

CURSO DE FUNDAMENTOS DE JAVA

EJERCICIO

DISEÑO DE CLASES EN JAVA



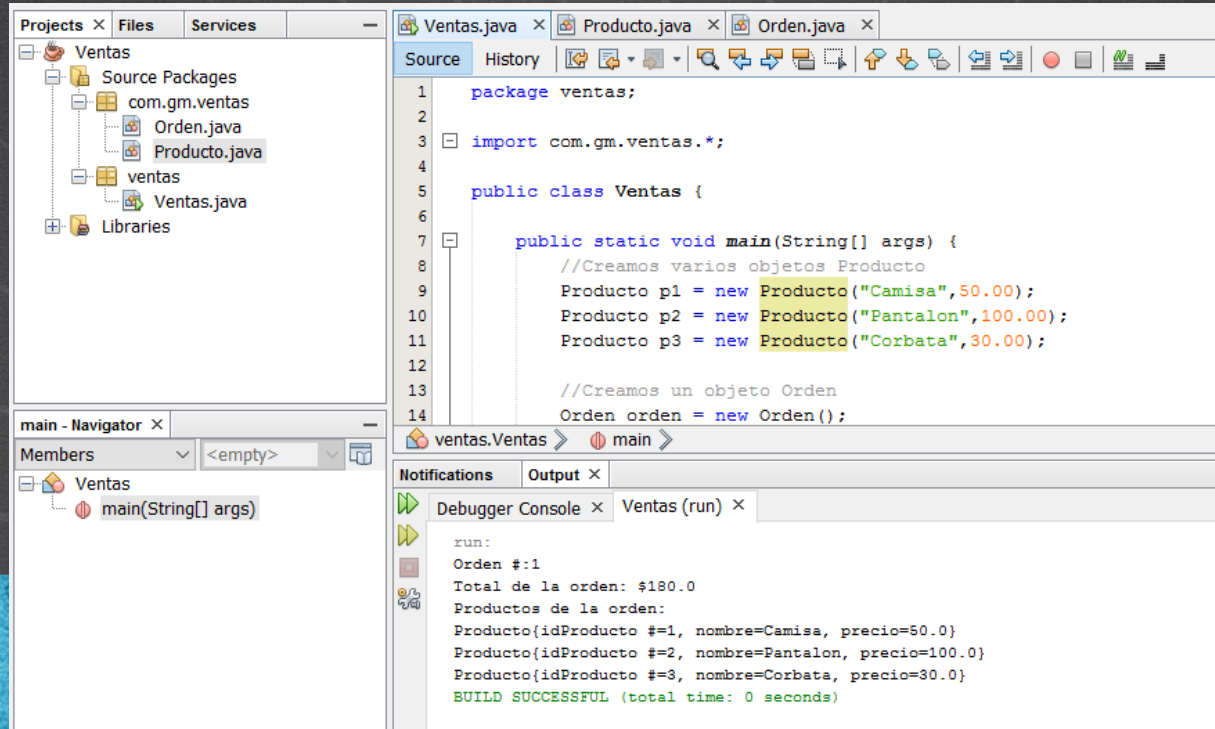
Experiencia y Conocimiento para tu vida

CURSO DE FUNDAMENTOS DE JAVA

www.globalmentoring.com.mx

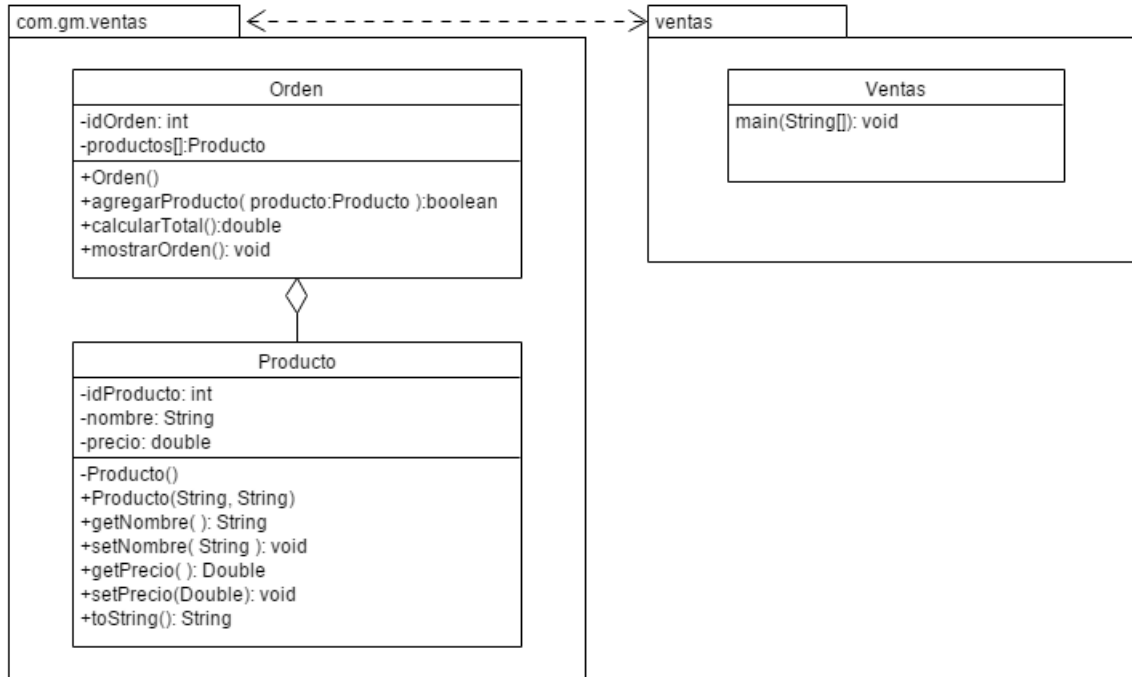
OBJETIVO DEL EJERCICIO

Con este ejercicio pondremos en práctica el concepto de diseño de Clases en Java. Al finalizar observaremos lo siguiente:



OBJETIVO DEL EJERCICIO

Crearemos el ejercicio basado en el siguiente diagrama:



PASO 1. CREACIÓN DEL PROYECTO

Vamos a crear el proyecto :

New Java Application

Steps

1. Choose Project
2. **Name and Location**

Name and Location

Project Name:

Project Location: Browse...

Project Folder:

☐ Use Dedicated Folder for Storing Libraries

Libraries Folder: Browse...

Different users and projects can share the same compilation libraries (see Help for details).

☒ Create Main Class

< Back Next > **Finish** Cancel Help

PASO 2. CREACIÓN CLASE

Creamos la clase Producto:

New Java Class

Steps

1. Choose File Type
2. **Name and Location**

Name and Location

Class Name:

Project:

Location:

Package:

Created File:

< Back Next > **Finish** Cancel Help

PASO 3. MODIFICAMOS EL CÓDIGO

Archivo Producto.java:

```
package com.gm.ventas;

public class Producto {

    private int idProducto;
    private String nombre;
    private double precio;
    private static int contadorProductos;

    //Constructor vacío
    private Producto() {
        //Asignamos el id producto único por cada objeto creado
        this.idProducto = ++contadorProductos;
    }

    //Constructor sobrecargado de dos argumentos
    public Producto(String nombre, double precio) {
        //Llamamos al constructor vacío para asignar el idProducto
        this();
        this.nombre = nombre;
        this.precio = precio;
    }
}
```

```
public String getNombre() {
    return nombre;
}

public void setNombre(String nombre) {
    this.nombre = nombre;
}

