



Gestión de Almacén		
1º DAW - ETS		
Departamento de Informática	Fecha	Luke Eric Marten Llorente

Descripción

Se desarrollará un sistema de gestión para un almacén mediante dos clases: **Producto** y **Almacén**. La clase **Producto** representará información sobre un artículo, mientras que **Almacén** permitirá administrar una colección de productos. También se implementará la clase **Main_Almacen** para probar su funcionamiento.

Requisitos

Clase **Producto**

Propiedades

- **String nombre**: Nombre del producto.
- **String descripcion**: Breve descripción del producto.
- **double precio**: Precio del producto.
- **int cantidad**: Cantidad disponible en el almacén.

Métodos

- **Constructores**:
 - Uno que inicializa todos los atributos.
 - Otro que inicializa **nombre**, **descripcion** y **precio**, asignando **0** a la cantidad.
- **Getters y Setters** para todos los atributos.
- **toString()**: Devuelve una cadena con la información del producto en el formato:
 - "Nombre: [nombre] - Descripción: [descripcion] - Precio: [precio] - Cantidad: [cantidad]".

Clase **Almacén**

Propiedades

- **Producto[] productos**: Array que almacena los productos en el almacén.
- **int[] estante**: Números de estante asignados a cada producto.

Métodos

- **Constructores**:
 - Uno que inicializa un almacén vacío.
 - Otro que recibe un array de **Producto**, lo almacena y asigna números de estante consecutivos comenzando desde 1.
- **void append(Producto nuevoProducto)**: Agrega un producto nuevo al almacén.
- **toString()**: Devuelve la lista de productos en formato alineado:
 - "Estante: [estante] - Nombre: [nombre] - Descripción: [descripcion] - Precio: [precio] - Cantidad: [cantidad]".

Clase **Main_Almacen**

- Crear instancias de **Producto** con distintos valores.
- Inicializar un array de productos y usarlo para crear un **Almacén**.
- Agregar productos adicionales con **append()**.
- Imprimir el almacén con **toString()**.