

PROYECTO 1 EJERCICIO N° 1

FECHA DE ENTREGA AL ESTUDIANTE: del 22 al 26 de marzo

FECHA DE ENTREGA AL DOCENTE: 16 de abril

PUNTAJE TOTAL: 100 Puntos

PORCENTAJE: Corresponde a un 20% de la nota final del curso

CONDICIONES DEL PROYECTO

El Proyecto de Aprendizaje del curso EIF204 Programación 2 pretende que el estudiante ponga en práctica los siguientes temas:

- Principios de Diseño
- Jerarquías de Clases
- Relaciones entre Clases

Se presenta un problema que el estudiante debe implementar en el IDE seleccionado por el profesor para desarrollar un proyecto en C++, y posteriormente subirlo al aula virtual en la sección que el profesor habilite para esto. El ejercicio se trabaja en grupos. Cualquier plagio se calificará con 0, como lo establece el artículo 24 del Reglamento General sobre los procesos de Enseñanza y Aprendizaje de la Universidad Nacional.

El código del programa debe compilar correctamente. Si el código no puede compilarse, el trabajo será calificado con una nota de 0 (cero). Para ser evaluado, tiene que cumplirse al menos un 50% de la funcionalidad solicitada. De lo contrario, el proyecto será considerado como no entregado y se penalizará la nota correspondiente

TEMAS EVALUADOS:

1. Mejores prácticas de programación
2. Herencia
3. Polimorfismo
4. Relaciones entre clases

ENUNCIADO DEL PROBLEMA

La empresa **Souvenirs de Costa Rica para el Mundo** (*SCORIM*) ha contratado a la empresa para la cual usted trabaja para que se le desarrolle un aplicativo con el que se administrarán las cajas registradoras de las tiendas de souvenirs que tienen distribuidas por todo el país incluyendo aeropuertos y estaciones de servicio.

La tienda además de vender souvenirs de todo tipo provee el servicio de personalización de estos, con los cuales pueden realizar impresiones especiales en tela, grabados en metal y transferencia de imágenes a cristalería, entre otros.

La aplicación debe contar con un menú para venta que se utiliza día a día, así como un menú para el administrador del punto que se encarga de configurar las reglas del negocio.

- Opción de administración: esta es solamente accesible por el administrador de la máquina por lo que este menú en particular permite:
 - Trabajar directamente con los productos con funciones tales como: insertar, modificar, borrar información.
 - Trabajar con el dinero registrado en el sistema de la caja registradora: consultar el saldo, agregar/retirar dinero de la caja.
- Opción de venta: esta opción del menú es ya para cuando se atiende al público en general por lo que solamente permite realizar compras y devolver el vuelto.

Clases

Clase Producto (10 puntos)

Esta clase es una generalidad que agrupa cualquier tipo de producto con cualquier tipo de comportamiento, por lo que en su estructura tiene métodos virtuales puros, que las clases derivadas deberán desarrollar.

Clase Artículo (5 puntos)

- A los productos: se le registra la fecha inicial de puesta en venta, si dicha fecha indica que el producto tiene más de 90 días de haber sido puesto en venta, se les aplica un 50 % de descuento.

Clase Servicio (5 puntos)

- A los servicios: se les puede aplicar un descuento que se establece por parte del administrador en el menú de administración.

Clase IAdministrador (5 puntos)

Esta clase es una interfaz, la cual tendrá los métodos que permiten administrar como tal la máquina. Esta clase es una interfaz y todos sus métodos virtuales puros.

Clase IVendedor (5 puntos)

Esta clase es una interfaz, la cual tendrá los métodos que permiten al usuario realizar la compra. Esta clase es una interfaz y todos sus métodos virtuales puros.

Clase Inventario (20 puntos)

Esta clase se adapta con facilidad al escenario donde los productos cambian, considerando que su dependencia con la clase producto es una dependencia con una abstracción, por lo que le permite a esta clase adaptarse a cualquier tipo de especialización de la interfaz producto. Es importante notar que esta clase tiene herencia múltiple ya que debe implementar las dos interfaces IAdministrador e IVendedor.

