10/19/2016

Practica 7

Alumnos: José Enrique Cruz González

Jorge Luis Macuitl Madrid

Ariel Ibarra Reza

Grado: 7° “A”

Docente: Mtro. Ismael Sánchez González

Matricula: 3514110345

Matricula: 3514110677

Introducción

El polimorfismo, en programación orientada a objetos, se refiere a la posibilidad de acceder a funciones distintas a través de la misma interfaz.

Por ejemplo, podemos realizar el proceso Dibujar en una clase de tipo Figura y utilizar el mismo nombre de proceso a realizar para otra clase heredada; es decir cada objeto tiene valores diferentes, aunque el proceso a realizar tenga el mismo nombre.

Es por ello que en esta práctica se establecerán algunos temas del polimorfismo y así concretar nuestro conocimiento.

QUÉ ES POLIMORFISMO.

El polimorfismo, en programación orientada a objetos, se refiere a la posibilidad de acceder a un variado rango de funciones distintas a través de la misma interfaz. O sea, un mismo identificador puede tener distintas formas (distintos cuerpos de función, distintos comportamientos) dependiendo del contexto en el que se halle.

El polimorfismo es muy parecido o más bien tiene sus bases en la capacidad de herencia que presentan los lenguajes orientados a objetos como es el caso de java.

Como sabemos con la herencia las clases padres comparten métodos con las clases hijas, bueno con el polimorfismo se hace prácticamente lo mismo, pero en vez de clases padres se tiene clases abstractas.

Las clases abstractas tienen métodos abstractos es decir métodos que solo están declarados sus nombres, pero su forma de “actuar” difiere de una clase hija a otra. Los métodos actúan dependiendo de la clase que haga mención del método declarado en la clase abstracta.

QUÉ ES ABSTRACCIÓN.

La abstracción es como se pueden representar los objetos en modo de código.

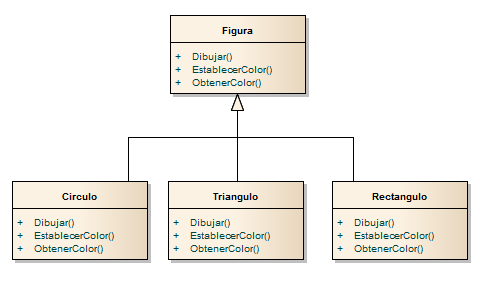
Es un método por el cual abstraemos, vale la redundancia, una determinada entidad de la realidad; sus características y funciones que desempeñan. Estos son representados en clases por medio de atributos y métodos de dicha clase.

Un ejemplo sencillo para comprender este concepto seria la abstracción de un Automóvil.

Acá vamos a sacar de esta entidad sus características, por ejemplo: color, año de fabricación, modelo, etc. Y ahora sacamos sus métodos o funciones típicas de esta entidad como, por ejemplo: frenar, encender, etc. A esto se le llama abstracción.

