

Descripción Proyecto Estructuras de Datos

Grupo 9: Jairo Rodríguez

Clase ProyectoESTD_1P (Main):

Se creó un atributo de tipo String llamado file para que guarde la ruta del archivo dentro de la carpeta recurso del proyecto, seguido de la instancia de la clase Cuadro para llamar a los métodos obtenerCuadro(), mostrarClusters() y llamar a la ventana de PintarCuadroVista.

Clase Cluster:

Se crearon los atributos para el número de cluster, numero de color, tamaño del cluster y List con las posiciones de los pixeles de cada cluster.

Clase Cuadro:

Los atributos matriz (arreglo doble) y clusters (ArrayList de objetos Cluster) son inicializados para darles uso en los métodos:

obtenerCuadro(): Lee el archivo Cuadro.txt, guarda el numero de filas y columnas y con un lazo for y separa cada número que está separado con comas y una vez generada la matriz, llama al método obtenerCuadro().

identificarClusters(): Va recorriendo la matriz en busca de clusters, para eso se utiliza un arreglo de booleanos para identificar que pixeles han sido colocados en un cluster (recorrido en filas y columnas) y llama a llenarCluster() para agrupar pixeles por color y se lo añade a la lista de clusters. También se define el orden a ser pintado ya que se solicita que sea por tamaño de mayor a menor (hace uso de sort comparando los tamaños, aunque mi implementación falla para los clusters con igual tamaño)

llenarCluster(): Método llamado desde identificarClusters() busca los pixeles que pertenecen al cluster por su color y posición en la matriz, para eso se utiliza Backtracing con pila para almacenar los pixeles que pertenecen al cluster que se está analizando en identificarCluster() moviéndose a la izquierda, derecha, arriba y abajo (va almacenando los pixeles dependiendo de los parámetros enviados en identificarCluster). Crea un objeto tipo Cluster haciendo uso del constructor en su clase, y lo devuelve para usarlo en el método anterior.

getClusters() retorna una lista de clusters identificados de la matriz que será usado después para ser pintado.

imprimirClusters(): imprime en consola cada cluster de tipo Cluster, por lo que hace referencia al toString() de la clase con el mismo nombre para mostrar el número de cluster, numero de color y tamaño;

`getMatriz()`: devuelve la matriz cargada desde el archivo para ser usada posteriormente en `PintarCuadroVista`.

Clase `PintarCuadroVista`:

Se crea una instancia de la clase `Cuadro`, un `JPanel` para la ventana del programa, `JLabel` para mostrar el número de cluster a pintar, combobox con la lista de colores y un botón para pintar después de seleccionar un color y un numero entero que ira aumentando en uno después de pintar cada cluster. Se establecen tamaños para la ventana y sus elementos. Con el método `paintCluster()` se pinta cada cluster con el color seleccionado del `HashMap` y con el método `drawCuadro` se representa la matriz en la ventana.

Observación: El programa arroja `OutOfMemoryError` después de que implementé el método `llenarCluster()`.