

Documentación

Dario Agustin Valeriano 110400

Clase Producto:

Representa un producto en una tienda, con atributos como nombre, precio, stock y si está en oferta. Proporciona funcionalidades para obtener información sobre el producto, como su nombre, precio, stock y si está en oferta.

Funciones:

Producto(): Constructor predeterminado que crea un nuevo objeto de tipo Producto.

Pre: Ninguna.

Post: Crea un nuevo objeto de tipo Producto.

Producto(string nombre, double precio, bool oferta, int stock, double precio_item):

Constructor que crea un nuevo objeto de tipo Producto con los valores asignados.

Pre: Ninguna.

Post: Crea un nuevo objeto de tipo Producto con los valores asignados.

~Producto(): Destructor de la clase Producto.

Pre: Ninguna.

Post: Libera cualquier recurso utilizado por el objeto Producto.

string obtener_nombre(): Devuelve el nombre almacenado del producto.

Pre: Ninguna.

Post: Devuelve el nombre almacenado del producto.

double obtener_precio(): Devuelve el precio del producto.

Pre: Ninguna.

Post: Devuelve el precio del producto.

bool obtener_oferta(): Devuelve true si el producto tiene oferta, de lo contrario devuelve false.

Pre: Ninguna.

Post: Devuelve true si el producto tiene oferta, de lo contrario devuelve false.

int obtener_stock(): Devuelve el stock del producto.

Pre: Ninguna.

Post: Devuelve el stock del producto.

double obtener_precio_item(): Devuelve el precio total del item, según la cantidad a adquirir.

Pre: Ninguna.

Post: Devuelve el precio total del item, según la cantidad a adquirir.

Clase Gondola:

Representa la góndola de productos de una tienda, donde se almacenan los productos disponibles para la venta. Esta clase gestiona un vector dinámico de productos, permitiendo agregar, eliminar y actualizar productos en la góndola, así como también obtener información sobre los productos almacenados y actualizar el archivo que contiene la información de los productos.

Funciones:

Gondola::Gondola(int capacidad): Constructor que inicializa una instancia de la clase Gondola con una capacidad inicial especificada.

Pre: Ninguna.

Post: Crea una nueva instancia de la clase Gondola, siendo un vector dinamico de objetos de clase Producto.

Gondola::~~Gondola(): Destructor que libera la memoria asignada dinámicamente para el vector de productos de la góndola.

Pre: Ninguna.

Post: Libera la memoria asignada dinámicamente para el vector, asegurando que no haya fugas de memoria.

void Gondola::ingresar_producto(string nombre, double precio, bool oferta, int stock): Agrega un nuevo producto a la góndola.

Pre: Ninguna.

Post: Agrega un nuevo producto al final del vector.

Producto Gondola::obtener_producto(int indice): Obtiene el producto almacenado en el índice especificado de la góndola.

Pre: indice debe ser un valor que esté dentro del rango de índices válidos para el vector gondola.

Post: Devuelve el producto almacenado en la posición especificada por indice en el vector gondola.

int Gondola::obtener_tamano(): Obtiene la cantidad de productos ingresados en la góndola.

Pre: Ninguna.

Post: Devuelve la cantidad de productos ingresados en el vector.

int Gondola::obtener_capacidad(): Obtiene la capacidad actual del vector de productos de la góndola.

Pre: Ninguna.

Post: Devuelve la capacidad actual del vector.

void Gondola::redimensionar(int nueva_capacidad): Redimensiona el vector de productos de la góndola con la nueva capacidad especificada.

Pre: nueva_capacidad debe ser mayor a la cantidad de productos ingresados.
Post: Redimensiona el vector con la nueva capacidad introducida por parámetro.

void Gondola::verificar_redimensionar(): Verifica si es necesario redimensionar el vector de productos de la góndola.

Pre: Ninguna.

Post: Verifica si se necesita redimensionar el vector.

void Gondola::actualizar_producto(int indice, int cantidad): Actualiza la cantidad de stock del producto almacenado en la posición especificada del vector góndola.

Pre: indice debe estar dentro de la capacidad de la góndola y cantidad debe ser un número válido.

Post: Actualiza el producto almacenado en la posición especificada por indice en el vector góndola.

void Gondola::eliminar_producto(int indice): Elimina el producto almacenado en la posición especificada del vector góndola.

Pre: indice debe ser un valor válido.

Post: Elimina el producto almacenado en la posición especificada por indice en el vector góndola.

void Gondola::imprimir_productos_gondola(): Muestra por pantalla los productos disponibles en la góndola junto con su información relevante.

Pre: Ninguna.

Post: Muestra por consola cada producto dentro del vector gondola.

void Gondola::actualizar_archivo_producto(ofstream& archivo): Actualiza el archivo que contiene la información de los productos de la góndola.

Pre: Ninguna.

Post: Actualiza el archivo ingresado por parámetro.

Clase Chango

Representa un carrito de compras en una tienda, donde los clientes pueden agregar productos. Esta clase gestiona un vector dinámico de productos, permitiendo agregar, eliminar y modificar productos en el carrito, así como también obtener información sobre los productos almacenados y el total de la compra. Proporciona funcionalidades para la administración y manipulación de productos dentro del carrito de compras.

Funciones:

Chango::Chango(int capacidad): Crea una nueva instancia de la clase Chango, con un vector dinámico de productos y una capacidad inicial especificada.

Pre: Ninguna.

Post: Crea una nueva instancia de la clase **Chango**, siendo un vector dinámico de objetos de clase **Producto**, con una capacidad inicial especificada.

Chango::~~Chango(): Libera la memoria asignada dinámicamente para el vector de productos del carrito.

Pre: Ninguna.

Post: Libera la memoria asignada dinámicamente para el vector, asegurando que no haya fugas de memoria.

Producto Chango::encontrar_producto(string nombre_producto, Gondola& gondola):

Busca un producto en la góndola con el nombre especificado y lo devuelve.

Pre: nombre_producto debe ser un nombre almacenado en un producto dentro de gondola.

Post: Devuelve el producto encontrado.

void Chango::agregar_al_carrito(Producto& product, int cantidad):

Agrega un producto al carrito junto con su cantidad correspondiente, calculando su precio total considerando cualquier descuento aplicable.

Pre: Ninguna.

Post: Agrega al vector chango un producto.

Producto Chango::obtener_producto_chango(int indice):

Obtiene el producto almacenado en el índice especificado del carrito.

Pre: indice debe estar dentro de la capacidad del vector chango.

Post: Devuelve el producto almacenado en el índice ingresado del vector chango.

int Chango::obtener_tamano_chango():

Devuelve la cantidad de productos almacenados en el carrito.

Pre: Ninguna.

Post: Devuelve la cantidad de productos ingresados en el chango, siendo esta el tamaño ocupado en el vector.

int Chango::obtener_indice_encontrado():

Devuelve el índice en el que se encontró el último producto buscado en la góndola.

Pre: Ninguna.

Post: Devuelve el índice en el que se encontro el producto buscado.

void Chango::imprimir_productos_chango():

Muestra por pantalla los productos en el carrito junto con su información relevante y el total de la compra.

Pre: Ninguna.

Post: Muestra por pantalla los productos dentro del vector chango.