

UNIVERSIDAD MADERO PUEBLA



Seleccionadora y control de cortinas para un invernadero.

Alumno: Emigdio Espinosa Jasso.

Num:7979763883.

Matricula: 20086534.

Carrera: Ingeniería de Software.

Materia: Patrones de diseño.

Maestro: Nicolás Arrioja Landa Cosio.

**Introducción.**

Este es un proyecto de la materia de patrones de diseño, que consiste en crear un programa aplicando patrones de diseño de software.

El proyecto va enfocado en ayudar a los productores de jitomate, aunque podría servir para otras hortalizas.

Elegí hacer este proyecto, ya que de esta manera podríamos ayudarles a muchos productores de jitomate a pagar nóminas, una manera de sacar costos para la construcción de nuevos invernaderos, y para vender su jitomate de una manera un poco más automatizada. Otro de los motivos por el cual elegí hacer este proyecto es porque mi familia es productora de jitomate, y nos damos cuenta de lo importante y necesario que es esta aplicación para nosotros y otros productores.

**Antecedentes**

Para saber cómo funcionan los precios dentro de la venta de jitomate, los precios de construir un invernadero, le hicimos una entrevista a un ing. Agrícola, el ing. Emigdio Espinosa Cruz, el cual nos dio información muy importante para poder hacer este proyecto, al igual, yo he crecido con la enseñanza de los invernaderos, ya que desde pequeño mis papás se dedicaron a eso.

**Descripción del problema.**

Cuando se vende el jitomate, a veces las personas que lo venden tienen un poco de conflicto en saber los precios de venta, ya que todos los precios derivan del precio de la caja XL del jitomate color rojo.

Otro de los problemas que tienen los productores de jitomate en invernadero es estimar costos de construcción de estos, ya que el precio varía por la superficie del invernadero, así como por cosas que se le pueden agregar a este, como sistemas de riego etc.

No tiene una forma segura de guardar la información de la nómina, así que con el programa guardara esa información en un proxy.

**Objetivo general.**

Ayudar a los productores a vender su jitomate de una forma más rápida.

Ver los costos de construcción de invernaderos.

Tener guardada la información de nómina en un lugar seguro

**Objetivo específico.**

Para la venta de jitomates ocupé el patrón de diseño Estrategia, con el cual puse varios métodos para generar los precios de forma automática, metiéndole solo un valor.

Para sacar los costos de construcción utilice el patrón de diseño Decorador, en el que se creaba un invernadero de tres posibles, y se le aumentaban componentes, con los que aumentaba el precio, todo se calculaba con distintos métodos y operaciones, todo dependía igual de la superficie a construir.

Para guardar la información de nómina utilice el patrón de diseño proxy, de esta forma se guardará de una forma segura esta información, y pedirá una contraseña para poder ver la información.

**Alcances.**

Podemos hacer procesos de ventas muy rápidos y efectivos, así como escatimar costos de construcción.

