

Tarea08

Nombre alumno o alumna: Podadera González, Andrés Samuel

Fecha de entrega: martes, 28 de marzo de 2023, 23:40

Retroalimentación global de la actividad:

Hola, Andrés. Muy bien. Excelente trabajo. Has sido el primero en entregar la tarea. Tan solo algunos pequeños detalles que te he indicado para mejorar tu código. Ahora, a descansar durante la Semana Santa y a la vuelta nos esperan los entornos gráficos. Ánimo, que ya estamos terminando y vas estupendamente. ¡Enhorabuena!

Puntuación orientativa de la actividad: 10,00

RESULTADOS DE APRENDIZAJE EVALUADOS EN ESTA ACTIVIDAD

RA5 Realiza operaciones de entrada y salida de información, utilizando procedimientos específicos del lenguaje y librerías de clases

		Punt. Máx	Punt. Obt.	Punt. Final (sobre 10)
RA5.d	Se han utilizado ficheros para almacenar y recuperar información.	61	61,00	10,0
RA5.e	Se han creado programas que utilicen diversos métodos de acceso al contenido de los ficheros.	61	61,00	10,0

RA6 Escribe programas que manipulen información seleccionando y utilizando tipos avanzados de datos.

		Punt. Máx	Punt. Obt.	Punt. Final (sobre 10)
RA6.h	Se han identificado las clases relacionadas con el tratamiento de documentos XML.	12	12,00	10,0
RA6.i	Se han realizado programas que realicen manipulaciones sobre documentos XML.	12	12,00	10,0

CORRECCIÓN POR EJERCICIO/ACTIVIDAD/ELEMENTO

Ejercicio 1: Lectura/Escritura de una agenda de contactos en ficheros de texto.

Puntos de control (elementos evaluables)		Punt. Máx.	Punt. Obt.	CE	CE	CE	CE	CE	CE
PC1	Se construyen los flujos de entrada utilizando las clases apropiadas que nos permitan leer archivos de texto (FileReader y BufferedReader).	2	2,0	RA5.d	RA5.e				
PC2	Lectura de líneas desde el fichero 'ListadoAficiones.txt' (bucle).	2	2,0	RA5.d	RA5.e				
PC3	Lectura de líneas desde el fichero 'ListadoContactos.txt' y extracción de los datos (bucle).	4	4,0	RA5.d	RA5.e				
PC4	Se cierran los flujos de entrada (manualmente o bien mediante try-with-resources).	2	2,0	RA5.d	RA5.e				
PC5	Se construye el flujo de salida utilizando las clases apropiadas que nos permitan escribir en archivos de texto (FileWriter y PrintWriter).	1	1,0	RA5.d	RA5.e				
PC6	Escritura en el fichero 'BDCAgendaContactos.txt' con el formato solicitado en el enunciado.	5	5,0	RA5.d	RA5.e				
PC7	Se cierran los flujos de salida (manualmente o bien mediante try-with-resources).	1	1,0	RA5.d	RA5.e				
PC8	Se capturan las excepciones de Entrada/Salida mostrando mensajes apropiados (IOException y FileNotFoundException)	2	2,0	RA5.d	RA5.e				
PC9	El programa funciona correctamente, no le falta ningún elemento necesario y no incluye elementos extraños, innecesarios o sin sentido.	4	4,0	RA5.d	RA5.e				

Retroalimentación:

Importante: si haces cinco veces **linea.split**, estás llevando a cabo el mismo análisis cinco veces y generando cinco arrays de resultado. Lo "razonable" y "eficiente" habría sido declarar una variable de tipo array de String si inicializar y almacenar en ella el resultado de **linea.split**.

Ejercicio 2: Lectura/Escritura de objetos. Serialización.

Puntos de control (elementos evaluables)		Punt. Máx.	Punt. Obt.	CE	CE	CE	CE	CE	CE
PC1	Método escribir: se construyen los flujos de salida necesarios para escribir objetos serializables (FileOutputStream y ObjectOutputStream).	1	1,0	RA5.d	RA5.e				
PC2	Método escribir: se utiliza el método apropiado para llevar a cabo la escritura.	1	1,0	RA5.d	RA5.e				
PC3	Método escribir: tratamiento de las excepciones con un mensaje apropiado.	1	1,0	RA5.d	RA5.e				
PC4	Se cierran los flujos de salida (manualmente o bien mediante try-with-resources).	1	1,0	RA5.d	RA5.e				
PC5	Método leer: se construyen los flujos de entrada necesarios para leer objetos serializables (FileInputStream y ObjectInputStream)	1	1,0	RA5.d	RA5.e				
PC6	Método leer: se utiliza el método apropiado para llevar a cabo la lectura.	1	1,0	RA5.d	RA5.e				
PC7	Método leer: tratamiento de las excepciones con un mensaje apropiado.	2	2,0	RA5.d	RA5.e				
PC8	Método leer: devolución del objeto almacenado en el archivo binario.	1	1,0	RA5.d	RA5.e				
PC9	Se cierran los flujos de entrada (manualmente o bien mediante try-with-resources).	1	1,0	RA5.d	RA5.e				
PC10	Se puede trabajar correctamente con instancias de la clase CAgendaContactosIO en el programa Ejercicio2.	2	2,0	RA5.d	RA5.e				

Retroalimentación:

Es conveniente que también captures la posible excepción **ClassCastException**, por si lo que hubiera en el archivo no fuera un objeto serializado.

