**Dia 1**

**Que es un applet?**

R// Es una pequeña aplicación que puede incrusta en una página web para crear contenido dinámico que se ejecuta dentro del navegador.

**En el Desarrollo GUI en Java se cuenta con AWT y SWING, en que difieren?**

R//

|  |  |  |
| --- | --- | --- |
| **DIFERENCIAS** | **AWT** | **SWING** |
| Tipo | Los componentes son más pesados. | Los componentes son más livianos. |
| Diferencias de la plataforma | Depende del SO. | No depende del SO. |
| Monitor | No admite una apariencia y tacto conectables. | Soporta una apariencia y tacto enchufable. |
| Componentes | Swing tiene componentes más avanzados que awt. | |
| Velocidad | La ejecución es más lenta. | La ejecución es más rápida. |
| MVC | No admite el patrón mvc. | Si admite el patrón mvc. |
| Espacio de memoria | Los componentes requieren más espacio de memoria. | Los componentes requieren menos espacio de memoria. |

Principal diferencia: AWT es el conjunto de herramientas de widgets, gráficos e interfaz de usuario dependientes de la plataforma java original, mientras que swing es un conjunto de herramientas de widget GUI para java que es una extensión de AWT.

**Que es Javadoc, identifique por lo menos 5 etiquetas**

R// Javadoc es una utilidad incluida en el JDK, que genera documentación automática en formato HTML, a partir de código fuente java.

1. @author
2. @version
3. @param
4. @return
5. @trhows

**Indique 5 características básicas de la POO**

R// Ocultación, Encapsulación, Abstracción, Polimorfismo, Herencia.

**Mencione al menos dos programas de la JDK**

R// javac.exe = Compilador de Java, appletviewer.exe = Visor de applets que genera sus vistas previas.

**Dia 2**

**Diferencia | o || y & o && y de un ejemplo de cada uno.**

R// &(And bitwise) o |(Or bitwise)

&&(And Operador Logico) o ||(Or Operador Logico) Short-Wired

& y | son utilizados para hacer And y Or bit por bit, es una operación lógica booleana, trabaja con dos tipos de operando, int y boolean

&& y || son operadores lógicos, trabajan con dos operandos de tipo boolean

El unitario evalua ambos lados de la expresión y el doble solo evalua un lado de la expresión.

**int** f = 12, g = 25;  
System.***out***.println(f & g); *// Salida = 8*System.***out***.println(f | g); *// Salida = 29*

**Evaluando (12 & 25)**

|  |  |
| --- | --- |
|  |  |
| 12 | 00001100 |
| 25 | 00011001 |
| 8 | 00001000 |

**Evaluando (12 | 25)**

|  |  |
| --- | --- |
|  |  |
| 12 | 00001100 |
| 25 | 00011001 |
| 29 | 00011101 |

**boolean** a = **true** && **false**;  
**boolean** b = **true** & **false**;  
System.***out***.println(**"A: "** + a + **" B: "** + b);  
  
**boolean** c = **true** || **false**;  
**boolean** d = **true** | **false**;  
System.***out***.println(**"C: "** + c + **" D: "** + d);

Salida:

A: false B: false

C: true D: true

**Día 4**

**Investigue la clase “Date”, identifique los métodos más comunes.**

R// Cuando se crea un objeto de la clase date representa un momento particular, cuando se necesite identificar atributos que representen una fecha, serán de tipo date.

La clase date representa un instante especifico en el tiempo, con precisión de milisegundos.

Métodos:

1. toString()
2. after()
3. before()
4. getTime()
5. equals()
6. setTime()
7. getDate()
8. getDay()
9. getMonth()
10. getYear()

**Investigue la clase “Calendar”.**

R// Es una clase abstracta que proporciona métodos para convertir entre un instante especifico en el tiempo y un conjunto de campos de calendario como año, mes, día\_de\_mes, etc, y para manipular los campos del calendario, como obtener fecha.

**Dia 10**

**Para que se utiliza el CallableStatament**

R// Se utiliza para hacer llamadas a procedimientos almacenados de la base de datos.

Para que la conexion hacia la base de datos funcione se debe ejecutar esta línea en la base de datos SET GLOBAL time\_zone=”+1:00”;

**import** java.sql.\*;  
  
**public class** ConexionMysql {  
  
 **private static** Connection *connection* = **null**;  
 **private static** CallableStatement *callableStatement* = **null**;  
 **private static** ResultSet *resultSet* = **null**;  
 **private static** String *url* = **"jdbc:mysql://localhost:3306/java?serverTimezone=GMT"**, *user* = **"root"**, *pass* = **""**;  
  
 **public static void** main(String[] args) {  
  
 **try** {  
 Class.*forName*(**"com.mysql.jdbc.Driver"**);  
 *connection* = DriverManager.*getConnection*(*url*, *user*, *pass*);  
  
 *callableStatement* = *connection*.prepareCall(**"{call** *MostrarUsuarios***(?)}"**);  
 *callableStatement*.setInt(1, 2);  
 *resultSet* = *callableStatement*.executeQuery();  
  
 **while** (*resultSet*.next()) {  
 System.***out***.println(*resultSet*.getString(**"nombre"**) + **" "** +  
 *resultSet*.getString(**"apellido"**) + **" "** +  
 *resultSet*.getString(**"telefono"**) + **" "** +  
 *resultSet*.getInt(**"id"**));  
 }  
 *connection*.close();  
  
 } **catch** (SQLException e) {  
 e.printStackTrace();  
 } **catch** (ClassNotFoundException e) {  
 e.printStackTrace();  
 }  
 }  
}