**Limitaciones.**

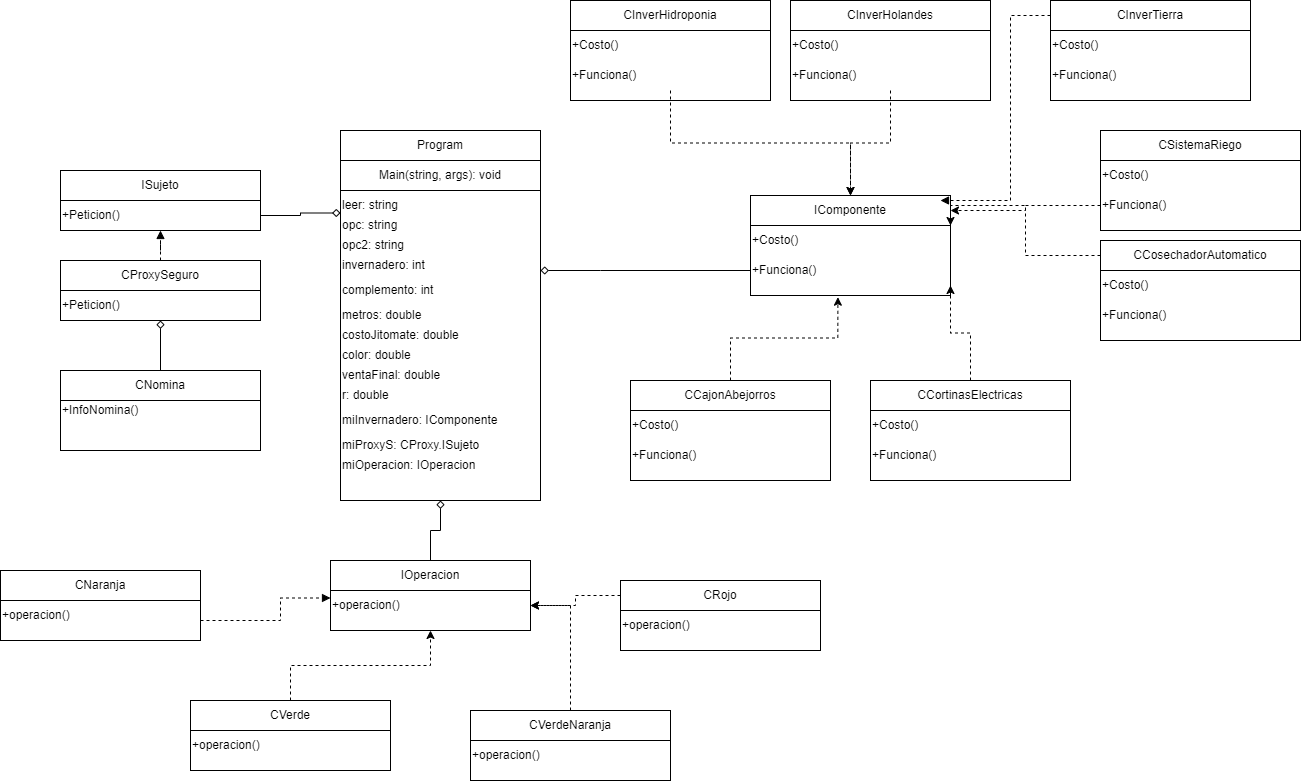
Es inflexible a la hora de manejar los precios, ya que a veces el cliente tratara de regatear, y el programa no puede saber eso.

Para la construcción de los invernaderos, el programa no puede saber si hay perdida de material, o si se descompone algo por lo que se tendría que cambiar, lo que no haría super definido un precio final.

**Bibliografía.**

*“Emigdio Espinosa Cruz (Ing. Agrónomo) (2022, 8 septiembre) Entrevista personal – Entrevista acerca de los costos del jitomate, así como de el costo de construcción de un invernadero.*

**Diagrama.**



**Código.**

using System;

namespace Proyecto1erParcial

{

///Se encarga de correr todo el programa

///Autor: Emigdio Espinosa Jasso

///Fecha: 14-09-2022

///Versión: 1.0

class Program

{

static void Main(string[] args)

{

//variables globales

string opc = "";

string leer = "";

//Interfaz de usuario

while (opc != "4")

{

Console.WriteLine("---------ROJOS-----------\n\r" +

"Bienvenido a la interfaz de usuario para la empresa ROJOS.\n\r" +

"Cuál es la acción que le gustaria realizar?\n\r" +

"1. Venta de jitomate \n\r" +

"2. Construir un nuevo invernadero \n\r" +

"3. Información nomina\n\r" +

"4. Salir");

opc = Console.ReadLine();

//1. Venta de jitomate //Patrón estrategia

if (opc == "1")

{

//Variables para guardar el numero de cajas de jitomates de los distintos tamaños

double j = 0;

double xl = 0;

double l = 0;

double m = 0;

double s = 0;

//Variable para mandar el numero de cajas a las clases

double r = 0;

//Variables extras que serviran para el buen funcionamiento.

