



Test final!!  rocksatesteban@gmail.com Cambiar cuenta	<u>∽</u>
* Indica que la pregunta es obligatoria	
Java  Cuántas veces se ejecutara el siguiente for?*  2 punt	tos
<pre>for (int i = 2; i &lt;= 128; i *= 2) {     System.out.println("i " + i); }</pre>	
<ul><li>Mostrará error</li><li>8</li></ul>	
<ul><li>128</li><li>7</li></ul>	
Los atributos static *	tos
Solo pueden ser usados dentro de metodos/funciones del tipo static	
<ul> <li>Solo pueden ser usados dentro de metodos/funciones del tipo NO static</li> <li>Son si o si son del tipo de alguna Clase (como por ejemplo Integer)</li> <li>Ninguna de las anteriores es correcta</li> </ul>	
Triliguilla de las aliteriores es correcta	
¿Qué es el encapsulamiento? *  Hace referencia a crear métodos para obtener y modificar información de un objeto, manteniendo oculto lo que hace un objeto por dentro.  Hace referencia a los métodos constructores con y sin parámetros  Hace referencia a separar el código en "cápsulas" para mejor lectura del mismo.	nto
¿Cuál de estas clases NO es parte del Collections * 2 punt	tos
Framework?  Collection	
<ul><li>Matriz</li><li>Iterator</li></ul>	
○ LinkedList	
¿Cuál de los siguientes métodos NO es parte de la * 1 pur clase String?	nto
.stringLength()  concat()	
.isEmpty() .toUpperCase()	
¿Qué significa el acrónimo UML?*	tos
Ninguna de las anteriores	
Lenguaje de Modelo Único  Lenguaje de Modelado Unificado	
Lenguaje de Mapas sobre Uniones	
¿Cuál de las siguientes no es una Relación de clases? * 2 punt  O Asociación	tos
<ul><li>Navegacion</li><li>Agregación</li></ul>	
Composición	
¿Qué es un constructor?*	nto
<ul><li>Un método void</li><li>Un método que nos define qué atributos tiene la clase</li></ul>	
Es un método que nos permite instanciar un objeto de la clase donde se encuentra	
¿Qué hace el siguiente Comparator?* 2 punt	tos
<pre>public static Comparator<estudiante> compararEstudiantes= new Comparator<estudiante>() {    @Override    public int compare(Estudiante e1, Estudiante e2) {       return e2.getNombre().compareTo(e1.getNombre());    } };</estudiante></estudiante></pre>	
Compara los estudiantes según su nombre de manera descendente  Ninguna de las anteriores	
<ul> <li>Ninguna de las anteriores</li> <li>Compara los estudiantes según su nombre de manera ascendente</li> <li>Compara los estudiantes según su edad de manera ascendente</li> </ul>	
<ul><li>Ninguna de las anteriores</li><li>Compara los estudiantes según su nombre de manera ascendente</li></ul>	tos
<ul> <li>Ninguna de las anteriores</li> <li>Compara los estudiantes según su nombre de manera ascendente</li> <li>Compara los estudiantes según su edad de manera ascendente</li> <li>¿Qué mostraría por consola? *</li> </ul>	tos
<pre>Ninguna de las anteriores Compara los estudiantes según su nombre de manera ascendente Compara los estudiantes según su edad de manera ascendente  ¿Qué mostraría por consola?*  try {     double resul = 10 / 0;     System.out.println("Exito"); } catch (ArithmeticException e) {     System.out.println("Error aritmetico"); } catch (Exception e) {</pre>	tos
<pre>Ninguna de las anteriores Compara los estudiantes según su nombre de manera ascendente Compara los estudiantes según su edad de manera ascendente  ¿Qué mostraría por consola?*  try {     double resul = 10 / 0;     System.out.println("Exito"); } catch (ArithmeticException e) {     System.out.println("Error aritmetico"); } catch (Exception e) {     System.out.println("Error Exception"); } finally {     System.out.println("Terminamos");</pre>	tos
<pre>Ninguna de las anteriores Compara los estudiantes según su nombre de manera ascendente Compara los estudiantes según su edad de manera ascendente  ¿Qué mostraría por consola?*  try {     double resul = 10 / 0;     System.out.println("Exito"); } catch (ArithmeticException e) {     System.out.println("Error aritmetico"); } catch (Exception e) {     System.out.println("Error Exception"); } finally {</pre>	tos