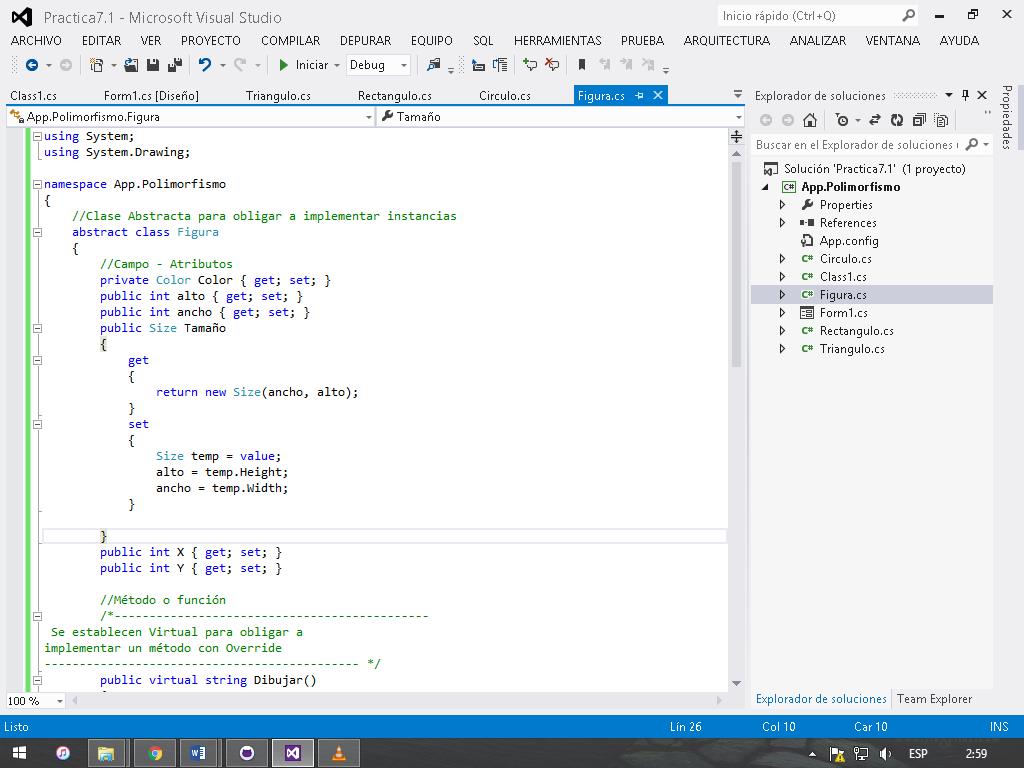
## Ejemplo de polimorfismo

Para proseguir con la práctica, es necesario realizar un ejemplo básico en el cual se basará en el siguiente diagrama:



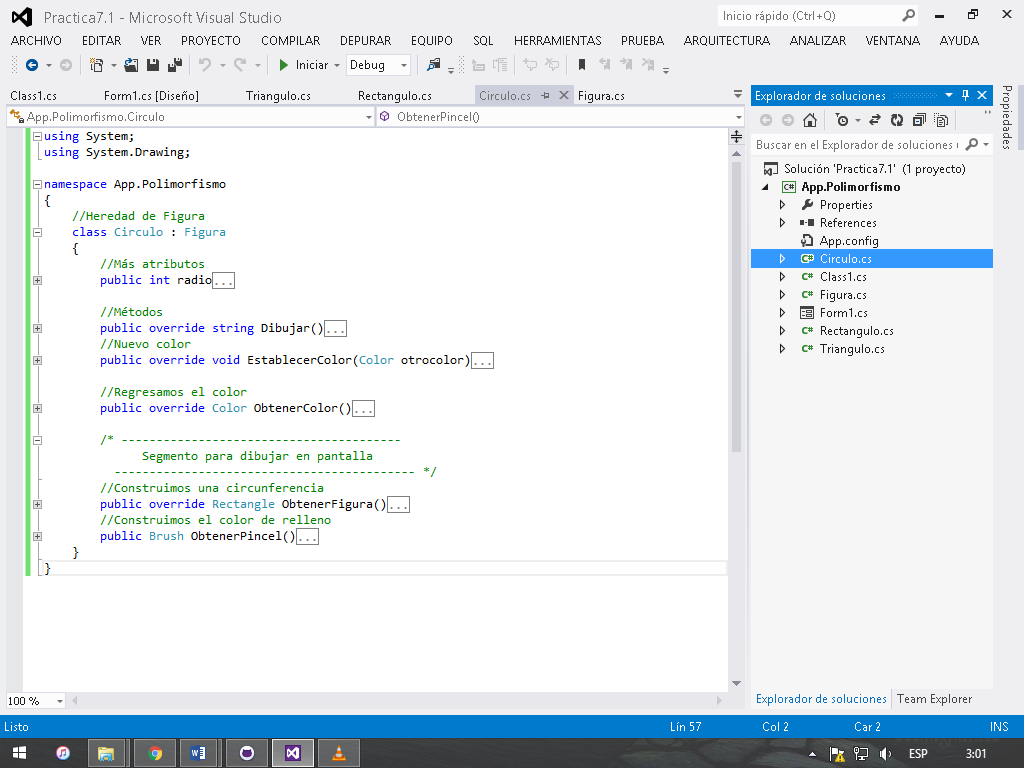
Teniendo como referencia este diagrama se podrá hacer el modelado en C#. Realizado el modelado se ejecutan las clases para verificar el resultado.

