**Trabajo Práctico 1: Programación II**

**Renzo Di Laudo – Parte De teoría**

**1. ¿Qué es un TAD?**

Un tipo de dato abstracto (TAD) es un modelo que define valores y las operaciones que se pueden realizar sobre ellos. Y se denomina abstracto ya que la intención es que quien lo utiliza, no necesita conocer los detalles de la representación interna o bien cómo están implementadas las operaciones.

**2. ¿Dónde se produce el encapsulamiento?**

El encapsulamiento es un principio de la POO que se refiere a la ocultación de los detalles de implementación de una clase y la exposición controlada de sus atributos y métodos. Se produce dentro de las mismas clases.

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

Las funciones son un conjunto de instrucciones encapsulados en un bloque, suelen recibir parámetros, realizan operaciones y retornan un valor. En java usan el modificador Static.

Por otro lado, los métodos pueden realzar exactamente las mismas tareas que las funciones y comportarse de igual manera, la diferencia es que los métodos son exclusivos de un objeto o clase, mientras que una función existe por si sola.

Y los procedimientos a diferencia de los anteriores, no retornan nada, usan el modificador void.

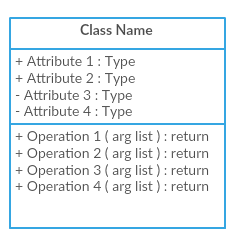
**4. ¿Qué es UML? ¿Y cómo se representa una clase en ese lenguaje?**

UML (Unified Modeling language) es un lenguaje estándar utilizado para visualizar, especificar, construir y documentar los artefactos de un sistema de software. Sirve para diseñar y representar sistemas complejos de manera visual.

Una clase en UML se representa mediante un rectángulo dividido en 3 secciones horizontales:

1. Parte superior: Nombre de la clase
2. Parte media: Atributos
3. Parte inferior: Métodos

Ejemplo:



**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 |
| Protected | X | X |  |  |
| Static |  |  |  | X |
| Final |  |  | X |  |
| Primera letra en minúscula |  | X |  |  |
| Primera letra en mayúscula |  |  | X |  |

**6. Verdadero o falso**

- Un constructor…

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

- Un método…

* Puede tener múltiples parámetros con el mismo nombre, siempre y cuando tengan tipos diferentes. **- Falso**
* Puede sobrecargarse. **- Verdadero**
* Puede sobreescribirse. **– Verdadero**
* Puede ser static. **- Verdadero**
* Puede ser tanto public como protected, pero no private. **- Falso**
* Un método puede tener un modificador de acceso final. **- Verdadero**

**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**

**a-** Se necesita que color se pueda acceder solamente para vehículo. Entonces, declaro color como: **private**