PRÁCTICA 5. Algoritmos “divide y vencerás”. Problema de la ordenación de un vector de N elementos.

La ordenación de un vector de N elementos

Dado un vector A[1..n] de n elementos. Ordenar estos elementos por orden ascendente.

Los algoritmos de ordenación toman decisiones para lograr la ordenación utilizando sólo comparaciones: Algunos de estos algoritmos son:

1. La ordenación por inserción es cuadrática, en el caso peor y en el promedio. Es más rápida si los datos ya están ordenados.

2. La ordenación por fusión (Mergesort), utiliza divide y vencerás para obtener un tiempo de ejecución O(NlogN).

3. La ordenación rápida (Quicksort) es un algoritmo recursivo divide y vencerás, rápido cuando se implementa adecuadamente. En la práctica es el algoritmo más rápido de ordenación basado en comparaciones.

Los dos últimos algoritmos utilizan la técnica recursiva “divide y vencerás”. Donde la recursión es la parte divide y la combinación de las soluciones es la parte vencerás.

Nota:

A.Junto con esta práctica van tres archivos:

1. La clase abstracta EsquemaDyV y las interfaces Problema y Solución. Esta clase representa la solución de los problemas recursivos resueltos mediante la técnica divide y vencerás.

2. Los distintos juegos de prueba de los ejercicios.

B.Consulte mediante la documentación de la API la clase Arrays que está en el paquete java.util.

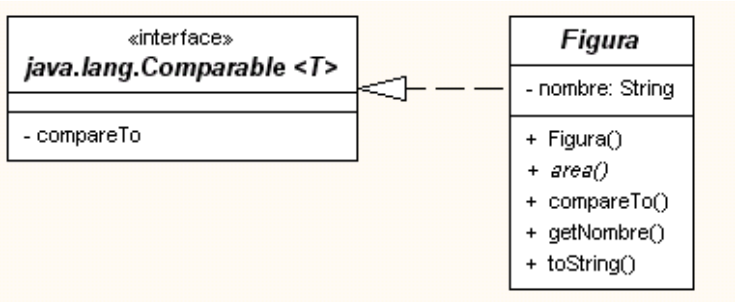
Encontrará métodos para manejar operaciones con arrays. Entre ellos métodos de ordenación, búsqueda y rellenado de arrays.

Ejercicios a realizar:

1) A partir del ejercicio 1 de la practica 2 (El ejercicio de las Figuras). Haga una copia de las clases creadas y documentadas a un nuevo paquete com.mp.practica5.ejercicio1.

   a) Modifique la clase Figura para que implemente la interface java.lang.Comparable: El método menorQue debe ser sustituido por CompareTo. (vea la documentación de la API).

   b) Modifique el método de ordenación por inserción para que utilice el método CompareTo.



Nota: Incluir en el paquete: **com.mp.practica5.ejercicio1**

Nota: Las pruebas de unidad se situaran en el paquete : **com.mp.practica5.ejercicio1.test**

2) Implemente los algoritmos de *inserción*, *mergesort* y *quicksort*, que resuelven la ordenación por comparación de un vector de N elementos, para obtener los datos empíricos de la siguiente tabla (tiempos de ejecución observados en milisegundos). El alumno deberá implementarlos. Utilice para las comparaciones objetos del tipo Entero. Además, para el cuarto algoritmo (Sort), deberá consultar el documento con la especificación de la API para la Java 2 Platform, Standard Edition, versión 1.3.1, y utilizar alternativamente el método *sort* de la clase *Arrays* del paquete *java.util*. Pruebe los algoritmos con un programa principal obteniendo los tiempos de ejecución observados de los diferentes algoritmos, construyendo la siguiente tabla:

N	Inserción	Mergesort	Quicksort	Sort
100				
200				
400				
....				