double costoJitomate = 0;

string fecha = DateTime.Now.Date.ToString();

double color = 0;

string salir = "";

double ventaFinal = 0;

//Nuestra variable que referencia al algoritmo

//Instanciamos con un default para permitir compilacion correcta

IOperacion miOperacion = new CVerde();

//Solo preguntamos el precio del jitomate rojo, ya que de ese se sacan los demas precios

Console.WriteLine("\n\r Bienvenido al area de venta de ROJOS" + "\n\rCuál es el precio de la caja de jitomate rojo en la central de abastos al dia de hoy {0}?", fecha);

leer = Console.ReadLine();

costoJitomate = Convert.ToDouble(leer);

while (salir != "2")

{

//El precio del jitomate cambia dependiendo que color se venda, pero todo en base al precio de la caja de jitomate XL del color rojo

Console.WriteLine("Qué color se va a vender?\n\r" +

"1. Verdes, 2. verde-naranja, 3. Naranjas, 4. Rojos ");

leer = Console.ReadLine();

color = Convert.ToInt32(leer);

Console.WriteLine("\n\rTamaños. Cuántas cajas tienes de?:" +

"Jumbo:");

leer = Console.ReadLine();

j = Convert.ToDouble(leer);

Console.WriteLine("\n\rXL:");

leer = Console.ReadLine();

xl = Convert.ToDouble(leer);

Console.WriteLine("\n\rL:");

leer = Console.ReadLine();

l = Convert.ToDouble(leer);

Console.WriteLine("\n\rM:");

leer = Console.ReadLine();

m = Convert.ToDouble(leer);

Console.WriteLine("\n\rSmall:");

leer = Console.ReadLine();

s = Convert.ToDouble(leer);

//Verdes

if (color == 1)

{

Console.WriteLine("El precio de la caja de jitomate XL verde es de{0}", costoJitomate / 2);

miOperacion = new CVerde();

r = miOperacion.operacion(j, xl, l, m, s, costoJitomate);

Console.WriteLine("Se vendio un total de {0} pesos de jitomate verde", r);

}

//verdeNaranja

if (color == 2)

{

Console.WriteLine("El precio de la caja de jitomate XL verde-naranja es de{0}", costoJitomate - 20);

miOperacion = new CVerdeNaranja();

r = miOperacion.operacion(j, xl, l, m, s, costoJitomate);

Console.WriteLine("Se vendio un total de {0} pesos de jitomate verde-naranja", r);

}

//Naranjas

if (color == 3)

{

Console.WriteLine("El precio de la caja de jitomate XL naranja es de{0}", costoJitomate + 20);

miOperacion = new CNaranja();

r = miOperacion.operacion(j, xl, l, m, s, costoJitomate);

Console.WriteLine("Se vendio un total de {0} pesos de jitomate naranja", r);

}

//Rojos

if (color == 4)

{

Console.WriteLine("El precio de la caja de jitomate XL rojo es el valor que usted nos proporciono, {0}", costoJitomate);

miOperacion = new CRojo();

r = miOperacion.operacion(j, xl, l, m, s, costoJitomate);

Console.WriteLine("Se vendio un total de {0} pesos de jitomate rojo", r);

}

ventaFinal += r;

Console.WriteLine("Deseas vender mas jitomates de otro color? 1.Si, 2.No");

salir = Console.ReadLine();

}

Console.WriteLine("\n\r Se obtuvieron {0} de ventas totales\n\r", ventaFinal);

}

//2. Construir un nuevo invernadero //DECORADOR

if (opc == "2")

{

string opc2 = "";

int invernadero = 0;

int complemento = 0;

double metros = 0;

//Variable IComponente para guardar el invernadero seleccionado.

