

Programación Básica 1 - Final 24 de Octubre 2023

Gestión de vivero

Nos han contratado para desarrollar un producto software que permita gestionar las operaciones necesarias para administrar un vivero.

Inicialmente podremos agregar plantas, modificar su stock, consultarlas y venderlas. En esta versión, no registramos ventas, solo manejaremos el saldo acumulado por las ventas realizadas.

Cada vivero tiene un cupo de macetas que representa la cantidad de plantas que podemos tener para vender.

Clase de prueba GestionDeVivero

Será la clase que contenga al método main y deberemos desarrollar lo siguiente para garantizar el correcto funcionamiento:

- **private static Planta ingresarPlanta():** Solicita el ingreso de los datos de una planta y devuelve una instancia de Planta. Considerar el uso del método ingresarTipoDePlanta(MENU_TIPO_DE_PLANTA).
- **private static MenuPrincipal obtenerOpcionDeMenuPrincipal(int numeroIngresado):** Devuelve la opción seleccionada por el usuario existente en el enum.
- **private static int ingresarNumeroEntero(String mensaje)**
- **private static String ingresarString(String mensaje)**
- **private static String ingresarString(String mensaje)**
- **private static double ingresarDouble(String mensaje):** Este grupo de métodos, muestra el mensaje suministrado por pantalla y devuelve el dato ingresado del tipo correspondiente.
- **public static void main(String[] args):** Contiene el algoritmo principal y se deberán completar las tareas solicitadas antes del switch y los casos del mismo.
 - **case AGREGAR_PLANTA:** Ingresar una planta utilizando el método ingresarPlanta() y luego mostrar si fue posible o no realizar la acción.
 - **case MODIFICAR_STOCK_PLANTA:** Solicitar el código y nuevo stock de la planta a la cual se le quiere actualizar el stock e indicar con mensajes si fue posible realizar la operación.

- **case BUSCAR_PLANTA_POR_CODIGO:** Ingresar el código de la planta para buscarla, en caso de existir mostrarla, caso contrario indicar con un mensaje que no se encontró.
- **case BUSCAR_PLANTAS_QUE_CONTIENEN:** Ingresar el texto a buscar en los nombres de las plantas y mostrar las que contenga el texto (se provee de un metodo mostrarPlantas()).
- **case VENDER_PLANTA:** Se debe solicitar el ingreso del código y cantidad a vender de la planta. Mostrar un mensaje de éxito o error según corresponda.
- **case OBTENER_PLANTAS_DEL_TIPO:** Ingresar el tipo de planta utilizando el método ingresarTipoDePlanta(MENU_TIPO_DE_PLANTA). Luego se debe obtener las plantas de ese tipo y mostrarlas (se provee de un método mostrarPlantas()).
- **case SALIR:** Antes de salir, mostrar el estado actual del vivero para visualizar el saldo.

Clase Vivero

Esta clase contiene la lógica principal con la cual se maneja el saldo y las plantas del vivero. Cuando se crea una instancia de vivero, se debe generar una contraseña aleatoria para poder ingresar al sistema.

Además es necesario resolver los siguientes métodos:

- **public boolean agregarPlanta(Planta planta):** Agrega la planta al array solo si no existe otra con el mismo código en el array de plantas. Considerar usar el método buscarPlantaPorCodigo(codigo).
- **public boolean venderPlanta(int codigo, int cantidadAVender):** Para vender una planta, primero debemos revisar que exista (buscando por código, por ejemplo) y luego verificar que tenga stock para satisfacer la cantidad a vender. Si es viable la venta, se deberá modificar el stock de la planta en el array de plantas (considerar el método modificarStockDePlantaPorCodigo()) para disminuir el mismo, y se deberá acumular el precio final de la venta al saldo del vivero. Ejemplo: Precio final de planta \$10, cantidad de plantas en la venta 10 unidades, entonces se acumulan \$100 al saldo del vivero.
- **public Planta buscarPlantaPorCodigo(int codigo):** Buscar la planta en el array de plantas utilizando el código de la planta. En caso de no existir, devolver null.
- **public Planta[] buscarPlantasQueContienen(String texto):** Generar un nuevo array con las plantas que en su nombre, contengan el texto suministrado. El array

no debe poseer espacios, es decir, las plantas deben ingresar ordenadamente al array nuevo (no permitir posiciones en null entre las plantas ingresadas).

