

Ordenar un ArrayList en Java

El proyecto de este post lo puedes descargar pulsando AQUI (<https://github.com/jarroba/SortArrayList>).

Una de las funcionalidades más interesantes a la hora de trabajar con estructuras de datos tipo "pila", "cola", "lista" etc. es su ordenación. En el post de "Ordenar Arrays en Java (Con sus índices)" (<https://jarroba.com/ordenar-arrays-en-java-con-sus-indices/>) implementamos un API que nos permitía ordenar Arrays y sus índices, que es una funcionalidad muy importante que no ha sido implementada por ningún API conocida y de forma sencilla.

En este post vamos a mostrar como ordenar ArrayList, bien sea con un ArrayList de datos atómicos o con un ArrayList de objetos.

Para realizar estas operaciones necesitaremos trabajar con la Clase "Collections" de Java. No vamos a explicar las características de esta clase, sino que simplemente vamos a utilizarla para ordenar ArrayList's. Como suele ser habitual en esta web, pasamos a mostrar esta funcionalidad con un ejemplo:

Vamos a tener un ArrayList llamado "arrayListInt" que es un array de enteros y que le pondremos los siguientes valores (sino sabéis el método add, mirar el post "ArrayList en Java, con ejemplos" (<https://jarroba.com/arraylist-en-java-ejemplos/>)):

```
private static ArrayList arrayListInt = new ArrayList<>();

// Guardo datos en el ArrayList
arrayListInt.add(3);    arrayListInt.add(4);
arrayListInt.add(2);    arrayListInt.add(6);
arrayListInt.add(5);    arrayListInt.add(1);
arrayListInt.add(7);
```

Si imprimimos este ArrayList en el orden en el que hemos guardado los elementos, tendremos que el primer elemento sera el número '3', el segundo el número '6' y si seguimos el orden, el último será el número '7'.

Este sitio web utiliza cookies para que usted tenga la mejor experiencia de usuario. Si continúa navegando está dando su consentimiento para la aceptación de las mencionadas cookies y la aceptación de nuestra política de cookies (<http://jarroba.com/faq/>)

(<http://jarroba.com/faq/>)

Para ordenar este ArrayList de menor a mayor, vamos a utilizar el método "sort" de la clase

ACEPTAR

Collections:

```
Collections.sort(arrayListInt);
```

Con este método nos devolverá el ArrayList ordenado de menor a mayor, siendo la salida la siguiente (en el proyecto que os podéis descargar, podéis ver los métodos que imprimen por pantalla los ArrayList):

```
Posicion(1) = 1  
Posicion(2) = 2  
Posicion(3) = 3  
Posicion(4) = 4  
Posicion(5) = 5  
Posicion(6) = 6  
Posicion(7) = 7
```

Para ordenarlo de mayor a menor, tenemos que crearnos un objeto de la clase Comparator para que compare los elementos y los ponga en orden inverso. Esto lo hacemos de la siguiente manera:

```
Comparator<Integer> comparador = Collections.reverseOrder();  
Collections.sort(arrayListInt, comparador);
```

Ejecutando esta parte del programa e imprimiendo por pantalla el ArrayList resultante, tenemos el ArrayList ordenado de mayor a menor:

```
Posicion(1) = 7  
Posicion(2) = 6  
Posicion(3) = 5  
Posicion(4) = 4  
Posicion(5) = 3  
Posicion(6) = 2  
Posicion(7) = 1
```

Pero ¿Que pasa si lo que tenemos es un ArrayList de objetos de una determinada clase y lo que queremos ordenar ese ArrayList por el valor de un determinado atributo?. Bueno pues hay una sencilla solución que vamos a mostrar a continuación y que consiste en sobrescribir un método de la clase Comparator.

Supongamos que tenemos una clase "Persona" de la siguiente forma:

Uso de cookies

Este sitio web utiliza cookies para que usted tenga la mejor experiencia de usuario. Si continúa navegando está dando su consentimiento para la aceptación de las mencionadas cookies y la aceptación de nuestra [política de cookies](http://jarroba.com/faq/) (<http://jarroba.com/faq/>)

(<http://jarroba.com/faq/>)

ACEPTAR

```
class Persona {  
  
    private String nombre;  
    private int edad;  
  
    public Persona() {  
    }  
  
    public Persona(String nombre, int edad) {  
        this.nombre = nombre;  
        this.edad = edad;  
    }  
  
    public String getNombre() {  
        return nombre;  
    }  
  
    public void setNombre(String nombre) {  
        this.nombre = nombre;  
    }  
  
    public int getEdad() {  
        return edad;  
    }  
  
    public void setEdad(int edad) {  
        this.edad = edad;  
    }  
  
    @Override  
    public String toString() {  
        return this.getNombre() + " - " + this.getEdad();  
    }  
}
```

Supongamos que tenemos un ArrayList llamado "personas" con los siguientes objetos de la clase persona:

```
private static ArrayList personas = new ArrayList<>();  
.....  
.....  
personas.add(new Persona("Pepe", 28));  
personas.add(new Persona("Juan", 32));  
personas.add(new Persona("Paco", 40));  
personas.add(new Persona("Susi", 24));  
personas.add(new Persona("Lola", 20));  
personas.add(new Persona("Jose", 28));  
personas.add(new Persona("Dani", 24));  
personas.add(new Persona("Sara", 36));
```

