Java Intermedio

ERICK:2, 15, 16,17

Para Yaxkin en hojas aparte:
Que es eso de Yaxkin? dejo una tarea aparte

Patrones de diseño(definición, que entiendo y bibliografia)
¿Tambièn el diagrama de gant, no?
que habia dejado el bato,andale esa madre
el diagrama? es mucho? creo que es sobre como organizas tu tiempo, asì que nel pastel
gracia s camaleon anonimo jajaj de nada Canguro Anonimo
Eso :v Lo de los patrones de diseño y el diagrama de Gant, pero no entendì si ese aùn :v
Lo del diagrama de Gant es para el proyecto de redes jajaja en serio es para el proecto de redes? :v y ese què pez? es a mano?
yo digo que andre se moche como la del puebla, si o no raza? Simon yo te apoyo

0-¿Còmo tenr una entrada de datos sin utilizar System.in?
usando la clase consola
Console consola = System.console();
String line=consola.readLine();

1. Instalar NetBeans
esta facil, solo descarguenlo de netbeans.org y ejecuten en archivo (abrir con bash) y se instala solo como por magia alv

2. ¿Qué es un hilo?
Un hilo es un flujo de control dentro de un programa. Creando varios hilos podremos realizar varias tareas simultáneamente.

/*Cada hilo tendrá sólo un contexto de ejecución (contador de programa, pila de ejecución). Es decir, a diferencia de los procesos UNIX, no tienen su propio espacio de memoria sino que acceden todos al mismo espacio de memoria común, por lo que será importante su sincronización cuando tengamos varios hilos accediendo a los mismos objetos.
*/

3. ¿Qué es un proceso?
En informàtica:
Un proceso puede informalmente entenderse como un programa en ejecución.
Formalmente un proceso es "Una unidad de actividad que se caracteriza por la ejecución de una secuencia de instrucciones, un estado actual, y un conjunto de recursos del sistema asociados"

En concepto general:
Un proceso es una secuencia de pasos dispuesta con algún tipo de lógica que se enfoca en lograr algún resultado específico

4. Diferencias entre hilo y proceso

En muchos de los sistemas operativos que dan facilidades a los hilos, es más rápido cambiar de un hilo a otro dentro del mismo proceso, que cambiar de un proceso a otro. Este fenómeno se debe a que los hilos comparten datos y espacios de direcciones, mientras que los procesos, al ser independientes, no lo hacen.

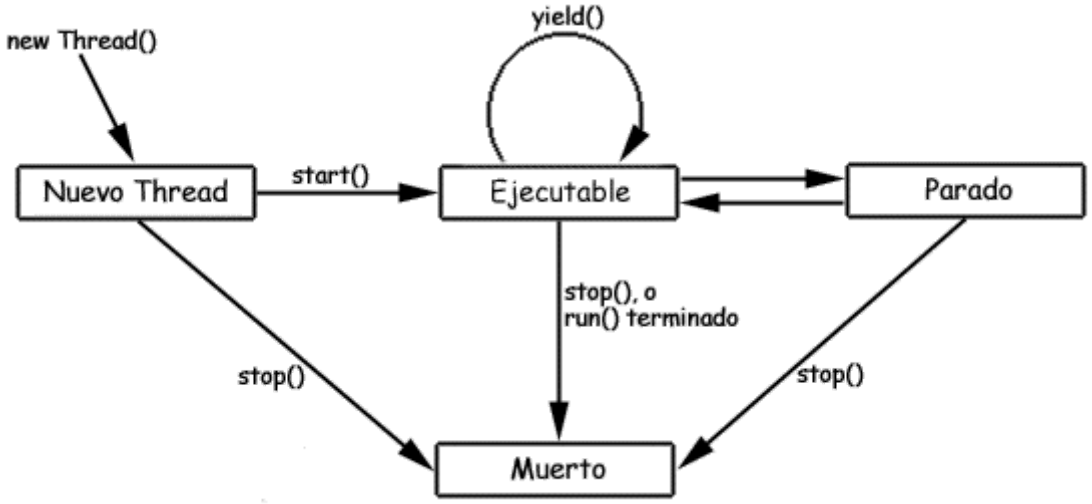
5. ¿Qué es la programación concurrente?

Hace referencia a las técnicas de programación que son utilizadas para expresar la concurrencia entre tareas y solución de los problemas de comunicación y sincronización entre procesos. La programación concurrente es la ejecución simultánea de múltiples tareas interactivamente. Estas tareas pueden ser un conjunto de procesos o hilos de ejecución creados por un único programa. Las tareas se pueden ejecutar en una sola CPU (multiprogramación), en varios procesadores, o en una red de computadores distribuidos.