IComponente miInvernadero = new CInverTierra("Tierra", 0);

while (invernadero != 4)

{

//Preguntamos a usuario que tipo de invernadero quiere

Console.WriteLine("Qué tipo de invernadero le gustaria construir? \n\r1.Invernadero de Tierra. \n\r2.Invernadero Hidroponico. \n\r3.Invernadero Holandes \n\r4. Salir");

opc = Console.ReadLine();

invernadero = Convert.ToInt32(opc);

if (invernadero == 4)

break;

//Preguntamos cuantos metros cuadrados va a construir

Console.WriteLine("Cuántos metros cuadrados tendra el invernadero?");

leer = Console.ReadLine();

metros = Convert.ToDouble(leer);

if (invernadero == 4)

break;

if (invernadero == 1)

{

//Le agregamos a miInvernadero el invernadero de tierra

miInvernadero = new CInverTierra("Invernadero de tierra", 30 \* metros);

Console.WriteLine(miInvernadero);

}

if (invernadero == 2)

{

//Le agregamos a miInvernadero el invernadero de tierra

miInvernadero = new CInverHidroponia("Invernadero hidroponico", 50 \* metros);

Console.WriteLine(miInvernadero);

}

if (invernadero == 3)

{

//Le agregamos a miInvernadero el invernadero de tierra

miInvernadero = new CInverHolandes("Invernadero holandes", 70 \* metros);

Console.WriteLine(miInvernadero);

}

//Preguntamos si quiere extras en su invernadero

while (complemento != 5)

{

Console.WriteLine("que cosas le gustaria agregar extra a su invernadero, 1.Cortinas electricas, \n\r2.Sistema automatico de riego, \n\r3.Cajon de abejorros, \n\r4.Robot cosechador,\n\r5.Salir");

opc2 = Console.ReadLine();

complemento = Convert.ToInt32(opc2);

if (complemento == 1)

{

//Decoramos el invernadero con cortinas electricas

miInvernadero = new CCortinasElectricas(miInvernadero, metros);

}

if (complemento == 2)

{

//Decoramos el invernadero con el sistema automatico de riego

miInvernadero = new CSistemaRiego(miInvernadero, metros);

}

if (complemento == 3)

{

//Decoramos el invernadero con el cajon de abejorros

miInvernadero = new CCajonAbejorros(miInvernadero);

}

if (complemento == 4)

{

//Decoramos el invernadero con un robot cosechador

miInvernadero = new CCosechadorAutomatico(miInvernadero);

}

}

}

Console.WriteLine(miInvernadero.Funciona());

Console.WriteLine("El costo total de construcción, ya con todos los componentes extra es de: " + miInvernadero.Costo());

//Console.WriteLine(miInvernadero);

}

//3. Pagar nomina // PROXY

if (opc == "3")

{

CProxy.ISujeto miProxyS = new CProxy.ProxySeguro();

miProxyS.Peticion(1);

Console.WriteLine("------");

}

}

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Proyecto1erParcial

{

///Interfase IOperacion

///Autor: Emigdio Espinosa Jasso

///Fecha: 14-09-2022

///Versión: 1.0

interface IOperacion

{

double operacion(double j, double xl, double l, double m, double s, double p);

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Proyecto1erParcial

{

interface IComponente

{

double Costo();

string Funciona();

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Proyecto1erParcial

{

///Clase CVerdeNaranja

///Implementa IOperacion

///Autor: Emigdio Espinosa Jasso

///Fecha: 14-09-2022

///Versión: 1.0

class CVerdeNaranja : IOperacion

{

///Autor: Emigdio Espinosa Jasso

///Fecha: 14-09-2022

///Versión: 1.0

/// <summary>

/// Metodo que calcula el precio de los distintos tamaños de jitomate del color verde-naranja

/// </summary>

/// <param name="j"></param>

/// <param name="xl"></param>

/// <param name="l"></param>

/// <param name="m"></param>

/// <param name="s"></param>

/// <param name="p"></param>

/// <returns></returns>

public double operacion(double j, double xl, double l, double m, double s, double p)

{

//Variable para calcular el precio del jitomate verde-naranja según el precio del rojo.

double costo = p - 20;

//Variable para retornar el resultado.

double resultado = (j \* (costo + 20)) + (xl \* costo) + (l \* (costo - 20)) + (m \* (costo - 40)) + (s \* (costo - 60));

return resultado;