Este ArrayList lo queremos ordenar (de menor a mayor edad) en función de la edad de las personas, por tanto tenemos que sobre escribir el método "compare" de la clase Comparator y llamar al método "sort" de la clase collections. Esto lo hacemos de la siguiente forma:

Uso de cookies

Este sitio web utiliza cookies para que usted tenga la mejor experiencia de usuario. Si continúa navegando está dando su consentimiento para la aceptación de las mencionadas cookies y la aceptación de nuestra política de cookies (<http://jarroba.com/faq/>)

(<http://jarroba.com/faq/>)

ACEPTAR

```

Collections.sort(personas, new Comparator() {
    @Override
    public int compare(Persona p1, Persona p2) {
        return new Integer(p1.getEdad()).compareTo(new Integer(p2.getEdad()));
    }
});

```

Como vemos lo que hace el método "compare" es comparar dos edades dadas. En este caso estamos comparando dos atributos de tipo "Integer". Si lo queremos hacer con un String, Double, etc. solo tenemos que cambiar el tipo de dato. Como salida a este código tenemos lo siguiente:

```

Posicion(1) = Lola   - 20
Posicion(2) = Dani   - 24
Posicion(3) = Susi   - 24
Posicion(4) = Pepe   - 28
Posicion(5) = Jose   - 28
Posicion(6) = Juan   - 32
Posicion(7) = Sara   - 36
Posicion(8) = Paco   - 40

```

Si ahora queremos ordenarlo de mayor a menor edad, solo tenemos que cambiar el método "compare" poniendo la comparación al revés:

```

Collections.sort(personas, new Comparator() {
    @Override
    public int compare(Persona p1, Persona p2) {
        // Aquí esta el truco, ahora comparamos p2 con p1 y no al revés como antes
        return new Integer(p2.getEdad()).compareTo(new Integer(p1.getEdad()));
    }
});

```

Como resultado de esta ordenación obtenemos lo siguiente:

```

Posicion(1) = Paco   - 40
Posicion(2) = Sara   - 36
Posicion(3) = Juan   - 32
Posicion(4) = Pepe   - 28
Posicion(5) = Jose   - 28
Posicion(6) = Dani   - 24
Posicion(7) = Susi   - 24
Posicion(8) = Lola   - 20

```

Esto es en resumen todo lo necesario para ordenar ArrayList, bien sean por un atributo de una serie de objetos o con datos atómicos (en realidad son objetos que representan datos atómicos, Integer, Double, etc.). No hemos entrado en detalle de explicar las clases Comparator y Collections ya que en principio la finalidad de este post es la de dar una solución a la ordenación de ArrayList.

Uso de cookies
Comparte esta entrada en:

Este sitio web utiliza cookies para que usted tenga la mejor experiencia de usuario. Si continúa navegando está dando su consentimiento para la aceptación de las mencionadas cookies y la aceptación de nuestra política de cookies. <http://jarroba.com/faq/>

(<http://jarroba.com/faq/>)

ACEPTAR

ps:/ ps:/ ps:/ ps:/



(https://creativecommons.org/licenses/by-nc-sa/3.0/es/)



(https://www.safecreative.org/userfeed/1401310112503)

Ordenar un ArrayList en Java (https://jarroba.com/ordenar-un-arraylist-en-java/) por "www.jarroba.com" esta bajo una licencia Creative Commons

Reconocimiento-NoComercial-CompartirIgual 3.0 Unported License.

Creado a partir de la obra en www.jarroba.com (https://jarroba.com/ordenar-un-arraylist-en-java/)

50 thoughts on "Ordenar un ArrayList en Java"



Edwing Camilo dice:

29/05/2018 a las 15:55 (https://jarroba.com/ordenar-un-arraylist-en-java/#comment-212374)

Buen tutorial pero tengo un inconveniente en el @Override me salen en los parametros la palabra Object y no me deja cambiarlo por el de la clase que se llama Almacenamiento

```
Collections.sort(datos, new Comparator() {
```

```
@Override
```

```
public int compare(Object t, Object t1) {
return new Integer(t.getEdad()).compareTo(new Integer(t1.getEdad()));
}
});
```

Responder



Michael dice:

02/01/2018 a las 03:36 (https://jarroba.com/ordenar-un-arraylist-en-java/#comment-191457)

Muy útil ! muchas gracias !

Responder



marie dice:

29/09/2017 a las 19:36 (https://jarroba.com/ordenar-un-arraylist-en-java/#comment-184959)

excelente... muchas gracias. !!!!

Responder



Yen dice:

29/09/2016 a las 06:47 (https://jarroba.com/ordenar-un-arraylist-en-java/#comment-65225)

(http://jarroba.com/faq/)

Por cierto felicidades muy bien explicado y muy interesante, salu2

Uso de cookies

Este sitio web utiliza cookies para que usted tenga la mejor experiencia de usuario. Si continúa navegando está dando su consentimiento para la aceptación de las mencionadas cookies y la aceptación de nuestra política de cookies. [ACEPTAR](#)

Responder



Yen dice:

29/09/2016 a las 06:45 (<https://jarroba.com/ordenar-un-arraylist-en-java/#comment-65224>)

Hola para ordenar de mayor a menor no hace falta el comparador se puede hacer mas facil

```
//se ordenan
```

```
Collections.sort(numeros);
```

```
// se invierte el orden
```

```
Collections.reverse(numeros);
```

```
//y listo solo imprimimos en pantalla
```

```
system.out.println("Numeros de mayor a menor"+numeros.toString);
```

Responder



Jesús Fonseca (<http://Anonnnymous.net>) dice:

31/01/2017 a las 05:29 (<https://jarroba.com/ordenar-un-arraylist-en-java/#comment-115739>)

Muchas graciass!!!

