

PRÁCTICA 1 DE ARRAYS

1) Diseña una clase denominada ArrayEnteros con las siguientes métodos:

- Constructor sin parámetros. Solicitamos por teclado el tamaño del array, creamos espacio para almacenar la información e inicializamos el número de elementos.
- Constructor que reciba el tamaño del array.
- Constructor copia.
- leerArray. Introduce por teclado el contenido del array hasta que el usuario teclee 999 o no quepan más datos.
- escribirArray. Visualiza en pantalla el contenido del array.
- insertar. Solicita la posición donde se desea insertar un dato, siendo ésta una de las que están ocupadas. Si la posición es válida lleva a cabo la inserción actualizando el número de elementos.
- Eliminar. Solicita la posición donde se desea eliminar un dato, siendo ésta una de las que están ocupadas. Si la posición es válida lleva a cabo la eliminación actualizando el número de elementos.
- busquedaSecuencial. Realiza un búsqueda secuencial. Devuelve la posición donde se encuentra el elemento, -1 si no se encuentra. Implementa tres versiones distintas.
- busquedaBinaria. Realiza una búsqueda binaria o dicotómica.
- ordenaBurbuja. Ordena de menor a mayor por el método de la burbuja.
- fusion. Fusiona o mezcla dos arrays previamente ordenados retornando la referencia de la lista obtenida como resultado de la fusión.
- ParesImpares. Visualiza en pantalla el número de pares e impares que hay en el array.
- Media. Visualiza en pantalla la media aritmética de los elementos del array.

PRÁCTICA 2 DE ARRAYS

Codifica los ejercicios siguientes:

- 1) Introduce por teclado un texto (pulse enter y a continuación ctrl.^d para terminar). Imprime 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.
- 2) Inicializa dos arrays de cadenas con nombre de personas.
Ordénalos por el método de la burbuja .
Fusiona o mezcla los dos arrays y visualízalos junto con el array mezclado.

PRÁCTICA 3 DE ARRAYS. ARRAYS DE CARACTERES

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 copia.
-
- leerCadena. Lee una cadena por teclado almacenando un '\0' después del último carácter. Dejamos de introducir caracteres a pulsar enter y control^D
-
- 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.
-
- esPalindromo. Un palíndromo es una palabra que se lee igual de izquierda a derecha que de derecha a izquierda

PRÁCTICA 4 DE ARRAYS. MATRICES

Diseña una clase denominada Matriz para manejar un array bidimensional de doubles. 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 en 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.
-
- Totales. Calcula los totales de cada fila y columna de la matriz almacenándoles en arrays unidimensionales. Visualiza la matriz y los totales para comprobar que se han calculado correctamente.

PRÁCTICA 5. ARRAYS DINÁMICOS

A continuación vamos a diseñar una aplicación para construir una agenda de contactos utilizando un array dinámico. La aplicación constará de tres clases:

- Contacto. Almacena el nombre, teléfono de una persona e email. Emplea un constructor con parámetros y los métodos set y get correspondientes. Sobrecarga los métodos toString y compareTo.
- Agenda. Sus miembros dato serán un array dinámico de referencias de tipo Contacto y una tamaño numElem que varía a lo largo de la ejecución del programa. Los métodos son:
 - Un constructor sin argumentos.
 -
 - añadir. Recibe una referencia de DatosPersona. Invoca a un método que solicita memoria para un más, y a otro que inserta la referencia en el espacio añadido.
 -
 - insertarEn. Recibe la referencia y la posición donde insertarla, si es válida.
 -
 - getContacto. Recibe la posición donde se encuentra la referencia que retorna, siempre que sea válida.
 -
 - unElementoMas. Recibe la referencia del array que asigna a un array auxiliar. Pide memoria para un elementos más, copia el contenido del array anterior y actualiza el tamaño.
 - buscar. Recibe una cadena y realiza una búsqueda secuencial por nombre . Retorna la posición, -1 si no está.
 - unElementoMenos. Utilizamos un array auxiliar que apuntea nuestro array. Pedimos memoria para una elemento menos. Copiamos en el array actual todas aquellas referencias no marcadas y actualizamos el tamaño.
 - agendaVacía. Retorna una boolean indicando si la agenda está vacía.
 - iniciarPrograma. Arranca la aplicación. Invoca a verMenu, altas, bajas, consultas o modificaciones.
 - verMenu. Visualiza un menú y retorna la opción validada.

Menú de operaciones:

1. Añadir contacto
 2. Editar contacto
 3. Eliminar contacto
 4. Consultar contacto
 5. Ver toda la agenda
 6. Abandonar la aplicación
- Elija la opción:

Tendremos que hacer un método para cada una de las 5 primeras operaciones. La agenda tiene que listarse ordenada alfabéticamente por el nombre del contacto

La clase Test Agenda estará constituida por los siguientes métodos:

- TestAgenda. Instancia un objeto agenda y arranca la aplicación.