UNIVERSIDAD TECMILENIO

Mauricio Estrada De la Garza - AL02976904
Igor Sung Min Kim Juliao - AL02829189
22/09/2023
Programación Orientada a Objetos
Reto 5

Reto 5: Implementación de Arreglos de dos Dimensiones y Manejo de Errores.

Objetivo de la actividad:

El alumno desarrollará un programa en el cual se implemente los conceptos:

1. **Creación de clases.**
1. **Implementación de Clases usando objetos.**
1. **Uso de atributos de un objeto y una clase.**
1. **Uso de métodos de un objeto y una clase.**

Requerimientos para la actividad:

Uso Java jdk a través de un IDE, preferentemente Netbeans. ☕

Instrucciones para el alumno:

Descripción:

Supongamos que estás trabajando en un sistema de inventario para una tienda en línea. Tu tarea es diseñar y crear una clase llamada "Producto" que represente un producto que se vende en la tienda. Cada objeto "Producto" debe tener ciertos atributos y métodos para administrar su información.

Requisitos:

- **Crea una clase llamada "Producto" con los siguientes atributos privados:**
- **id (int): El identificador único del producto.**
- **nombre (String): El nombre del producto.**
- **precio (double): El precio del producto.**
- **stock (int): La cantidad en stock del producto.**

- **Implementa un constructor por defecto que inicialice todos los atributos con valores predeterminados. (X)**
- **Implementa un constructor sobrecargado que permita la creación de objetos "Producto" con valores personalizados para todos los atributos. (X)**
- **Implementa métodos públicos para acceder a los atributos privados (getters) y para modificar los atributos (setters). Cada atributo debe tener su propio getter y setter. (X)**
- **Agrega un método público llamado vender (int cantidad) que permita restar la cantidad especificada del stock del producto. Asegúrate de verificar que la cantidad vendida no sea mayor que el stock disponible.**
- **Agrega un método público llamado reabastecer(int cantidad) que permita aumentar la cantidad en stock del producto**

Permitir a los usuarios agregar más productos utilizando un menú. Los productos agregados deben almacenarse en un arreglo (o lista) de productos.

Requisitos Adicionales:

- **Crea un arreglo (o lista) de objetos "Producto" para almacenar los productos agregados.**
- **Implementa un menú que permita al usuario realizar las siguientes acciones:**
- **Agregar un nuevo producto (proporcionando el ID, nombre, precio y stock).**
- **Vender un producto existente.**
- **Reabastecer un producto existente.**
- **Mostrar la información de todos los productos en el inventario.**
- **Utiliza un bucle para que el menú se ejecute continuamente hasta que el usuario decida salir. (X)**

Conceptos a calificar:

Definición de funciones

Creación de Clases(s)

Definición y creación de atributos

Definición y creación de métodos.

Creación de objetos.

Funcionalidad

🍪🍪 ***Brownie points, si se utilizan excepciones*** 🍪🍪😃😃 **(X)**

1. **Reflexión: Discutir y reflexionar con diversos ejemplos de uso de objetos y sus diferencias con la programación estructurada.**

Entregables:

- **Código Fuente, archivos con extensión .java (Recuerda, cualquier otro tipo de archivo no es permitido, i.e. .class, .zip, .jar, etc)**
- **Compilación sin errores.**
- **Cumplimiento de Especificaciones.**
- **Reporte.**
- **¡¡¡¡¡ Reflexión individual por integrante del equipo. !!!!!**