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Proyecto1erParcial

{

///Clase CVerdeNaranja

///Implementa IOperacion

///Autor: Emigdio Espinosa Jasso

///Fecha: 14-09-2022

///Versión: 1.0

class CVerdeNaranja : IOperacion

{

///Autor: Emigdio Espinosa Jasso

///Fecha: 14-09-2022

///Versión: 1.0

/// <summary>

/// Metodo que calcula el precio de los distintos tamaños de jitomate del color verde-naranja

/// </summary>

/// <param name="j"></param>

/// <param name="xl"></param>

/// <param name="l"></param>

/// <param name="m"></param>

/// <param name="s"></param>

/// <param name="p"></param>

/// <returns></returns>

public double operacion(double j, double xl, double l, double m, double s, double p)

{

//Variable para calcular el precio del jitomate verde-naranja según el precio del rojo.

double costo = p - 20;

//Variable para retornar el resultado.

double resultado = (j \* (costo + 20)) + (xl \* costo) + (l \* (costo - 20)) + (m \* (costo - 40)) + (s \* (costo - 60));

return resultado;

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Proyecto1erParcial

{

class CSistemaRiego : IComponente

{

//Variable de referencia hacia quien decoramos

private IComponente decoramosA;

private double metros;

//Pasamos al objeto que va a ser decorado

public CSistemaRiego(IComponente pComponente, double pMetros)

{

decoramosA = pComponente;

metros = pMetros;

}

public override string ToString()

{

return "Sistema de riego automatico \r\n" + decoramosA.ToString();

}

//Metodos de la interfaz

public double Costo()

{

//Calculamos el costo

//El costo de lo que decoro mas mi costo como componente

return decoramosA.Costo() + (6 \* metros);

}

public string Funciona()

{

return decoramosA.Funciona() + "\n\rSe agrego un sistema de riego automatico, lo que hace mas eficientes los riegos";

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Proyecto1erParcial

{

///Clase CRojo

///Implementa IOperacion

///Autor: Emigdio Espinosa Jasso

///Fecha: 14-09-2022

///Versión: 1.0

class CRojo : IOperacion

{

///Autor: Emigdio Espinosa Jasso

///Fecha: 14-09-2022

///Versión: 1.0

/// <summary>

/// Metodo que calcula el precio de los distintos tamaños de jitomate del color naranja

/// </summary>

/// <param name="j"></param>

/// <param name="xl"></param>

/// <param name="l"></param>

/// <param name="m"></param>

/// <param name="s"></param>

/// <param name="p"></param>

/// <returns></returns>

public double operacion(double j, double xl, double l, double m, double s, double p)

{

//Variable para calcular el precio del jitomate verde-naranja según el precio del rojo.

double costo = p;

//Variable para retornar el resultado.

double resultado = (j \* (costo + 20)) + (xl \* costo) + (l \* (costo - 20)) + (m \* (costo - 40)) + (s \* (costo - 60));

return resultado;

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Proyecto1erParcial

{

///Clase CInverTierra

///Implementa IComponente

///Autor: Emigdio Espinosa Jasso

///Fecha: 14-09-2022

///Versión: 1.0

public class CProxy

{

///Interface ISujeto

///Implementa IComponente

///Autor: Emigdio Espinosa Jasso

///Fecha: 14-09-2022

///Versión: 1.0

public interface ISujeto

{

void Peticion(int pOpcion);

}

///Clase CProxySeguro

///Implementa IComponente

///Autor: Emigdio Espinosa Jasso

///Fecha: 14-09-2022

///Versión: 1.0

public class ProxySeguro : ISujeto

{

private CNomina nomina;

///Autor: Emigdio Espinosa Jasso

///Fecha: 14-09-2022

///Versión: 1.0

/// <summary>

/// Metodo de interfaz que implementamos

/// </summary>

/// <param name="pOpcion"></param>

public void Peticion(int pOpcion)

{

string password;

Console.WriteLine("Dame el password");

password = Console.ReadLine();

if (password == "ROJOS")

