

ARREGLOS EN PARALELO

(VECTORES)

**VECTORES EN PARALELO**

 **Definición de Vectores en Paralelo:**

Los vectores en paralelos se definen como un mecanismo que asocia la información de una entidad determinada bajo un mismo índice para más de un vector.

 **Características de un vector en paralelo:**

1. Utilizan un mismo índice para agrupar la información de una entidad determinada.

2. El ordenamiento de la información se refleja con base a uno o más atributos de la entidad.

3. Cada atributo de la entidad es un vector.

4. Disponen de un mismo tamaño en cada vector asociado a los atributos de la entidad.

Par entender mejor el concepto de vectores en paralelo, analicemos la siguiente situación: imaginemos que se requiere almacenar la información asociada a los productos de un supermercado, como son:

o Código del producto o Nombre del producto o Precio del Producto

o Cantidad en existencia

Para la solución se puede tener una clase llamada Productos con los siguientes atributos: codigoProducto, nombreProducto, precioProducto y cantidad. Sin embargo, qué ocurriría si deseamos almacenar la información de 5 productos; se tendrían que crear en nuestra vista 5 objetos de la clase, que representarían cada producto. Como podemos apreciar no se tendría una solución óptima desde el punto de vista del programador, debido a que la información se encuentra dispersa en distintos objetos y se hace más complicado hacer consultas o referencias específicas de la información de la entidad.

Una mejor solución al problema, es utilizar el concepto de **Vectores en Paralelo** permitiendo almacenar los diferentes atributos en vectores asociados a una entidad (Podrían utilizar el concepto de **Vectores de Objetos**, esta temática será tratada más adelante).

Entonces en la solución utilizando Vectores en Paralelo se tendrían que declarar cuatro vectores, para almacenar en cada uno de ellos la información asociada a los atributos de la clase Productos. Es decir, un vector de tipo entero para registrar los códigos de los productos, otro vector de tipo cadena para registrar los nombres de cada producto, uno para los precios y un vector para las cantidades.

Los cuatro vectores tendrán un mismo tamaño y existirá una correspondencia entre los índices de cada vector para identificar la información correspondiente a un producto específico. En la siguiente figura se aprecia cómo se almacena la información bajo el esquema de vectores en

paralelo para un producto.

**0 1**



Vector codigoProducto

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1232 | 2213 | 3812 | 4115 | 3214 |

**2 3 4**

Vector nombreProducto

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Leche | Azucar | Aceite | Huevos | Carne |

Vector precioProducto

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 2500 | 1400 | 5000 | 500 | 6500 |

Vector cantidad

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 20 | 5 | 8 | 30 | 10 |

**Figura 1. Esquemas Vectores en Paralelo**



Como se muestra en la Figura 1, la información de un producto específico se determina basado en los índices correspondientes; es decir la información del producto 1; está definida por los valores en los arreglos codigoProducto[1], nombreProducto[1], precioProducto[1] y cantidad[1] en la

posición o índice igual a 1 (Ver Figura 2).

Vector codigoProducto

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **0** | **1** | **2 3 4** | | |
| 1232 | 2213 | 3812 | 4115 | 3214 |

Vector nombreProducto



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Leche | Azucar | Aceite | Huevos | Carne |

Vector precioProducto

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 2500 | 1400 | 5000 | 500 | 6500 |

Vector cantidad

**Figura 2. Producto específico con sub\_índice 1**



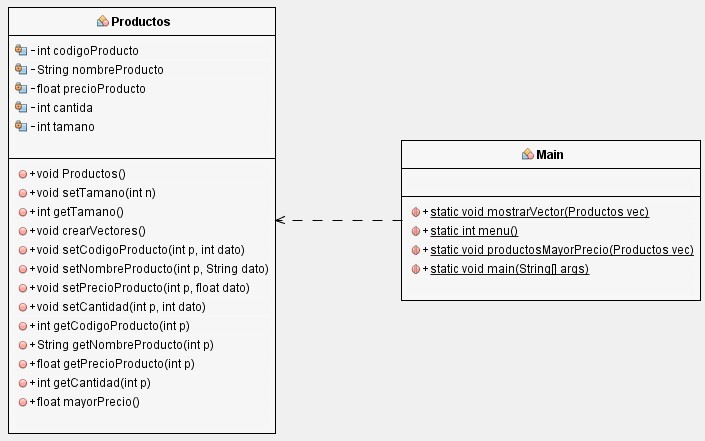
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 20 | 5 | 8 | 30 | 10 |

**IMPLEMENTACIÓN DE VECTORES EN PARALELO**

 **Ejercicio propuesto:**

Teniendo en cuenta la problemática plateada anteriormente se requiere registrar la información de los productos de un supermercado y obtener el nombre del producto o los productos de mayor valor.

 **Diseño de clases UML de la solución**



 **Implementación de la clase Productos en el fichero Productos.java**

En la clase Productos se declaran los cuatro vectores para registrar la información de los productos, además se implementa el método que devuelve el mayor precio registrado.

public class Productos {

//Se declaran los cuatro vectores para registrar la información de los productos. private int codigoProducto[];

private String nombreProducto[]; private float precioProducto[]; private int cantida[];

private int tamano; //Atributo para asignar el tamaño de los 4 vectores.

