

**Nombre:**

Onasis Alexander Pereyra Jiménez

**Matrícula:**

2024-1676

**Día de clase:**

Viernes

1. Considera estás desarrollando un programa donde necesitas trabajar con objetos de tipo Persona. Define una clase Persona, pero en este caso considerando los siguientes atributos de clase: nombre (String), apellidos (String), edad (int), casado (boolean), numeroDocumentoIdentidad(String) y 3 metodos como acciones diferentes por persona de acuerdo a una profesión. Define un constructor y los métodos para poder establecer y obtener los valores de los atributos. Mínimo 7 personas diferentes con acciones diferentes.

using System.Reflection.Metadata;

namespace FirstTask

{

public class Program

{

static void Main(string[] args)

{

//1. Profesor de Historia:

Persona historyProfessor = new Persona("Carlos", "Almonte", 36, false, "890-44347-09-3");

historyProfessor.HistoryProfessorAction();

//2. Odontólogo:

Persona dentist = new Persona("Alejandro", "Hughes", 30, true, "110-41387-03-2");

dentist.DentistAction();

//3. Psicólogo

Persona psychologist = new Persona("Pedro", "Pereyra", 40, false, "120-51317-01-7");

psychologist.PsychologistAction();

//4. Ingeniero Mecánico

Persona mechanicalEng = new Persona("Jose", "Perez", 32, true, "190-13312-01-4");

mechanicalEng.MechanicalEngAction();

//5. Ingeniero Civil

Persona civilEng = new Persona("Alexander", "Jiménez", 27, false, "130-12812-01-1");

civilEng.CivilEngAction();

//6. Ingeniero Eléçtrico

Persona electricalEng = new Persona("Adrián", "Henríquez", 25, true, "900-18219-01-2");

electricalEng.ElectricalEngAction();

//7. Ingeniero Químico

Persona chemicalEng = new Persona("Gonzalo", "Aponte", 28, false, "490-16332-01-4");

chemicalEng.ChemicalEngAction();

Console.ReadKey();

}

}

public class Persona {

public string Name { get; set;}

public string Lastname { get; set; }

public int Age { get; set; }

public bool Married { get; set; }

public string IdentityDocumentNumber { get; set; }

public Persona (string name, string lastname, int age, bool married, string identityDocumentNumber)

{

this.Name = name;

this.Lastname = lastname;

this.Age = age;

this.Married = married;

this.IdentityDocumentNumber = identityDocumentNumber;

}

public void HistoryProfessorAction()

{

Console.WriteLine($"""

El Profesor de historia {Name} {Lastname} de {Age} años de edad

va a dar una introducción a la materia Historia Universal.

""");

}

public void DentistAction()

{

Console.WriteLine($"""

El Odóntologo {Name} {Lastname} de {Age} años de edad

va a chequearle una muela a un paciente.

""");

}

public void PsychologistAction()

{

Console.WriteLine($"""

El Psicólogo {Name} {Lastname} de {Age} años de edad

va a realizar una terapia familiar.

""");

}

public void MechanicalEngAction()

{

Console.WriteLine($"""

El Ingeniero Mecánico {Name} {Lastname} de {Age} años de edad

va a diseñar una pieza para un robot.

""");

}

public void CivilEngAction()

{

Console.WriteLine($"""

El Ingeniero Civil {Name} {Lastname} de {Age} años de edad

va a diseñar una carretera.

""");

}

public void ElectricalEngAction()

{

Console.WriteLine($"""

El Ingeniero Eléctrico {Name} {Lastname} de {Age} años de edad

va a evaluar unos equipos electrónicos.

""");

}

public void ChemicalEngAction()

{

Console.WriteLine($"""

El Ingeniero Químico {Name} {Lastname} de {Age} años de edad

va a producir nuevos fármacos.