Responder



Jav dice:

02/06/2016 a las 15:53 (<https://jarroba.com/ordenar-un-arraylist-en-java/#comment-39496>)

Buenas, quizá vengo tarde pero hay un pequeño error en el código donde haces los comparador. Tendría que ser así

```
Collections.sort(personas, new Comparator<Personas>(){
```

```
//El metodo compare
```

Uso de cookies

Este sitio web utiliza cookies para que usted tenga la mejor experiencia de usuario. Si continúa navegando está dando su consentimiento para la aceptación de las mencionadas cookies y la aceptación de nuestra política de cookies (<http://jarroba.com/faq/>)

(<http://jarroba.com/faq/>)

Ya que si el Comparator se pone como está publicado, los parametros del metodo compare

ACEPTAR

seran del tipo Object y no del tipo Persona

Responder



gge dice:

21/08/2016 a las 19:03 (<https://jarroba.com/ordenar-un-arraylist-en-java/#comment-54346>)

gracias bro, no sabia porque me tiraba el error cuando ya habia implementado el metodo "compare"

Responder



diego Babativa dice:

13/07/2017 a las 13:09 (<https://jarroba.com/ordenar-un-arraylist-en-java/#comment-171443>)

Heyyyyy thank you so much bro.

Regards from Colombia.

Responder



Edwing Camilo dice:

29/05/2018 a las 16:22 (<https://jarroba.com/ordenar-un-arraylist-en-java/#comment-212377>)

Gracias!!! acabas de Salvar el semestre de 20 personas T-T

Responder



maria dice:

13/05/2016 a las 19:00 (<https://jarroba.com/ordenar-un-arraylist-en-java/#comment-38086>)

hola soy nuevo en esto de programar me gustaria saber como aria esta pregunta en la codificacion ¿Se requiere que los valores de las edades y los nombres de las personas sean introducidos por el usuario? gracias

Responder



Ramón [Admin Jarroba] dice:

16/05/2016 a las 19:06 (<https://jarroba.com/ordenar-un-arraylist-en-java/#comment-38339>)

Pues depende de lo que se necesite. Puedes optar por consola o por interfaz de usuario, luego para guardar los datos hay un sin fin de maneras de tratarlos: Arrays, Objetos, List, Maps.

Uso de cookies

Este sitio web utiliza cookies para que usted tenga la mejor experiencia de usuario. Si continúa navegando está dando su consentimiento para la aceptación de las mencionadas cookies y la aceptación de nuestra política de cookies (<https://jarroba.com/faq/>)

(<http://jarroba.com/faq/>)

ACEPTAR

Responder



Neptali dice:

05/11/2015 a las 18:49 (<https://jarroba.com/ordenar-un-arraylist-en-java/#comment-26338>)

si la explicacion esta muy bien, pero sin utilizar ArrayList y mucho menos String y lo realizo con objetos, como lo puedo ordenar los datos insertado....??????????????

Responder



Ramón [Admin Jarroba] dice:

05/11/2015 a las 21:11 (<https://jarroba.com/ordenar-un-arraylist-en-java/#comment-26360>)

Para poder ordenar algo tiene que estar en alguna estructura como array o List, sino son objetos sin relación ninguna. Para relacionar objetos del List como indicamos más arriba puedes utilizar compare()

Responder



Rostan dice:

12/08/2015 a las 18:53 (<https://jarroba.com/ordenar-un-arraylist-en-java/#comment-22001>)

Ramón buenos dias,

Tengo una duda respecto a terminos en Java, cuando dices "Sobreescribir el Metodo" te refieres a ir hasta la clase Compare.java y meterle esas lineas de codigo (basado en los objetos que estoy usando)...

De antemano muchas gracias!...

No ha sido la primera vez que me has salvado, pero en esta me he quedado atorado. (:

Responder



Ramón [Admin Jarroba] dice:

12/08/2015 a las 19:05 (<https://jarroba.com/ordenar-un-arraylist-en-java/#comment-22005>)

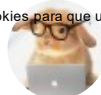
No, en este caso no me refiero a sobrescribir métodos. Me he expresado mal, quería decir sustituir un valor por otro con put: nombreMap.put(K clave, V valor);

Responder

Uso de cookies

Este sitio web utiliza cookies para que usted tenga la mejor experiencia de usuario. Si continúa navegando está dando su consentimiento para la aceptación de las mencionadas cookies y la aceptación de nuestra política de cookies (<http://jarroba.com/faq/>)

(<http://jarroba.com/faq/>)



Rostan dice:

ACEPTAR

12/08/2015 a las 19:39 (<https://jarroba.com/ordenar-un-arraylist-en-java/#comment-22007>)

oh listo! bueno me tocara googlear lo del put: y el map haha (:

Responder



Ramón [Admin Jarroba] dice:

12/08/2015 a las 20:06 (<https://jarroba.com/ordenar-un-arraylist-en-java/#comment-22010>)

Tienes como funciona HashMap en <http://jarroba.com/map-en-java-con-ejemplos/> (<http://jarroba.com/map-en-java-con-ejemplos/>) 🙄

Responder



lola dice:

25/06/2015 a las 01:55 (<https://jarroba.com/ordenar-un-arraylist-en-java/#comment-19494>)

hola tengo un array list cuyo codigo es el siguiente

```
public class New {
public static int frec;

public static void main(String[] args) throws IOException {

ArrayList<Person> peoplelist=new ArrayList<Person>();

javax.swing.JFileChooser j= new javax.swing.JFileChooser();
j.showOpenDialog(j);
File f;

String path= j.getSelectedFile().getAbsolutePath();
String lectura="";
f = new File(path);
FileReader fr = new FileReader(f);
BufferedReader br = new BufferedReader(fr);
String line=null;
```

String aux;

Uso de cookies

Este sitio web utiliza cookies para que usted tenga la mejor experiencia de usuario. Si continúa navegando está dando su consentimiento para la aceptación de las mencionadas cookies y la aceptación de nuestra [política de cookies](http://jarroba.com/faq/) (<http://jarroba.com/faq/>)

(<http://jarroba.com/faq/>)

ACEPTAR

```
while((line=br.readLine())!=null){

    String [] peopleInfo=line.split(",");
    String fecha=peopleInfo[0];
    String serie=peopleInfo[1];
    String lote=peopleInfo[2].trim();
    String frecuencia=peopleInfo[3].trim();
    String diferencia=peopleInfo[4].trim();
    String Q=peopleInfo[5].trim() ;
    String nom_operador=peopleInfo[6];
    String num_operador=peopleInfo[7];

    frec=Integer.parseInt(frecuencia);

    double q=Double.parseDouble(Q);
    peoplelist.add(new
    Person(fecha,serie,lote,frecuencia,diferencia,Q,nom_operador,num_operador));

}

}

}
```

se trata de obtener el numero mayor en el campo frecuencia e imprimirlo como podria hacerlo? de antemano gracias

 Responder



Diego dice:

30/05/2015 a las 18:23 (<https://jarroba.com/ordenar-un-arraylist-en-java/#comment-17540>)

Eres un grande, muy bien explicado tío

Uso de cookies

Este sitio web utiliza cookies para que usted tenga la mejor experiencia de usuario. Si continúa navegando está dando su consentimiento para la aceptación de las mencionadas cookies y la aceptación de nuestra [política de cookies](http://jarroba.com/faq/) (<http://jarroba.com/faq/>)

 Responder

(<http://jarroba.com/faq/>)

ACEPTAR



Mario Obed dice:

26/04/2015 a las 21:45 (<https://jarroba.com/ordenar-un-arraylist-en-java/#comment-15974>)

Que chilera la forma en como lo explican, me ha ayudado a comprender mucho, pero tengo otra duda con esto específicamente:

tengo una clase donde se guardan todas mi ArrayList ejemplo:
`ArrayList<Persona>c_persona=new ArrayList<Persona>();`

y tengo mi clase persona:
 String nombre:
 String Apellido:
 int Edad.

tengo mi JFrame e otra clase distinta donde introduzco los datos que necesito, toda la parte de ingreso de datos ya lo tengo pero necesito hacer un ordenamiento con su atributo por edad y mostrarlo, e leído como lo tienen pero no logro comprender: podrían ayudarme. Gracias desde ya

Responder



Gabriel dice:

02/03/2015 a las 00:14 (<https://jarroba.com/ordenar-un-arraylist-en-java/#comment-11183>)

Gracias por aportar tus conocimientos, saludos..

Responder



telecodesesperado dice:

16/12/2014 a las 10:22 (<https://jarroba.com/ordenar-un-arraylist-en-java/#comment-1646>)

Hola, tengo un ArrayList de características fisiológicas como presión arterial y glucosa en sangre, se supone que cada una de estas mediciones van asociadas a una fecha. Quería saber como devolver una lista en las que estas características estén ordenadas por fecha. Gracias:)

Responder



Ramón [Admin Jarroba] dice:

16/12/2014 a las 22:23 (<https://jarroba.com/ordenar-un-arraylist-en-java/#comment-1698>)
Uso de cookies
 Este sitio web utiliza cookies para que usted tenga la mejor experiencia de usuario. Si continúa navegando está dando su consentimiento para la aceptación de las mencionadas cookies y la aceptación de nuestra política de cookies (<http://jarroba.com/faq/>)

(<http://jarroba.com/faq/>)

Hola,
 Si tienes un ArrayList con objetos con las características fisiológicas, junto otros datos como

fechas, etc. Lo único que tienes que hacer es un sort con un nuevo Comparator() que compare por la variable de fecha.

Responder



hugo dice:

24/11/2014 a las 20:20 (<https://jarroba.com/ordenar-un-arraylist-en-java/#comment-1253>)

Buenas tarde tengo que realizar una venta de un libro y me quede trabado en la impresión de la lista del libro me podrian dar una mano? desde ya muchas gracias!

```
package Main;

import Clientes.Cliente;
import Gestores.GestorVentaDeLibro;
import Comprar.VentaDeLibro;
import Gestores.GestorClienteComun;
import Gestores.GestorClienteVip;
import Gestores.GestorLibrosDigitales;
import Gestores.GestorLibrosImpresos;
import Libros.LibrosDigitales;
import Libros.LibrosImpresos;

import java.util.ArrayList;
import java.util.List;

import javax.swing.text.html.HTMLDocument.Iterator;

import Libros.Libro;

public class VenderLibros {

    public static void main(String[] args) {

        // CREAR UN CLIENTE COMUN
        GestorClienteComun gcc = new GestorClienteComun();
        Cliente clienteComun = gcc.registrarClienteComun(«Jesus», «Davila», «Aconcagua», 239, «CABA»,
        «CABA»,49877458, «DODO», 32230461, «Jebuss», 500);

        //CREAR UN CLIENTE VIP
        GestorClienteVip gcv = new GestorClienteVip();
        Cliente clienteVip = gcv.registrarClienteVip(«Hugo», «Ferrand», «Pringles», 4239, «Lanus», «BSAS»,
        1145254781, «Kratos», 26230461, «ZaZa», 200);

        //CREAR 1 LIBRO DIGITAL
        GestorLibrosDigitales ld = new GestorLibrosDigitales();
```