- **public boolean modificarStockDePlantaPorCodigo(int codigo, int stock):** Iterar el array de plantas y en caso de existir alguna y que coincida con el código suministrado, actualizar el stock de la misma.
- **private String generarContraseña():** Generar una contraseña aleatoria y devolverla. La misma debe poseer: 4 caracteres (en el orden que desee) los cuales deben ser: 1 número, 1 letra mayúscula y 2 letras minúsculas. Considerar el método obtenerCaracterAleatorio(1,10).
- **private char obtenerCaracterAleatorio(int posicionInicial, int posicionFinal):** Generar un carácter aleatorio considerando las posiciones de los caracteres en la tabla ASCII.
- **private Planta[] ordenarPlantasPorNombreAscendente(Planta[] plantas):** Ordenar las plantas en el array suministrado por nombre de manera ascendente y devolver el array.

Clase Planta

Representa la abstracción de una planta y se deberán completar los siguientes métodos, incluyendo su constructor:

- **public double obtenerPrecioFinal():** Calcular y devolver el precio final de la planta. Para ello, se deberá revisar el TipoDePlanta considerando: Hierba o Mata: se incrementa un 20% el precio base. Arbusto: se incrementa un 60% el precio base. Árbol: se incrementa un 130% el precio base. Usar las constantes
- **public String toString():** Mostrar la información actual de la planta incluyendo el precio final. Considerar el método: obtenerPrecioFinal().

Lineamientos

- No es posible modificar la firma de los métodos existentes, pero es posible agregar todos los métodos que se consideren necesarios.
- Se pueden aplicar las mejoras que considere necesarias, siempre cumpliendo el punto anterior.
- Se deberán completar los métodos y código faltante para que el sistema funcione correctamente.

- Deberá ajustar el nombre del proyecto suministrado indicando sus apellidos y nombres en donde dice “ApellidosNombres”:
PB1-FinalOctubre2023-**ApellidosNombres**.
- La versión de java del proyecto deberá ser ajustada a la del laboratorio.

Condiciones de aprobación

Se considerará desaprobado todo proyecto que:

- a. no compile, o,
- b. no se identifiquen los atributos necesarios para resolver lo solicitado, o,
- c. los tipos de datos elegidos para los atributos no sean los adecuados, o,
- d. no cumpla con las condiciones mínimas descritas a continuación.

Se considerará aprobado aquel examen que cumpla con el 70% de puntaje.

Clase de prueba GestionDeVivero	private static Planta ingresarPlanta()	5%
	private static MenuPrincipal obtenerOpcionDeMenuPrincipal(int numeroIngresado)	5%
	private static int ingresarNumeroEntero(String mensaje) private static String ingresarString(String mensaje) private static String ingresarString(String mensaje) private static double ingresarDouble(String mensaje)	5%
	public static void main(String[] args)	15%
Clase Vivero	public boolean agregarPlanta(Planta planta)	5%
	public boolean venderPlanta(int codigo, int cantidadAVender	10%
	public Planta buscarPlantaPorCodigo(int codigo)	5%
	public Planta[] buscarPlantasQueContienen(String texto)	5%
	public boolean modificarStockDePlantaPorCodigo(int codigo, int stock)	5%
	private String generarContrasenia()	5%

	private char obtenerCaracterAleatorio(int posicionInicial, int posicionFinal)	10%
	private Planta[] ordenarPlantasPorNombreAscendente(Planta[] plantas)	10%
Clase Planta	Constructor, getters y setters necesarios	5%
	public double obtenerPrecioFinal()	5%
	public String toString()	5%
Total		100%