Ejercicios

Dentro de la clase Empresa, expresada en UML, Figura 1, se agregaran tres métodos (copiEmps, ordEmpsNom e impEmpsNom), con el fin de imprimir la lista de empleados ordenada por nombres. El actual atributo de la lista de empleados empDir[ ] se encuentra ordenada por clave de empleado.

Figura 1

|  |
| --- |
| Empresa |
| i- nombre : String  i- direccion : String  i- dueño: String  i- MAX: final int = 10  i- empsDir: Empleadp [ ]  i- cantEmpss: int = 0 |
| c+ Empresa(String, String, String)  i+ altaEmp(Empleadp) : boolean  i+ reporteEmps(Empleado) : String  i+ buscaEmpPos(int) : int  i+ aumentoEmp(int, int) : Boolean  i- copiEmps( ) : Empleado[ ]  i- ordEmpsNom(Empleado[ ]) : void  i+ impEmpsNom( ) : void |

El método privado copiEmps( ) se encargará de hacer una copia del empsDir[ ] en un nuevo arreglo que devolverá (return) como resultado.

Figura 2

|  |
| --- |
| // Copia la lista de empleados, previamente ordenada por clave  private Empleado [ ] copiEmps() {  Empleado [ ] arrEmps = new Empleado[MAX];  // Copia empsDir[ ] en arrEmps[ ]  // - - -    return arrEmps;  } |

El método privado ordEmpsNom( ), Figura 2, recibirá como parámetro el nuevo arreglo para ordenarlo por el nombre del empleado usando el algoritmo de ordenamiento de Selección Directa cuyo diagrama de flujo se muestra en el archivo “AlgoritmosDeOrdenamientoYBusqueda.pdf”. Una explicación sobre este algoritmo viene al final de este documento, como Apéndice A.

Figura 3

|  |
| --- |
| // Ordena el nuevo arreglo por nombre del empleado.  private void ordEmpsNom(Empleado [] arrEmpsC) {  // Ordena arrEmpsC[] por Nombre, con el  // algoritmo de Seleccion Directa  // - - -  } |

Por último, el método público impEmpsNom( ) invocará (llamará) a los dos métodos anteriores para después imprimir la lista de empleados ordenada por nombre.

Figura 4

|  |
| --- |
| // Imprime el arreglo de Empleados ordenado por Nombre  public void impEmpsNom() {  Empleado [] arrEmpsNom;  arrEmpsNom = copiEmps();  ordEmpsNom(arrEmpsNom);  // Imprimir la lista de Empleados por Nombre  // - - -  } |

Por supuesto hay que probar desde main() el uso del método impEmpsNom( ).

Continúa en la siguiente página.

Algoritmo de ordenamiento de Selección Directa

El diagrama de este algoritmo es el primero que aparece en el archivo “AlgoritmosDeOrdenamientoYBusqueda.pdf”. Se trata de un arreglo A[ ] de N elementos entero (int) en cualquier orden.

El diagrama consta de dos ciclos, uno, el externo, controlado por el rombo I<N-1, y el otro, el interno, controlado por el rombo J<N. Las variables I y J actúan como índices de A[ ].

El índice I cambia de valores entre 0 y N-2, mientras va ciclando, cada vez que I toma un valor la J es iterada desde I-1 hasta N-1.

Suponga A[ ] = {40, 30, -28, 70, 60}.

Esto hace que cuando I=0, J itera desde 1 hasta N-1, permitiendo comparar A[I] contra los A[J], dicho de otra manera va comparando A[0] contra los A[1], A[2] hasta A[N-1], dejando al final del primer ciclo de I=0, el menor de los elementos comparados en A[0].

Quedaría A[ ] = {-28, 40, 30, 70, 60}.

Luego, cuando I=1, J itera desde 2 hasta N-1, permitiendo comparar A[I] contra los A[J], dicho de otra manera va comparando A[1] contra los A[2], A[3] hasta A[N-1], dejando al final del segundo ciclo de I=1, el segundo menor de los elementos comparados en A[1].

Quedaría A[ ] = {-28, 30, 40, 70, 60}.

Después, cuando I=2, J itera desde 3 hasta N-1, permitiendo comparar A[I] contra los A[J], dicho de otra manera va comparando A[2] contra los A[3], A[4] hasta A[N-1], dejando al final del tercer ciclo de I=2, el tercer menor de los elementos comparados en A[2].

Quedaría A[ ] = {-28, 30, 40, 70, 60}.

Así hasta que I=N-2, J itera sólo como N-1, permitiendo comparar A[I] contra los A[J], dicho de otra manera va comparando A[N-2] contra A[N-1], dejando al final de este último ciclo de I=N-2, el penúltimo menor en A[N-2].

Quedaría A[ ] = {-28, 30, 40, 60, 70}.

La variable MIN siempre lleva el índice donde se encuentra el menor de los elementos, que es A[MIN].

Volviendo a los ejercicios.

Usted tendrá que hacer las adaptaciones correspondientes ya que el arreglo que usted ordenará es un arreglo de objetos Empleado.