco

Descrição gerada automaticamente

**B - Ler 8 elementos em uma matriz A tipo vetor. Construir uma matriz B de mesma dimensão com os elementos da matriz A multiplicados por 3. O elemento B[i] deverá ser implicado pelo elemento A[i]\*3, o elemento B[2] implicado pelo elemento A[2]\*3 e assim por diante, até 8. Apresentar o vetor B.**

/\*

\* Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license

\* Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template

\*/

package ADO01;

import javax.swing.JOptionPane;

/\*\*

\*

\* @author nicolas.vsilva2

\*/

public class ADO\_B {

public static int[] vetorA;

public static int[] vetorB;

public static int mult = 3;

public static int[] CriaVetorA(int[] vetorA) {

for (int i = 0; i < vetorA.length; i++) {

vetorA[i] = Integer.parseInt(JOptionPane.showInputDialog("Digite o " + String.format("%dº ", (i + 1)) + "numero do vetor: "));

}

return vetorA;

}

public static int[] CriaVetorB(int[] vetorA, int[] vetorB, int mult) {

for (int i = 0; i < vetorB.length; i++) {

vetorB[i] = vetorA[i] \* mult;

}

return vetorB;

}

public static void ExibeVetores(int[] vetorA, int[] vetorB) {

String saida1 = "";

for (int i = 0; i < vetorA.length; i++) {

saida1 += vetorA[i];

saida1 += " ";

saida1 += vetorB[i];

saida1 += "\n";

}

JOptionPane.showMessageDialog(null, saida1);

}

public static void main(String[] args) {

vetorA = new int[8];

vetorB = new int[8];

vetorA = CriaVetorA(vetorA);

vetorB = CriaVetorB(vetorA, vetorB, mult);

ExibeVetores(vetorA, vetorB);

}

}

Tela de computador com texto preto sobre fundo branco

Descrição gerada automaticamente

**C - Ler duas matrizes A e B do tipo vetor com 20 elementos. Construir uma matriz C, onde cada elemento de C é a subtração do elemento correspondente de A com B. Apresentar a matriz C.**

/\*

\* Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license

\* Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template

\*/

package ADO01;

import javax.swing.JOptionPane;

/\*\*

\*

\* @author nicolas.vsilva2

\*/

public class ADO\_C {

public static int[] vetorA;

public static int[] vetorB;

public static int[] vetorC;

public static int[] CriaVetorA(int[] vetorA) {

for (int i = 0; i < vetorA.length; i++) {

vetorA[i] = Integer.parseInt(JOptionPane.showInputDialog("Digite o " + String.format("%dº ", (i + 1)) + "numero do vetor: "));

}

return vetorA;

}

public static int[] CriaVetorB(int[] vetorB) {

for (int i = 0; i < vetorB.length; i++) {

vetorB[i] = Integer.parseInt(JOptionPane.showInputDialog("Digite o " + String.format("%dº ", (i + 1)) + "numero do vetor: "));

}

return vetorB;

}

public static int[] CriaVetorC(int[] vetorA, int[] vetorB, int[] vetorC) {

for (int i = 0; i < vetorC.length; i++) {

vetorC[i] = vetorA[i] - vetorB[i];

}

return vetorC;

}

public static void ExibeVetores(int[] vetorA, int[] vetorB, int[] vetorC) {

String saida1 = "";

for (int i = 0; i < vetorA.length; i++) {

saida1 += vetorA[i];

saida1 += " ";

saida1 += vetorB[i];

saida1 += " ";

saida1 += vetorC[i];

saida1 += "\n";

}

JOptionPane.showMessageDialog(null, saida1);

}

public static void main(String[] args) {

vetorA = new int[20];

vetorB = new int[20];

vetorC = new int[20];

CriaVetorA(vetorA);

CriaVetorB(vetorB);

CriaVetorC(vetorA, vetorB, vetorC);

ExibeVetores(vetorA, vetorB, vetorC);

}

}

Interface gráfica do usuário, Aplicativo

Descrição gerada automaticamente

**D – Ler 15 elementos de uma matriz tipo vetor. Construir uma matriz B de mesmo tipo, observando a seguintes lei de formação: “Todo elemento de B deverá ser o quadrado do elemento de A correspondente”. Apresentar as matrizes A e B.**

package ADO01;

import javax.swing.JOptionPane;

public class ADO\_D {

public static int[] vetorA;

public static int[] vetorB;

public static int[] CriaVetorA(int[] vetorA) {

for (int i = 0; i < vetorA.length; i++) {

vetorA[i] = Integer.parseInt(JOptionPane.showInputDialog("Digite o " + String.format("%dº ", (i + 1)) + "numero do vetor: "));

}

return vetorA;

}

public static int[] CriaVetorB(int[] vetorA, int[] vetorB) {

for (int i = 0; i < vetorB.length; i++) {

vetorB[i] = (int) Math.pow(vetorA[i], 2);

}

return vetorB;

}

public static void ExibeVetores(int[] vetorA, int[] vetorB) {

String saida1 = "";

for (int i = 0; i < vetorA.length; i++) {

saida1 += vetorA[i];

saida1 += " ";

saida1 += vetorB[i];

saida1 += "\n";

}

JOptionPane.showMessageDialog(null, saida1);

}

public static void main(String[] args) {

vetorA = new int[15];

vetorB = new int[15];

CriaVetorA(vetorA);

CriaVetorB(vetorA, vetorB);

ExibeVetores(vetorA, vetorB);

}

}

Tela de computador com texto preto sobre fundo branco

Descrição gerada automaticamente

**E – Ler duas matrizes A e B do tipo vetor com 15 elementos cada. Construir uma matriz C, sendo esta a junção das duas outras matrizes. Desta forma, C deverá ter o dobro de elementos, ou seja, 30. Apresentar a matriz C.**

package ADO01;

import javax.swing.JOptionPane;

public class ADO\_E {

public static int[] vetorA;

public static int[] vetorB;

public static int[] vetorC;

public static int[] CriaVetorA(int[] vetorA) {

for (int i = 0; i < vetorA.length; i++) {

vetorA[i] = Integer.parseInt(JOptionPane.showInputDialog("Digite o " + String.format("%dº ", (i + 1)) + "numero do vetor: "));

}

return vetorA;

}

public static int[] CriaVetorB(int[] vetorB) {

for (int i = 0; i < vetorB.length; i++) {

vetorB[i] = Integer.parseInt(JOptionPane.showInputDialog("Digite o " + String.format("%dº ", (i + 1)) + "numero do vetor: "));

}

return vetorB;

}

public static int[] CriaVetorC(int[] vetorA, int[] vetorB, int[] vetorC) {

for (int i = 0; i < vetorC.length; i++) {

if (i < vetorA.length) {

vetorC[i] = vetorA[i];

} else {

vetorC[i] = vetorB[i-vetorA.length];

}

}

return vetorC;

}

public static void ExibeVetores(int[] vetorA, int[] vetorB, int[] vetorC) {

String saida1 = "";

for (int i = 0; i < vetorC.length; i++) {

saida1 += vetorC[i];

saida1 += "\n";

}

JOptionPane.showMessageDialog(null, saida1);

}

public static void main(String[] args) {

vetorA = new int[15];

vetorB = new int[15];

vetorC = new int[30];

vetorA = CriaVetorA(vetorA);

vetorB = CriaVetorB(vetorB);

vetorC = CriaVetorC(vetorA, vetorB, vetorC);

ExibeVetores(vetorA, vetorB, vetorC);

}

}

Interface gráfica do usuário, Aplicativo, PowerPoint

Descrição gerada automaticamente