**UNIVERSIDAD DE SAN CARLOS DE GUATEMALA**

**FACULTAD DE INGENIERÍA**

ESCUELA DE CIENCIAS Y SISTEMAS

ING.

1ER SEMESTRE 2022

**U DRAWING PAPER**

PROYECTO FASE 1, ESTRUCTURAS DE DATOS

INGENIERÍA EN CIENCIAS Y SISTEMAS

ABNER ALEXANDER TOL TURCIOS 201314462

**GUATEMALA, 20 DE FEBRERO, 2022**

**Manual de Técnico**

**U Drawing Paper**

**Requisitos de sistema**

* Windows 10 o superior
* 2 GB de ram
* Intel Celeron o equivalente AMD
* 100 MB HDD

**Librerías e instalaciones**

* Json-Simple 1.0
* Graphviz
* Java 7 o superior (JDK)
* Netbeans 10 o superior(Opcional)

**Descripción de la solución**

La aplicación combina una interfaz gráfica para el menú principal y la impresión en consola para los resultados/estados de cada módulo. También se incluye una función para generar comandos que pueden ser interpretados por la librería Graphviz.

**Configuraciones**

* El archivo Jar se puede ejecutar desde cualquier directorio
* El proyecto para netbeans se puede instalar en cualquier directorio
* Debe existir una carpeta con el nombre “Graphiz” y estar bajo la ruta “…Usuario\Documents\Graphviz”

**Descripción general de la estructura del código**

Se debe tomar en cuenta, para todos los cambios futuros, que todas las listas son estructuras de datos, (listas enlazadas, doblemente enlazadas, circulares, listas de listas)

**Clases principales**

* UDrawingPaper.java:
  + Main: Crea las clase paso, y ejecuta el menú principal.
* Paso.java:

Esta clase tiene conexión con casi todas las clases secundarias, pues es la que gestiona todos los flujos y subflujos

* + EjecutarPaso():
  + CrearUsuarios(): Ejecuta lo métodos de la clase ReadJsonFile para crear los usuarios
  + AgregarUsuarioAColaRecepción(): Genera un número aleatorio para seleccionar un nuevo usuario de la clase ReadJsonFile, y lo asigna a la lista de usuarios en cola de recepción.
  + HacerPasarSiguienteUsuarioAVentanilla(): Recorre la lista de ventanillas, cuando encuentra una vacía, toma el usuario más antiguo en cola de recepción y lo ingresa en dicha ventanilla
  + CrearVentanillas(): Muestra un menú en pantalla para ingresar la cantidad desada de ventanillas y las crea
  + ImprimirListadeUsuariosRecepcion(): Llama al método imprimirLista de la clase ListaColaRecepción
  + ImprimirListaVentanillas(): Llama al método imprimirLista de la clase ListaVentanillas
  + AgregarImagenAVentanillaCorrespondiente(): Toma una imagen del usuario en cierta ventanilla y lo ingresa a la lista de dicha ventanilla
  + ImprimirListaClientesEnEspera(): Llama al método imprimirLista de la clase ListaClientesEnEspera
  + ImprimirListaDeImagenesEnVentanilla(): Llama al método imprimirLista de la clase Ventanilla para ver las ventanillas y sus respectivas imágenes
  + ImprimirListaDeClientesAtendido()s: Llama al método imprimirLista de la clase ListaClientesAtendidos

