

PRUEBA DE EVALUACIÓN CONTINUA 2

13 de DICIEMBRE de 2023

- Lee detenidamente y de forma completa cada uno de los problemas planteados y responde únicamente a lo que se te pide.
- Para la resolución de este examen no está permitido el uso de bucles ni la modularización del código. Para que los problemas obtengan la máxima puntuación, **habrá que tratar las excepciones obligatorias usando try/catch con sus excepciones específicas y mostrando mensajes legibles**, estas excepciones no pueden provocar que se aborte el programa con la traza del código. Además el código será claro, legible, deberán explicarse usando comentarios y estará correctamente tabulado.
- Una vez descargado el enunciado no se permite el uso de ninguna red de datos ni WIFI ni por cable. Los únicos recursos disponibles son los pdf del tema y la documentación de las clases. Si se incumple esta norma es motivo de invalidación del examen.
- **Entrega:** Crea un proyecto que se llamen ExamenT2_INI y dentro de este crea un fichero java por cada ejercicio que se llame EjN_INI.java donde N es el número del ejercicio e INI son tus iniciales. Si esta estructura no es correcta no se corregirá el examen. Finalmente para entregar el examen comprime la carpeta del proyecto en un fichero zip y entregalo en la plataforma moodle en la tarea habilitada para ello.

1. **(3.25 puntos) Haz una aplicación que genere un informe sobre un directorio pasado como parámetro:**

- En primer lugar el programa pedirá una ruta al usuario para que la introduzca por teclado. A continuación debe comprobar que esa ruta exista y que sea un directorio. Si no existe o no es un directorio informará del error y el programa acabará.
- Si la ruta es correcta el programa obtendrá la lista de archivos de ese directorio y generará un resumen con los 3 primeros archivos del directorio. Si hay menos de 3 solo escribirá la información de esos sin dar error. Este resumen se guardará en el archivo de texto **“resumen_INI.txt”** y tendrá el siguiente formato:

Resumen de archivos de:
ruta introducida

nombre	tipo	tamaño
nombre1	tipo	tamaño
nombre1	tipo	tamaño

Ejemplo:

Resumen de archivos de:
C:/datos/

juegos	carpeta	123546
datos.txt	archivo	468
foto.jpg	archivo	758

2. **(3.5 puntos) Usando la clase LocalDate vamos a simular un programa de reservas de billetes de tren:**

- Realiza un programa que pregunte por teclado la fecha de salida (día, mes y año) y la fecha de vuelta (día, mes y año), También preguntará el numero de plazas a reservar. El programa deberá comprobar que los datos leídos sean válidos (1-31,1-12,>=2023) y plazas (1-10). Si alguno de los datos no es válido se informará del error y el programa acabará.
- En segundo lugar creará los objetos LocalDate y se comprobará que la fecha de salida es posterior a hoy y que la fecha de vuelta es posterior a la de ida (usando métodos de la clase). Si no se cumple alguna de estas condiciones el programa informará del error y acabará.
- Si todo es correcto el programa simulará la comprobación de plazas libres del tren generando 2 números aleatorios entre 0 y 50, uno para la ida y otro para la vuelta. Comprobará que el numero de plazas solicitadas es inferior a las disponibles para ambos trayectos. Si en alguno de ellos no hay plazas suficientes informará de ello y el programa finalizará.
- Si hay plazas disponibles generará 1 archivos de texto con el nombre “reserva-fechaIda.txt” donde fecha es la fecha de ida del viaje. Que contenga lo siguiente:

Reserva Tren
TuNombreCompleto

Fecha Ida: “dd-mm-yyyy”
Fecha Vuelta: “dd-mm-yyyy”
Número de plazas: plazas

Ejemplo:

Reserva Tren
Alberto Zagalaz Anula

Fecha Ida: “14-12-2023”
Fecha Vuelta: “06-01-2024”
Número de plazas: 3

3. **(3,25 puntos) Realiza un programa que lea los datos de un usuario de un fichero y compruebe que son válidos:**

- En primer lugar leerá un archivo de texto preguntado por teclado. Si el archivo no existe o no se puede leer informará del error y el programa terminará.
- Se deberá leer el fichero que contendrá el siguiente formato

Nombre: Nombre
Apellidos: Apellidos
Email: mail
Contraseña: Contraseña
Contraseña2: Contraseña

Ejemplo:

Nombre: Alberto
Apellidos: Zagalaz Anula
Email: azagalaz@ieshlanz.es
Contraseña: secreto1235
Contraseña2: secreto1235

- Si se produce algún error durante la lectura se informará del error ocurrido y el programa acabará.
- En caso contrario el programa realizará las siguientes comprobaciones:
 - Nombre y apellidos: Que no estén en blanco
 - Email: Que contenga solamente una @, que haya al menos 5 caracteres delante de esta y al menos 7 detrás de esta. Que exista al menos un punto detrás de la @ y que después de este (si hay más de uno el último) haya al menos 2 caracteres más.
 - Contraseña: Eliminará posibles espacios que haya delante o detrás de esta y comprobará que sean iguales tanto ignorando mayúsculas.
- Si alguna de las comprobaciones falla se informará del error y el programa continuará.
- Si todo es correcto escribirá solamente los datos en un fichero que se llame "BD.txt" preguntando si quiere añadir o sobrescribir dicho fichero de la siguiente forma, para que el usuario se añada al final del fichero o se sobrescriba:

Nombre Completo
Email
Contraseña

Ejemplo:

Alberto Zagalaz Anula
azagalaz@ieshlantz.es
secreto1235
