1. ¿Qué es un TAD?

**Un Tipo de Dato Abstracto es un modelo que define valores y las operaciones que se pueden realizar sobre ellos**.

1. ¿Dónde se produce el encapsulamiento?

**El encapsulamiento se produce al declarar los atributos en una clase.**

1. ¿Cuáles son las semejanzas y las diferencias entre funciones, procedimientos y métodos?

|  |  |  |  |
| --- | --- | --- | --- |
| **Característica** | **Funciones** | **Procedimientos** | **Métodos** |
| Estructura | Bloque de código que realiza una tarea especifica | Bloque de código que realiza una tarea especifica | Bloque de código que realiza una tarea especifica |
| Reutilización de código | Se puede reutilizar | Se puede reutilizar | Se puede reutilizar |
| Retorno de valor | Devuelven un valor | NO devuelven un valor | NO devuelven un valor |
| Contexto | Pueden ser independientes y no necesitan estar asociadas a una clase o instancia | Pueden ser independientes o asociados a un modulo | Están asociadas con clases y objetos en la programación orientada a objetos |
| Visibilidad de los datos | No tienen acceso a datos internos de objetos o clases | Pueden acceder y modificar los datos internos de objetos o clases | Pueden acceder y modificar los datos internos de objetos o clases |
| Encapsulamiento | No forman parte de la encapsulación de objetos | No forman parte de la encapsulación de objetos | Pueden formar parte de la encapsulación de objetos |

1. ¿Qué es UML? ¿Y cómo se representa una clase en ese lenguaje?

**Es un lenguaje de modelado, de propósito general, usado para la visualización, especificación, construcción y documentación de sistemas orientados a objetos.**

Representación de una clase:

-legajo

-nombre

-domicilio

Alumno

1. Marcar con una 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 |
| Protected |  |  |  | X |
| Static |  |  | X |  |
| Final |  |  |  | X |
| Primera letra en minúscula |  | X |  |  |
| Primera letra en mayúscula |  |  | X |  |

1. Verdadero o falso

-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 diferentes. F
* Puede sobrecargarse. F
* Puede sobreescribirse. V
* Puede ser static. F
* Puede ser tanto public como protected, pero no private. V
* Un método puede tener un modificador de acceso final. F

1. Calificadores de acceso. Completa
2. Se necesita que cualquiera pueda acceder al color de un vehículo. Entonces, declaro color como: **PUBLIC**
3. Se necesita que color se pueda acceder a través no solo de vehículo, si 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**
4. Se necesita que color se pueda acceder solamente para vehículo. Entonces, declaro color como: **PRIVATE**

Ejercicios 8,9 y 10 en GitHub

**Enlace al repositorio:** <https://github.com/Vecxzz/Programacion-IIB>