6. ¿Qué es la programación en paralelo?

La programación paralela es el uso de varios procesadores trabajando en conjunto para dar solución a una tarea en común, lo que hacen es que se dividen el trabajo y cada procesador hace una porción del problema al poder intercambiar datos por una red de interconexión o atraves de memoria.

7. Diagrama de estados de los threads



8. Investigar la clase Thread

clase que encapsula todo el control necesario sobre los hilos de ejecución (threads).

9. Investigar la interfaz Runnable

implementando la interfaz Runnable se usa para crear un hilo, al iniciar el hilo se llama al método de ejecución del objeto en ese hilo que se ejecuta por separado.

10. ¿Que es la sincronización de hilos?

Cuando en un programa tenemos varios hilos corriendo simultáneamente es posible que varios hilos intenten acceder a la vez a un mismo sitio (un fichero, una conexión, un array de datos) y

es posbible que la operación de uno de ellos entorpezca la del otro. Para evitar estos problemas, hay que sincronizar los hilos. Por ejemplo, si un hilo con vocación de Cervantes escribe en fichero "El Quijote" y el otro con vocación de Shakespeare escribe "Hamlet", al final quedarán todas las letras entremezcladas. Hay que conseguir que uno escriba primero su Quijote y el otro, luego, su Hamlet.

11. ¿Qué es un socket?

Los sockets son basicamente formas en las que podemos interconectar 2 (o mas) programas mediante el uso de la internet. En java se utilizan para poder crear conexiones utilizando basicamente una IP/hostname y un puerto para establecer la conexión.
El modelo mas basico de los sockets consta de 2 simples programas, un **servidor** y un **cliente.** Basicamente el programa servidor comienza a "escuchar" en un puerto determinado(nosotros lo especificamos), y posteriormente el programa que la hace de "cliente" debe conocer la ip o nombre de dominio/hostname del servidor y el puerto que esta escuchando, al saber esto simplemente solicita establecer una conexión con el servidor. Es aqui cuando el servidor acepta esa conexión y se puede decir que estos programas estan "conectados", de este modo pueden intercambiar información.

12. Investigar las clases: Socket y SocketServer

La clase Socket del paquete java.net, java oculta las complejidades derivadas del establecimiento de la conexión de red y del envío de datos a través de ella. En esencia, el paquete java.net proporciona la misma interfaz de programación que se utiliza cuando se trabaja con archivos.

La clase ServerSocket es la que se utiliza a la hora de crear servidores, al igual que como se ha visto, la clase Socket se utiliza para crear clientes.

En java se utilizan para poder crear conexiones utilizando básicamente una IP/hostname y un puerto para establecer la conexión

13. ¿Qué es comunicación síncrona y asíncrona?

comunicación sincrónica es el intercambio de información por Internet en tiempo real.
asincrona aquella que permite la **comunicación** por Internet entre personas de forma no simultánea.

14. ¿Qué es GUI?

programa informático que actúa de interfaz de usuario, utilizando un conjunto de imágenes y objetos gráficos para representar la información y acciones disponibles en la interfaz.