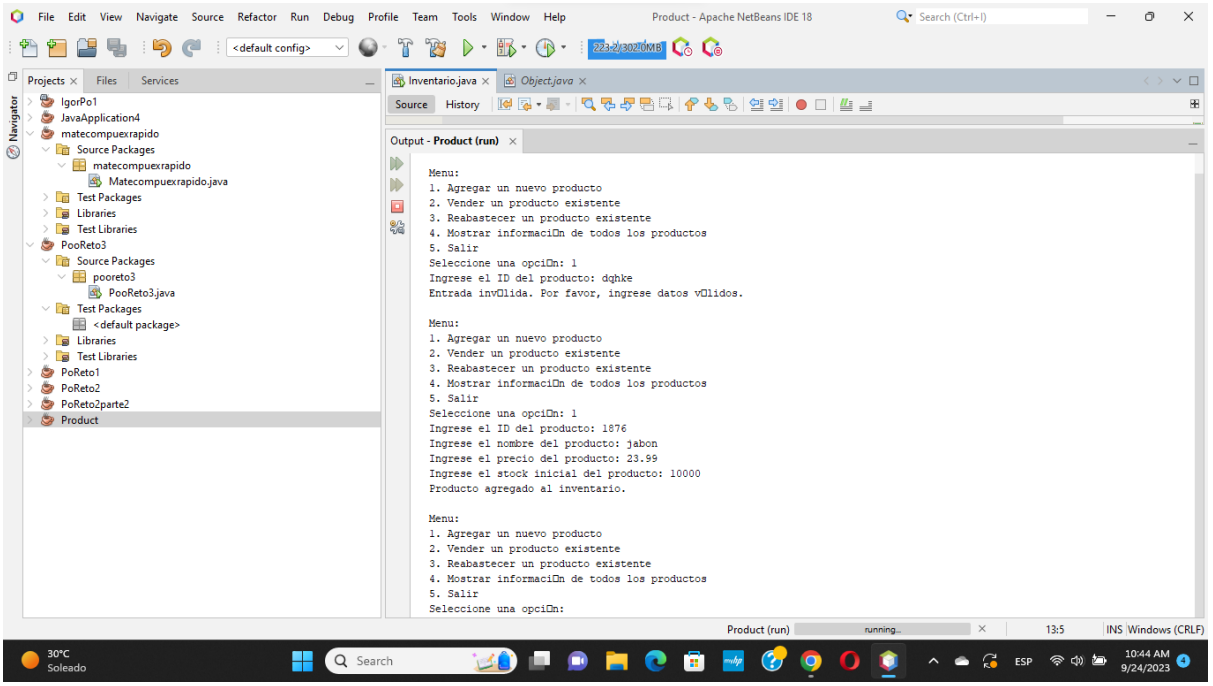
Importante: Solo se revisa la última entrega

*** Sin reporte y/o código fuente, ningún porcentaje es considerado ⇒ Calificación NE**

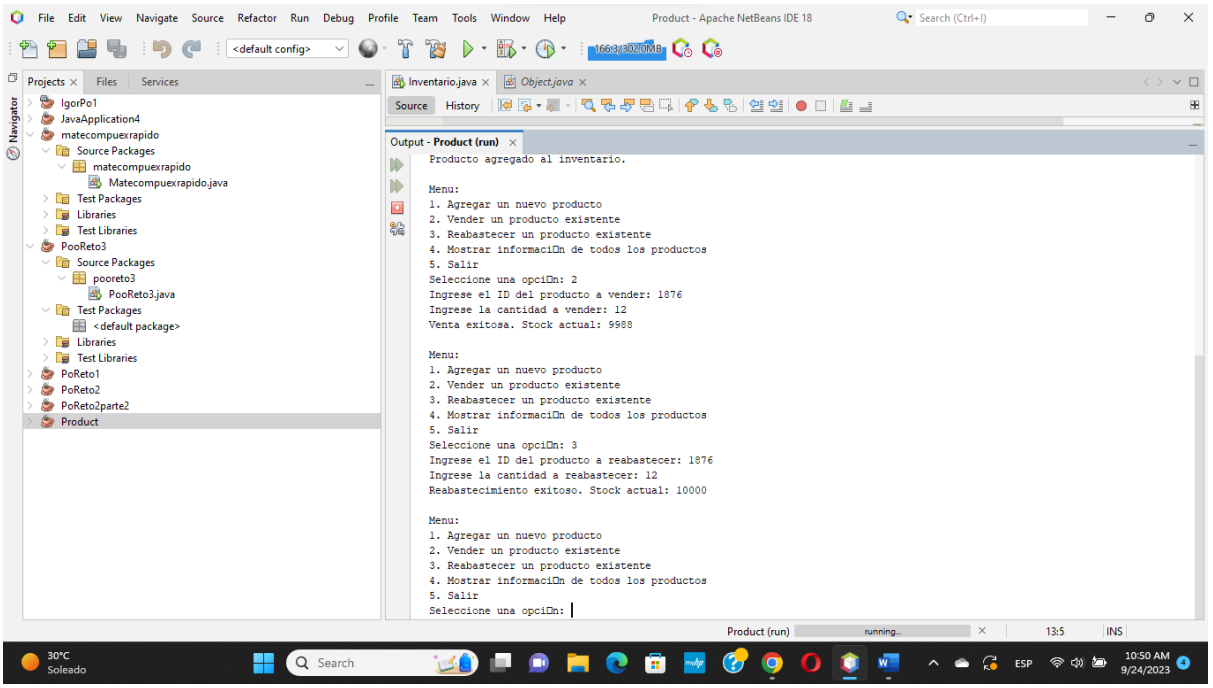
Restricción: Equipos de no más de tres personas, no menos de dos. 😀