Clase Caja (10 puntos)

La clase caja se encarga de manejar el dinero de la máquina. Dentro de esta clase existe un método que se encarga de calcular el vuelto a ser entregado al usuario cuando este cancela (*desgloseVuelto*) este método debe seguir el algoritmo voraz para devolver la menor cantidad de monedas cuando se calcula el vuelto. Este método devuelve en texto el detalle completo del desglose.

Los productos tienen precios cerrados con valores definidos entre 1000, 3000, 7000. El cliente normalmente paga con billetes de 20000, 10000 o 5000. Por lo que la máquina debe calcular el desglose en monedas de 100, monedas de 500, billetes de 1000 y billetes de 2000 cuál es la mejor forma de entregar el vuelto al cliente, este detalle es lo que debe devolver el método.

Clase MenuPrincipal

Esta clase tiene dos características de tipo MenuAdministrador y MenuVenta. Esta clase debe velar por la existencia de todos los menús de la aplicación, para mostrar adecuadamente al usuario y capturar las selecciones hechas por el usuario, así como de manejar su interacción correspondiente. Recuerde que el menú debe ser un ciclo sin fin hasta que le usuario decida salir. El comportamiento es similar para los submenús de administrador y cobro.

Clase MenuAdministrador (10 puntos)

Este menú debe contar con las administraciones básicas de la máquina, tales como insertar, modificar, borrar información, ingresar dinero y retirar dinero. Este menú requiere ingresar la clave “1234” para poder acceder a sus opciones.

Clase MenuVenta (10 puntos)

Las opciones del menú son:

1. Mostrar Producto: aquí se muestra en pantalla la lista de productos y su respectiva posición.
2. Seleccionar Producto: aquí el usuario selecciona la posición del producto escogido. Esta opción muestra la cantidad disponible y el precio del producto que el usuario desea. Y se le consulta al usuario si desea realizar la compra o cancelar:
 - a. Realizar la compra: Si el cliente desea continuar con la compra, la máquina ingresa a un sub-menú que le indica que debe introducir el dinero, capturando vía pantalla el monto recibido. De nuevo el sistema consulta si desea continuar o cancelar. En el caso que haya continuado, el sistema recibe el dinero, corrobora que sea suficiente, si no es suficiente le informa al usuario el dinero faltante y se cancela la operación. Si el dinero es suficiente, se entrega el producto, lo que muestra en pantalla el nombre del producto vendido y el vuelto entregado, reduce el inventario del producto, y aumenta el dinero recolectado en la billetera de la máquina.
 - i. Tome en cuenta que es justo en este paso donde la venta se concreta que el sistema debe informar al usuario del desglose de su vuelto. Recuerde que La clase Máquina está compuesta por una Caja que a su vez internamente tiene el método *desgloceVuelto*.
3. Salir

Requerimientos adicionales de diseño

El arquitecto de software a cargo del proyecto le ha pedido además que cumpla con los siguientes principios de diseño:

- Responsabilidad Única: **(5 puntos)**

Argumente cómo se respeta el principio de Responsabilidad Única de acuerdo con el problema planteado. Debe incluir su argumento detallado como comentario dentro del código fuente en la sección o secciones donde se cumple con este principio.

- Inversión de Dependencias: **(5 puntos)**

Argumente cómo se respeta el principio de Inversión de Dependencias de acuerdo con el problema planteado. Debe incluir su argumento detallado como comentario dentro del código fuente en la sección o secciones donde se cumple con este principio.

- Abierto/Cerrado: **(5 puntos)**

Argumente cómo se respeta el principio de Abierto Cerrado de acuerdo con el problema planteado. Debe incluir su argumento detallado como comentario dentro del código fuente en la sección o secciones donde se cumple con este principio.

- Segregación de Interfaces **(5 puntos)**

Argumente cómo se respeta el principio de Segregación de Interfaces de acuerdo con el problema planteado. Debe incluir su argumento detallado como comentario dentro del código fuente en la sección o secciones donde se cumple con este principio.

Diagrama UML del Proyecto

