

Tarea 4

T_ipos de D_atos A_bstractos



Tipos de **D**atos **A**bstractos (TDA) / **A**bstract **D**ata **T**ypes (ADT)

Objetivo de la tarea:

Implementación de TDA usando clases para reutilizar software

Estudio

Trabajo

Responsabilidad

Honestidad

¡Y ante todo DIOS!



INSTRUCCIONES:

- Nombres de los programas:
 - Programa 1: Tarea4_su_nombre_Clase_Estudiante (50 puntos)
 - Programa 2: Tarea4_su_nombre_Clase_Tiempo (50 puntos)
- Enviar un archivo comprimido (Tarea4_su_nombre.zip) con todos los objetos necesarios para ejecutar los 2 programas.
- Recuerde usar buenas prácticas de programación como nombres significativos, reutilización de código (funciones), documentación interna, eficiencia, etc.
- No valide datos, estos se darán según las especificaciones.
- Enviar al tecDigital / EVALUACIONES / TAREAS. Verifique el envío al tecDigital.
- Fecha de entrega: 25 de setiembre, 11pm
- Requisito para calificar la tarea:
 - Tener documentación interna, al menos en cada función debe poner comentarios de lo que hace la función, sus entradas y salidas.

RECOMENDACIONES:

- Lea cuidadosamente el problema a resolver (primer paso de la metodología de solución de problemas: entender el problema).
- En la etapa de diseño del algoritmo se sugiere hacer un esquema donde determine el comportamiento del algoritmo para luego proceder con su desarrollo en la computadora.

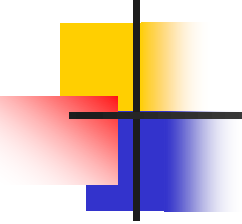


■ Programa 1: Clase Estudiante

- Definir interfaz: copie el encabezado de la clase Estudiante y genere el archivo de encabezado: Estudiante.h
- Definir implementación: copie la implementación de la clase Estudiante y genere el archivo de implementación de la clase: Estudiante.cpp

Haga un programa cliente que use esta clase para:

- Crear objetos (al menos 4 objetos)
- Asignar valores a esos objetos (puede usar datos constantes)
- Almacenar los objetos en un vector (usando <vector>)
- Hacer una función que tome los objetos del vector y los imprima accediendo directamente las variables miembro de Estudiante, use una función friend.



■ Programa 2: Clase Tiempo

- Definir interfaz: archivo de encabezado (.h)
- Definir implementación: archivo de implementación (.cpp)

Haga un programa cliente que use esta clase para

- Crear dos objetos de esa clase: tiempo1, tiempo2.
- Asignar valores a esos objetos (puede usar datos constantes).
- Usar el operador + sobrecargado para sumar esos objetos:
$$\text{suma_tiempos} = \text{tiempo1} + \text{tiempo2};$$
- Imprimir el valor retornado por suma_tiempos con el método respectivo
- Usar la función tiempo_mayor e imprimir su resultado



clase Tiempo

- Datos miembro (Atributos)
 - horas (de 00 a 23)
 - minutos (de 00 a 59)
 - segundos (de 00 a 59)



■ Funciones miembro (Métodos)

- Constructor: hacer un constructor para asignar valores a todos los atributos
- `imprime_tiempo`: en formato `hh:mm:ss`
- Sobrecargar operador `+` (operator `+`): suma dos objetos y retorna otro objeto conteniendo una hora que representa la suma de los tiempos en la operación. Todos los objetos van a ser de la clase `Tiempo`. Si el resultado es `>` a 24 horas retornar -1 en todos los datos del resultado. Ejemplo:

Objeto 1 → 13, 20, 30

Objeto 2 → 2, 10, 50

Tiempo retornado → 15, 31, 20

- `tiempo_mayor`: recibe dos tiempos, uno es el implícito (`this`) y otro por parámetro. Retorna otro objeto conteniendo la hora que es mayor. En el ejemplo anterior sería Objeto 1.