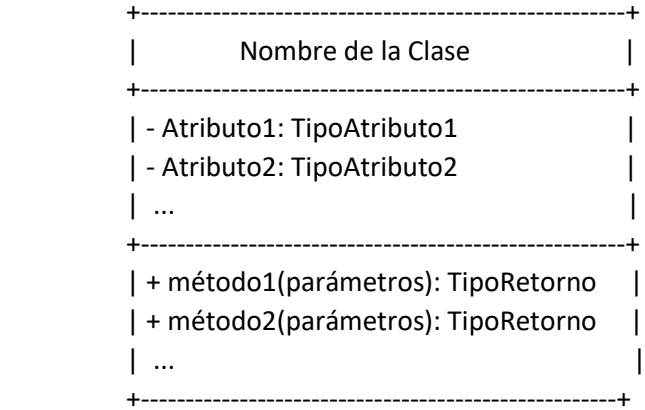
Trabajo practico N°1

1. TAD (Tipo Abstracto de Datos) es un concepto que se refiere a una estructura de datos abstracta definida por un conjunto de operaciones permitidas y sus comportamientos asociados, sin revelar los detalles de cómo se implementan esas operaciones. Es decir, un TAD define un tipo de dato mediante una interfaz, especificando qué operaciones se pueden realizar con ese tipo de dato y qué resultados se pueden esperar de esas operaciones, pero sin especificar cómo se llevan a cabo internamente esas operaciones.
   Un TAD proporciona una manera de abstraer la representación de los datos y las operaciones sobre esos datos, lo que permite a los programadores trabajar a un nivel más alto de abstracción y facilita el diseño modular y la reutilización de código.

2. El encapsulamiento se produce dentro de la implementación del Tipo Abstracto de Datos (TAD). Es una de las características fundamentales de la programación orientada a objetos (POO) que permite ocultar los detalles internos de un objeto y solo exponer una interfaz pública consistente en los métodos y propiedades relevantes para interactuar con ese objeto.
   En el contexto de un TAD, el encapsulamiento implica que la estructura interna de los datos y los detalles de implementación de las operaciones definidas en el TAD están ocultos al usuario o cliente del TAD. Solo se permite el acceso a través de los métodos y funciones proporcionados por la interfaz pública del TAD.

3. Semejanzas y diferencias entre funciones, procedimientos y métodos:
   - **Semejanzas:**
     - Todos representan unidades de código: Tanto las funciones, procedimientos y métodos son unidades de código que realizan una tarea específica.
     - Encapsulación de la lógica: Cada uno encapsula un conjunto de instrucciones que se ejecutan secuencialmente para realizar una tarea específica.
     - Reutilización de código: Permiten la reutilización de código al llamarlos o invocarlos desde diferentes partes del programa.
   - **Diferencias**:
     - Funciones: Son unidades de código que devuelven un valor después de su ejecución. Se utilizan para realizar un cálculo y devolver un resultado.
     - Procedimientos: Son unidades de código que realizan una tarea, pero no devuelven un valor.
     - Métodos: Son funciones o procedimientos asociados a un objeto o una clase en la programación orientada a objetos (POO).

4. UML (Unified Modeling Language) es un lenguaje de modelado visual utilizado en el desarrollo de software para especificar, visualizar, construir y documentar los artefactos de un sistema de software. Proporciona un conjunto de notaciones y diagramas estandarizados que permiten a los desarrolladores y diseñadores comunicarse de manera efectiva entre sí, así como con los stakeholders del proyecto.
   Una de las partes fundamentales en UML es la clase, que se utiliza para representar una abstracción de un concepto del dominio del problema en el sistema. Una clase en UML se representa de la siguiente manera:



Donde:

- Nombre de la Clase es el nombre de la clase que se está representando.
- AtributoX son los atributos de la clase, con su respectivo tipo de dato.
- métodoX son los métodos de la clase, con su respectiva lista de parámetros y tipo de retorno.
- Los prefijos +, - y # indican la visibilidad del elemento:
- +: Público (los miembros son accesibles desde cualquier parte del sistema).
- -: Privado (los miembros solo son accesibles dentro de la propia clase).
- #: Protegido (los miembros son accesibles dentro de la propia clase y sus subclases).