""");

}

}

}

1. Crea una clase Cuenta con los métodos ingreso, reintegro y transferencia. La clase contendrá un constructor por defecto, un constructor con parámetros y los métodos getters y setters para mostrar e ingresar.

using System.Reflection.Metadata.Ecma335;

using System.Security.Cryptography.X509Certificates;

namespace SecondTask

{

public class Program

{

static void Main(string[] args)

{

try

{

Cuenta account = new Cuenta(0, 0, 0);

account.HacerIngreso();

account.HacerTransferencia();

account.Reintegro();

account.MostrarValores();

}

catch (FormatException)

{

Console.WriteLine("El formato que ha ingresado no es válido.");

}

Console.ReadKey();

}

public class Cuenta

{

public decimal Enter { get; set; }

public decimal Withdrawal { get; set; }

public decimal Transference { get; set; }

public int WishToMakeWithdrawalOrTransference{ get; set; }

public Cuenta()

{

Enter = 0;

}

public Cuenta(decimal withdrawal, decimal transference, int wishToMakeWithdrawalOrTransference)

{

this.Withdrawal = withdrawal;

this.Transference = transference;

this.WishToMakeWithdrawalOrTransference = wishToMakeWithdrawalOrTransference;

}

public void HacerIngreso()

{

Console.WriteLine("Por favor digite la cantidad que desea ingresar a su cuenta.");

Enter = Convert.ToDecimal(Console.ReadLine());

if (Enter == 0)

{

Console.WriteLine("Debe digitar una cantidad mayor que 0 para poder ingresar dinero a su cuenta.");

}

}

public void HacerTransferencia()

{

Console.WriteLine("¿Desea hacer una transferencia? 1.Sí 2.No");

WishToMakeWithdrawalOrTransference = Convert.ToInt32(Console.ReadLine());

if (WishToMakeWithdrawalOrTransference == 1)

{

Console.WriteLine("Por favor ingrese la cantidad que desea transferir:");

Transference = Convert.ToDecimal(Console.ReadLine());

}

else if (WishToMakeWithdrawalOrTransference > 2 || WishToMakeWithdrawalOrTransference == 0)

{

Console.WriteLine("Debe seleccionar la opción 1 si desea hacer una transferencia.");

}

}

public void Reintegro()

{

Console.WriteLine("¿Desea realizar un reintegro a su cuenta? 1.Sí 2.No");

WishToMakeWithdrawalOrTransference = Convert.ToInt32(Console.ReadLine());

if (WishToMakeWithdrawalOrTransference == 1)

{

Console.WriteLine("Por favor ingrese la cantidad que desea reintegrar:");

Withdrawal = Convert.ToDecimal(Console.ReadLine());

}

else if (WishToMakeWithdrawalOrTransference > 2 || WishToMakeWithdrawalOrTransference == 0)

{

Console.WriteLine("Debe seleccionar la opción 1 si desea hacer un reintegro.");

}

}

public void MostrarValores()

{

Console.WriteLine($"""

Dinero total ingresado:{Enter}

Dinero total transferido: {Transference}

Dinero total reintegrado: {Withdrawal}

"""

);

}

}

}

}

1. Crea una clase Contador con los métodos para incrementar y decrementar el contador. La clase contendrá un constructor por defecto, un constructor con parámetros, y los métodos getters y setters.

using System.ComponentModel.DataAnnotations;

using System.Diagnostics.Metrics;

using System.Reflection;

namespace ThirdTask

{

public class Program

{

static void Main(string[] args)

{

Contador counter = new Contador();

Console.WriteLine($"Contador: {counter.Counter}");

Console.WriteLine("Incremento del contador en 5: ");

counter.Incrementar();

Console.WriteLine("Decremento del contador en 5: ");

counter.Decrementar();

}

}

public class Contador

{

public int Counter { get; set; }

public Contador()

{

Counter = 0;

}

public Contador(int counter)

{

this.Counter = counter;

}

public void Incrementar()

{

Console.WriteLine(Counter += 5);

}

public void Decrementar()

{

Console.WriteLine(Counter -= 5);

}

}

}

1. Crea una clase Libro con los métodos préstamo, devolución y ToString. La clase contendrá un constructor por defecto, un constructor con parámetros y los métodos getters y setters.