public double getPrecio() {
    return precio;
}

public void setPrecio(double precio) {
    this.precio = precio;
}

@Override
public String toString() {
    return "Producto{" + "idProducto #" + idProducto + ",
nombre=" + nombre + ", precio=" + precio + "'";
}
}
```

PASO 4. CREACIÓN CLASE

Creamos la clase Orden:

New Java Class

Steps

1. Choose File Type
2. **Name and Location**

Name and Location

Class Name:

Project:

Location:

Package:

Created File:

< Back Next > **Finish** Cancel Help

PASO 5. MODIFICAMOS EL CÓDIGO

Archivo Orden.java:

```
package com.gm.ventas;

public class Orden {

    private final int idOrden;
    private final Producto productos[];
    private static int contadorOrdenes;
    private int contadorProductos;
    private static final int maxProductos = 10;

    public Orden() {
        this.idOrden = ++contadorOrdenes;
        productos = new Producto[maxProductos];
    }

    public void agregarProducto(Producto producto) {
        //Si los productos agregados no superan al máximo
        // de productos, agregamos el nuevo producto
        if (contadorProductos < maxProductos) {
            //Agregamos el nuevo producto al arreglo
            //e incrementamos el contador de productos
            productos[contadorProductos++] = producto;
        }
        else{
            System.out.println("Se ha superado el máximo de
productos: " + maxProductos);
        }
    }
}
```

```
public double calcularTotal() {
    double total = 0;
    for (int i = 0; i < contadorProductos; i++) {
        total += productos[i].getPrecio();
    }
    return total;
}

public void mostrarOrden() {
    System.out.println("Orden #:" + idOrden);
    System.out.println("Total de la orden #"
        + idOrden + ": $" + calcularTotal());
    System.out.println("Productos de la orden #"
        + idOrden + ":");
    for (int i = 0; i < contadorProductos; i++) {
        System.out.println(productos[i]);
    }
}
```

