TAREA PARA PROG08

Detalles de la tarea de esta unidad.

Introducción.

En esta unidad se ha aprendido la utilidad de las **estructuras de almacenamiento**. Esta tarea consta de varias actividades para trabajar de manera independiente estas estructuras y mejorar en razonamiento lógico.

Para ello se proponen **5 ejercicios**. Hay que entregar los 4 primeros (el 5 es opcional y sirve además de para aprender y pasar un buen rato, para compensar posibles fallos en los 4 primeros ejercicios)

Todas las clases tienen que estar **comentadas** con vuestro nombre, así con la **fecha** de creación, siguiendo, preferiblemente estructura de javadoc.

Además, se realizará un **documento** con **algún** pantallazo en el que se vean todos los ejercicios realizados y vuestra **identificación** del aula virtual. (no es necesario ir explicando, sólo demostrar identidad, **una o dos páginas**)

Si se quiere realizar alguna anotación o comentario al profesor, se puede realizar también en ese pantallazo.

Enunciado. (todos los paquetes dentro de un mismo proyecto de Netbeans)

1. **Operaciones con cadenas. (paquete: ejercicio1.maestre)**

   Crea una clase en Java, que se llame **UtilidadesCadenas** y tenga 4 métodos que realicen lo siguiente:

   a) Reciba 2 cadenas y devuelve una cadena resultado de unir las dos anteriores.
   b) Reciba una cadena y una letra, y devuelva cuántas veces está dicha letra en la cadena.
   c) Reciba una cadena y diga si es palíndroma. (si tiene espacios se eliminan dentro del método, antes de decidir si es palíndroma. Para eliminar espacios se utiliza el método que tiene la clase String para ello.)
   d) Reciba una cadena, y la devuelva girada, es decir al revés.

   Desde un Lanzador que contendrá el main, probarás los 4 métodos y mostrarás los resultados en consola.

   Es indiferente si los métodos se hacen estáticos (se recomienda), o no, siempre que luego se utilice adecuadamente desde el Lanzador.

2. **Operaciones con arrays unidimensionales. (paquete: ejercicio2.maestre)**

   Juego: Donde está la mosca.

   (piensa las estructuras que necesitas, sabiendo que se hace con arrays de una dimensión)

   Vamos a intentar cazar una mosca. La mosca será un valor que se introduce en una posición de un array de 8 posiciones (el jugador no sabe dónde está, pero el programa tendrá un panel oculto donde la tiene posicionada aleatoriamente al empezar el juego)

0	0	0	1	0	0	0	0

   Puedes usar un array de boolean, char o lo que prefieras, esto es sólo un ejemplo.

En cada ronda, el jugador dirá una posición.

Si la mosca está en esa posición se acaba el juego, mosca cazada y se da un mensaje de enhorabuena.

Si la mosca no está en esa posición pueden ocurrir dos cosas:

- que la mosca esté en casillas adyacentes a la casilla indicada por el jugador, en cuyo caso la mosca revolotea y se sitúa en otra casilla al azar (se le dice al jugador que la mosca ha revoloteado)
- que la mosca no esté en casillas adyacentes, en este caso la mosca permanece donde está. (se le dice al jugador que la mosca sigue donde estaba y que elija otra posición)
- Si el jugador caza la mosca, se le mostrará un mensaje dando la enhorabuena y diciendo el número de rondas utilizadas.
- Si en 5 rondas no la ha cazado, se termina la partida indicando al jugador que ha perdido.
    - Tanto el tamaño del vector, como las rondas para terminar el juego, las puedes variar para que te resulte más o menos fácil.

3. Operaciones con arrays multidimensionales. (paquete: ejercicio3.maestre)

Escribir un programa que lea una matriz de enteros de dimensión NxN. Posteriormente, calcular el valor máximo de cada fila y la media de los máximos. Por ejemplo:

			Máximo	Media
3	5	1	5	
6	7	2	7	7
8	1	9	9	

Lo primero que tiene que hacer el programa es rellenar automáticamente la matriz con valores aleatorios entre 1 y 9.

En un ejemplo, para una matriz 3x3, y una vez rellena, se irá recorriendo la matriz buscando el máximo de cada fila, al encontrarlo, los máximos se guardarán en un array unidimensional de 3 posiciones y posteriormente se mostrará y calculará su media, también mostrándola.

Realiza el programa para un tamaño genérico (N se pide al inicio)

4. Operaciones con ArrayList. (paquete: ejercicio4.maestre)

Sobre la clase **CuerpoCeleste**.java que creaste en la Unidad 6, crea un Lanzador, que contenga un main, que pida al usuario cuerpos celestes, y que tendrán que irse almacenando en un ArrayList de CuerposCeleste.