{

if (nomina == null)

{

Console.WriteLine("Activando el sujeto");

nomina = new CNomina();

}

if (pOpcion == 1)

nomina.InfoNomina();

}

else

{

Console.WriteLine("Acceso denegado");

}

}

}

///Clase CNomina

///Implementa IComponente

///Autor: Emigdio Espinosa Jasso

///Fecha: 14-09-2022

///Versión: 1.0

private class CNomina

{

///Autor: Emigdio Espinosa Jasso

///Fecha: 14-09-2022

///Versión: 1.0

/// <summary>

/// Metodo que guarda el nombre de los empleados y cuanto es su salario.

/// </summary>

public void InfoNomina()

{

Console.WriteLine("Cristobal: 3000");

Console.WriteLine("Angel: 2500");

Console.WriteLine("Omar: 2400");

Console.WriteLine("Armando: 4000");

Console.WriteLine("Nancy: 3500");

Console.WriteLine("Ricardo: 4000");

Console.WriteLine("Daniel: 3500");

Console.WriteLine("Dinero en caja: 40,000");

}

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Proyecto1erParcial

{

///Clase CNaranja

///Implementa IOperacion

///Autor: Emigdio Espinosa Jasso

///Fecha: 14-09-2022

///Versión: 1.0

class CNaranja : IOperacion

{

///Autor: Emigdio Espinosa Jasso

///Fecha: 14-09-2022

///Versión: 1.0

/// <summary>

/// Metodo que calcula el precio de los distintos tamaños de jitomate del color naranja

/// </summary>

/// <param name="j"></param>

/// <param name="xl"></param>

/// <param name="l"></param>

/// <param name="m"></param>

/// <param name="s"></param>

/// <param name="p"></param>

/// <returns></returns>

public double operacion(double j, double xl, double l, double m, double s, double p)

{

//Variable para calcular el precio del jitomate naranja según el precio del rojo.

double costo = p + 20;

//Variable para retornar el resultado.

double resultado = (j \* (costo + 20)) + (xl \* costo) + (l \* (costo - 20)) + (m \* (costo - 40)) + (s \* (costo - 60));

return resultado;

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Proyecto1erParcial

{

///Clase CInverTierra

///Implementa IComponente

///Autor: Emigdio Espinosa Jasso

///Fecha: 14-09-2022

///Versión: 1.0

class CInverTierra : IComponente

{

private string nombre;

private double costo;

///Autor: Emigdio Espinosa Jasso

///Fecha: 14-09-2022

///Versión: 1.0

/// <summary>

/// Este es el metodo constructor de la clase CInverTierra

/// </summary>

/// <param name="pNombre"></param>

/// <param name="pCosto"></param>

public CInverTierra(string pNombre, double pCosto)

{

nombre = pNombre;

costo = pCosto;

}

///Autor: Emigdio Espinosa Jasso

///Fecha: 14-09-2022

///Versión: 1.0

/// <summary>

/// Metodo para sobreesvribir mi string

/// </summary>

/// <returns></returns>

public override string ToString()

{

return string.Format("Invernadero: {0} \r\n", nombre);

}

///Autor: Emigdio Espinosa Jasso

///Fecha: 14-09-2022

///Versión: 1.0

/// <summary>

/// Metodo de interfaz que implementamos

/// </summary>

/// <returns></returns>

public double Costo()

{

return costo;

}

///Autor: Emigdio Espinosa Jasso

///Fecha: 14-09-2022

///Versión: 1.0

/// <summary>

/// Metodo de interfaz que implementamos

/// </summary>

/// <returns></returns>

public string Funciona()

{

return "Se construyo un invernadero de tierra";

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Proyecto1erParcial

{

///Clase CInverHolandes

///Implementa IComponente

///Autor: Emigdio Espinosa Jasso

///Fecha: 14-09-2022

///Versión: 1.0

class CInverHolandes : IComponente

{

private string nombre;

private double costo;