AMENTOS DE JAVA

ntoring.com.mx

PASO 6. MODIFICAMOS EL CÓDIGO

Archivo Ventas.java:

```
package ventas;

import com.gm.ventas.*;

public class Ventas {

    public static void main(String[] args) {
        //Creamos varios objetos Producto
        Producto p1 = new Producto("Camisa", 50.00);
        Producto p2 = new Producto("Pantalon", 100.00);
        Producto p3 = new Producto("Corbata", 30.00);

        //Creamos un objeto Orden
        Orden orden1 = new Orden();

        //Agregamos los productos a la orden
        orden1.agregarProducto(p1);
        orden1.agregarProducto(p2);
        orden1.agregarProducto(p3);

        //Imprimimos la orden
        orden1.mostrarOrden();

        //Creamos una segunda orden
        Orden orden2 = new Orden();

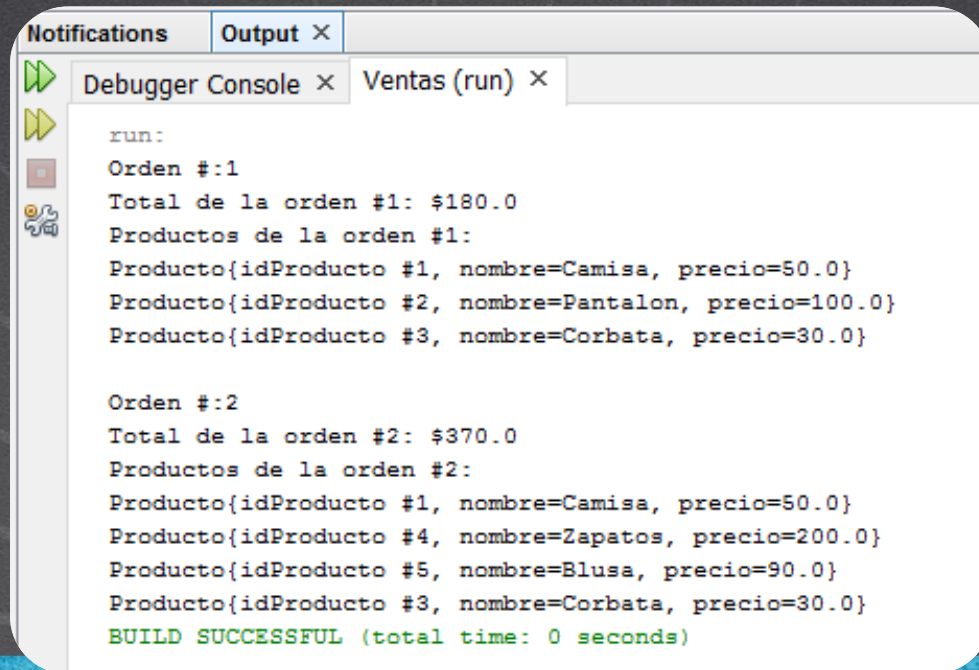
        //Creamos nuevos productos
        Producto p4 = new Producto("Zapatos", 200);
        Producto p5 = new Producto("Blusa", 90);

        //Agregamos productos a la orden 2
        orden2.agregarProducto(p1);
        orden2.agregarProducto(p4);
        orden2.agregarProducto(p5);
        orden2.agregarProducto(p3);

        //Imprimimos la orden 2
        System.out.println("");
        orden2.mostrarOrden();
    }
}
```

PASO 7. EJECUTAMOS EL PROYECTO

El resultado es como sigue:



```
run:
Orden #:1
Total de la orden #1: $180.0
Productos de la orden #1:
Producto{idProducto #1, nombre=Camisa, precio=50.0}
Producto{idProducto #2, nombre=Pantalon, precio=100.0}
Producto{idProducto #3, nombre=Corbata, precio=30.0}

Orden #:2
Total de la orden #2: $370.0
Productos de la orden #2:
Producto{idProducto #1, nombre=Camisa, precio=50.0}
Producto{idProducto #4, nombre=Zapatos, precio=200.0}
Producto{idProducto #5, nombre=Blusa, precio=90.0}
Producto{idProducto #3, nombre=Corbata, precio=30.0}
BUILD SUCCESSFUL (total time: 0 seconds)
```

CURSO DE FUNDAMENTOS DE JAVA

www.globalmentoring.com.mx

TAREAS EXTRA DEL EJERCICIO

- Probar con el modo debug del IDE y verificar paso a paso.
- Crear más ordenes y productos, y probar con distintos valores y verificar el resultado.



CURSO DE FUNDAMENTOS DE JAVA

www.globalmentoring.com.mx

CONCLUSIÓN DEL EJERCICIO

- Con este ejercicio hemos puesto en práctica el concepto diseño de clases en Java, y como apoyarnos en la relación entre ellas para poder comenzar a crear sistemas cada vez más complejos.
- Este tema es apenas el principio de lo que estudiaremos en cursos posteriores, pero lo importante es tener las bases de Java para seguir avanzando con pasos firmes en el aprendizaje de este fabuloso lenguaje de programación.
- Con esta lección concluimos el curso de Fundamentos de Java y los invitamos a seguir su especialización Java y estudio del siguiente curso: *Programación con Java*. Los esperamos



CURSO DE FUNDAMENTOS DE JAVA

www.globalmentoring.com.mx

CURSO ONLINE

FUNDAMENTOS DE JAVA

Por: Ing. Ubaldo Acosta



Experiencia y Conocimiento para tu vida

CURSO DE FUNDAMENTOS DE JAVA

www.globalmentoring.com.mx