<pre>Ninguna de las anteriores Compara los estudiantes según su nombre de manera ascendente Compara los estudiantes según su edad de manera ascendente  ¿Qué mostraría por consola?*  try {     double resul = 10 / 0;     System.out.println("Exito"); } catch (ArithmeticException e) {     System.out.println("Error aritmetico"); } catch (Exception e) {     System.out.println("Error Exception"); } finally {     System.out.println("Terminamos"); }</pre>	tos
<pre>Ninguna de las anteriores Compara los estudiantes según su nombre de manera ascendente Compara los estudiantes según su edad de manera ascendente  try {     double resul = 10 / 0;     System.out.println("Exito"); } catch (ArithmeticException e) {     System.out.println("Error aritmetico"); } catch (Exception e) {     System.out.println("Error Exception"); } finally {     System.out.println("Terminamos"); } </pre> <pre>     "Error aritmetico", "Error Exception", "Terminamos"     "Exito", "Terminamos"</pre>	tos
<pre>Ninguna de las anteriores Compara los estudiantes según su nombre de manera ascendente Compara los estudiantes según su edad de manera ascendente  try {     double resul = 10 / 0;     System.out.println("Exito"); } catch (ArithmeticException e) {     System.out.println("Error aritmetico"); } catch (Exception e) {     System.out.println("Error Exception"); } finally {     System.out.println("Terminamos"); } </pre> <pre> "Error aritmetico", "Error Exception", "Terminamos" "Exito", "Terminamos" "Error Exception", "Terminamos"</pre>	
O Ninguna de las anteriores Compara los estudiantes según su nombre de manera ascendente Compara los estudiantes según su edad de manera ascendente  ¿Qué mostraría por consola? *  try {     double resul = 10 / 0;         System.out.println("Exito"); } catch (ArithmeticException e) {         System.out.println("Error aritmetico"); } catch (Exception e) {         System.out.println("Error Exception"); } finally {         System.out.println("Terminamos"); }  "Error aritmetico", "Error Exception", "Terminamos"  "Error Exception", "Terminamos"  "Error Exception", "Terminamos"  "Error aritmetico", "Terminamos"	
Ninguna de las anteriores  Compara los estudiantes según su nombre de manera ascendente  Compara los estudiantes según su edad de manera ascendente  ¿Qué mostraría por consola?*  try {     double resul = 10 / 0;     System.out.println("Exito"); } catch (ArithmeticException e) {     System.out.println("Error aritmetico"); } catch (Exception e) {     System.out.println("Error Exception"); } finally {     System.out.println("Terminamos"); }  "Error aritmetico", "Error Exception", "Terminamos"  "Error Exception", "Terminamos"  "Error aritmetico", "Terminamos"  La clase padre de todas las excepciones y errores, es * 2 punt la clase?  Throwable	
Ninguna de las anteriores  Compara los estudiantes según su nombre de manera ascendente  Compara los estudiantes según su edad de manera ascendente  try {     double resul = 10 / 0;     System.out.println("Exito"); } catch (ArithmeticException e) {     System.out.println("Error aritmetico"); } catch (Exception e) {     System.out.println("Error Exception"); } finally {     System.out.println("Terminamos"); }  "Error aritmetico", "Error Exception", "Terminamos"  "Exito", "Terminamos"  "Error aritmetico", "Terminamos"  Error aritmetico", "Terminamos"  Throwable  Ninguna de las anteriores  Error  Exception	tos
Ninguna de las anteriores  Compara los estudiantes según su nombre de manera ascendente  Compara los estudiantes según su edad de manera ascendente  ¿Qué mostraría por consola?*  try {     double resul = 10 / 0;         System.out.println("Exito");     } catch (ArithmeticException e) {         System.out.println("Error aritmetico");     } catch (Exception e) {         System.out.println("Error Exception");     } finally {         System.out.println("Terminamos");     }  "Error aritmetico", "Error Exception", "Terminamos"  "Exito", "Terminamos"  "Error Exception", "Terminamos"  ¿La clase padre de todas las excepciones y errores, es * 2 punt la clase?  Throwable  Ninguna de las anteriores  Error  Excepction  ¿Las clases en UML se representan con un?*  2 punt	tos