///Autor: Emigdio Espinosa Jasso

///Fecha: 14-09-2022

///Versión: 1.0

/// <summary>

/// Este es el metodo constructor de la clase CInverHolandes

/// </summary>

/// <param name="pNombre"></param>

/// <param name="pCosto"></param>

public CInverHolandes(string pNombre, double pCosto)

{

nombre = pNombre;

costo = pCosto;

}

///Autor: Emigdio Espinosa Jasso

///Fecha: 14-09-2022

///Versión: 1.0

/// <summary>

/// Metodo para sobreesvribir mi string

/// </summary>

/// <returns></returns>

public override string ToString()

{

return string.Format("Invernadero: {0} \n\r", nombre);

}

///Autor: Emigdio Espinosa Jasso

///Fecha: 14-09-2022

///Versión: 1.0

/// <summary>

/// Metodo de interfaz que implementamos

/// </summary>

/// <returns></returns>

public double Costo()

{

return costo;

}

///Autor: Emigdio Espinosa Jasso

///Fecha: 14-09-2022

///Versión: 1.0

/// <summary>

/// Metodo de interfaz que implementamos

/// </summary>

/// <returns></returns>

public string Funciona()

{

return "Se construyo un invernadero holandes";

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Proyecto1erParcial

{

///Clase CInverHidroponia

///Implementa IComponente

///Autor: Emigdio Espinosa Jasso

///Fecha: 14-09-2022

///Versión: 1.0

class CInverHidroponia : IComponente

{

private string nombre;

private double costo;

///Autor: Emigdio Espinosa Jasso

///Fecha: 14-09-2022

///Versión: 1.0

/// <summary>

/// Este es el metodo constructor de la clase CInverHidroponia

/// </summary>

/// <param name="pNombre"></param>

/// <param name="pCosto"></param>

public CInverHidroponia(string pNombre, double pCosto)

{

nombre = pNombre;

costo = pCosto;

}

///Autor: Emigdio Espinosa Jasso

///Fecha: 14-09-2022

///Versión: 1.0

/// <summary>

/// Metodo para sobreesvribir mi string

/// </summary>

/// <returns></returns>

public override string ToString()

{

return string.Format("Invernadero: {0} \n\r", nombre);

}

///Autor: Emigdio Espinosa Jasso

///Fecha: 14-09-2022

///Versión: 1.0

/// <summary>

/// Metodo de interfaz que implementamos

/// </summary>

/// <returns></returns>

public double Costo()

{

return costo;

}

///Autor: Emigdio Espinosa Jasso

///Fecha: 14-09-2022

///Versión: 1.0

/// <summary>

/// Metodo de interfaz que implementamos

/// </summary>

/// <returns></returns>

public string Funciona()

{

return "Se construyo un invernadero de hidroponia";

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Proyecto1erParcial

{

///Clase CCosechadorAutomatico

///Implementa IComponente

///Autor: Emigdio Espinosa Jasso

///Fecha: 14-09-2022

///Versión: 1.0

class CCosechadorAutomatico : IComponente

{

//Variable de referencia hacia quien decoramos

private IComponente decoramosA;

///Autor: Emigdio Espinosa Jasso

///Fecha: 14-09-2022

///Versión: 1.0

/// <summary>

/// Este es el metodo constructor de la clase CCosechadorAutomatico

/// </summary>

/// <param name="pComponente"></param>

public CCosechadorAutomatico(IComponente pComponente)

{

decoramosA = pComponente;

}

///Autor: Emigdio Espinosa Jasso

///Fecha: 14-09-2022

///Versión: 1.0

/// <summary>

/// Metodo para sobreesvribir mi string

/// </summary>

/// <returns></returns>

public override string ToString()

{

return "Cosechador automatico\r\n" + decoramosA.ToString();

}

///Autor: Emigdio Espinosa Jasso

///Fecha: 14-09-2022

///Versión: 1.0

/// <summary>

/// Metodo de interfaz que implementamos

/// </summary>

/// <returns></returns>

public double Costo()

