
FS-0432: Primer Parcial Práctico III-24

Martes 4 de Febrero, 2025

Fecha límite de Entrega: Martes 11 de Febrero a las 17:00

Instrucciones generales: Utilice cualquier recurso necesario para responder y resolver los siguientes problemas. Cada ítem indica que se debe entregar en cada caso. Los documentos se entregan en el apartado respectivo en mediación virtual, titulado "Primer Parcial Práctico III-24". El/la estudiante debe demostrar comprensión en los apartados pertinentes. Cualquier ítem que explícitamente utilice el término "explique", se refiere que además del código debe haber un análisis del método y de la implementación.

1 ÚNICA PARTE: DESARROLLO (100%)

Responda las siguientes preguntas de la forma más completa posible. Puntos intermedios se otorgan de acuerdo a la respuesta.

1. Programación orientada a objetos (65%):

Escriba una clase `Tuple`, la cual maneja recursos de memoria. Básicamente, `Tuple` define una estructura de datos que contiene N valores de punto flotante de doble precisión. Para esta parte del examen debe entregar dos archivos: `tuple.hpp`, `tuple.cpp` y `main.cpp`. Debe evitar que un usuario/a ponga el estado del objeto en un estado inconsistente, i.e., más/menos elementos en memoria de los que el están asignados en sus atributos. Explique en comentarios sobre la implementación de cada método, atributo y clase.

a) Genere tres archivos: (5%)

- `tuple.hpp`: este archivo contiene la declaración de la clase
- `tuple.cpp`: este archivo contiene la implementación de la clase
- `main.cpp`: este archivo invoca métodos y construye objetos

b) Declare una clase llamada `Tuple` que contiene dos atributos: El contenedor (un puntero a donde se ubican los objetos en memoria) y el número de elementos en el contenedor (5%)

c) Genere un constructor por defecto que inicializa la estructura a contener un valor: 0.0 (5%)

d) Genere un constructor personalizado que asigna la memoria e inicializa la estructura de datos a contener solo los valores 0.0 (5%)

- e) Genere un constructor copia apropiado (7.5%)
- f) Genere un operador de asignación apropiado (7.5%)
- g) Genere un destructor apropiado (5%)
- h) Sobrecargue el operador `+` que suma los elementos de dos `tuple`. Debe comprobar que los objetos a ser sumados son del mismo tamaño. Debe hacer esta comprobación con manejo de excepciones (5%)
- i) Sobrecargue el operador `-` que resta los elementos de dos `tuple`. Debe comprobar que los objetos a ser sumados son del mismo tamaño. Debe hacer esta comprobación con manejo de excepciones (5%)
- j) Sobrecargue el operador `*` que multiplica elemento por elemento de dos `tuple`. Debe comprobar que los objetos a ser sumados son del mismo tamaño. Debe hacer esta comprobación con manejo de excepciones (5%)
- k) Sobrecargue el operador `/` que multiplica elemento por elemento de dos `tuple`. Debe comprobar que los objetos a ser sumados son del mismo tamaño. Debe hacer esta comprobación con manejo de excepciones (5%)
- l) Realice pruebas de los nueve métodos anteriores en el `main.cpp` (5%)

2. Integración (35%):

Resuelva la integral

$$I = \int_0^{\infty} dt e^{-t^2}. \quad (1.1)$$

Para esto, puede ser útil la transformación $z = t/(1+t)$. Realice su estimación utilizando cuadratura Gaussiana y realice un gráfico de I como función de N , desde $N = 2$ hasta $N = 50$. Se entrega un jupyter-notebook `gauss.jpnb` con su método, análisis e implementación.