

Reporte de lectura - Capítulo 6

Los objetos son un conjunto de datos relacionados entre sí, es decir, estos datos pueden pertenecer a un ente o cosa, como por ejemplo un objeto para perro. A pesar de que, todos los perros son distintos, en este objeto estarán los datos iniciales que se interesaría saber de cualquier perro, como su nombre, raza, edad y si está vacunado contra la rabia.

Entonces, por otro lado, tenemos los métodos, que vienen siendo como las acciones o comportamientos, en este caso de nuestro perro sería como comer, hacer sus necesidades y dormir. Podemos ver que, tenemos como objeto perro y sus distintos métodos que realizan estos, entonces ya con eso, podemos usar este objeto y sus métodos, podemos crear una clase, pues los objetos son instancias de una clase y allí es donde se definen.

Una vez tengamos nuestra clase, podemos crear instancias de esta para los distintos perros que queramos albergar en nuestro sistema, sin la necesidad de tener que crear las variables y métodos para cada uno, sino más bien, creando una clase para todos y creamos instancias de esta para cada uno. En fin, podemos decir que una alberga las características y métodos (acciones) de un objeto.

Ya dentro de las propiedades de una clase, luego de crear esta, declaramos las variables que contendrá la clase, que vienen siendo los atributos de la clase. Las variables que pueden ser declaradas como públicas o privadas, esto significa que, si una variable es privada solo puede ser manipulada dentro de la clase. Mientras que, por su parte las variables declaradas como públicas, pueden ser manipuladas por código fuera de la clase. Esta manera de elegir si se quieren declarar las variables públicas o privadas es fundamental para la programación orientada a objetos. Puesto que, gracias a esto podemos encapsular los datos, para evitar que accedan a los datos cierto tipo de usuarios, asegurar la integridad de los datos, entre otros.

Además de declarar los atributos, podemos crear los métodos de la clase que definirán su comportamiento, como mencionamos al inicio. Los métodos pueden retornar o no valores, esto lo decide el programador. Para los métodos que puede retornar, debemos de especificar el tipo de valor que este retornará, cuando creamos el método, el tipo del retorno puede ser: entero, flotante, cadena, booleano, entre otros. Mientras que, cuando definimos un método sin retorno, le colocamos void en vez de colocar el tipo de dato de retorno.

Dentro de los métodos, también tenemos el método constructor, que lleva como nombre el mismo que la clase. Este método constructor es llamado desde el momento en que creamos una

instancia de objeto de una clase. La función de estos constructores es permitir que se le asignen valores por defecto a los atributos creados. Estos métodos no incluyen un tipo de datos de retorno.

De igual forma, es importante mencionar que los métodos pueden estar parametrizados. Esto quiere decir, que lo podemos pasar parámetros a los métodos, estos parámetros vienen siendo tipos de datos. La manera de hacer esto, es que cuando creamos el método le especificamos la cantidad de datos y los tipos de cada uno de estos. Esto se hace mediante la declaración de estos, pero debemos saber que estas no son variables que podemos usar fuera del método en la clase. Para poder guardar estas y usarla en la clase, debemos contenerla en alguna variable que hayamos declarado a la hora de crear los atributos.

Para finalizar con los métodos, también debemos conocer el concepto de métodos de sobrecarga. Estos nos es más la creación de varios métodos con el mismo nombre. Lo que diferencia a estos, son los parámetros que se le pasan, puede ser la cantidad o el tipo de datos en los parámetros que obtiene como entrada. De igual forma, podemos hacer sobrecarga no solo cambiando el parámetro, sino también, cambiando el tipo de dato que retorna el método. Por ejemplo, podemos crear un método que reciba como parámetro dos números de tipo entero y que retorne la suma de estos de tipo entero; y podemos hacer un método de sobrecarga con el mismo nombre, cambiando el tipo de los parámetros de entrada a flotante y retornando el valor en tipo flotante. El compilador es capaz de especificar cual de los métodos será usado.

En conclusión, las clases son parecidas a los planos, creamos cosas en base a esta. Nos referimos que las clases son bases, pues cuando creamos objetos donde podemos encasillar diferentes instancias, que son diferentes, pero comparten el mismo núcleo; entonces, en este caso las instancias de una clase tienen como base de su estructura a esta. Estas instancias pueden ser modificadas, pero las clases viene siendo la esencia, como ejemplo de esto tenemos a la raza humano y las instancias son las distintas personas en el mundo. Entonces, todos los humanos compartimos rasgos similares, pero cada uno somos distinto, lo mismo que pasa con la clase y sus distintas instancias. Pero no debemos olvidar que la clase es solo un esqueleto, que los atributos y métodos le dan vida. Pues con solo crear una clase, no significa que hemos creado un objeto, ya que necesitamos que la tengan la estructura necesaria para convertirse en un objeto. Lo positivo de la programación orientada a objetos es la encapsulación de datos en las clases que nos da más seguridad en nuestro programa y la reducción de código innecesario, pues

al tener las clases, solo debemos crear instancias de esta para los objetos que la utilizan como base.

Referencias

Clark, D. (2013). Beginning C# Object-Oriented Programming (2nd Ed.). New York: Apress.