6.F. Actividades propuestas.

Sitio: Formación Profesional a Distancia
Curso: Programación
Libro: 6.F. Actividades propuestas.
Imprimido por: Iván Jiménez Utiel
Día: martes, 7 de enero de 2020, 22:37

Tabla de contenidos

1. ActividadUT06-1: Clase ContarVocales.

Crear un clase que tenga un método principal que incorpore el conteo del número de vocales de una cadena de texto introducida por teclado. Tener en cuenta que debe considerar tanto mayúsculas como minúsculas, así como sus variantes con tilde.

2. ActividadUT06-2: Clase ConteoAbecedario.

Crear una clase que permita Introducir una cadena por teclado y que imprima en pantalla el número de veces que aparece cada letra del abecedario en el texto. Pista: emplea un array de contadores donde se asocie la posición con el contador correspondiente. (‘a’ – ‘a’ = 0, ‘b’ – ‘a’ = 1 y así sucesivamente). En una línea aparecerán las letras del abecedario y en la siguiente el número de veces que aparece cada letra.

Nota: Haremos el ejercicio de forma que no distinga letras minúsculas de mayúsculas (no case sensitive).

3. ActividadUT06-3: Clase ArraysCadenas.

Crear una clase llamada **ArrayCadenas** que disponga de un método principal que Inicialice dos arrays de cadenas. Dispondrá de métodos para visualizar el contenido de cada array, para ordenarlos alfabéticamente por el método de la burbuja y para fusionarlos o mezclar los dos arrays y visualízar el array resultante de la mezcla.

Nota: en este ejercicio usaremos métodos estáticos en todos los casos.

Tendremos una salida similar a la siguiente:

---ARRAYS SIN ORDENAR---

Array de Cadenas 1

Array[0] = Pepe

Array[1] = Ana

Array[2] = Carlos

Array de Cadenas 2

Array[0] = Cristina

Array[1] = Monica

Array[2] = Jose

---ARRAYS ORDENADOS---

Array de Cadenas 1

Array[0] = Ana

Array[1] = Carlos

Array[2] = Pepe

Array de Cadenas 2

Array[0] = Cristina

Array[1] = Jose

Array[2] = Monica

---ARRAYS FUSIONADOS---

Array de Cadenas 3

Array[0] = Ana

Array[1] = Carlos

Array[2] = Cristina

Array[3] = Jose

Array[4] = Monica

Array[5] = Pepe

4. ActividadUT06-4: Clase ManipularCadenas.

Diseña una clase denominada **Cadena** para manipular cadenas. Los caracteres se almacenarán en un array de caracteres. La clase dispondrá de los siguientes métodos:

- Un constructor que reciba el tamaño máximo de la cadena y pida memoria para el tamaño máximo +1, puesto que después del último carácter almacenado en el array guardaremos un '\0' que al igual que en lenguaje C indique fin de cadena.
- Un constructor de copia.
- **leerCadena()**. Lee una cadena por teclado almacenando un '\0' después del último carácter. Dejamos de introducir caracteres al pulsar un salto de línea.
- **escribirCadena()**. Envía el contenido de la cadena al monitor.
- **longitud()**. Retorna el número de caracteres que contiene la cadena.
- **copiar()**. Copia el contenido de una cadena en otra.
- **esVocal()**. Retorna un valor que indique si una letra es vocal.
- **eliminarVocales()**. Elimina las letras vocales de la cadena moviendo los caracteres siguientes una posición atrás.
- **contieneCadena()**. Retorna un valor que indique si cadena creada contiene la cadena recibida.
- **convertirAmayusculas()**. Convierte las letras minúsculas a mayúsculas.
- **esPalindromo()**. Un palíndromo es una palabra que se lee igual de izquierda a derecha que de derecha a izquierda

5. ActividadUT06-5: Clase Matriz.

Diseña una clase denominada **Matriz** para manejar un array bidimensional de elementos de tipo double.

Además de un constructor, la clase debe contar con los siguientes métodos:

- **leerMatriz()**. Almacena en una matriz los datos introducidos por teclado.
- **escribirMatriz()**. Visualiza en pantalla el contenido de la matriz.
- **sumaDiagonal()**. Visualiza en pantalla la suma de los elementos que están situados por en la diagonal principal de la matriz.
- **sumaEncimaDiagonal()**. Visualiza en pantalla la suma de los elementos que están situados por encima de la diagonal principal. Implementar dos versiones.
- **sumaDebajoDiagonal()**. Visualiza en pantalla la suma de los elementos que están situados por debajo de la diagonal principal. Implementar dos versiones.
- **traspuesta()**. Construye la traspuesta de una matriz.
- **maximo()**. Visualiza el máximo de la matriz.
- **totales()**. Calcula los totales de cada fila y columna de la matriz almacenándoles en arrays unidimensionales. Visualiza las matriz y los totales para comprobar que se han calculado correctamente

6. ActividadUT06-6: Comprobación de formato de fecha con patrones.

Crear una clase llamada **ComprobacionFormatoFecha** que, haciendo uso de las expresiones regulares, lea una fecha, y no la dé como buena hasta que no tenga un formato del tipo "1/ene/2019". Por ejemplo "1/1/2019" no será validada. El día podrá expresarse con uno o dos dígitos, dando por bueno "05/mar/2019". El año será necesariamente de cuatro dígitos.

Los meses serán: "ene", "feb", "mar", "abr", "may", "jun", "jul", "ago", "sep", "oct", "nov" y" dic". Además permitiremos que no sea "case sensitive" en dicha especificación de mes, pudiendo aprobar: "ENE", "Ene", o incluso "eNe" o "enE".

El programa pedirá indefinidamente una fecha mientras esta no sea correcta.