**Clases secundarias:**

* ListaClientesEnEspera.java:
  + nodo: Clase anidada, contiene el nodo, que a su vez contendrá las referencias hacia los demás nodos.
  + insertarFinal(): Toma un nodo y lo ingresa a la lista asociada para esta clase.
  + imprimirLista(): Recorre la lista e imprime la información contenida
  + tamañoLista(): Devuelve el tamaño actual de la lista, el cual es útil para determinar la dimensión de la misma en recorridos, etc.
  + construirComandoGrafo(): Recorre la lista completa, generando a su vez el código necesario para el comando Graphviz
* Imagen.java:
  + Imagen(): Constructor, recibe el id del usuario, la ruta de la imagen y el timpo de imagen para crear un Nodo, elemento.
  + getAllInfo(): Devuelve toda la información del elemento, en forma de una cadena de texto
  + getTipoImagen(): Devuelve el tipo de imagen, BW para blanco y negro, o Color para color.
  + getIDImagen: Devuelve el id asociado al usuario propietario de la imagen
* impresoraByN.java:
  + nodo: Clase anidada, contiene el nodo, que a su vez contendrá las referencias hacia los demás nodos.
  + insertarFinal(): Toma un nodo y lo ingresa a la lista asociada para esta clase.
  + imprimirLista(): Recorre la lista e imprime la información contenida
  + tamañoLista(): Devuelve el tamaño actual de la lista, el cual es útil para determinar la dimensión de la misma en recorridos, etc.
  + construirComandoGrafo(): Recorre la lista completa, generando a su vez el código necesario para el comando Graphviz
* ImpresoraColor.java:
  + nodo: Clase anidada, contiene el nodo, que a su vez contendrá las referencias hacia los demás nodos.
  + insertarFinal(): Toma un nodo y lo ingresa a la lista asociada para esta clase.
  + imprimirLista(): Recorre la lista e imprime la información contenida
  + tamañoLista(): Devuelve el tamaño actual de la lista, el cual es útil para determinar la dimensión de la misma en recorridos, etc.
  + construirComandoGrafo(): Recorre la lista completa, generando a su vez el código necesario para el comando Graphviz
* ListaClientesAtendidos.java:
  + nodo: Clase anidada, contiene el nodo, que a su vez contendrá las referencias hacia los demás nodos.
  + insertarFinal(): Toma un nodo y lo ingresa a la lista asociada para esta clase.
  + imprimirLista(): Recorre la lista e imprime la información contenida
  + tamañoLista(): Devuelve el tamaño actual de la lista, el cual es útil para determinar la dimensión de la misma en recorridos, etc.
  + construirComandoGrafo(): Recorre la lista completa, generando a su vez el código necesario para el comando Graphviz
* ListaClientesEnEspera.java:
  + nodo: Clase anidada, contiene el nodo, que a su vez contendrá las referencias hacia los demás nodos.
  + insertarFinal(): Toma un nodo y lo ingresa a la lista asociada para esta clase.
  + imprimirLista(): Recorre la lista e imprime la información contenida
  + tamañoLista(): Devuelve el tamaño actual de la lista, el cual es útil para determinar la dimensión de la misma en recorridos, etc.
  + construirComandoGrafo(): Recorre la lista completa, generando a su vez el código necesario para el comando Graphviz
* ListaColaRecepción.java:
  + nodo: Clase anidada, contiene el nodo, que a su vez contendrá las referencias hacia los demás nodos.
  + insertarFinal(): Toma un nodo y lo ingresa a la lista asociada para esta clase.
  + imprimirLista(): Recorre la lista e imprime la información contenida
  + tamañoLista(): Devuelve el tamaño actual de la lista, el cual es útil para determinar la dimensión de la misma en recorridos, etc.
  + construirComandoGrafo(): Recorre la lista completa, generando a su vez el código necesario para el comando Graphviz
* ListaImagen.java:
  + nodo: Clase anidada, contiene el nodo, que a su vez contendrá las referencias hacia los demás nodos.
  + insertarFinal(): Toma un nodo y lo ingresa a la lista asociada para esta clase.
  + imprimirLista(): Recorre la lista e imprime la información contenida
  + tamañoLista(): Devuelve el tamaño actual de la lista, el cual es útil para determinar la dimensión de la misma en recorridos, etc.
  + construirComandoGrafo(): Recorre la lista completa, generando a su vez el código necesario para el comando Graphviz
* ListaVentanilla.java:
  + nodo: Clase anidada, contiene el nodo, que a su vez contendrá las referencias hacia los demás nodos.
  + insertarFinal(): Toma un nodo y lo ingresa a la lista asociada para esta clase.
  + imprimirLista(): Recorre la lista e imprime la información contenida
  + tamañoLista(): Devuelve el tamaño actual de la lista, el cual es útil para determinar la dimensión de la misma en recorridos, etc.
  + construirComandoGrafo(): Recorre la lista completa, generando a su vez el código necesario para el comando Graphviz
* Usuario.java:
  + Usuario (): Constructor, recibe el id del usuario, su nombre, cantidad de imágenes a color y a blanco y negro, para crear un Nodo, elemento.
  + getAllInfo(): Devuelve toda la información del elemento, en forma de una cadena de texto
  + getUserId: Devuelve el id asociado al usuario
  + getusername: Devuelve el nombre asociado al usuario
  + getBWImages():Devuelve la cantidad de imágenes del usuario, asociadas a este tipo
  + getColorImages(): Devuelve la cantidad de imágenes del usuario, asociadas a este tipo
  + getListaImagen(): Devuelve la lista de imágenes asociadas al usuario, solo se ocupa este método en la clase ListaClientesEnEspera
* Ventanilla.java:
  + Ventanilla (): Constructor, recibe el código de ventanilla, el usuario, una lista vacía de imágenes, una referencia hacia listaClientesEnEspera y las referencias para las impresoras, todo esto para crear un Nodo, elemento.
  + getAllInfo(): Devuelve toda la información del elemento, en forma de una cadena de texto
  + getusername: Devuelve el nombre asociado al usuario
  + verificarusuario(): Devuelve la referencia del usuario activo o null si aún está vacía la ventanilla
  + recibirusuario(): Recibe el usuario que ocupará la ventanilla durante su descarga de imágenes
  + setTypes(): Actualiza los campos color y bw, correspondientes al número de imágenes de cada tipo
  + AgregarImagen(): Agrega un nodo imagen a ListaImagenes asociada a la ventanilla actual
  + VaciarVentanilla(): Elimina lista asociada a la ventanilla actual
  + EnviarImagenesAImpresoras(): Envía al usuario a ListaClientesEnEspera y sus imágenes a las impresoras, clasificándolas por color o blanco y negro según corresponda.
  + ImprimirListaImagenes():Recorre la lista e imprime la información contenida
  + ConstruirComandoGrafo():Recorre la lista completa, generando a su vez el código necesario para el comando Graphviz

**Otras clases**

* Graphviz.java:
  + RunBat: Ejecuta el bat que llama el compilador de Graphviz, el cual lee el archivo .dot que contiene el comando para generar la gráfica
  + OpenImage: Ejecuta un bat que abre la imagen generada por la librería graphviz
  + DelteImage: Ejecuta un bat Elmina la imagen anterior
  + WriteFile: Escribe un archivo .dot con el comando para Graphviz
* ReadJsonFile.java:
  + CreateUsers: Lee un archivo JSON que contiene la información para los usuarios que simularán el funcionamiento en la aplicación. Los almacena en un arraylist.
  + GetUsers: Recibe como parámetro un número, el cual representa la posición de un usuario en el arraylist, y devuelve dicho usuario.
  + GetSize: Devuelve el tamaño actual de la lista de usuarios

Acerca de:

Ultima versión estable 2.0 (al 20 de febrero de 2022)

Dudas/Comentarios

Contacto:

[2340994290301@ingenieria.usac.edu.gt](mailto:2340994290301@ingenieria.usac.edu.gt)

54798248