CÓDIGO:

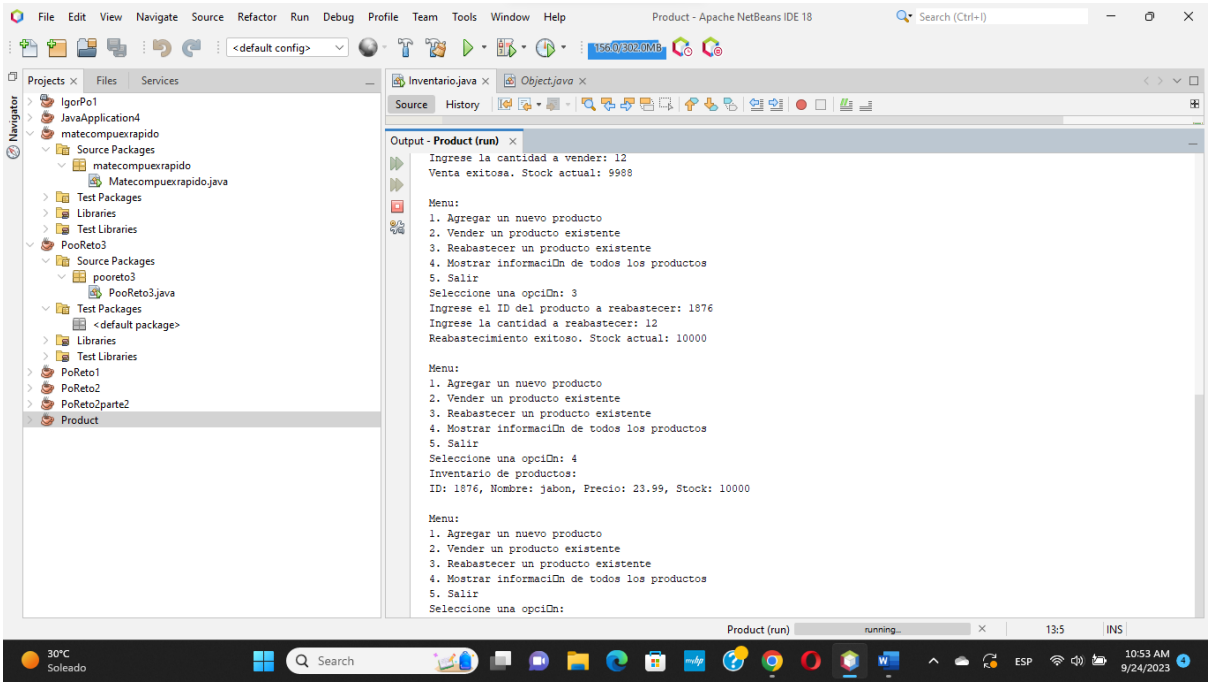
PRUEBAS:



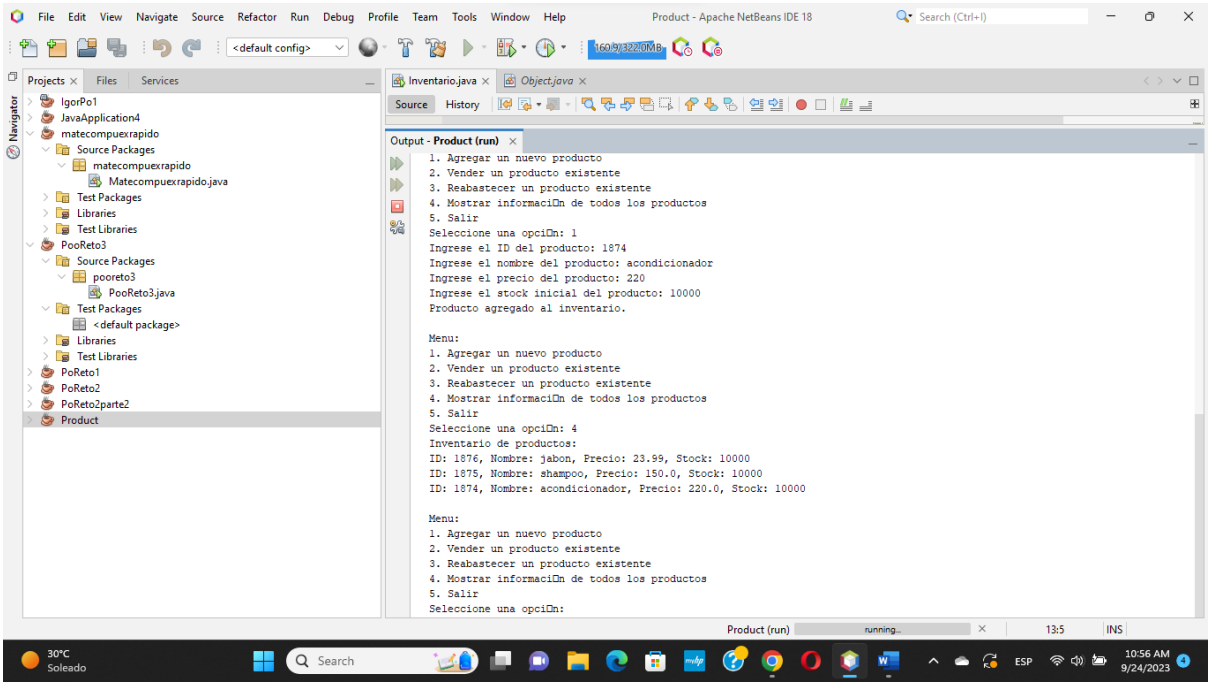
En esta captura se muestran tres funciones del codigo, una es él uso de la excepcion mismatch como parte inicial del run, la segunda es él agregar un producto él id del producto es 1876 y es jabon, él agregado fueron 10000 piezas.