5. Marcar con cruz.

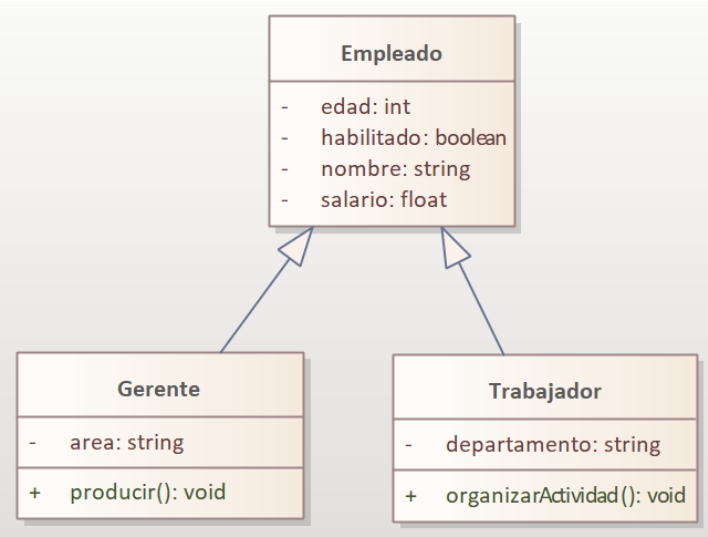
	No se aplica a clases	Solo se aplica a atributos	Solo se aplica a clases	Se aplica a atributos, métodos y clases
Public				X
Private	X	X		
Protected				X
Static				X
Final				X
Primera letra en minúscula	X	X		
Primera letra en mayúscula			X	

6. - Un constructor…
   - Es el método principal para ejecutar un programa. F
   - Crea instancias. V
   - Devuelve el valor de un atributo privado. F
   - Tiene sentencia return. F
   - Siempre existe uno por defecto, sin parámetros ni inicializaciones de atributos. V
   - Se puede sobreescribir. F
   - Se puede sobrecargar. V
   - Su nombre se escribe con mayúscula. V
   - Su calificador de acceso es static. F
   - Su tipo de devolución no se indica y corresponde a la clase. V

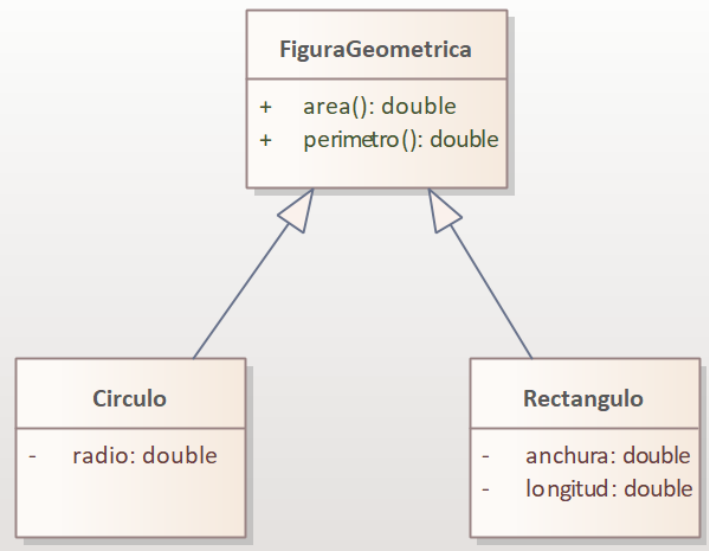
   - Un método…
   - Puede tener múltiples parámetros con el mismo nombre, siempre y cuando tengan tipos diferentesF
   - Puede sobrecargarse. F
   - Puede sobreescribirse. V
   - Puede ser static. V
   - Puede ser tanto public como protected, pero no private. F
   - Un método puede tener un modificador de acceso final V

7. Calificadores de acceso. Completa.
   a. Se necesita que cualquiera pueda acceder al color de un vehículo. Entonces, declaro color como: public
   b. Se necesita que color se pueda acceder a través no sólo de vehículo, sí no ahora también de Buses, y como todos sabemos un bus es un tipo de vehículo, entonces también deberá tener acceso a color. Entonces, declaro color como: protected
   c. Se necesita que color se pueda acceder solamente para vehículo. Entonces, declaro color como: private

8. UML



9. UML



10. UML