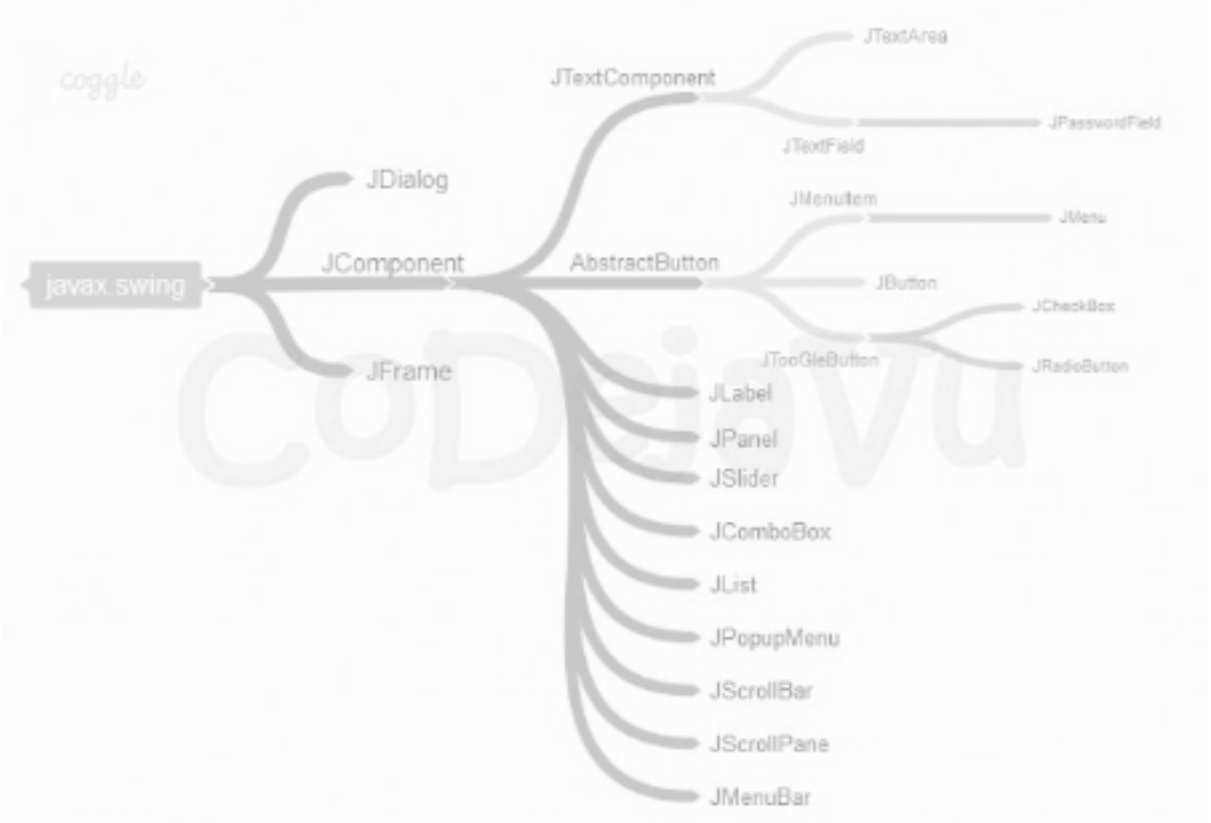
15. ¿Qué es SWING?

Swing es un toolkit para Java. Es una biblioteca de interfaces gráficas de usuario (GUI). Incluye widgets para interfaz gráfica de usuario tales como cajas de texto, botones, desplegables y tablas.

/*Sus características incluyen:

- Todas las prestaciones de AWT
- Componentes 100% Java de las versiones de los componentes de AWT
- Un rico conjunto de componentes de alto nivel (listas en árbol, paneles de pestañas, etc...)
- Un diseño Java puro, no depende de terceros.
- Look and feel intercambiable
-

*/
/*los Swing se reconocen porque anteponen la letra J antes del nombre, por ejemplo JButton*/
//por interés



http://codejavu.blogspot.mx/2013/08/que-es-java-swing.html

16. ¿Qué es AWT?

La larga: AWT son las siglas de Abstract Window ToolKit, las herramientas originales de Java para crear interfaces gráficas.
Es una librería GUI portable para aplicaciones autónomas y/o applets, proporciona la conexión entre nuestra aplicación y el GUI nativo.

La corta: es un conjunto de librerias enfocadas a la construcción de interfaces

/*Las prestaciones de AWT incluyen:

- Un amplio grupo de componentes de usuario

- Un modelo de manejo de eventos robusto
- Herramientas gráficas y de imágenes (clases de Formas, colores y fuentes)
- Manejadores de diseño que no dependen del tamaño de pantalla o resolución
- Clases de transferencia de datos, para copiar-pegar a través del portapapeles de la plataforma

*/

/*si queremos que nuestra aplicación corra en cualquier entorno, independientemente de la antigüedad del mismo, deberíamos usar AWT*/

http://codejavu.blogspot.mx/2013/08/que-es-java-swing.html
http://swing-facil.blogspot.mx/2012/03/java-gui-swing-o-awt.html

17. Investigar:

- **JFrame** – Es la Ventana de aplicación, el contenedor principal
- **JDialog** – Una ventana de tipo Ventana de diálogo, tambien puede ser un contenedor principal.
- **JPanel** – Permite la creación de paneles independientes donde se almacenan otros componentes.
- **JScrollPane** – permite la vinculación de barras de desplazamiento en un contenedor.
- **JSplitPane** – permite la creación de un contenedor dividido en 2 secciones.
- **JTabbedPane** – Permite la creación de pestañas, cada pestaña representa un contenedor independiente.
- **JDesktopPane** – Permite crear ventanas dentro de una ventana principal
- **JToolBar** – Permite introducir una Barra de herramientas

