

Title: *Introducción a las clases y los objetos*

Keyword

*Clases
objetos
métodos
programación
Atributos
Variables de instancia*

Topic: *Clases, objetos, métodos, propiedades y variables de instancia.*

El Método: Son los mecanismos que se encargan de realizar sus tareas; y ocultan al usuario, las tareas complejas que se realizan.

Clase: A cada un método, se proporcionan uno más métodos, que están diseñados para realizar las tareas de esa clase.

También debemos construir un objeto de una clase para poder hacer que una aplicación realice las tareas descritas por la clase. Estas son una de las razones por las cuales C# se conoce como un lenguaje de programación orientado a objetos.

Las propiedades contienen descriptores de acceso get para leer los valores de las variables y descriptores de acceso set para almacenar valores en ellos.

Questions

*¿Qué son los objetos?
¿Qué son las clases?*

Summary: *Se explica el concepto de clases, objetos y métodos en programación para construir un objeto*

NAME	PAGES	SPEAKER/CLASS	DATE - TIME
Diego Vargas	2-6	F. programación II	20/1/2024

Title: Introducción a las clases y los objetos

Keyword
 Clases
 métodos
 C#
 propiedades

Topic: Declaración de una clase con un método e instanciando el objeto de una clase
 Declaración de la clase 'libroCalificaciones' que contiene un método llamado 'mostrar mensajes'. La clase se inicia en la línea 5 y utiliza la palabra clave 'public' como modificador de acceso. El cuerpo de la clase está encerrado entre llaves izquierda y derecha.

El método 'mostrar mensajes' se encuentra en las líneas 9-11 y muestra un mensaje en la pantalla, siendo la línea 10 la que ejecuta esta acción.

Questions

¿Cómo se utilizan las propiedades de un objeto para acceder a sus atributos en lugar de la variable de instancia directa?

Cada clase tenía un método llamado 'main', pero en este caso, la clase 'libroCalificaciones' tiene el método 'mostrar mensajes'. El método 'main' es especial y se llama automáticamente al ejecutar una aplicación.

Summary: Se destaca la importancia de los métodos para realizar tareas, la programación en C#

NAME
Diego Vargas

PAGES
3-6

SPEAKER/CLASS
F. programación II

DATE - TIME
20/1/2024

Title: Introducción a las clases y los objetos

Keyword

Tipos de datos

Clases

Uml

(unified modeling language)

Parámetros

Topic: Declaración de un método con un parámetro

Al declarar un método, es necesario especificar si requiere datos para realizar su tarea, lo cual se logra mediante la lista de parámetros en las paréntesis después del nombre del método. Esta lista puede contener cualquier número de parámetros, incluso ninguno. Las paréntesis vacías indican que el método no requiere parámetros.

El diagrama de clases Uml actualizado para la clase libro Colecciones muestra la operación Mostrar Mensaje, opera con un parámetro llamado Nombre Curso de tipo string.

Aunque Uml tiene tipos de datos similares a C++, prefiere utilizar tipos de C++ en los diagramas Uml para evitar confusiones.

Questions

¿Cómo se marcan los métodos con múltiples parámetros?

¿Cuál es la función de la lista de parámetros en la declaración de un método y dónde se encuentra?

Summary:

Un método puede tener múltiples parámetros, separados por Comas. Al llamar a un método, el número y tipo de argumentos deben coincidir con la lista de parámetros en la declaración del método.

NAME Diego Vargas	PAGES 4-6	SPEAKER/CLASS F. Programación II	DATE - TIME 20/1/2024
----------------------	--------------	-------------------------------------	--------------------------

Title: Introducción a las clases y los objetos

Keyword
libros
Categorías
instancias
declaración

Topic: Variables de instancias y propiedades

En la clase libro Categorías, se mantiene el nombre del curso como una variable de instancia, y se introduce una propiedad llamada NombreCurso, junto con un método MostrarMensaje. La variable instancia NombreCurso es de tipo string y se declara como private, lo que significa que solo es accesible dentro de la clase.

El método MostrarMensaje ahora utiliza la propiedad NombreCurso para obtener el nombre del curso. Se destaca la práctica de utilizar propiedades para manipular variables de instancia en lugar de acceder directamente a ellas.

La declaración de variables de instancia con la palabra clave private y la explicación de que el modificador de acceso private limita el acceso a los variables y métodos solo a la propia clase.

Questions

¿Qué es una variable de instancia?

Summary: Se aborda la idea de modificación de variable de instancia private mediante métodos get y set en otros lenguajes, pero se menciona que C# proporciona una solución más elegante a través del uso de propiedades.

NAME
Diego Vargas

PAGES
5-6

SPEAKER/CLASS
F. Programación

DATE - TIME
20/1/2024

Title: Introducción a las clases y objetos

Keyword

UML
Atributos
Visibilidad pública
Propiedad
Método público
Nombre Curso

Topic: Diagramas de clases de UML con una propiedad

Las propiedades se modelan como atributos, siendo representados en un diagrama de clases con el signo más (+) para indicar visibilidad pública. Se utiliza la palabra "property" entre los signos < y >, que actúan como esterotipo. El tipo de propiedad se especifica con un signo de dos puntos y el tipo después del nombre de la propiedad.

Cuando se modelan variables de instancia privada que no son propiedades, se utilizan atributos en el Compartimiento Central del diagrama de clases. Se usa el símbolo menos (-) para indicar visibilidad privada.

Ejemplo: la variable de instancia nombreCurso se modelaría como "- nombreCurso : string". El diagrama de clases se centra en el diseño de la clase, omitiendo detalles de implementación como la variable de instancia.

Questions

¿Cómo se representan las propiedades en un diagrama de clases UML?
¿Cuál es el símbolo utilizado para indicar visibilidad pública en UML?

Summary:

En C#, la propiedad se nombra generalmente igual que la variable de instancia que manipula, pero con la primera letra en mayúscula. El diagrama de clases proporciona una guía para diseñar la clase sin mostrar todos los detalles de implementación, dejando al programador implementar la variable de instancia según sea necesario.

Diego Vargas

NAME

6-6

PAGES

SPEAKER/CLASS

F. programación II

DATE - TIME

20/1/2024

Title: Introducción a las clases y los objetos

Keyword

privada
acceso
Propiedades
Validar
método
manipulación

Topic: Ingeniería de software con propiedades y los descriptores de acceso set y get

El uso de propiedades en C# permite controlar el acceso a variable de instancia privada, manteniendo la encapsulación de datos. A través de descriptores de acceso get y set, se pueden traducir formato de datos y aplicar validaciones.

Este enfoque garantiza que los datos privados permanezcan seguros y que su manipulación esté controlada. Además, se obtiene la conveniencia de que los propios métodos de la clase utilicen las propiedades para manipular variable de instancia privada, lo que hace que la clase sea más robusta y fácil de mantener en caso de cambios en la representación interna de los datos.

Los cambios en la representación interna de la clase pueden manejarse sin modificar los métodos que utilizan las propiedades.

Questions

¿Cómo se modelan las propiedades en un diagrama de clases UML?
¿Por qué es crucial mantener por variable de instancia como privada en la programación orientada a objetos?

Summary:

Se permite el poder controlar el acceso a variable de una instancia privada, los propios métodos de la clase utilizan las propiedades para manipular variable de instancia privada.