Ninguna de las anteriores Compara los estudiantes según su nombre de manera ascendente Compara los estudiantes según su edad de manera ascendente  try {     double resul = 10 / 0;     System.out.println("Exito"); } catch (ArithmeticException e) {     System.out.println("Error aritmetico"); } catch (Exception e) {     System.out.println("Error Exception"); } finally {     System.out.println("Terminamos"); }  "Error aritmetico", "Error Exception", "Terminamos" "Error Exception", "Terminamos" "Error aritmetico", "Terminamos"  "Error aritmetico", "Terminamos"  La clase padre de todas las excepciones y errores, es * 2 punt la clase?  Throwable Ninguna de las anteriores  Error Exception  Las clases en UML se representan con un? * 2 punt la clase clase en umana la clase en umana con un? * 2 punt la clase clase en umana con un? * 2 punt la clase en umana clas anteriores  Las clases en umana clas anteriores  Circulo Ninguna de las anteriores	tos
Ninguna de las anteriores  Compara los estudiantes según su nombre de manera ascendente  Compara los estudiantes según su edad de manera ascendente  ¿Qué mostraría por consola?*  try {     double resul = 10 / 0;     System.out.println("Exito"); } catch (arithmeticException e) {     System.out.println("Error aritmetico"); } catch (Exception e) {     System.out.println("Error Exception"); } finally {     System.out.println("Terminamos"); }  "Error aritmetico", "Error Exception", "Terminamos"  "Exito", "Terminamos"  "Error Exception", "Terminamos"  ¿La clase padre de todas las excepciones y errores, es * 2 punt la clase?  Ninguna de las anteriores  Error  Excepction  ¿Las clases en UML se representan con un? *  ¿ punt Triangulo  Circulo	tos
Ninguna de las anteriores Compara los estudiantes según su nombre de manera ascendente Compara los estudiantes según su edad de manera ascendente  try {     double resul = 10 / 0;     System.out.println("Exito"); } catch (ArithmeticException e) {     System.out.println("Error aritmetico"); } catch (Exception e) {     System.out.println("Error Exception"); } finally {     System.out.println("Terminamos"); }  "Error aritmetico", "Error Exception", "Terminamos" "Error Exception", "Terminamos" "Error aritmetico", "Terminamos"  "Error aritmetico", "Terminamos"  La clase padre de todas las excepciones y errores, es * 2 punt la clase?  Throwable Ninguna de las anteriores  Error Exception  Las clases en UML se representan con un? * 2 punt la clase clase en umana la clase en umana con un? * 2 punt la clase clase en umana con un? * 2 punt la clase en umana clas anteriores  Las clases en umana clas anteriores  Circulo Ninguna de las anteriores	tos
Ninguna de las anteriores Compara los estudiantes según su nombre de manera ascendente Compara los estudiantes según su edad de manera ascendente  ¿Qué mostraría por consola?*  try {     double resul = 10 / 0;         System.out.println("Exito");     catch (ArithmeticException e) {         System.out.println("Error aritmetico");     } catch (Exception e) {         System.out.println("Brror Exception");     } finally {         System.out.println("Terminamos");     }  "Error aritmetico", "Error Exception", "Terminamos"  "Error Exception", "Terminamos"  ¿La clase padre de todas las excepciones y errores, es * 2 punt la clase?  Throwable Ninguna de las anteriores Error Excepction  ¿Las clases en UML se representan con un? *  ¿Las clases en Comparatores Error Excepction  ¿Las clases en UML se representan con un? *  ¿Las clas	tos
Ninguna de las anteriores  Compara los estudiantes según su nombre de manera ascendente  Compara los estudiantes según su edad de manera ascendente  ¿Qué mostraría por consola? *  2 punt  try {     double resul = 10 / 0;         System. println ("Exito");     } catch (ArithmeticException e) {         System. println ("Error aritmetico");     } catch (Exception e) {         System. println ("Error Exception");     } finally {         System. println ("Terminamos");     }  "Error aritmetico". "Error Exception", "Terminamos"  "Exito". "Terminamos"  "Error Exception", "Terminamos"  "Error Exception", "Terminamos"  La clase?  Throwable  Ninguna de las anteriores  Error  Exception  ¿Las clases en UML se representan con un? *  ¿Las clases en UML se representan con un? *  ¿Cómo se llama el archivo que usa Maven para construir sus dependencias?  Dependencies-propperties.xml	tos