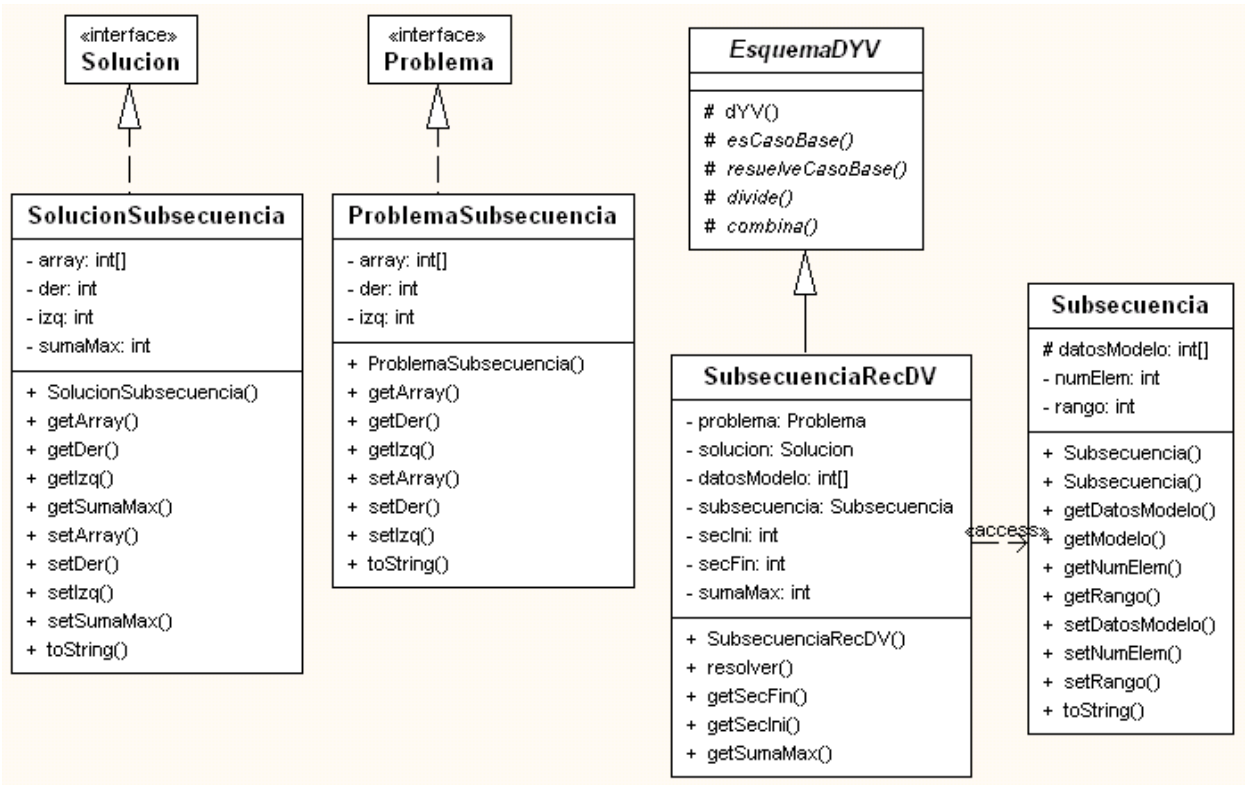
Nota: Incluir en el paquete: **com.mp.practica5.ejercicio2**

Nota: Las pruebas de unidad se situaran en el paquete : **com.mp.practica5.ejercicio2.test**

De los dos ejercicios siguientes el alumno deberá realizar uno de ellos a su elección.

3) Resolver el problema de la subsecuencia, ya resuelto en la práctica anterior, utilizando la técnica recursiva divide-vencerás.

   Para ello, adapte la solución ya implementada en la práctica anterior, a una solución que utilice la clase abstracta EsquemaDyV heredando e implementando sus métodos abstractos.

