

Universidad Mariano Gálvez de Guatemala
Boca del Monte

Ingeniería en Sistemas. Ciclo IV
Jornada Sábado.

PROGRAMACIÓN II
JONATHAN JOSUE GARCIA CUQUE



Nombre: Luis Fernando Lima Ixcuná
Carné: 7690-20-17409

Problema #1

Paquete e Importaciones:

El código pertenece al paquete problema1.

Importa la clase Scanner para leer la entrada del usuario desde la consola.

Clase Principal:

Define la clase pública problema1.

Método main:

Define el método main, que es el punto de entrada del programa.

Inicialización del Scanner:

Crea una instancia de Scanner para leer la entrada del usuario desde la consola.

Lectura de Datos de la Primera Persona:

Solicita y lee el nombre de la primera persona.

Solicita y lee el año de nacimiento de la primera persona.

Consume la nueva línea pendiente después de leer un entero.

Lectura de Datos de la Segunda Persona:

Solicita y lee el nombre de la segunda persona.

Solicita y lee el año de nacimiento de la segunda persona.

Cálculo de la Diferencia de Edad:

Calcula la diferencia absoluta entre los años de nacimiento de las dos personas.

Comparación de Edades y Salida:

Compara los años de nacimiento para determinar quién es mayor y muestra el resultado correspondiente.

Si los años de nacimiento son iguales, indica que tienen la misma edad.

Cierre del Scanner:

Cierra el Scanner para liberar los recursos asociados.

Problema #2

Paquete e Importaciones:

El código pertenece al paquete problema2.

Importa las clases ArrayList y Scanner de la biblioteca estándar de Java, aunque no se utilizan en el fragmento proporcionado.

Clase Libro:

Define la clase pública Libro.

Atributos Privados:

titulo: Almacena el título del libro.

autor: Almacena el autor del libro.

disponible: Indica si el libro está disponible para préstamo (booleano).

Constructor:

public Libro(String titulo, String autor): Inicializa un nuevo objeto Libro con un título y un autor.

Verifica que el título no sea nulo ni vacío.

Verifica que el autor no sea nulo ni vacío.

Inicializa el atributo disponible a true.

Método prestar:

public void prestar(): Marca el libro como prestado si está disponible.

Si el libro está disponible, cambia disponible a false y muestra un mensaje indicando que el libro ha sido prestado.

Si el libro no está disponible, lanza una excepción IllegalStateException.

Método devolver:

public void devolver(): Marca el libro como disponible.

Cambia disponible a true y muestra un mensaje indicando que el libro ha sido devuelto.

Método consultar_disponibilidad:

public void consultar_disponibilidad(): Consulta y muestra la disponibilidad del libro.

Si el libro está disponible, muestra un mensaje indicando que el libro está disponible para préstamo.

Si el libro no está disponible, muestra un mensaje indicando que el libro no está disponible para préstamo.

Método getTitulo:

public String getTitulo(): Devuelve el título del libro.

Problema #3

Método para Vender Cantidad

El método verifica si la cantidad a vender es válida. Si la cantidad es mayor que 0 y menor o igual a la cantidad disponible en el inventario, se reduce la cantidad en el inventario y se imprime un mensaje de confirmación. Si la cantidad a vender es mayor que la cantidad disponible, se imprime un mensaje indicando que no hay suficiente inventario. Si la cantidad a vender es menor o igual a 0, se imprime un mensaje indicando que la cantidad debe ser mayor que cero.

Clase Principal

La clase principal contiene el método main, que es el punto de entrada del programa. Se inicializa un objeto Scanner para leer la entrada del usuario y una lista de productos que representa el inventario.

Bucle Principal

El programa entra en un bucle infinito que muestra un menú con diferentes opciones para el usuario: consultar el inventario, agregar cantidad, vender cantidad, registrar un nuevo producto y salir del programa. El usuario selecciona una opción ingresando un número.

Opciones del Menú

Consultar Inventario: Recorre la lista de productos y llama a un método para consultar el inventario de cada producto.

Agregar Cantidad: Solicita al usuario el nombre del producto y la cantidad a agregar. Si el producto se encuentra en el inventario, se agrega la cantidad especificada; de lo contrario, se muestra un mensaje de error.

Vender Cantidad: Solicita al usuario el nombre del producto y la cantidad a vender. Si el producto se encuentra en el inventario, se procede a vender la cantidad especificada; de lo contrario, se muestra un mensaje de error.

Este código es parte de un sistema de gestión de inventario básico que permite consultar, agregar y vender productos, así como registrar nuevos productos y salir del programa.

Problema #4

Método actualizarEmpleado

Este método permite actualizar la información de un empleado existente.

Solicitar ID del Empleado:

Se solicita al usuario que ingrese el ID del empleado que desea actualizar.

Se lee el ID ingresado y se busca el empleado correspondiente en la lista de empleados.

Buscar Empleado:

Se llama al método buscarEmpleadoPorId para encontrar el empleado con el ID especificado.

Si el empleado es encontrado, se procede a actualizar su información; de lo contrario, se muestra un mensaje indicando que el empleado no fue encontrado.

Actualizar Información del Empleado:

Empleado por Horas:

Si el empleado es una instancia de EmpleadoPorHoras, se solicita al usuario que ingrese el nuevo nombre, las nuevas horas trabajadas y la nueva tarifa por hora.

Se actualiza la información del empleado con los nuevos datos ingresados.

Empleado Asalariado:

Si el empleado es una instancia de EmpleadoAsalariado, se solicita al usuario que ingrese el nuevo nombre, el nuevo salario base, las nuevas comisiones y los nuevos bonos.

Se actualiza la información del empleado con los nuevos datos ingresados.

Confirmación de Actualización:

Se imprime un mensaje confirmando que la información del empleado ha sido actualizada exitosamente.

Método buscarEmpleadoPorId

Este método busca un empleado en la lista de empleados utilizando su ID.

Recorrer Lista de Empleados:

Se recorre la lista de empleados.

Si se encuentra un empleado con el ID especificado, se retorna dicho empleado.

Empleado No Encontrado:

Si no se encuentra ningún empleado con el ID especificado, se retorna null.