- **JLabel** – Permite Vincular Etiquetas, tanto de texto como de imagenes
- **JButton** – Permite vincular Botones simples.
- **JCheckBox** – Son Casilla de verificación, ideal para selección múltiples.
- **JRadioButton** – Permite presentar opciones de selección similares a las checkbox, solo que el enfoque de estas es de única selección.
- **JToggleButton** – Botón que al oprimirlo se quedará presionado hasta que se ejecute otro evento.
- **JComboBox** – Permite mostrar una lista de elementos como un combo de selección.
- **JScrollBar** – Permite mostrar una barra de desplazamiento, regularmente usada en Areas de texto o paneles donde el contenido es mayor que el tamaño del componente.
- **JSeparator** – Permite separar

- **JToggleButton** – Botón que al oprimirlo se quedará presionado hasta que se ejecute otro evento.
- **JComboBox** – Permite mostrar una lista de elementos como un combo de selección.
- **JScrollBar** – Permite mostrar una barra de desplazamiento, regularmente usada en Areas de texto o paneles donde el contenido es mayor que el tamaño del componente.
- **JSeparator** – Permite separar opciones, es una barra simple.
- **JSlider** - Permite vincular un Deslizador en nuestra ventana.
- **JSpinner** – permite vincular una caja de texto con botones integrados para seleccionar algún valor.
- **JProgressBar** – Establece una barra de progreso.

- **JTextField** – Permite introducir un campo de texto simple.
- **JFormattedTextField** – Permite introducir un campo de texto con formato, (si definimos que solo recibe números no permitirá letras...)
- **JPasswordField** – Campo de texto que oculta los caracteres ingresados.
- **JTextArea** – Permite vincular un área de texto donde el usuario ingresara información o simplemente para presentar cadenas de texto.
- **JEditorPane** –Permite vincular un área de texto con propiedades de formato.
- **JTextPane** – Similar al anterior, permitiendo otras opciones de formato, colores, iconos entre otros.

Componentes de Menus.

Estos componentes permiten vincular opciones de menú en nuestras ventanas, tipo menú principal, como por ejemplo el conocido Inicio, Archivo, Edición etc..

- **JMenuBar** – Permite vincular una barra de menús.
- **JMenu**– Permite vincular botones o enlaces que al ser pulsados despliegan un menú principal.
- **JMenuItem** – Botón u opción que se encuentra en un menú.
- **JCheckBoxMenuItem**– Elemento del menú como opciones de checkbox.
- **JRadioButtonMenuItem**– Elemento del menú como botón de selección.
- **JPopupMenu**– Opciones de menú emergentes.

- **JTable** – Permite vincular una tabla de datos con sus respectivas filas y columnas.
- **JTree** - Carga un árbol donde se establece cierta jerarquía visual, tipo directorio.
- **JList** – Permite cargar una lista de elementos, dependiendo de las propiedades puede tenerse una lista de selección múltiple.
- **JFileChooser** – Es un componente que permite la búsqueda y selección de ficheros entre otras.
- **JColorChooser** – Componente que permite cargar un panel selector de color
- **JOptionPane** – No es algo complejo sino mas un componente independiente que permite mostrar un cuadro de diálogo personalizable.

denle en el link, ahì están todas https://www.javatpoint.com/java-jbutton. Si pueden pasenlas aquí pero en ingles
EN ESPAÑOL http://codejavu.blogspot.mx/2013/09/componentes-java-swing.html

- JFrame
La clase javax.swing.JFrame es un tipo de contenedor que hereda la clase java.awt.Frame. JFrame funciona como la ventana principal donde se agregan componentes como etiquetas, botones, campos de texto para crear una GUI.

- JDialog
(Contenedor) El control JDialog representa una ventana de nivel superior con un borde y un título utilizado para tomar alguna forma de entrada del usuario. Hereda la clase Dialog.

A diferencia de JFrame, no tiene maximizar y minimizar botones.

- JOptionPane is used to provide standard dialog boxes such as message dialog box, confirm dialog box and input dialog box. These dialog boxes are used to display information or get input from the user.
JOptionPane hace que sea más fácil mostrar un cuadro de diálogo estándar que solicita a los usuarios un valor o les informa algo

- JPanel

- JTabbedPane

- JToolBar

- JLayeredPane
admite otros componentes de forma que se superponen unos sobre otros
- JButton

- JCheckbox

- JRadioButton
Una implementación de un botón de opción: un elemento que se puede seleccionar o deseleccionar, y que muestra su estado al usuario

- JComboBox
Un componente que combina un botón o campo editable y una lista desplegable. El usuario puede seleccionar un valor de la lista desplegable, que aparece a petición del usuario. Si hace que el cuadro combinado sea editable, el cuadro combinado incluye un campo editable en el que el usuario puede escribir un valor.