Este sitio web utiliza cookies para que usted tenga la mejor experiencia de usuario. Si continúa navegando está dando su consentimiento para la aceptación de las mencionadas cookies y la aceptación de nuestra política de cookies (<http://jarroba.com/faq/>)

(<http://jarroba.com/faq/>)

ACEPTAR

```
LibrosDigitales libroDigital = Id.registrarLibroDigital(«Expreso de media noche»,»Billy Hayes»,120.50,
«kjjkgkjg»,»Aconcagua»,2);
```

```
//CREAR UN LIBRO IMPRESO
GestorLibrosImpresos li = new GestorLibrosImpresos();
LibrosImpresos libroImpreso = li.registrarLibroImpreso(«Cianuro espumoso»,»Agatha
Christie»,180.90, «juerr3",»Molino»,165);
```

```
// VEDER UN LIBRO A UN CLIENTE
GestorVentaDeLibro cc = new GestorVentaDeLibro();
List libros = new ArrayList();
libros.add(libroDigital);
libros.add(libroImpreso);
```

```
//Esta es la parte que no entiendo como hacerlo!!!
```

```
VentaDeLibro cld = cc.venderLibro(«15/Nov/2014», 1, libros, clienteComun);
System.out.println(«La fecha es: » );
System.out.println(«Libro» + libros.size()+libroImpreso);
System.out.println(«Libro » + libros.size());
```

```
// VENDER OTRO LIBRO A OTRO CLIENTE
```

```
}
```

```
}
```

Responder



Ramón [Admin Jarroba] (<http://www.jarroba.com>) dice:

26/11/2014 a las 01:14 (<https://jarroba.com/ordenar-un-arraylist-en-java/#comment-1254>)

Hola Hugo,

Por lo que leo lo que haces tan solo es añadir los libros a un ArrayList, por lo que entiendo que actuaría como de «carro de la compra». Si esto es así (es lo que he entendido), el GestorVentaDeLibro será quien se encargue de hacer el resto y le tendrías que pasar los parámetros en el método venderLibro() (esto es un foreach al ArrayList libros), y así por cada cliente. No sé si es lo que necesitas.

Responder



Sebastian dice:

Uso de cookies

Este sitio web utiliza cookies para que usted tenga la mejor experiencia de usuario. Si continúa navegando está dando su consentimiento para la aceptación de las mencionadas cookies y la aceptación de nuestra política de cookies (<http://jarroba.com/faq/>)

(<http://jarroba.com/faq/>) 18/11/2014 a las 22:08 (<https://jarroba.com/ordenar-un-arraylist-en-java/#comment-1251>)

ACEPTAR

una pregunta, tengo un arraylist de planes turisticos, en donde tengo en cada posicion muchos datos como destino, cupos, costo, fecha, lo que necesito es ordenar cada posicion del array solo con costo para que se me organicen los planes turisticos del que tiene un mayor costo a un menor costo para que un cliente pueda verlos, ahi como se realizaria.

Gracias por adelantado 😊

Responder



Ramón [Admin Jarroba] (http://www.jarroba.com) dice:

22/11/2014 a las 13:21 (https://jarroba.com/ordenar-un-arraylist-en-java/#comment-1252)

Hola Sebastián,

lo tienes muy sencillo. Supongo que tendrás una estructura de datos del tipo:

```
List miLista = ArrayList<PlanTuristico>();
```

y en cada objeto de cada PlanTuristico tienes los datos que indicas.

Tan solo tienes que hacer sort() con un nuevo Comparator(), al igual que en este ejemplo, donde compares cada precio. Y lo tienes hecho 😊

Responder



jesus dice:

09/11/2014 a las 22:49 (https://jarroba.com/ordenar-un-arraylist-en-java/#comment-1249)

una pregunta como hago para ordenar una lista de forma alfabética me explico mi lista es de articulo

mi clases es la siguiente

```
class xxx{
String autor;
String titulo;
String fecha;
}
```

y me pide que lo ordene alfabéticamente por el titulo

Responder



Ramón [Admin Jarroba] (http://www.jarroba.com) dice:

10/11/2014 a las 16:55 (https://jarroba.com/ordenar-un-arraylist-en-java/#comment-1250)

Hola Jesús,

desde el sort con el comparador puedes ir preguntando cuál es mayor que el otro. Un ejemplo sería.

Este sitio web utiliza cookies para que usted tenga la mejor experiencia de usuario. Si continúa navegando está dando su consentimiento para la aceptación de las mencionadas cookies y la aceptación de nuestra política de cookies (http://jarroba.com/faq/)

(http://jarroba.com/faq/)

```
Collections.sort(lista, new Comparator() {
@Override
```

ACEPTAR

```
public int compare(final xxx obj1, final xxx obj2) {
return obj1.getTitulo().compareTo(obj2.getTitulo());
}
} );
```

Responder



Vitmar Aliaga dice:

02/02/2015 a las 21:00 (<https://jarroba.com/ordenar-un-arraylist-en-java/#comment-9017>)

es necesario ponerle el «final»,,?