//Método constructor de la clase Productos.

public void Productos(){ codigoProducto = null; nombreProducto = null; precioProducto = null; cantida = null;

tamano = 0;

}

//Método selector y modificador para asignar el tamaño de los vectores. public void setTamano(int n){

tamano = n;

}

public int getTamano(){

return tamano;

}

//Método para crear los 4 vectores. public void crearVectores(){

codigoProducto = new int[tamano]; nombreProducto = new String[tamano]; precioProducto = new float[tamano];

cantida = new int[tamano];

}

//Métodos modificadores para cada vector.

public void setCodigoProducto(int p, int dato){

codigoProducto[p] = dato;

}

public void setNombreProducto(int p, String dato){

nombreProducto[p] = dato;

}

public void setPrecioProducto(int p, float dato){

precioProducto[p] = dato;

}

public void setCantidad(int p, int dato){

cantida[p] = dato;

}

//Métodos selectores para cada vector. public int getCodigoProducto(int p){

return codigoProducto[p];

}

public String getNombreProducto(int p){

return nombreProducto[p];

}

public float getPrecioProducto(int p){

return precioProducto[p];

}

public int getCantidad(int p){

return cantida[p];

}

//Método que devuelve el mayor precio. public float mayorPrecio(){

float mayor = getPrecioProducto(0); for (int i=0; i<=getTamano()-1; i++){ if (getPrecioProducto(i) > mayor){

mayor = getPrecioProducto(i);

}

}

return mayor;

}

}

 **Implementación de la clase Main en el fichero Main.java**

public class Main {

//Método que muestra el contenido de los cuatro vectores, manejando una correlación en su posición. public static void mostrarVector(Productos vec){

String datosVector = "";

for(int i=0; i<=vec.getTamano()-1; i++){

datosVector = datosVector+String.valueOf("CODIGO: "+vec.getCodigoProducto(i)+" "+ "NOMBRE: "+vec.getNombreProducto(i)+" "+"PRECIO: "+vec.getPrecioProducto(i)+" "+ "CANTIDAD: "+vec.getCantidad(i)+"\n");

}

JOptionPane.showMessageDialog(null, "=========== PRODUCTOS REGISTRADOS EN EL VECTOR ==========="+"\n\n"+datosVector);

}

//Método que muestra el menú de opciones al usuario. public static int menu(){

int opcion = 0;

do{

opcion = Integer.parseInt(JOptionPane.showInputDialog("===== SELECCIONE SELECCIONE UNA OPCIÓN DEL MENÚ =====\n\n"+

"1. Registrar información de los productos \n"+"2. Mostar los productos más costosos \n"+

"3. Mostrar los productos registrados \n"+"4. Salir"+ "\n \n Seleccione una opción del 1 al 4"));

}while(opcion <= 0 || opcion > 4);

return opcion;

}

//Método que lista los nombres de los productos de mayor precio. public static void productosMayorPrecio(Productos vec){

String datosVector = "";

for(int i=0; i<=vec.getTamano()-1; i++){

if(vec.getPrecioProducto(i) == vec.mayorPrecio()){

datosVector = datosVector+String.valueOf("NOMBRE: "+vec.getNombreProducto(i)+"\n");

}

}

JOptionPane.showMessageDialog(null, "=========== PRODUCTOS MAS COSTOSOS EN EL VECTOR ==========="+"\n"+datosVector);

}

public static void main(String[] args) { Productos vec = new Productos(); int opcion;

do{

opcion = menu(); switch(opcion) { case 1:

int numeroElementos = Integer.parseInt(JOptionPane.showInputDialog(null, "Digite el Número de Productos a Registrar:"));

vec.setTamano(numeroElementos);

vec.crearVectores();

for(int i=0; i<=vec.getTamano()-1; i++){

int codigo = Integer.parseInt(JOptionPane.showInputDialog(null, "Digite el Código del

Producto "+i+": "));

String nombre = JOptionPane.showInputDialog(null, "Digite el Nombre del Producto "+i+": ");

float precio = Float.parseFloat(JOptionPane.showInputDialog(null, "Digite el Precio del

Producto "+i+": "));

int cantidad = Integer.parseInt(JOptionPane.showInputDialog(null, "Digite la cantidad de

Productos "+i+": "));

vec.setCodigoProducto(i, codigo); vec.setNombreProducto(i, nombre); vec.setPrecioProducto(i, precio);

vec.setCantidad(i, cantidad);

}

JOptionPane.showMessageDialog(null,"Productos Registrados.... ");

mostrarVector(vec);

break;

case 2: productosMayorPrecio(vec); break;

case 3: mostrarVector(vec); break;

case 4:

break;

}

}while(opcion != 4);

}

}

**ACTIVIDAD**

1. Teniendo en cuenta el ejercicio implementado en el tema de **Vectores en Paralelo**, agregar la siguiente funcionalidad a la aplicación:

 Generar un Informe con los nombres de los productos que superen un precio de 2500 y tengan menos de 5 productos en existencia.

 Buscar la información de un producto pasando como paramento su código. Se recomienda implementar un método que devuelva la posición del correspondiente índice del producto buscado.

 Utilizando otro vector auxiliar, almacenar el valor total que cobrara el supermercado por la venta cada producto en existencia. Se debe tener en cuenta el precio de cada producto y la cantidad en existencia.

**BIBLIOGRAFÍA**

 Zahonero, I., y Joyanes Aguilar, L. (1999). Estructura de Datos - Algoritmos, Abstracción y

Objetos. España: McGraw-Hill.

 Allen Weiss, M. (2004). Estructuras de datos en Java. España: Addison Wesley - Pearson.

776 pp.