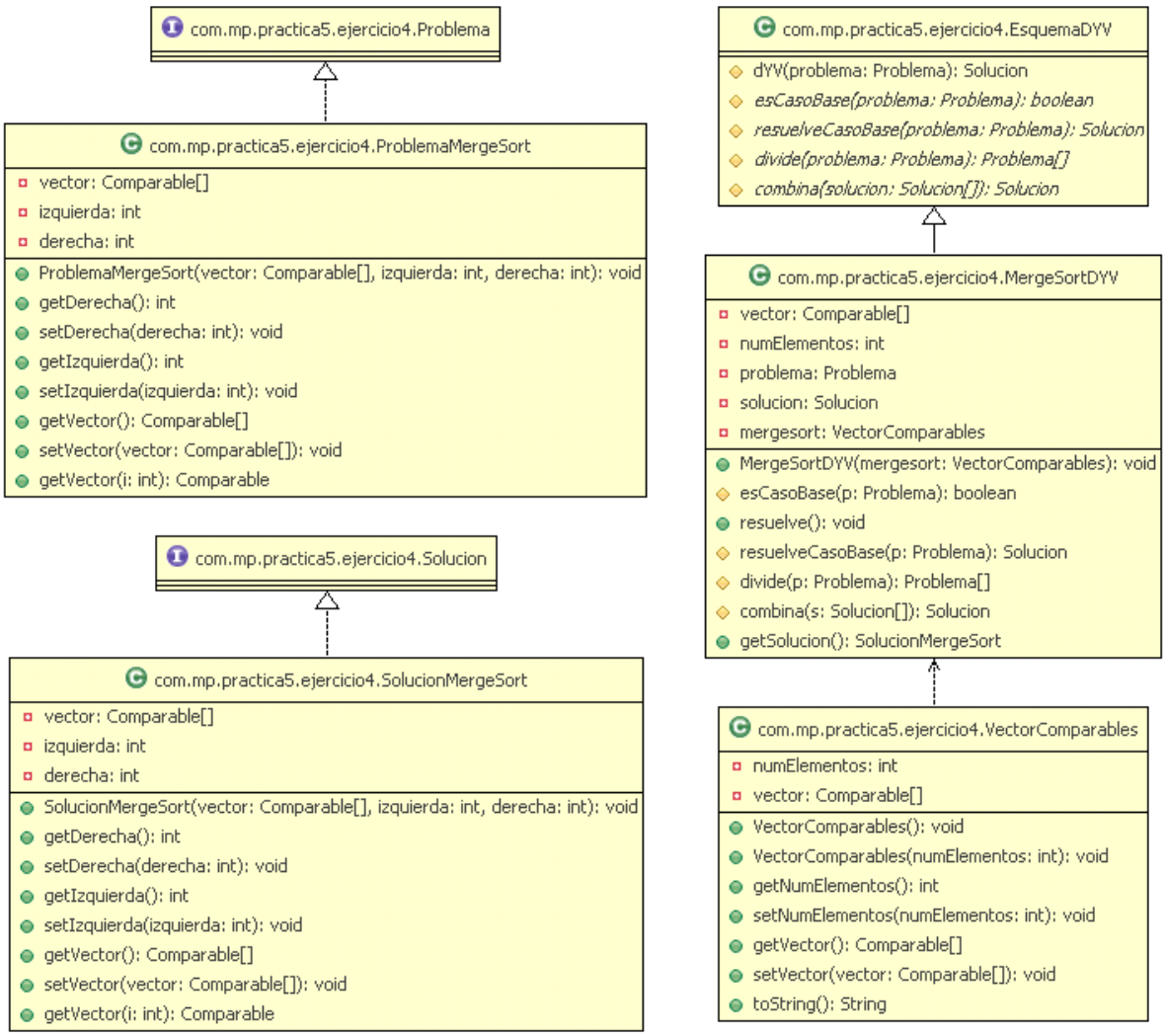


Nota: Incluir en el paquete: **com.mp.practica5.ejercicio3**

Nota: Las pruebas de unidad se situaran en el paquete : **com.mp.practica5.ejercicio3.test**

4) Resolver el problema de ordenación utilizando el algoritmo mergesort, utilizando la técnica recursiva divide-vencerás.

   Para ello, adapte la solución ya implementada en la práctica anterior, a una solución que utilice la clase abstracta EsquemaDyV heredando e implementando sus métodos abstractos.



Nota: Incluir en el paquete**: com.mp.practica5.ejercicio4**

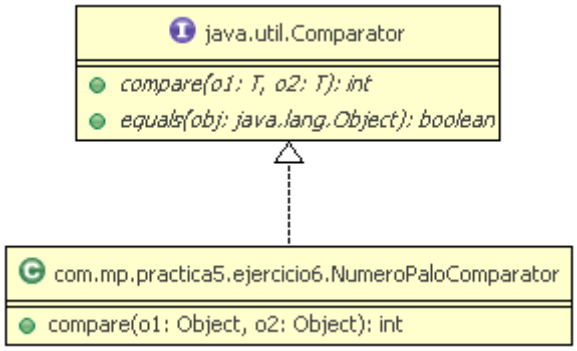
Nota: Las pruebas de unidad se situaran en el paquete : **com.mp.practica5.ejercicio4.test**

5) Queremos ordenar una baraja española de 40 cartas según varios criterios. Termine de implementar las clases Carta y Baraja.

   Ordenar la baraja en base al orden natural: primero palo y dentro del mismo palo los números, es decir; O 1, O2, O3 ... O12 , C 1, C2, C3 ... C12 , E 1, E2, OE ... E12 , B1, B2, B3 ... B12. Carta deberá implementar la interface java.lang.Comparable.

   Ordenar la baraja en base a nuevo orden: primero número y dentro del mismo número los palos {O,C,E,B}, es decir, O 1, C 1, E 1, B 1, O 2, C 2, E 2, B 2 ……. O 12,C 12,E 12,B 12. Para ello, deberá implementar la clase NumeroPaloComparator que implementa la interface java.util.Comparator.

   En el paquete de tests para este ejercicio deberá situar el juego de pruebas BarajaTest.



Nota: Incluir en el paquete: **com.mp.practica5.ejercicio5**

Nota: Las pruebas de unidad se situaran en el paquete : **com.mp.practica5.ejercicio5.test**