Ninguna de las anteriores Compara los estudiantes según su nombre de manera ascendente Compara los estudiantes según su edad de manera ascendente  try {     double resul = 10 / 0;     System.out.println("Exito");     catch (Arithmeticexception e) {         System.out.println("Error aritmetico");     } catch (Exception e) {         System.out.println("Error Exception");     } tinally {         System.out.println("Terminamos");     }      "Error aritmetico", "Error Exception", "Terminamos"     "Error Exception", "Terminamos"     "Error Exception", "Terminamos"      "Error Exception", "Terminamos"      "Error Exception", "Terminamos"      "Error Exception", "Terminamos"      "Error Exception", "Terminamos"      "Error Exception e) {         Circulo	tos
Ninguna de las anteriores  Compara los estudiantes según su nombre de manera ascendente  Compara los estudiantes según su edad de manera ascendente  Ly (	tos
Ninguna de las anteriores  Compara los estudiantes según su nombre de manera ascendente  Compara los estudiantes según su edad de manera ascendente  Las double resul = 10 / 0; System.out.println ("Exito"); Catch (Arithmeticexception e) ( System.out.println ("Error aritmetico"); Catch (Exception e) ( System.out.println ("Error Exception"); Catch (Exception e) ( System.out.println ("Terminamos"); Catch (Exception e) ( System.out.println ("Terminamos"); Catch (Exception e) ( System.out.println ("Terminamos"); Catch (Exception e) ( Catch (Exception	tos
Ninguna de las anteriores  Compara los estudiantes según su nombre de manera ascendente  Compara los estudiantes según su edad de manera ascendente  ¿Qué mostraría por consola? *  zami  try {     double resul = 10 / 0;         Systemui.println("exito"); } catch (ArithmeticException e) {         Systemui.println("Error aritmetico"); } catch (Exception e) {         Systemui.println("Error Exception"); } finally {         Systemui.println("Terminamos"); }  "Error aritmetico", "Terminamos"  "Error Exception", "Terminamos"  "Error aritmetico", "Terminamos"  ¿La clase padre de todas las excepciones y errores, es * ² aum la clase?  Throwable  Ninguna de las anteriores  Error  Exception  ¿Las clases en UML se representan con un? *  ¿Las clases en UML se representan con un? *  ¿Cómo se llama el archivo que usa Maven para construir sus dependencias?  Dependencies propperties xml  POM.xml  Application-context.xml  Para decir que una clase hereda a otra se usa la palabra:  Ninguna de las anteriores  Extends  Para decir que una clase hereda a otra se usa la palabra:  Ninguna de las anteriores  Extends	tos
Ninguna de las anteriores  Compara los estudiantes según su nombre de manera ascendente  Compara los estudiantes según su edad de manera ascendente  Las clases en uma la printin ("Estro");  Systemuiprintin ("Estro");  Catch (ArithmeticException e) ( Systemuiprintin ("Estror aritmetico");  Catch (Exception e) ( Systemuiprintin ("Estror Exception");  Inally ( Systemuiprintin ("Terminamos");  Iterror aritmetico", "Terminamos"  Las clase padre de todas las excepciones y errores, es la clase?  Throwable  Ninguna de las anteriores  Error  Exception  Las clases en UML se representan con un?*  Las clases en UML se representan con un?*  Las clases en UML se representan con un?*  Pomo estangulo  Circulo  Ninguna de las anteriores  Rectangulo  Circulo  Ninguna de las conteriores  Rectangulo  Pomo mul  Application-context.xml  Pomo mul  Application-context.xml  Para decir que una clase hereda a otra se usa la palabra:  Ninguna de las anteriores  Exends  Implements  Ninguna de las anteriores	tos
