ESCUELA DE CIENCIAS DE LA
COMPUTACIÓN E INFORMÁTICA
UNIVERSIDAD DE COSTA RICA

CI-0113 Programción 2
II semestre 2018
Prf.: Alan Calderón Castro

Segundo laboratorio guiado de programación

Objetivos

Al completar este laboratorio el estudiante debería ser capaz de:

1. Explicar los conceptos básicos sobre plantillas.
2. Explicar principios esenciales del diseño de plantillas.
3. Programar y depurar plantillas.

Descripción del problema

Se deberá construir una plantilla de clase que represente una matriz rala o matriz esparcida. Una matriz rala es aquélla que tiene un “alto porcentaje” de elementos igual a cero, por lo que se omite guardar los ceros para ahorrar espacio y sólo se representan los valores diferentes de cero. La plantilla de clase “matriz_rala< T >” permitirá almacenar en general valores de tipos numéricos estándar tales como: int, double, float, long int, etc, pero también permitirá almacenar valores de tipos numéricos no estándar, como podrían “Racional” o “Complejo” para representar números racionales (fracciones) o números complejos. Se deberá aplicar el programa de pruebas provisto para asegurar el funcionamiento correcto de la plantilla principal “matriz_rala< T >” y las otras dos plantillas asociadas: “fila_rala< T >” y “itr_matriz_rala< T >” que representan una fila con muchos ceros y un iterador que permitirá recorrer los valores diferentes de cero de una matriz rala.

De acuerdo con lo anterior usted deberá programar las siguientes plantillas de clase:

Nombre de la clase	Función que cumple
“matriz_rala< T >”	Representa una matriz rala genérica, de dimensiones MxN (M es la cantidad de filas y N la cantidad de columnas) que permite almacenar cualquier tipo numérico estándar o no estándar.
“fila_rala< T >”	Representa una fila de una matriz rala. La intención es que esta plantilla sólo pueda usarse con “matriz_rala< T >”, NUNCA por separado.
“itr_matriz_rala< T >”	Representa un iterador que recorre todas los valores **diferentes de cero** de una matriz rala. Sigue el orden usual de todo recorrido de matriz: de fila 0 a fila M – 1, y en cada fila de columna 0 a columna N – 1.

Procedimiento

Para aplicar la metodología “pruebas-primero” se deberá:
1. programar la plantilla “fila_rala< T >”,
2. por cada constructor o método de la plantilla “matriz_rala< T >”, se aplicarán las pruebas correspondientes del programa de pruebas,
3. programar la plantilla de iterador “itr_matriz_rala< T >”,
4. aplicar las pruebas correspondientes al iterador.

Criterios de evaluación

La nota final de su trabajo dependerá de si en su programa se:

1. Respetan las reglas de estilo del código: márgenes, nombres de objetos empiezan en minúsculas, nombres de clases empiezan en mayúsculas, nombres de métodos también empiezan en minúscula pero se usan mayúsculas para concatenar palabras, comentarios para los atributos y variables de métodos. Formato automático dado por NetBeans.
2. El código es lógicamente lo más simple posible.
3. Implementan los métodos de manera eficiente.
4. Hace uso óptimo de la memoria aplicando las estructuras de datos discutidas en clase para cada una de las clases.
5. Respeta la división de responsabilidades entre el controlador-modelo de manera que sólo el main() se ocupa de la entrada de datos, generar mensajes de error, el despliegue de ciertos resultados por la "Consola", así como invocar a los demás objetos para que realicen todos los procesamientos necesarios.

Cronograma

1. Del 24 de octubre al 30 de octubre programación y depuración de las plantillas “matriz_rala< T >” y de “fila_rala< T >”.
2. Del 31 de octubre al 7 de noviembre, programar y depurar la plantilla “itr_matriz_rala< T >”.

Fecha de entrega: miércoles 7 de noviembre a las 23:55 por medio del enlace en el sitio del curso. SÓLO DEBERÁ SUBIR LOS ARCHIVOS DE CÓDIGO FUENTE (*.h y *.cpp) Y DATOS.

Evaluación

Criterio	%Prc
Eficacia comprobada y eficiencia de “matriz_rala< T >” y de “fila_rala< T >”	60
Eficacia comprobada y eficiencia de “itr_matriz_rala< T >”	40
Hasta 5/100 puntos extra por un buen reporte de errores cuando no funcione	

Notas importantes:
1. Este proyecto deberá realizarse idealmente y a lo más en parejas. NO SE ACEPTARÁ NINGÚN TRABAJO ELABORADO POR MÁS DE DOS PERSONAS.
2. Por CADA HORA de atraso en la entrega se restará 1/100 a la nota final obtenida.
3. A TODOS LOS ESTUDIANTES INVOLUCRADOS EN UN FRAUDE SE LES APLICARÁ EL ARTÍCULO #5 INCISO C DEL "Reglamento de Orden y Disciplina de los Estudiantes de la Universidad de Costa Rica".
4. NO SUBA ningún otro archivo que no sea de código fuente (*,h y *.cpp) o de datos para evitar la dispersión de virus. SI SU TRABAJO TIENE VIRUS, SERÁ PENALIZADO CON 20 puntos MENOS DE LA NOTA QUE SE LE ASIGNE.