Responder



Vitmar Aliaga dice:

02/02/2015 a las 21:01 (<https://jarroba.com/ordenar-un-arraylist-en-java/#comment-9018>)

asi estaria bien??...

```
Collections.sort(lpf, new Comparator() {
```

```
@Override
```

```
public int compare(PersonaForm o1, PersonaForm o2) {
return o1.getPaterno().compareTo(o2.getPaterno());
}
});
```

Responder



Ramón [Admin Jarroba] dice:

03/02/2015 a las 18:59 (<https://jarroba.com/ordenar-un-arraylist-en-java/#comment-9125>)

Hola Vitmar,

Estaría correcto también.

Te aclaro lo del final en el parámetro de la función; para que veas que no te influye. No sé si sabes el uso de «final» en variables globales, pero funciona igual en parámetros de función. El «final» convierte a una variable en NO modificable, por ejemplo para constantes (el valor del número pi no lo vamos a querer modificar). Al aplicarlo al parámetro de una función estamos diciendo que no vamos a modificar el dato que nos entre a la función. ¿Podríamos no ponerlo? Podríamos, pero al poner «final» a una variable el ordenador consume menos recursos (por tanto menos batería si nuestro programa se ejecuta en un móvil por ejemplo)

Uso de cookies

Este sitio web utiliza cookies para que usted tenga la mejor experiencia de usuario. Si continúa navegando está dando su consentimiento para la aceptación de las mencionadas cookies y la aceptación de nuestra [política de cookies](http://jarroba.com/faq/) (<http://jarroba.com/faq/>)

(<http://jarroba.com/faq/>)

Responder

ACEPTAR



Vimar Aliaga dice:

03/03/2015 a las 01:07 (<https://jarroba.com/ordenar-un-arraylist-en-java/#comment-11270>)

Gracias...



Christian dice:

05/11/2014 a las 18:49 (<https://jarroba.com/ordenar-un-arraylist-en-java/#comment-1248>)

Gracias me sirvio mucho asi "Collections.sort(personas, new Comparator())"...

Responder



Simon Mosqueda dice:

23/09/2014 a las 04:37 (<https://jarroba.com/ordenar-un-arraylist-en-java/#comment-1246>)

Hola muchas gracias por el tutorial de verdad me ayudo mucho, si me pudieras ayudar con este punto te lo agradeceria aun mas.

tengo un arraylist que esta cargado con informacion de libro, es decir cada posicion en el arraylist tiene los siguientes datos: (int CodigodelLibro, String Titulo, String Autor, String Editorial) y necesito que ingresando un autor el programa recorra el arraylist e imprima los libros que sean del mismo autor.

muchas gracias por adelantado

Responder



Ramón [Admin Jarroba] (<http://www.jarroba.com>) dice:

23/09/2014 a las 22:38 (<https://jarroba.com/ordenar-un-arraylist-en-java/#comment-1247>)

Hola Simon,
entiendo que tienes un ArrayList de objetos Libro. Si fueran objetos simples te diría que utilizaras contains(), pero como son objetos complejos tendrás que recorrerlos con un foreach e ir buscando uno a uno (mira en <http://docs.oracle.com/javase/6/docs/api/java/util/ArrayList.html> (<http://docs.oracle.com/javase/6/docs/api/java/util/ArrayList.html>))

Responder



Jose Manuel dice:

13/09/2014 a las 01:44 (<https://jarroba.com/ordenar-un-arraylist-en-java/#comment-1245>)

Este sitio web utiliza cookies para que usted tenga la mejor experiencia de usuario. Si continúa navegando está dando su consentimiento para la aceptación de las mencionadas cookies y la aceptación de nuestra política de cookies (<http://jarroba.com/faq/>)

Buenas, estoy creando una clase para ordenar una lista de objetos según su atributo
El objeto de la lista se llama cliente.

ACEPTAR

Hago el método para ordenar por nombre y sobrescribo, pero me salen 3 errores:

- Me sugiere que declare la clase abstracta
- Me dice que el método sort no es aplicable para esos argumentos
- Me dice el método compare debería sobrescribir o implementar un método «supertype»

Adjunto el código y agradezco la ayuda.

```
import java.util.Collections;
import java.util.Comparator;

public class Ordenar implements Comparable {
    public Cliente[] tempClientes;
    public Artículo[] tempArticulos;

    public Ordenar(Cliente[] clientes, Artículo[] articulos){
        tempClientes = clientes;
        tempArticulos = articulos;
    }

    public Cliente[] sortNombre(Cliente[] lista){
        tempClientes = lista;
        Collections.sort(tempClientes, new Comparator(){
            @Override
            public int compare(Cliente p1, Cliente p2){
                return new String(p1.getNombre()).compareTo(new String(p2.getNombre()));
            }
        });
        return tempClientes;
    }
}
```

Responder



Ronald Orantes dice:

26/08/2014 a las 23:03 (<https://jarroba.com/ordenar-un-arraylist-en-java/#comment-1244>)

Debo agradecerte bastante por este tutorial, me ha servido muchísimo, adelante con tu labor.

