INFORMATICA PARA INGENIEROS PARADIGMA ORIENTADO A OBJETOS

LECTURAS SEPTIMA SESION

ABSTRACCION

Empezaremos conociendo a qué le llamamos abstracción dentro de la Programación orientada a objetos.

Una abstracción puede definirse como:

Las características específicas de un objeto, aquellas que lo distinguen de los demás tipos de objetos y que logran definir límites conceptuales respecto a quien está haciendo dicha abstracción del objeto.

Una abstracción se enfoca en la visión externa de un objeto, separa el comportamiento específico de un objeto, a esta división que realiza se le conoce como la barrera de abstracción, la cual se consigue aplicando el principio de mínimo compromiso.

Pero… ¿Qué es el principio de mínimo compromiso? Se refiere al proceso por el cuál la interfaz de un objeto muestra su comportamiento específico y nada más, absolutamente nada más.

Pero… ¿Qué es una Interfaz? Una interfaz de objeto permite crear código con el cuál se específica que métodos serán implementados por una clase sin necesidad de definir que harán estos métodos, dichos métodos deben ser públicos.

Pero… Existe también el principio de mínima sorpresa, en el cuál una abstracción obtiene el comportamiento completo de algún objeto y por ende no ofrece sorpresas o efectos laterales que lleguen más allá del ámbito de la abstracción.

Hay una alta gama de abstracciones que existen desde los objetos que modelan muy cerca de entidades, a objetos que no tienen razón para existir, vamos a hacer una rápida mención de ello.

Abstracción de Entidades: Es un objeto que representa un modelo útil de una entidad que se desea.

Abstracción de Acciones: Un objeto que representa un conjunto de operaciones y todas ellas desempeñan funciones del mismo tipo.

Abstracción de Máquinas virtuales: Un objeto que agrupa operaciones, todas ellas virtuales, utilizadas por algún nivel superior de control u operaciones (entre ellos podríamos hablar de un circuito).

Abstracción de coincidencia: Un objeto que almacena un conjunto de operaciones que no tienen relación entre sí.

Modelo contractual de programación

En dicho modelo podemos mencionar que la vista exterior de cada objeto define una interfaz del que puedan depender otros objetos, esta interfaz como bien lo habíamos mencionado, establece todas las suposiciones que pueda hacer un objeto cliente acerca del comportamiento de un objeto servidor, es decir la interfaz abarca las responsabilidades de un objeto, dicho en otras palabras, abarca el comportamiento del que se le considera responsable.

Un objeto puede actuar y reaccionar de diferentes formas, un conjunto de operaciones que puede realizar un cliente sobre un objeto se le denomina protocolo.

No está demás mencionar que toda abstracción tiene propiedades estáticas y dinámicas.

Propiedades estáticas podemos mencionar, el nombre, el tamaño, en algunas ocasiones su contenido.

Propiedades dinámicas podemos mencionar peso, tamaño, contenido.

Dependiendo el contexto que se está analizando el contenido u otras propiedades pueden ser dinámicas como estáticas.

Realizaremos un breve ejemplo de una abstracción sobre un vehículo.

Supongamos que en este ejemplo nos están solicitando la abstracción de un vehículo en cuanto a su comportamiento.

La pregunta que hago es ¿Cuáles son los comportamientos que tienen todos los vehículos? con base a esta pregunta, automáticamente fluye nuestra abstracción del comportamiento de un vehículo.

Encender Vehículo

Apagar Vehículo

Acelerar Vehículo

Frenar Vehículo

Retroceder Vehículo

Parabrisas Vehículo

Y podría listar el comportamiento del vehículo, ahora para que ustedes puedan ejercitar lo aprendido, pueden realizar abstracciones de objetos, incluyendo sus características y el comportamiento.

FUNCIONALIDAD (METODOS)

Métodos: conjunto de instrucciones a las que se les asocia un nombre de modo que si se desea ejecutarlas, sólo basta o referenciarlas a través de dicho nombre en vez de tener que escribirlas.

Dentro de estas instrucciones es posible acceder con total libertad a la información almacenada en los campos, pertenecientes a la clase dentro de la que el método se ha definido. Por lo que los métodos premiten manipular los datos almacenados en los objetos.

Sintaxis:

()

{

instrucciones;

}

Tipo devuelto: Todo método puede devolver un objeto como resultado de la ejecución de las insrucciones que lo forma, aquí de indica el tipo del dato al que pertenece este objeto.

Si no devuelve nada se indica “void” y si devuelve algo es obligatorio finalizar la ejecución de sus instrucciones con “return ” que indica el objeto a devolverse.

Parámetro: opcionalmente todo método puede recibir en cada llamada una lista de objetos a los que podrá acceder durante la ejecución de sus instrucciones. Aquí se indica cuáles son los tipos de datos de estos objetos y cuál es el nombre con el que harán referencia las instrucciones de método a cada uno de ellos.

Aún que los objetos que puede recibir el método pueden ser diferentes, cada vez que se solicite su ejecución siempre han de ser los mismos tipos y han de seguir el orden establecido en parámetros.

Ejemplo:

```
class persona

{

string nombre;

int edad;

string RFC;

void cumpleaños ()

{

edad ++;

}

}
```

Sintaxis para llamar a los métodos de un objeto:

.(parámetro)

Parámetro: valores que se desean dar a los parámtros del método al hacer la llamada. Si el método no toma parámtros se deja vacío.

ejemplo: llamada al método cumpleaños de un objeto persona llamado p

```
p.cumpleaños();
```