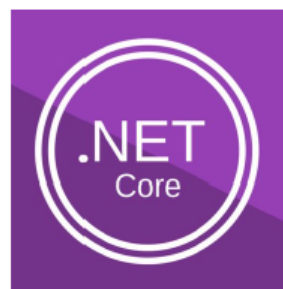


Módulo 2

Programación Orientada a Objetos en .NET Core (C#)



Parte 2 – ejercicios

Curso de Programación en .NET

Ejercicios

Cuando trabajemos con clases diferentes y no sea una continuación del ejercicio anterior, es recomendable crear un proyecto nuevo para mantener cierta limpieza de código.

1. Crea una clase llamada **Tienda** con el atributo **precio** y un método llamado **Bienvenida** que imprima por consola "¡Bienvenido a nuestra tienda!". Crea otra clase llamada **TiendaLicores** que herede de **Tienda** el **precio** y que además tenga otro atributo llamado **impuesto**. Crea un objeto del tipo **TiendaLicores** y llama al método **Bienvenida**.
2. Sobrescribe el método **Bienvenida** en la clase **TiendaLicores**. Deberá hacer lo mismo que la clase base y añadir el mensaje "Si eres menor de 18 años, fuera de aquí". Crea un método **ComprarBebida** al que se le pase la cantidad y que imprima por consola el precio total ($\text{precio} \times (1 + \text{impuesto}) \times \text{cantidad}$).

Ejemplo. *Has comprado 5 bebidas y has gastado 60 euros.*

3. Crea la clase **Persona** con los atributos **dni**, **nombre** y **edad** (junto a los getters y setters que vayas necesitando). Sobrescribe el método **ToString**, que mostrará los datos de la persona, el método **Equals** (y **GetHashCode**) que utilizarán solamente el **DNI** para comparar si una persona es la misma que otra, y haz un método que se llame **Clone** que devolverá una copia de la persona (usa el método **MemberwiseClone** heredado de **Object**).

Crea 4 personas. De una de ellas crea una copia usando **Clone** y cambiándole el nombre. A continuación crea un array que contenga a esas 4 personas (la copia no). Finalmente recorre el array comparando la copia con cada posición del array usando **Equals**. Cuando encuentres a la persona imprime el mensaje: "Persona encontrada en la posición N" (siendo N la posición del array). Imprime también los datos de la persona del array y de la copia con el método **ToString()**.

4. Crea una clase llamada **Empleado** que tendrá las propiedades **nombre** y **salario**. Otra clase llamada **Direccion** que contendrá la **calle** y el **número**. Y finalmente otra clase llamada **Empresa** que tendrá una **dirección** (objeto de la clase **Direccion**), un **nombre**, un **array** de **empleados** y el **número** de **empleados** actual. El constructor de la clase **Empresa** recibirá el **nombre**, la **dirección**, y la **capacidad** máxima de **empleados**.

Crea un método en **Empresa** llamado **Clone** que haga una copia en profundidad, copiando además de los datos de la empresa, los objetos que contiene como la **dirección** o el **array** de **empleados** (para ello implementa también el método **Clone** en la clase **Direccion** y en la clase **Empleado**). Implementa también el método **ToString** que muestre todos los datos de la empresa, incluyendo su **dirección** y la **lista** de **empleados**. Crea también los métodos que vayas necesitando.

Para probar que está todo correcto, crea una empresa (con capacidad para 5 empleados) y añádele 3 empleados, a continuación genera una copia. En la copia cambia el nombre al primer empleado, cambia el número de la calle, y añade otro empleado. A continuación muestra los datos de ambas empresas (ToString) y comprueba que lo que has cambiado en la copia no ha afectado al original (no se ha mantenido ninguna referencia al mismo objeto).

Ejemplo código para cambiar el nombre del empleado, añadir otro y cambiar el número de la dirección:

```
copiaEmpresa.GetEmpleado(0).SetNombre("Marcelino");
copiaEmpresa.AddEmpleado(new Empleado("Isabel", 23000));
copiaEmpresa.GetDireccion().SetNumero(21);
```

5. Crea una clase llamada Figura. Esta clase será abstracta y tendrá los métodos Perímetro y Area como abstractos también. Crea la clase Cuadrado y Circulo que hereden de Figura e implementa dichos métodos junto a los atributos que necesites: lado para el cuadrado y radio para el círculo.
6. Haz lo mismo que en el ejercicio anterior pero usando una interfaz (IFigura) en lugar de una clase abstracta.
7. Crea un array que contenga 3 círculos y 3 cuadrados pero no los pongas seguidos. Para ello tendrás que recurrir al polimorfismo y usar una clase o interfaz que compartan. Recorre dicho array mostrando el área y el perímetro de cada figura así:

```
Area: 12. Perímetro: 14
```

8. A partir del ejercicio anterior, debes detectar cuando recorres el array si se trata de un círculo o un cuadrado, y en función de eso imprimir el tipo de figura y el tamaño del lado o del radio. Ejemplo:

```
*** Círculo - Radio: 3 ***
Área: 28,27. Perímetro: 18,84
*** Cuadrado - Lado: 3 ***
Área: 9. Perímetro: 12
```