

Desarrollo de Software IV

Laboratorio # 7 – Prof. Regis Rivera

Objetivo: Introducción a POO en C#

PARAMETROS

- IDE: Visual Studio .NET 2022
- Tipo de Aplicación: Consola
- Solución: Laboratorio7
- Proyectos: Laboratorio71, 72... [Según cada Ejemplo]
- Lenguaje: C#

Lab 7-1

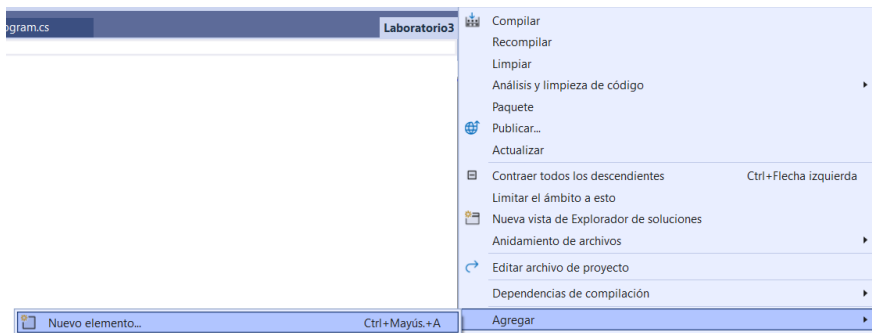
Un banco tiene 3 clientes que pueden hacer depósitos y extracciones. También el banco requiere que al final del día calcule la cantidad de dinero que hay depositada.

La Solución tendrá el siguiente esquema: Debemos definir los atributos y los métodos de cada clase:

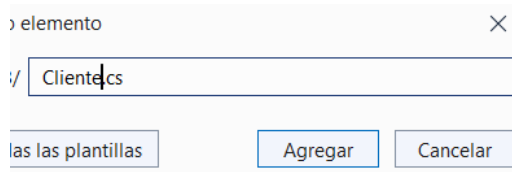
Cliente
atributos
nombre
monto
métodos
constructor
Depositar
Extraer
RetornarMonto

Banco
atributos
3 Cliente (3 objetos de la clase Cliente)
métodos
constructor
Operar
DepositosTotales

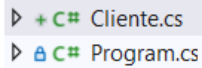
¿Cómo agregamos una nueva clase en Visual Studio.NET para C#? Click derecho sobre el proyecto, elegir la opción agregar, luego “nuevo elemento”



Colocarle el nombre Cliente y luego click en aceptar



Observar aparece un nuevo archivo con extensión .cs en su proyecto, adicional del tradicional ya conocido Program.cs



Codificar la clase cliente:

```
class Cliente
{
    private string nombre;
    private int monto;

    public Cliente(string nom)
    {
        nombre = nom;
        monto = 0;
    }

    public void Depositar(int m)
    {
        monto = monto + m;
    }

    public void Extraer(int m)
    {
        monto = monto - m;
    }

    public int RetornarMonto()
    {
        return monto;
    }

    public void Imprimir()
    {
        Console.WriteLine(nombre + " tiene depositado la suma de " + monto);
    }
}
```

Adicional, crear la clase Banco:

```
class Banco
{
    private Cliente cliente1, cliente2, cliente3;

    public Banco()
    {
        cliente1 = new Cliente("Yhonas");
        cliente2 = new Cliente("Ana");
        cliente3 = new Cliente("Pedro");
    }

    public void Operar()
    {
        cliente1.Depositar(100);
        cliente2.Depositar(150);
        cliente3.Depositar(200);
        cliente3.Extraer(150);
    }

    public void DepositosTotales()
    {
        int t = cliente1.RetornarMonto() +
            cliente2.RetornarMonto() +
            cliente3.RetornarMonto();
        Console.WriteLine("El total de dinero en el banco es:" + t);
        cliente1.Imprimir();
        cliente2.Imprimir();
        cliente3.Imprimir();
    }
}
```

Y codificar el método principal (bajo Program.cs)

```
static void Main(string[] args)
{
    Banco banco1 = new Banco();
    banco1.Operar();
    banco1.DepositosTotales();
    Console.ReadKey();
}
```

Al ejecutar el código muestra el siguiente resultado:

```
El total de dinero en el banco es:300
Yhonas tiene depositado la suma de 100
Ana tiene depositado la suma de 150
Pedro tiene depositado la suma de 50
```

Lab 7-2

Plantear un programa que permita jugar a los dados. Las reglas de juego son: se tiran tres dados si los tres salen con el mismo valor mostrar un mensaje que "gano", sino "perdió".

Lo primero que hacemos es identificar las clases y luego los atributos y los métodos de cada clase:

Dado

atributos

valor

métodos

constructor

Tirar

Imprimir

RetornarValor

JuegoDeDados

atributos

3 Dado (3 objetos de la clase Dado)

métodos

constructor

Jugar

```
class Dado
{
    private int valor;
    private static Random aleatorio;

    public Dado()
    {
        aleatorio = new Random();
    }

    public void Tirar()
    {
        valor = aleatorio.Next(1, 7);
    }

    public void Imprimir()
    {
        Console.WriteLine("El valor del dado es:" + valor);
    }

    public int RetornarValor()
    {
        return valor;
    }
}
```

```

class JuegoDeDados
{
    private Dado dado1, dado2, dado3;

    public JuegoDeDados()
    {
        dado1 = new Dado();
        dado2 = new Dado();
        dado3 = new Dado();
    }

    public void Jugar()
    {
        dado1.Tirar();
        dado1.Imprimir();
        dado2.Tirar();
        dado2.Imprimir();
        dado3.Tirar();
        dado3.Imprimir();
        if (dado1.RetornarValor() == dado2.RetornarValor() &&
            dado1.RetornarValor() == dado3.RetornarValor())
        {
            Console.WriteLine("Ganó");
        }
        else
        {
            Console.WriteLine("Perdió");
        }
        Console.ReadKey();
    }
}

```

Y el programa principal:

```

static void Main(string[] args)
{
    JuegoDeDados j = new JuegoDeDados();
    j.Jugar();
}

```

Al ejecutar el código muestra el siguiente resultado

```

El valor del dado es:4
El valor del dado es:2
El valor del dado es:5
Perdió

```

Guardar su proyecto y comprimir la carpeta del mismo (zip/rar/7z), por ej.: **Laboratorio7.zip**