namespace FourthTask

{

public class Program

{

static void Main(string[] args)

{

try

{

Libro libro = new Libro("", "", "", "", "");

libro.FuncionPrestamo();

libro.FuncionDevolucion();

libro.ToString();

}

catch (FormatException)

{

Console.WriteLine("El formato que ha ingresado no es válido.");

}

}

}

public class Libro

{

public int WantToBorrowOrReturnABook { get; set; }

public string BorrowedBook { get; set; }

public string ReturnedBook { get; set; }

public string IsBorrowed { get; set; }

public string IsReturned { get; set; }

public string BookTitle { get; set; }

public Libro()

{

WantToBorrowOrReturnABook = 0;

}

public Libro(string borrowedBook, string returnedBook, string isBorrowed, string isReturned, string bookTitle)

{

this.BorrowedBook = borrowedBook;

this.ReturnedBook = returnedBook;

this.IsBorrowed = isBorrowed;

this.IsReturned = isReturned;

this.BookTitle = bookTitle;

}

public void FuncionPrestamo()

{

Console.WriteLine("¿Desea tomar un libro prestado? 1.Sí 2.No");

WantToBorrowOrReturnABook = Convert.ToInt32(Console.ReadLine());

if (WantToBorrowOrReturnABook == 1)

{

Console.WriteLine("Por favor ingrese el nombre del libro que desea tomar prestado:");

BorrowedBook = Convert.ToString(Console.ReadLine());

IsBorrowed = "Sí";

BookTitle = BorrowedBook;

} else

{

IsBorrowed = "No ha tomado un libro prestado.";

}

}

public void FuncionDevolucion()

{

Console.WriteLine("¿Desea devolver un libro? 1.Sí 2.No");

WantToBorrowOrReturnABook = Convert.ToInt32(Console.ReadLine());

if (WantToBorrowOrReturnABook == 1)

{

Console.WriteLine("Por favor ingrese el nombre del libro que desea devolver:");

ReturnedBook = Convert.ToString(Console.ReadLine());

IsReturned = "Sí";

BookTitle = ReturnedBook;

} else

{

IsReturned = "No ha devuelto un libro.";

}

if (ReturnedBook.ToLower() == BorrowedBook.ToLower())

{

BorrowedBook = string.Empty;

IsBorrowed = "No";

IsReturned = "Sí";

}

if (ReturnedBook == string.Empty && BorrowedBook == string.Empty)

{

IsBorrowed = "No ha tomado un libro prestado.";

IsReturned = "No ha devuelto un libro.";

}

}

public void ToString()

{

Console.WriteLine($"""

Titulo del libro: {BookTitle}

¿Esta Prestado? {IsBorrowed}

¿Esta devuelto? {IsReturned}

""");

}

}

}

1. Crea una clase Fracción con métodos para sumar, restar, multiplicar y dividir fracciones.

using System.Reflection.Metadata;

namespace FifthTask

{

public class Program

{

static void Main(string[] args)

{

try

{

Fraccion fraction = new Fraccion();

fraction.OperacionSeleccionada();

}

catch (DivideByZeroException)

{

Console.WriteLine("No es posible dividir entre 0.");

}

catch (FormatException)

{

Console.WriteLine("El formato ingresado no es válido.");

}

}

}

public class Fraccion

{

public decimal UserFirstFractionAndNumerator { get; set; }

public decimal UserFirstFractionAndDenominator { get; set; }

public decimal UserSecondFractionAndNumerator { get; set; }

public decimal UserSecondFractionAndDenominator { get; set; }

public decimal FractionCalc { get; set; }

public int SelectedOperation { get; set; }

public Fraccion()

{

this.UserFirstFractionAndNumerator = 0;

this.UserFirstFractionAndDenominator = 0;

this.UserSecondFractionAndNumerator = 0;

this.UserSecondFractionAndDenominator = 0;

this.FractionCalc = 0;

}

public void OperacionSeleccionada()

{

Console.WriteLine("""

Por favor seleccione la operación que desee realizar de fracciones:

1.Suma 2.Resta 3.Multiplicación 4.División

""");

SelectedOperation = Convert.ToInt32(Console.ReadLine());

switch (SelectedOperation)

{

case 1:

PreguntasParaHacerOperacionesFraccionarias();

Console.WriteLine($"""

El resultado de la operación es:

{SumarFracciones(UserFirstFractionAndNumerator, UserFirstFractionAndDenominator, UserSecondFractionAndNumerator, UserSecondFractionAndDenominator, FractionCalc)}

""");

break;

case 2:

PreguntasParaHacerOperacionesFraccionarias();

Console.WriteLine($"""

El resultado de la operación es:

{RestarFracciones(UserFirstFractionAndNumerator, UserFirstFractionAndDenominator, UserSecondFractionAndNumerator, UserSecondFractionAndDenominator, FractionCalc)}

""");

break;

case 3:

PreguntasParaHacerOperacionesFraccionarias();

Console.WriteLine($"""

El resultado de la operación es:

{MultiplicarFracciones(UserFirstFractionAndNumerator, UserFirstFractionAndDenominator, UserSecondFractionAndNumerator, UserSecondFractionAndDenominator, FractionCalc)}

""");

break;

case 4:

PreguntasParaHacerOperacionesFraccionarias();

Console.WriteLine($"""

El resultado de la operación es:

{DividirFracciones(UserFirstFractionAndNumerator, UserFirstFractionAndDenominator, UserSecondFractionAndNumerator, UserSecondFractionAndDenominator, FractionCalc)}

""");

break;

}

}

public void PreguntasParaHacerOperacionesFraccionarias()

{

Console.WriteLine("Por favor ingrese el numerador de la primera fracción: ");

UserFirstFractionAndNumerator = Convert.ToDecimal(Console.ReadLine());

Console.WriteLine("Por favor ingrese el denominador de la primera fracción: ");

UserFirstFractionAndDenominator = Convert.ToDecimal(Console.ReadLine());

Console.WriteLine("Por favor ingrese el numerador de la segunda fracción: ");

UserSecondFractionAndNumerator = Convert.ToDecimal(Console.ReadLine());

Console.WriteLine("Por favor ingrese el denominador de la segunda fracción: ");

UserSecondFractionAndDenominator = Convert.ToDecimal(Console.ReadLine());

}

public static decimal SumarFracciones(decimal userFirstFractionAndNumerator, decimal userFirstFractionAndDenominator, decimal userSecondFractionAndNumerator, decimal userSecondFractionAndDenominator, decimal fractionCalc)

{

fractionCalc = (userFirstFractionAndNumerator / userFirstFractionAndDenominator) + (userSecondFractionAndNumerator / userSecondFractionAndDenominator);

return fractionCalc;

}

public static decimal RestarFracciones(decimal userFirstFractionAndNumerator, decimal userFirstFractionAndDenominator, decimal userSecondFractionAndNumerator, decimal userSecondFractionAndDenominator, decimal fractionCalc)

{

fractionCalc = (userFirstFractionAndNumerator / userFirstFractionAndDenominator) - (userSecondFractionAndNumerator / userSecondFractionAndDenominator);

return fractionCalc;

}

public static decimal MultiplicarFracciones(decimal userFirstFractionAndNumerator, decimal userFirstFractionAndDenominator, decimal userSecondFractionAndNumerator, decimal userSecondFractionAndDenominator, decimal fractionCalc)

{

fractionCalc = (userFirstFractionAndNumerator / userFirstFractionAndDenominator) \* (userSecondFractionAndNumerator / userSecondFractionAndDenominator);

return fractionCalc;

}

public static decimal DividirFracciones(decimal userFirstFractionAndNumerator, decimal userFirstFractionAndDenominator, decimal userSecondFractionAndNumerator, decimal userSecondFractionAndDenominator, decimal fractionCalc)

{

fractionCalc = (userFirstFractionAndNumerator / userFirstFractionAndDenominator) / (userSecondFractionAndNumerator / userSecondFractionAndDenominator);

return fractionCalc;

}

}

}