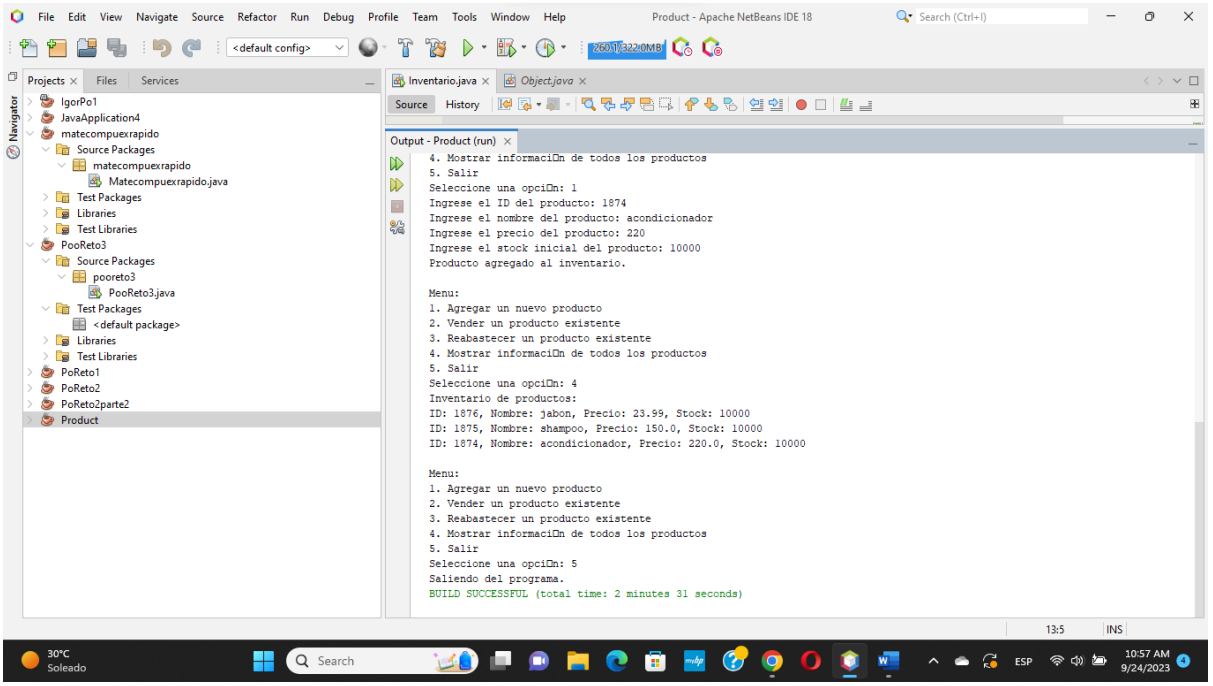
En esta captura se muestra como es que se vendieron 12 piezas de jabon, y como es que se abastece un producto ya existente en este caso él jabon.



en esta captura se muestra la información de todos los productos existentes



en esta captura agregue nuevos productos para mostrar como es que funciona cuando tienes más productos agregados.



y en esta ultima captura salí del programa.

REPORTE:

Este código tiene dos clases principales una llamada Product y otra llamada ManejarInventario,
la clase product tiene 4 propiedades, las que son id, nombre, precio y stock,
siendo id el número de identificación del producto, nombre el nombre del producto, precio el precio del producto y stock la cantidad de productos en él inventario.

Product

- dentro de esta clase tiene las siguientes funciones(o métodos si así prefieren llamarlos) Product() que es la función constructora que inicializa los valores del producto.
- Product(int id, String nombre, double precio, int stock), que es la funcion constructora que permite crear nuevos productos(objetos) con valores personalizados.
- getID(), que permite obtener el valor de la propiedad id del objeto creado
- getNombre(), que permite obtener el valor de la propiedad nombre del objeto creado
- getPrecio(), que permite obtener el valor de la propiedad precio del objeto creado
- getStock(), que permite obtener el valor de la propiedad stock del producto
- setID(int id), establece el valor obtenido dentro de la propiedad ID del objeto creado
- setNombre(String nombre), establece el valor obtenido dentro de la propiedad Nombre del objeto creado
- setPrecio(double precio), establece el valor obtenido dentro de la propiedad Precio del objeto creado
- setStock(int stock), establece él valor obtenido dentro de la propiedad Stock del objeto creado
- vender(int cantidad), resta la cantidad establecida a la cantidad del stock, tiene una excepcion para cualquier cantidad que sea mayor a la cantidad guardada dentro del stock o en caso de que sea menor o igual a 0

- reabastecer(int cantidad), suma la cantidad establecida al stock, tiene una excepcion en caso de que la cantidad establecida sea menor o igual a 0
- toString(), representa en una cadena cualquiera de las propiedades anteriormente vistas