Responder



Iki dice:

30/07/2014 a las 16:56 (<https://jarroba.com/ordenar-un-arraylist-en-java/#comment-1242>)

Hola

Uso de cookies

Este sitio web utiliza cookies para que usted tenga la mejor experiencia de usuario. Si continúa navegando está dando su consentimiento para la aceptación de las mencionadas cookies y la aceptación de nuestra [política de cookies](http://jarroba.com/faq/) (<http://jarroba.com/faq/>)

(<http://jarroba.com/faq/>)

Muchas gracias por la explicación, me ha servido bastante.

ACEPTAR

Tengo una duda. Estoy haciendo un proyecto en NetBeans e hice una clase en la cual sobre escribí el compare de la clase Comparator como lo mencionas.

NetBeans me dice que mi clase no es abstracta y que no sobre escribo el método compare en Comparator.

¿Qué está sucediendo?

Responder



Ramón [Admin Jarroba] (<http://www.jarroba.com>) dice:

30/07/2014 a las 20:27 (<https://jarroba.com/ordenar-un-arraylist-en-java/#comment-1243>)

Hola,

fíjate que el «new Comparator(){...}» lo estés pasando como evento (como parámetro de la función sort()). Y dentro del cuerpo de «Comparator(){aquí entre las llaves}» sea donde hagas la sobrescritura (@override) del método compare(). Si lo pones fuera te dirá lo de que tu clase no es abstracta 😊

Responder



bek dice:

25/06/2014 a las 20:30 (<https://jarroba.com/ordenar-un-arraylist-en-java/#comment-1237>)

nesesito un ejemplo de obtener valores numericos de un joptionpane solo uno... y este que almacene todos los valores numericos en un array ya k ya llevo hecho hasta los operandos solo me falta almacenar los valores numericos para k asi me haga la operacion como si fuera una calculadora

Responder



Ramón [Admin Jarroba] (<http://www.jarroba.com>) dice:

25/06/2014 a las 21:20 (<https://jarroba.com/ordenar-un-arraylist-en-java/#comment-1238>)

Hola bek,

no es el tema del artículo, pero en la página oficial de Java tienes un ejemplo de JOptionPane (ver en <http://docs.oracle.com/javase/7/docs/api/javax/swing/JOptionPane.html> (<http://docs.oracle.com/javase/7/docs/api/javax/swing/JOptionPane.html>)). El resto es utilizar los valores como te comentamos 😊

Uso de cookies

Este sitio web utiliza cookies para que usted tenga la mejor experiencia de usuario. Si continúa navegando está dando su consentimiento para la aceptación de las mencionadas cookies y la aceptación de nuestra política de cookies (<http://jarroba.com/faq/>)

(<http://jarroba.com/faq/>)

ACEPTAR



ramon dice:

11/06/2014 a las 16:02 (<https://jarroba.com/ordenar-un-arraylist-en-java/#comment-1234>)

Excelente solución, ahora, me queda una duda, ¿como puedo lograr eso mismo pero alfabéticamente?

Tengo que ordenar por unidadEjecutora y por entidad varios elementos

unidadejecutora | entidad

128 Guanajuato
128 Aguascalientes
101 Aguascalientes
101 DistritoFederal

tendría que quedar así.

unidadejecutora | entidad

101 Aguascalientes
101 DistritoFederal
128 Aguascalientes
128 Guanajuato

apreciaría tu ayuda. gracias!

 Responder



Ramón [Admin Jarroba] (<http://www.jarroba.com>) dice:

11/06/2014 a las 21:05 (<https://jarroba.com/ordenar-un-arraylist-en-java/#comment-1235>)

Hola Ramón,

se hace igual, con el método `sort()`, ya que ordena también listas de Strings. Para tu ejemplo:

```
List miLista = new ArrayList();  
miLista.add(«128 Guanajuato»);  
miLista.add(«128 Aguascalientes»);  
miLista.add(«101 Aguascalientes»);  
miLista.add(«101 DistritoFederal»);  
Collections.sort(miLista);
```

Devuelve el resultado ordenado de:

101 Aguascalientes
101 DistritoFederal
128 Aguascalientes
128 Guanajuato

Uso de cookies

Este sitio web utiliza cookies para que usted tenga la mejor experiencia de usuario. Si continúa navegando está dando su consentimiento para la aceptación de las mencionadas cookies y la aceptación de nuestra política de cookies (<http://jarroba.com/faq/>)

(<http://jarroba.com/faq/>)

ACEPTAR

Responder



Ramón [Admin Jarroba] (<http://www.jarroba.com>) dice:

11/06/2014 a las 21:29 (<https://jarroba.com/ordenar-un-arraylist-en-java/#comment-1236>)

Te respondo otra vez, ya que creo que te refieres a que tienes los datos en objetos Pojo (y dejo el anterior comentario, por si te sirviera). Es decir, que tendrías una clase que voy a llamar «Entidad.java» en la que guardas un número entero y un nombre. Por lo que podemos rellenar la lista del siguiente modo:

```
List miLista = new ArrayList();
miLista.add(new Entidad(128, «Guanajuato»));
miLista.add(new Entidad(128, «Aguascalientes»));
miLista.add(new Entidad(101, «Aguascalientes»));
miLista.add(new Entidad(101, «DistritoFederal»));
```

Luego, para ordenar, solo has de seguir el tutorial y verás que puedes sobrescribir Comparator como sigue:

```
Collections.sort(miLista, new Comparator() {
@Override
public int compare(Entidad ent1, Entidad ent2) {
return new Integer(ent1.id).compareTo(ent2.id);
}
});
```

La anterior devuelve lo que necesitas, ya que ordena por id. Si quisieras ordenar además por nombre, tendrías que jugar con los enteros devueltos por el método compareTo(). Aunque te recomendaría que implementaras la clase Comparable dentro de la clase Pojo y que comparases uno a uno como el id del ejemplo anterior (y lo mismo para el String del nombre):

```
public static Comparator compararId = new Comparator() {
public int compare(Entidad ent1, Entidad ent2) {
return Integer(ent1.id).compareTo(ent2.id);
}
};
```

Responder



Ramón Rangel dice:

01/07/2014 a las 20:10 (<https://jarroba.com/ordenar-un-arraylist-en-java/#comment-1239>)

Te agradezco la ayuda tocayo. Lo checo y te comento en breve!