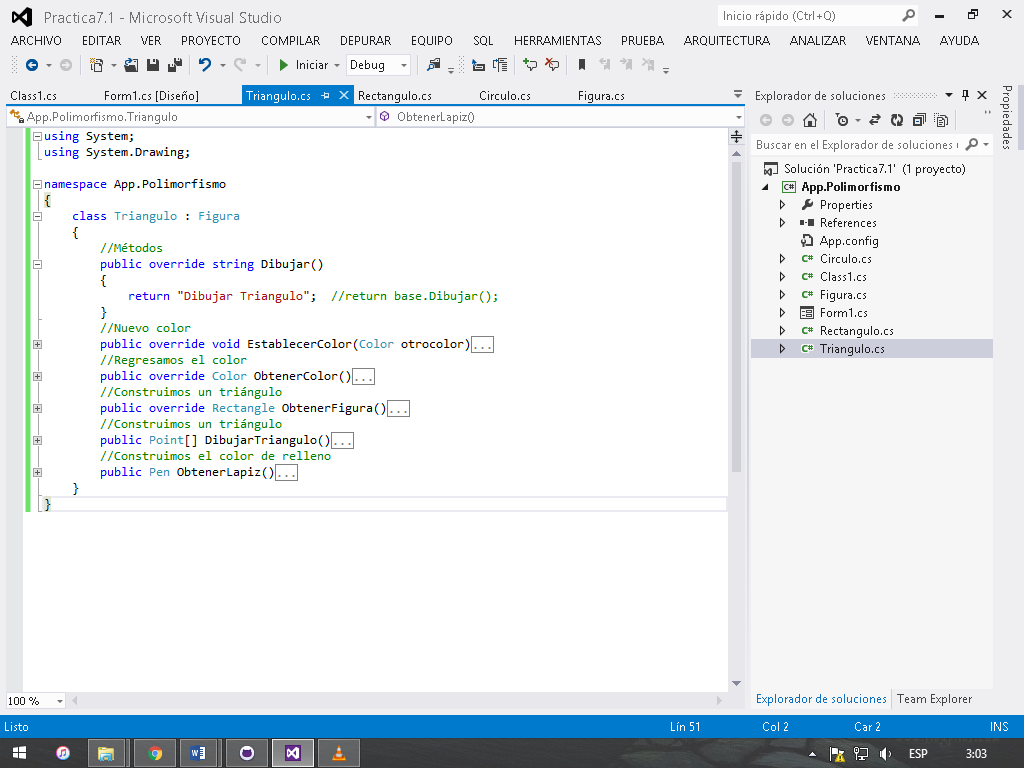
Dentro de esta primera se crea y se codifica la clase Figura, en la cual se establecen sus métodos.