ManejarInventario
Dentro de esta clase está el main él cual esta conformado por un arraylist para hacer una lista de los objetos que van a entrar dentro del inventario, y tambien esta el menu de operaciones que funciona usando un while como bucle o ciclo para que mientras el usuario siga queriendo realizar operaciones pueda hacerlas, recopila los datos dentro del int y utiliza un switch dentro de un try para recopilar los datos y seleccionar que operacion realizar siendo 1 agregar producto donde utiliza la funcion scanner para agregar un nuevo producto a la lista de inventario, 2 vender un producto existente donde se pide él id del producto a vender y se utiliza un for para utilizar la funcion vender mientras que él producto electo este dentro de la lista del inventario, 3 reabastecer un producto existente usando él for de la isma manera que en él caso 2 solo que esta vez usando la funcion reabastecer, 4 mostrar él inventario de todos los productos otra vez usando él for para ver si él producto esta dentro de la lista del inventario para luegohacer un print de todas las propiedades del producto usando la funcion tostring dentro de un sout y 5 salir, teniendo como default la petición de realizar la operación de nuevo por que eligio una operacion no valida, tambien tiene dos catch una que es una exception para un illegalargument, como por ejemplo en él caso de que coloque una cantidad mayor a la que hay en él stock a la hora de vender un producto, y una exception del tipo input mismatch para cuando coloca un dato invalido, como por ejemplo colocar letras cuando pida él id de un producto que es un string.

CONCLUSIÓN:

-Mau: Fue algo nuevo para mi hacer un ejercicio ya de Programación Orientada a Objetos, fue al principio un poco confuso pero poco a poco le iré agarrando mas la onda, asi estaba igual el semestre pasado con los nuevas temas que veíamos en C#. También me he dado cuenta que el trabajo en equipo es importante ya que también puedes aprender de los demás y es una competencia necesaria en el mundo laboral, nosotros como programadores tendremos que arreglar bugs que a lo mejor no comprendemos del todo y por eso ocupamos apoyarnos de un equipo.

-Igor: La verdad que mi experiencia con programación orientada a objetos no fue algo muy nuevo que digamos, porque ya tenía experiencia en ello, ya que estudie un semestre en la anahuac y ahi vi programacion orientada a objetos en c++, pero aunque los fundamentos base son los mismos, lo que si se me hizo diferente es como java maneja la programacion orientada a objetos, ya que java lo maneja de una manera un poco más sotisficada, en c++ los metodos son muy brutos por así decirlo y uno tiene que o crear su propio codigo para hacer algo o tendria que llenar el archivo de librerias para poder hacer las mismas cosas

que se hacen aqui, recuerdo que la ultima vez que programe en c++ fue para hacer un ajedrez y tuve que agregar alrededor de 12 o 13 librerias distintas para poder programar, mientras que en java solo necesite las librerias para la funcion scanner, para hacer él arraylist y para las excepciones. senti que fue mucho más simple que en c++ y la gran diferencia que hay entre orientada a objetos y programacion estructurada en mi opinion es que en la programacion orientada a objetos tienes mucha más libertad en él manejo del codigo, por lo menos en él sentido que no tienes que inicializar variables para cada cosa que hagas, y puedes en vez de tener que agregar algo dentro del codigo fuente cada vez que quieras colocar algo diferente, solo tienes que construir él objeto, por ejemplo si estamos hablando de que vamos a hacer un sistema de banco y tenemos que hacerlo a partir de programacion estructurada tendriamos que agregar una variable para cada propiedad, y tendriamos que hacer una variable por propiedad por cliente, y no se podrian agregar más clientes, mientras que en poo solo se necesita crea un objeto llamado cliente, agregar las propiedades que tendria él cliente, y usar un ciclo para colocar la funcion constructora para cada cliente nuevo que queremos agregar y poder modificar cada cliente distinto, tambien se tendria que usar una lista para guardar los objetos pero aun así te ahorrarias muchas lineas de codigo. En otras palabra mientras tengas él objeto llamado cliente podras tener cuantos clientes tu quieras sin tener que agregar más lineas de codigo, solo tendrias que hacer una manipulacion de datos para que se guarden los datos en algún archivo o en la metadata de un servidor, aqui fue lo mismo, tendria que haber creado la propiedad para cada producto individual que quisiera vender, mientras que usando él objeto product me ahorre muchas lineas solo agregando las propiedades dentro del objeto, en mi opinion no esta más dificil los fundamentos base son los mismos solo que hacer la programacion orientada a objetos hace que sea mucho más practico, tambien ayuda mucho a la hora de hacer debugging porque solo tienes que checar una vez por cada objeto de clase distinto, en vez de checar linea por linea él codigo. solo tendrias que checar si salio bien o mal dentro del objeto, o en caso de lo que este mal no este dentro del objeto, checar dentro del main.