Uso de cookies

Este sitio web utiliza cookies para que usted tenga la mejor experiencia de usuario. Si continúa navegando está dando su consentimiento para la aceptación de las mencionadas cookies y la aceptación de nuestra [política de cookies](http://jarroba.com/faq/) (<http://jarroba.com/faq/>)

Responder

(<http://jarroba.com/faq/>)

ACEPTAR

**Ramón Rangel** dice:

03/07/2014 a las 16:47 (<https://jarroba.com/ordenar-un-arraylist-en-java/#comment-1240>)

Apliqué lo siguiente...
Tengo mi lista..

```

        unidadejecutora | entidad
128 Guanajuato
128 Aguascalientes
101 Aguascalientes
101 DistritoFederal

```

```

//implementé comparable como me dijiste en mi clase Pojo
public class EstructuraReporte implements Comparable {
    private Integer unidad_ejecutora;
    private String entidad;

```

```

        public Integer getUnidad_ejecutora() {
            return unidad_ejecutora;
        }

```

```

        public void setUnidad_ejecutora(Integer unidad_ejecutora) {
            this.unidad_ejecutora = unidad_ejecutora;
        }

```

```

        public String getEntidad() {
            return entidad;
        }

```

```

        public void setEntidad(String entidad) {
            this.entidad = entidad;
        }

```

```

//aplique el compareTo
public int compareTo(EstructuraReporte o) {
    //concatenando el valor texto de unidad_ejecutora y la entidad
    String a=new String(String.valueOf(this.unidad_ejecutora)+this.entidad);
    String b=new String(String.valueOf(o.unidad_ejecutora)+o.entidad);
    //comparo y listo.
    return a.compareTo(b);
}
}

```

Uso de cookies

Este sitio web utiliza cookies para que usted tenga una mejor experiencia de navegación. Si continúa navegando está dando su consentimiento para la aceptación de las mencionadas cookies y la aceptación de nuestra política de cookies (<http://jarroba.com/faq/>)

(<http://jarroba.com/faq/>)

Collection.sort(lista);

obtuve la lista con el orden que quería

unidadejecutora | entidad

101 Aguascalientes

101 DistritoFederal

128 Aguascalientes

128 Guanajuato

Agradezco la orientación...

Responder



Ramón [Admin Jarroba] (<http://www.jarroba.com>) dice:

03/07/2014 a las 19:38 (<https://jarroba.com/ordenar-un-arraylist-en-java/#comment-1241>)

Estupendo Ramón, ahora a seguir adelante 😊



oscar dice:

06/01/2014 a las 00:17 (<https://jarroba.com/ordenar-un-arraylist-en-java/#comment-1233>)

interesante

Responder

Deja un comentario

Tu dirección de correo electrónico no será publicada. Los campos obligatorios están marcados con *

Comentario

Nombre *

Uso de cookies

Este sitio web utiliza cookies para que usted tenga la mejor experiencia de usuario. Si continúa navegando está dando su consentimiento para la aceptación de las mencionadas cookies y la aceptación de nuestra política de cookies (<http://jarroba.com/faq/>)

Correo electrónico *

(<http://jarroba.com/faq/>)

ACEPTAR

Web

[Publicar comentario](#)

Este sitio usa Akismet para reducir el spam. Aprende cómo se procesan los datos de tus comentarios (<https://akismet.com/privacy/>).



(/)

Siguenos en:

<https://www.youtube.com/user/JarrobaWeb><https://twitter.com/JarrobaWeb><https://www.facebook.com/jarrobaWeb><https://github.com/jarroba?tab=repositories><https://plus.google.com/103352032205227460108/>Contacto: jarrobaweb@gmail.com (<mailto:jarrobaweb@gmail.com>)

[FAQ \(HTTPS://JARROBA.COM/FAQ/\)](https://jarroba.com/faq/)

[VISITAS \(HTTPS://JARROBA.COM/VISITAS/\)](https://jarroba.com/visitas/)

[GASTOS](#)

[MAPA DEL SITIO \(HTTPS://JARROBA.COM/SITEMAP_INDEX.XML\)](https://jarroba.com/sitemap_index.xml)

[HERRAMIENTAS \(HTTPS://JARROBA.COM/TOOLS/\)](https://jarroba.com/tools/)

Copyright © 2019 Jarroba.com (<https://www.jarroba.com>) - Todos los derechos reservados

By Reimon & Richard

Uso de cookies

Este sitio web utiliza cookies para que usted tenga la mejor experiencia de usuario. Si continúa navegando está dando su consentimiento para la aceptación de las mencionadas cookies y la aceptación de nuestra [política de cookies](http://jarroba.com/faq/) (<http://jarroba.com/faq/>)

(<http://jarroba.com/faq/>)

[ACEPTAR](#)