No se sabe cuántos cuerpos quiere insertar el usuario, se irán guardando en el ArrayList, hasta que el usuario decida dejar de insertarlos. (la finalización o no, se le pregunta al usuario cada vez que inserta un Cuerpo)

Una vez termine de insertar, se mostrarán todos los Cuerpos que se insertaron (recorriendo el ArrayList)

5. VOLUNTARIO: Juego del MasterMind (paquete: ejercicio5.maestre).

- Aquí puedes utilizar las estructuras que estimes oportuno.
- Puedes realizarlo en entorno gráfico o bien por consola.

Escribe un programa que juegue al Rojo – Amarillo – Verde.

El programa genera tres dígitos aleatorios distintos entre 0 y 9.

A estos dígitos se les asignan las posiciones 1, 2 y 3.

El objetivo del juego es adivinar los números y sus posiciones correctas en el menor número de intentos posibles.

Para cada intento el jugador proporciona tres números para las posiciones 1, 2 y 3.

El programa responde con una pista que consta de rojo, amarillo y verde:

- Si un dígito está en la posición correcta, la respuesta es verde.
- Si el dígito adivinado está en una posición incorrecta, la respuesta es amarillo.
- Si el dígito no coincide con ninguno de los tres dígitos la respuesta es rojo.

Se darán 7 oportunidades para adivinarlo, si dentro de esas opciones, el jugador lo adivina, el programa dará la enhorabuena e indicará las rondas que ha tardado. Si no, el programa dirá que el jugador ha perdido y mostrará un mensaje de despedida.

Ejemplo: 6 5 8 es la combinación a averiguar

Intento Pista

1 2 5 r r a

8 5 3 a v r

8 5 6 a v a

6 5 8 v v v

Entrega

Se deberá entregar el proyecto con el **código fuente** sin errores para que cada ejercicio pueda ser corregido.

Todos los ejercicios irán en un **único proyecto de NetBeans**, en el que habrá, diferentes **paquetes**, uno con cada ejercicio. El proyecto se llamará **Tarea8NombreApellidos**. Si no se realiza así la entrega, y se generan proyectos diferentes, no se cumple el formato de entrega y se dará por erróneo.

Dentro de cada paquete, cada ejercicio dispondrá de su main, en una clase Lanzadora y de alguna otra clase si se necesitase. Además, se valorará una correcta modularización y control de excepciones genéricas, o entrada de datos correctos, etc. como se ha ido realizando en ejercicios anteriores.

Antes de cada método o clase, se debe indicar entre comentarios el **autor del código y la fecha de realización del mismo**. (preferiblemente siguiendo la nomenclatura de javadoc)

El código fuente debe de estar bien **indentado** y la nomenclatura de clases y variables ser correcto.

Además, se deberá realizar **un documento en PDF demostrando la identidad,** con algún pantallazo, como se ha realizado en las prácticas anteriores, es decir, mostrando identificación

del alumno en el aula virtual. (un documento breve, sin demasiadas explicaciones salvo dónde lo veas oportuno)

Además, en ese documento se puede realizar algún comentario de algún ejercicio, si se necesita.

Criterios de puntuación. Total 10 puntos.

Requisito para evaluar, que el documento demuestre la identificación del alumno, mostrando de fondo en los pantallazos, el aula virtual del curso junto con la identidad.

- Cada ejercicio del uno al cuatro vale **2.5 puntos**, restando 0.5 puntos por cada error importante o ausencia de algún requisito pedido.
- El **ejercicio 5 es voluntario**, sólo servirá para subir nota, nunca pudiendo exceder la nota máxima de 10. El ejercicio 5, **puede subir hasta 2** puntos, si está perfecto.
- Si un ejercicio contiene **errores de compilación**, está vacío o no hace lo que se pide, se calificara con 0 puntos.

Recursos necesarios para realizar la Tarea.

- Ordenador personal.
- Sistema operativo Windows o Linux.
- JDK y JRE de Java (preferiblemente últimas versiones)
- NetBeans IDE 6.9.1 o superior. (preferiblemente Apache NetBeans 12.5)

Consejos y recomendaciones.

Dentro de la carpeta de NetBeans, se habrá creado automáticamente la carpeta del proyecto. Comprueba que efectivamente está la carpeta y dentro todo el contenido que has creado.

Para entregar, **comprime en un único** archivo la **carpeta del proyecto** con todo su contenido, junto con el **documento PDF** y **pantallazos con las explicaciones que demuestran la identidad**.

Indicaciones de entrega.

Se enviará un **único archivo comprimido que contendrá el proyecto, y un documento PDF**, con pantallazos que muestren identidad y explicaciones necesarias indicadas más arriba, sobre el programa.

En ese documento tendrá algún pantallazo demostrando identidad y si se necesita se podrá hacer alguna explicación o comentario.

El envío se realizará a través de la plataforma y el archivo se nombrará siguiendo las siguientes pautas:

apellido1_apellido2_nombre_SIGxx_Tarea

Asegúrate que el nombre no contenga la letra ñ, tildes ni caracteres especiales extraños. Así por ejemplo la alumna Begoña Sánchez Mañas para la quinta unidad del MP de PROG, debería nombrar esta tarea como:

sanchez_manas_begona_PROG08_Tarea.