- JMenu
Una implementación de un menú: una ventana emergente que contiene JMenuItems que se muestra cuando el usuario selecciona un elemento en JMenuBar. Además de JMenuItems, un JMenu también puede contener JSeparators.

En esencia, un menú es un botón con un JPopupMenu asociado. Cuando se presiona el "botón", aparece el JPopupMenu. Si el "botón" está en JMenuBar, el menú es una ventana de nivel superior. Si el "botón" es otro elemento del menú, entonces el JPopupMenu es el menú "pull-right".

Los menús pueden ser configurados, y hasta cierto punto controlados, por Acciones. El uso de una acción con un menú tiene muchos beneficios más allá de la configuración directa de un menú.

- JSlider
Un componente que permite al usuario seleccionar gráficamente un valor deslizando una perilla dentro de un intervalo delimitado. La perilla siempre se posiciona en los puntos que coinciden con los valores enteros dentro del intervalo especificado.

El control deslizante puede mostrar las dos marcas de graduación principales y las marcas de graduación menores entre las más importantes. El número de valores entre las marcas se controla con setMajorTickSpacing y setMinorTickSpacing. La pintura de las marcas está controlada por setPaintTicks.

Los controles deslizantes también pueden imprimir etiquetas de texto a intervalos regulares (o en ubicaciones arbitrarias) a lo largo de la pista del control deslizante. La pintura de etiquetas está controlada por setLabelTable y setPaintLabels.

- JLabel

Un área de visualización para una cadena de texto corto o una imagen, o ambos. Una etiqueta no reacciona a los eventos de entrada. Como resultado, no puede enfocar el teclado. Sin embargo, una etiqueta puede mostrar una alternativa de teclado como una conveniencia para un componente cercano que tiene una alternativa de teclado pero no puede mostrarla.

Un objeto JLabel puede mostrar texto, una imagen o ambos. Puede especificar en qué parte del área de visualización de la etiqueta se alinean los contenidos de la etiqueta estableciendo la alineación vertical y horizontal. Por defecto, las etiquetas se centran verticalmente en su área de visualización. Las etiquetas de solo texto están alineadas al borde de ataque, por defecto; las etiquetas de solo imagen están centradas horizontalmente, por defecto.

También puede especificar la posición del texto en relación con la imagen. De forma predeterminada, el texto está en el borde posterior de la imagen, con el texto y la imagen alineados verticalmente.

El borde inicial y posterior de una etiqueta se determinan a partir del valor de su propiedad ComponentOrientation. Actualmente, la configuración predeterminada de ComponentOrientation asigna el borde anterior a la izquierda y el borde posterior a la derecha.

Finalmente, puede usar el método setIconTextGap para especificar cuántos píxeles deben aparecer entre el texto y la imagen. El valor predeterminado es 4 píxeles.

- JToolTip

es una herramienta de ayuda visual, que funciona al situar el cursor sobre algún elemento gráfico, mostrando una ayuda adicional para informar al usuario de la finalidad del elemento sobre el que se encuentra.

- JProgressBar

Un componente que muestra visualmente el progreso de alguna tarea. A medida que la tarea avanza hacia la finalización, la barra de progreso muestra el porcentaje de finalización de la tarea. Este porcentaje típicamente se representa visualmente mediante un rectángulo

que comienza vacío y gradualmente se rellena a medida que la tarea avanza. Además, la barra de progreso puede mostrar una representación textual de este porcentaje.

JProgressBar usa un BoundedRangeModel como su modelo de datos, con la propiedad value representando el estado "actual" de la tarea, y las propiedades mínima y máxima que representan los puntos inicial y final, respectivamente.

Para indicar que se está ejecutando una tarea de longitud desconocida, puede poner una barra de progreso en modo indeterminado. Mientras la barra está en modo indeterminado, se anima constantemente para mostrar que el trabajo está ocurriendo. Tan pronto como pueda determinar la duración y la cantidad de progreso de la tarea, debe actualizar el valor de la barra de progreso y volver a cambiar al modo determinado

- JColorChooser
nos ayuda a seleccionar un color de una forma mas atractiva para el usuario.
- JFileChooser
nos permite mostrar fácilmente una ventana para la selección de un fichero.
18. Realizar un programa que, utilizando flujos, pueda realizar la misma acción que la clase Scanner, es decir, que pueda introducir datos por medio de teclado hacia su programa.

usen la clase console ()

19. Terminar su programa de copia de archivos

esta en corto, si alguien no lo tiene pidalo por el wa