**UNIVERSIDAD ABIERTA PARA ADULTOS**

**(UAPA)**



# **Asignatura:**

# **Programación III**

# **Tema:**

# **Tarea I**

# **Facilitador:**

# **Gregorio De La Rosa**

# **Participante:**

# **Jochimin Contreras Garcia.**

# **Mat. 2019-05041**

**Después de consultar el texto básico de la asignatura y otros documentos sugeridos por su facilitador, en el contenido de la Unidad I, realiza las actividades sugeridas:**

**1. Investigar en la web acerca de los lenguajes de programación orientados a objetos, características, ventajas y desventajas.**

Los lenguajes de programación orientados a objetos tratan a los programas como conjuntos de objetos que se ayudan entre ellos para realizar acciones. Entendiendo como objeto al entidades que contienen datos. Permitiendo que los programas sean más fáciles de escribir, mantener y reutilizar.

Los objetos tienen toda la información (atributos) que los diferencia de otros pertenecientes a otra clase. Por medio de unos métodos se comunican los objetos de una misma o diferente clase produciendo el cambio de estado de los objetos. Esto hace que a los objetos se les trate como unidades indivisibles en las que no se separan la información ni los métodos usados en su tratamiento.

Los lenguajes de programación orientados a objetos tienen su origen en un lenguaje que fue diseñado por los profesores Ole-Johan Dahl y Kristen Nygaard en Noruega. Este lenguaje de programación orientado a objetos fue el “Simula 67” que fue un lenguaje creado para hacer simulaciones de naves.

Los lenguajes de programación orientadas a objetos son lenguajes dinámicos en los que estos objetos se pueden crear y modificar sobre la marcha. Esta programación orientada a objetos (POO) tomo auge a mediados de los años ochenta debido a la propagación de las interfaces gráficas de usuarios, para lo que los lenguajes de programación orientados a objetos están especialmente dotados.

Los principales lenguajes de programación orientados a objetos son:

Ada, C++, C#, VB.NET, Clarion, Delphi, Eiffel, Java, Lexico (en castellano), Objective-C, Ocaml, Oz, PHP, PowerBuilder, Python, Ruby y Smalltalk.

No todos estos lenguajes de programación orientados a objetos son específicamente orientados a objetos. Sino que algunos de ellos se le han añadido extensiones orientadas a objetos.

**Entre sus características estan:**

**Abstracción**

Denota las características esenciales de un objeto, donde se capturan sus comportamientos.

[**Encapsulamiento**](https://es.wikipedia.org/wiki/Encapsulamiento_(programaci%C3%B3n_orientada_a_objetos))

Significa reunir todos los elementos que pueden considerarse pertenecientes a una misma entidad, al mismo nivel de abstracción.

**Polimorfismo**

Comportamientos diferentes, asociados a objetos distintos, pueden compartir el mismo nombre; al llamarlos por ese nombre se utilizará el comportamiento correspondiente al objeto que se esté usando. O, dicho de otro modo, las referencias y las colecciones de objetos pueden contener objetos de diferentes tipos, y la invocación de un comportamiento en una referencia producirá el comportamiento correcto para el tipo real del objeto referenciado.

**Herencia**

Las clases no se encuentran aisladas, sino que se relacionan entre sí, formando una jerarquía de clasificación. Los objetos heredan las propiedades y el comportamiento de todas las clases a las que pertenecen.

**Modularidad**

Se denomina "modularidad" a la propiedad que permite subdividir una aplicación en partes más pequeñas (llamadas módulos), cada una de las cuales debe ser tan independiente como sea posible de la aplicación en sí y de las restantes partes.

**Principio de ocultación**

Cada objeto está aislado del exterior, es un módulo natural, y cada tipo de objeto expone una "interfaz" a otros objetos que especifica cómo pueden interactuar con los objetos de la clase.

**Recolección de basura**

La recolección de basura (garbage collection) es la técnica por la cual el entorno de objetos se encarga de destruir automáticamente, y por tanto desvincular la memoria asociada, los objetos que hayan quedado sin ninguna referencia a ellos

**2. Elaborar informe de lectura, mapa conceptual o cuadro comparativo acerca de: clase, atributos, abstraccion, y polimorfismo.**

|  |  |  |  |
| --- | --- | --- | --- |
| **Clase** | **Atributos** | **Abstracción** | **Polimorfismo** |
| La clase es un modelo o prototipo que define las variables y métodos comunes a todos los objetos de cierta clase. También se puede mencionar que una clase es una plantilla genérica para un conjunto de objetos de similares características. Una clase define el estado y el comportamiento que todos los objetos creados a partir de esa clase tendrán. | Es una característica de un objeto, que ayuda a definir su estructura y permite diferenciarlo de otros objetos. Se define con un identificador y un tipo, el cual indica los valores que puede almacenar. El conjunto de valores de los campos definen el estado del objeto. | La abstracción es un proceso de interpretación y diseño que implica reconocer y enfocarse en las características importantes de una situación u objeto, y filtrar o ignorar todas las particularidades no esenciales. | Son comportamientos diferentes, asociados a objetos distintos, pueden compartir el mismo nombre; al llamarlos por ese nombre se utilizará el comportamiento correspondiente al objeto que se esté usando. |

**3. Crear una clase pública, con atributos y métodos.**