{

//Calculamos el costo

//El costo de lo que decoro mas mi costo como componente

return decoramosA.Costo() + 100000;

}

///Autor: Emigdio Espinosa Jasso

///Fecha: 14-09-2022

///Versión: 1.0

/// <summary>

/// Metodo de interfaz que implementamos

/// </summary>

/// <returns></returns>

public string Funciona()

{

return decoramosA.Funciona() + "\n\rSe agrego un cosechador automatico, es muy caro, pero nos ahorraremos mucha mano de obra";

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Proyecto1erParcial

{

///Clase CCortinasElectricas

///Implementa IComponente

///Autor: Emigdio Espinosa Jasso

///Fecha: 14-09-2022

///Versión: 1.0

class CCortinasElectricas : IComponente

{

//Variable de referencia hacia quien decoramos

private IComponente decoramosA;

private double metros;

///Autor: Emigdio Espinosa Jasso

///Fecha: 14-09-2022

///Versión: 1.0

/// <summary>

/// Este es el metodo constructor de la clase CCortinasElectricas

/// </summary>

/// <param name="pComponente"></param>

public CCortinasElectricas(IComponente pComponente, double pMetros)

{

decoramosA = pComponente;

metros = pMetros;

}

///Autor: Emigdio Espinosa Jasso

///Fecha: 14-09-2022

///Versión: 1.0

/// <summary>

/// Metodo para sobreesvribir mi string

/// </summary>

/// <returns></returns>

public override string ToString()

{

return "Cortinas electricas \r\n" + decoramosA.ToString();

}

///Autor: Emigdio Espinosa Jasso

///Fecha: 14-09-2022

///Versión: 1.0

/// <summary>

/// Metodo de interfaz que implementamos

/// </summary>

/// <returns></returns>

public double Costo()

{

//Calculamos el costo

//El costo de lo que decoro mas mi costo como componente

return decoramosA.Costo() + (8 \* metros);

}

///Autor: Emigdio Espinosa Jasso

///Fecha: 14-09-2022

///Versión: 1.0

/// <summary>

/// Metodo de interfaz que implementamos

/// </summary>

/// <returns></returns>

public string Funciona()

{

return decoramosA.Funciona() + "\n\rSe instalaron Cortinas electricas para abrir y cerrar automaticamente el invernadero";

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Proyecto1erParcial

{

///Clase CCajonAbejorros

///Implementa IComponente

///Autor: Emigdio Espinosa Jasso

///Fecha: 14-09-2022

///Versión: 1.0

class CCajonAbejorros : IComponente

{

//Variable de referencia hacia quien decoramos

private IComponente decoramosA;

///Autor: Emigdio Espinosa Jasso

///Fecha: 14-09-2022

///Versión: 1.0

/// <summary>

/// Este es el metodo constructor de la clase CCortinasElectricas

/// </summary>

/// <param name="pComponente"></param>

public CCajonAbejorros(IComponente pComponente)

{

decoramosA = pComponente;

}

///Autor: Emigdio Espinosa Jasso

///Fecha: 14-09-2022

///Versión: 1.0

/// <summary>

/// Metodo para sobreesvribir mi string

/// </summary>

/// <returns></returns>

public override string ToString()

{

return "Cajón de abejorros\r\n" + decoramosA.ToString();

}

///Autor: Emigdio Espinosa Jasso

///Fecha: 14-09-2022

///Versión: 1.0

/// <summary>

/// Metodo de interfaz que implementamos

/// </summary>

/// <returns></returns>

public double Costo()

{

//Calculamos el costo

//El costo de lo que decoro mas mi costo como componente

return decoramosA.Costo() + 8000;

}

///Autor: Emigdio Espinosa Jasso

///Fecha: 14-09-2022

///Versión: 1.0

/// <summary>

/// Metodo de interfaz que implementamos

/// </summary>

/// <returns></returns>

public string Funciona()

{

return decoramosA.Funciona() + "\n\rSe agrego un cajón de abejorros al invernadero, lo que mejorara la polinización de las plantas";

}

}

}

**Corridas.**