Ninguna de las anteriores  Compara los estudiantes según su nombre de manera ascendente  Compara los estudiantes según su edad de manera ascendente  Ly {  double resul = 10 / 0;  Systemut.println ("Exito"); } catch (Arithmeticexception e) {  Systemut.println ("Error aritmetico"); } catch (Exception e) {  Systemut.println ("Error Exception"); } finally {  Systemut.println ("Terminamos"); }  "Error aritmetico", "Error Exception: "Terminamos"); }  "Error Exception", "Terminamos"  "Error Exception", "Terminamos"  La clase padre de todas las excepciones y errores, es * 2 para la clase?  Throwable  Ninguna de las anteriores  Error  Exception  Las clases en UML se representan con un? * 2 para construir sus dependencias?  Circulo  Ninguna de las anteriores  Rectangulo  ¿Cómo se llama el archivo que usa Maven para construir sus dependencias?  Dependencies-propperties.xml  POM.xml  Application-propperties.xml  Application-contest.xml  Para decir que una clase hereda a otra se usa la palabra:  Ninguna de las anteriores.  Extends  Implements  Implements  Implements  Implements	tos
Ninguna de las anteriores	tos
Ningura de las anteriores  Compara los estudiantes según su nombre de manera ascendente  Cumpara los estudiantes según su redad de manera ascendente  ¿Qué mostraria por consola? *  try {	tos
Ninguna de las anteriores  Campara los estudiantes según su rumben de numero acandente  Compara los estudiantes según su edad de manera ascendente  ¿Qué mostraría por consola? * * * * * * * * * * * * * * * * * * *	tos
Ninguna dicias anteriores  Compara los estudiantes según su nombre de macera ascendente  Compara los estudiantes según su cedad de manera ascendente  ¿Qué mostraría por consola?*  try {     double resul = 10 / 0;     Systemouprintln("Exception e) {         Systemouprintln("Exception e) {             Systemouprintln("Exror aritmetico");         } catch (Exception e) {             Systemouprintln("Error Exception");         } finally {             Systemouprintln("Error Exception");         } finally {             Systemouprintln("Terminamos");         }              "Fror aritmetico", "Fror Exception", "Terminamos"              "Error Exception "Terminamos"              "Error Exception", "Terminamos"              "Error Exception              "Error Exception "Terminamos"              "Error Exception "Term	tos
Ninguna de las anteriores  Compara luc estudiantes según su nontre de manera ascendente  Compara luc estudiantes según su nontre de manera ascendente  Compara luc estudiantes según su nontre de manera ascendente  Compara luc estudiantes según su nontre de manera ascendente  Compara luc estudiantes según su nontre de manera ascendente  Compara luc estudiantes según su nontre de manera ascendente  Compara luc estudiantes según su nontre de manera ascendente  Compara luc estudiantes según su nontre de manera ascendente  Compara luc estudiantes según su nontre de manera ascendente  Compara luc estudiantes según su nontre de manera ascendente  Compara luc estudiantes según su nontre de manera ascendente  Compara luc estudiantes según su nontre de manera ascendente  Compara luc estudiantes según su nontre de manera ascendente  Compara luc estudiantes según su nontre de manera ascendente  Compara luc estudiantes según su nontre la luc estudiante luc estudiantes luc e	tos
Ninguna dicias anteriores  Compara los estudiantes según su nombre de macera ascendente  Compara los estudiantes según su cedad de manera ascendente  ¿Qué mostraría por consola?*  try {     double resul = 10 / 0;     Systemouprintln("Exception e) {         Systemouprintln("Exception e) {             Systemouprintln("Exror aritmetico");         } catch (Exception e) {             Systemouprintln("Error Exception");         } finally {             Systemouprintln("Error Exception");         } finally {             Systemouprintln("Terminamos");         }              "Fror aritmetico", "Fror Exception", "Terminamos"              "Error Exception "Terminamos"              "Error Exception", "Terminamos"              "Error Exception              "Error Exception "Terminamos"              "Error Exception "Term	tos tos

Página 2 de 8 Borrar formulario Siguiente Atrás

¿Cuál de estas herencias no es soportada en Java?\*

2 puntos

Con el símbolo \_ luego del nombre del objeto

Herencia Multinivel

Herencia Jerarquica

Herencia Múltiple

Nunca envíes contraseñas a través de Formularios de Google.

Herencia Simple