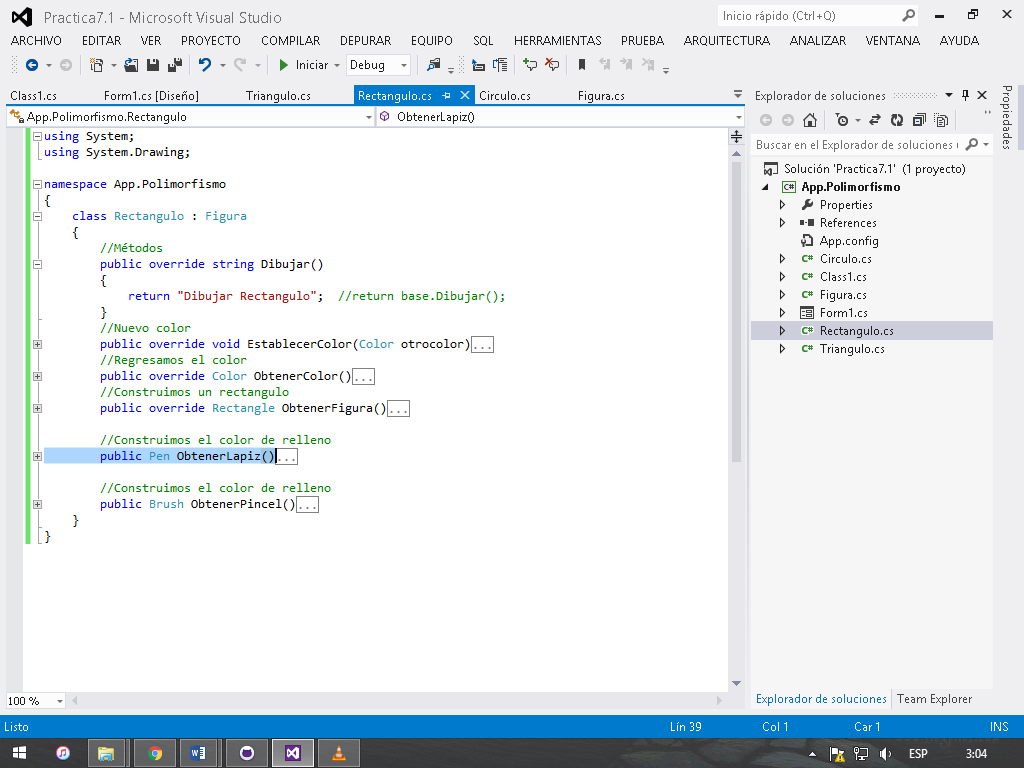
Creada esta primera clase, la cual es de vital importancia en la práctica, se pueden crear las clases de las cuales van a heredar los métodos de la clase Figura, como dibujar, establecer color, dibujar color.

Clase Circulo:



Clase Triangulo:

Clase Rectángulo:



Todos estos métodos en su contenido, tienen el mismo concepto (todo es igual) de acuerdo a cada clase. Pero, para aplicar el concepto e polimorfismo. Es necesario hacer que las clases Circulo, Triangulo y Rectángulo hereden los métodos de la clase Figura. De esta manera todos podrán hacer referencia a una misma acción.

El término del polimorfismo establece que todas las acciones pueden ser iguales (métodos) pero cada uno de ellos lo va a ejecutar con diferentes escenarios (casos posibles).

Para ello en cada uno de los métodos, el contenido fue cambiado y de esta manera se cumpliera una regla el polimorfismo, en la que, aunque tenga el mismo nombre el método se le pueden programar acciones diferentes a cada uno de estos.

Conclusiones

Durante la realización de este trabajo, se ejecutaron diversas clases, las cuales tenían un contenido específico.

En programación orientada a objetos se denomina polimorfismo a la capacidad que tienen objetos de diferentes clases de responder al mismo mensaje o evento. Esto significa que puede haber muchos mensajes con el mismo nombre, en diferentes clases. Cada Clase responde al mensaje con su código propio (o método).

También se puede aplicar a la propiedad que poseen algunas operaciones de tener un comportamiento diferente dependiendo del objeto (o tipo de dato) sobre el que se aplican.

El polimorfismo sólo es aplicable si cualquiera de los posibles tipos de objetos que se invoquen tienen definida la operación empleada, y los tipos de datos de entrada requeridos y los valores devueltos, más allá de cómo se empleen o calculen, son compatibles entre sí