**Instituto Superior de Formación Técnica Nº 151 logo151-trans.gif  
Carrera: Analista de Sistemas  
1 Año. Algoritmos y Estructuras de Datos I**

|  |  |
| --- | --- |
| **Trabajo Práctico Nº 4** | **Unidad 4** |
| **Modalidad:** Semi-Presencial | **Estratégica Didáctica:** Trabajo Grupal. |
| **Metodología de Desarrollo:** acordar | **Metodología de Corrección:** acordar docente |
| **Carácter de Trabajo:** Obligatorio – con Nota | **Fecha Entrega:** A confirmar por el Docente. |

**Marco Teórico:**

1. ¿Qué es abstracción en programación?
2. ¿Qué relación hay entre Modelo y Abstracción?
3. ¿Por qué usaría un “Proceso de Abstracción” para resolver un Problema?
4. ¿Qué es la descomposición en el proceso de Abstracción?
5. Asocie abstracción con Descomposición
6. Explicar los Mecanismos de Abstracción
7. Explicar los Tipos de Abstracción
8. ¿Qué entiende por TDA?
9. ¿Qué mejoras aportan los TDA?
10. Explicar la Visión Externa y la Visión Interna de los TDA
11. ¿Qué es Encapsulamiento?
12. ¿Qué entiende Por Interface?

Lic. Oemig José Luis.

**Marco Teórico:**

1. ¿Qué es abstracción en programación?

Se entiende por abstracción a la operación intelectual que ignora selectivamente partes de un todo para facilitar su comprensión. El hecho de centrar la mirada en los aspectos más generales permite que la solución propuesta no sea solo para un problema sino para una familia de problemas.

1. ¿Qué relación hay entre Modelo y Abstracción?

Un Modelo es el resultado de una Abstracción. De hecho hay autores que hablan de Abstracción en dos sentidos: el primero, como se comentó en el punto 1 es decir como una acción (verbo), el segundo es como sustantivo. En este último sentido, el Modelo es justamente sinónimo de Abstracción.

1. ¿Por qué usaría un “Proceso de Abstracción” para resolver un Problema?

Por lo comentado en el final del punto 1: porque el Proceso de Abstracción hará posible que el Modelo de Solución que se plantee permita resolver no sólo el problema que se plantea sino toda una familia de problemas que compartan las características generales de aquel.

1. ¿Qué es la descomposición en el proceso de Abstracción?

La Descomposición en el proceso de Abstracción se refiere al proceso de identificación de todas aquellas estructuras (con sus respectivos comportamientos) que pueden funcionar de manera independiente y que se pueden utilizar / sincronizar para resolver no sólo el problema que se nos plantea sino una familia de problemas que comparta todas o algunas de estas partes.

1. Asocie abstracción con Descomposición

Es justamente el proceso de abstracción el que permite la descomposición más adecuada ya que aquellas partes individualizadas con la suficiente generalidad permiten una mayor reutilización.

1. Explicar los Mecanismos de Abstracción

**Abstracción por parametrización**. Se introducen parámetros para abstraer un número infinito de implementaciones/especializaciones.

Ejemplo: Cálculo de cos(x) : Podremos calcular el coseno de 30º, 27º 82º, etc con sólo darle diferentes valores (especializaciones) al parámetro ‘x’.

**Abstracción por especificación**. Permite abstraerse de la implementación concreta de un procedimiento asociándole una descripción precisa de su comportamiento.

Ejemplo: double sqrt(double a);

requisitos: a > 0;

efecto: devuelve una aproximación de la raíz cuadrada de a.

La **especificación** es un comentario lo suficientemente definido y explícito como para poder usar el procedimiento sin necesitar conocer otros elementos.

1. Explicar los Tipos de Abstracción

**1. Abstracción Procedimental**. Se define (encapsula) un conjunto de operaciones (procedimientos) que se comporta como una sola operación.

**2. Abstracción de Datos (TDA)**. Tenemos un conjunto de datos y un conjunto de operaciones que caracterizan el comportamiento del conjunto. Las operaciones están vinculadas a los datos del tipo.

**3. Abstracción de Iteración**. Abstracción que permite trabajar sobre colecciones de objetos sin tener que preocuparse por la forma concreta en que se organizan.

1. ¿Qué entiende por TDA?

Un **Tipo de Dato Abstracto (TDA)** es una entidad abstracta formada por un conjunto de datos y una colección de operaciones asociadas a dichos datos.

1. ¿Qué mejoras aportan los TDA?

Las mejoras aportadas por los TDA son múltiples:

En principio apuntan a representar de una manera más fiel todo lo concerniente al espacio del problema que los originó. Otro de los pilares es que el usuario de un TDA necesita centrarse en *qué* hace éste y no *cómo* lo hace. Esto no sólo le permite al programador abstraerse de la implementación sino que es más eficiente a la hora de actualizar un TDA debido a que se puede mejorar el funcionamiento sin tener que cambiar la forma en que se implementa. Por otra parte, en grandes proyectos, se hace absolutamente imprescindible el uso de los TDA dado que, además de las ventajas enumeradas, le dan mayor robustez y legibilidad al código.

1. Explicar la Visión Externa y la Visión Interna de los TDA

La Visión Externa se refiere a la especificación.

La Visión Interna hace referencia a la representación e implementación.

Las ventajas de esta separación son:

* Se puede cambiar la visión interna sin afectar a la externa.
* Facilita la labor del programador permitiéndole concentrarse en cada fase por separado.

1. ¿Qué es Encapsulamiento?

El Encapsulamiento es el proceso de almacenar en una misma sección los elementos de una abstracción que constituyen su estructura y su comportamiento; sirve para separar el interfaz contractual de una abstracción y su implementación.

1. ¿Qué entiende por Interface?

Interfaz es el enlace de conexión en un límite compartido que permite que sistemas independientes se satisfagan y actúen o se comuniquen entre sí. También es la descripción formal del propósito de un subprograma y el mecanismo para comunicarse con él.