Ejercicio 3: Lectura/Escritura de objetos. XML.**Puntos de control (elementos evaluables)**

		Punt. Máx.	Punt. Obt.	CE	CE	CE	CE	CE	CE
PC1	Método escribir: serialización del objeto CAGendaContactos utilizando el método toXML de una forma adecuada.	1	1,0	RA6.h	RA6.i				
PC2	Método escribir: se construye el flujo de salida utilizando las clases apropiadas que nos permitan escribir la cadena en formato XML (FileWriter y PrintWriter).	1	1,0	RA6.h	RA6.i				
PC3	Método escribir: tratamiento de las excepciones con un mensaje apropiado.	1	1,0	RA6.h	RA6.i				
PC4	Se cierran los flujos de salida (manualmente o bien mediante try-with-resources).	1	1,0	RA6.h	RA6.i				
PC5	Método leer: se construyen los flujos de entrada necesarios para leer desde el archivo de texto que contiene el XML (FileReader y BufferedReader)	1	1,0	RA6.h	RA6.i				
PC6	Método leer: se utiliza el método apropiado para llevar a cabo la lectura y se almacena en una única cadena de caracteres (String).	1	1,0	RA6.h	RA6.i				
PC7	Método leer: des-serialización de la cadena que contiene la estructura XML por medio del método fromXML.	1	1,0	RA6.h	RA6.i				
PC8	Método leer: tratamiento de las excepciones con un mensaje apropiado.	1	1,0	RA6.h	RA6.i				
PC9	Método leer: devolución del objeto almacenado en el archivo de texto.	1	1,0	RA6.h	RA6.i				
PC10	Se cierran los flujos de entrada (manualmente o bien mediante try-with-resources).	1	1,0	RA6.h	RA6.i				
PC11	Se puede trabajar correctamente con instancias de la clase CAGendaContactosXML en el programa Ejercicio3.	2	2,0	RA6.h	RA6.i				

Retroalimentación:

Correcto.

Ejercicio 4: De imagen a texto**Puntos de control (elementos evaluables)**

		Punt. Máx.	Punt. Obt.	CE	CE	CE	CE	CE	CE
PC1	Método leerArchivomágenes: se construyen los flujos de entrada utilizando las clases apropiadas que nos permitan leer archivos de texto (FileReader y BufferedReader).	2	2,0	RA5.d	RA5.e				
PC2	Método leerArchivomágenes: lectura de líneas desde el fichero "ListadoCategorías.txt" y almacenamiento de los datos en el array "imágenes" (bucle).	2	2,0	RA5.d	RA5.e				
PC3	Método leerArchivomágenes: se cierran los flujos de entrada (manualmente o bien mediante try-with-resources).	2	2,0	RA5.d	RA5.e				
PC4	Método leerArchivomágenes: tratamiento de las excepciones con un mensaje apropiado.	2	2,0	RA5.d	RA5.e				
PC5	Método leerImagen: se construyen los flujos de entrada utilizando las clases apropiadas que nos permitan leer archivos binarios (FileInputStream, ImageInputStream y BufferedImage).	3	3,0	RA5.d	RA5.e				
PC6	Método leerImagen: lectura desde el fichero de imagen correspondiente a la categoría y obtención de las características de la imagen (altura y anchura).	2	2,0	RA5.d	RA5.e				
PC7	Método leerImagen: se recorre la imagen y se convierte cada píxel en un carácter en función del color. Los caracteres se van agregando a la cadena de caracteres final.	5	5,0	RA5.d	RA5.e				
PC8	Método leerImagen: se cierran los flujos de entrada.	3	3,0	RA5.d	RA5.e				
PC9	Método leerImagen: tratamiento de las excepciones con un mensaje apropiado.	2	2,0	RA5.d	RA5.e				
PC10	Método leerImagen: devolución de la cadena que representa la imagen.	1	1,0	RA5.d	RA5.e				
PC11	Se puede trabajar correctamente con instancias de la clase CagendaContactosImagen en el programa Ejercicio4.	2	2,0	RA5.d	RA5.e				

Retroalimentación:

Correcto.