Conociendo [En lab03.doc y bodyTic.asta ]

1. En el directorio descarguen los archivos contenidos en bodyTic.zip y bodyTic.asta. Revisen el código del programa a) ¿Cuántos paquetes tiene? b) ¿Cuántas clases tiene en total? c) ¿Cuál es la clase ejecutiva? ¿Por qué?
2. ¿Cuántos paquetes tiene? Dos paquetes aplicacion y presentacion
3. ¿Cuántas clases tiene en total? Tiene 6 paquetes(ensalon,salon,persona,deportista,bola,bodyticgui).
4. )¿Cuál es la clase ejecutiva? BodyTicGUI; porque es la que ejecuta la interfaz.
5. Ejecuten el programa.

¿Qué funcionalidades ofrece?Ofrece entren,inicien,paren,decidan,salgan

¿Qué hace actualmente? Solo muestra una interfaz en blanco

¿Por qué? Falta implementar y completar las funcionalidades.

***Arquitectura general.***

1. Consulten el significado de las palabras package e import de java. Explique su uso con un ejemplo de este programa.

2. ¿Qué es un paquete? ¿Para qué sirve? Inicien el diseño con un diagrama de paquetes en el que se presente los componentes y las relaciones entre ellos.

3. Revisen el contenido del directorio de trabajo y sus subdirectorios. Describa su contenido. ¿Qué coincidencia hay entre paquetes y directorios?

1)Package en java es para identificar las clases que estan contenidas dentro de esta, para usarlo se utiliza la palabra reservada package NombrePaquete arriba de todo el codigo, esto me indica que la(s) clases estan dentro de este paquete, por ejemplo en la clase BodyTicGUI esta package presentacion, es decir que esta clase se encuentra dentro de el paquete (o carpeta) presentacion.

Import en java es para importar las clases publicas de un paquete complete, por ejemplo para poder acceder a aplicacion desde presentacion utilizacmos import aplicacion.\*

2)Los paquetes son una forma de organizar a un grupo de clases, sirven para tener mas organizado un conjunto de clases y poder verlos de una manera mas simplificada y simple.

En nuestro codigo tenemos dos paquetes, presentacion y aplicacion, en presentacion se encuentra la(s) clases que permiten lo visual (JPanel, JFrame) y aplicacion es aquel que contiene las clases que permiten la interaccion entre objetos que estan en el JFrame.

3)Dentro de la carpeta BodyTic Podemos encontrar las carpetas (o paquetes) presentacion y aplicacion, como presentacion y aplicacion son paquetes entonces se crea una carpeta con sus respectivos nombres donde contendran las clases correspondientes.

**Ciclo 1. Iniciando con los deportistas normales [En lab03.doc y \*.java]** (NO OLVIDE BDD - MDD)

1. Estudie la clase Salon ¿Qué tipo de colección usa para albergar los elementos? ¿Puede recibir deportistas? ¿Por qué?

Tiene una colección llamada elementos. Si puede recibir deportistas porque tiene una ArrayList<EnSalon> elementos, que es de la interfaz EnSalon con una Array FORMAS que recibe personas; donde él deportista es una persona.

1. Estudie el código de la clase Deportista, ¿qué otros componentes software la definen? ¿cómo?

La definen con la implementación que él deportista esta en un salón al utilizar implements.

1. Por ser una Persona ¿qué saben hacer? ¿qué no puede hacer distinto? ¿qué deben aprender a hacer? Justifique su respuesta.

Puede mover él brazo(muevaBrazo),pierna(muevaPierna), moverse(muévase). No puede hacer distinto él muévase ya que cualquiera de la clase persona se mueve igual. Debe aprender como hacer sus movimientos ya que dependiendo de unos char I,S,B,D O P mueve sus extremidades de diferente fomas.

1. Por comportarse como un EnSalon, ¿qué sabe hacer? ¿qué no puede hacer distinto? ¿qué debe aprender a hacer? Justifique su respuesta.

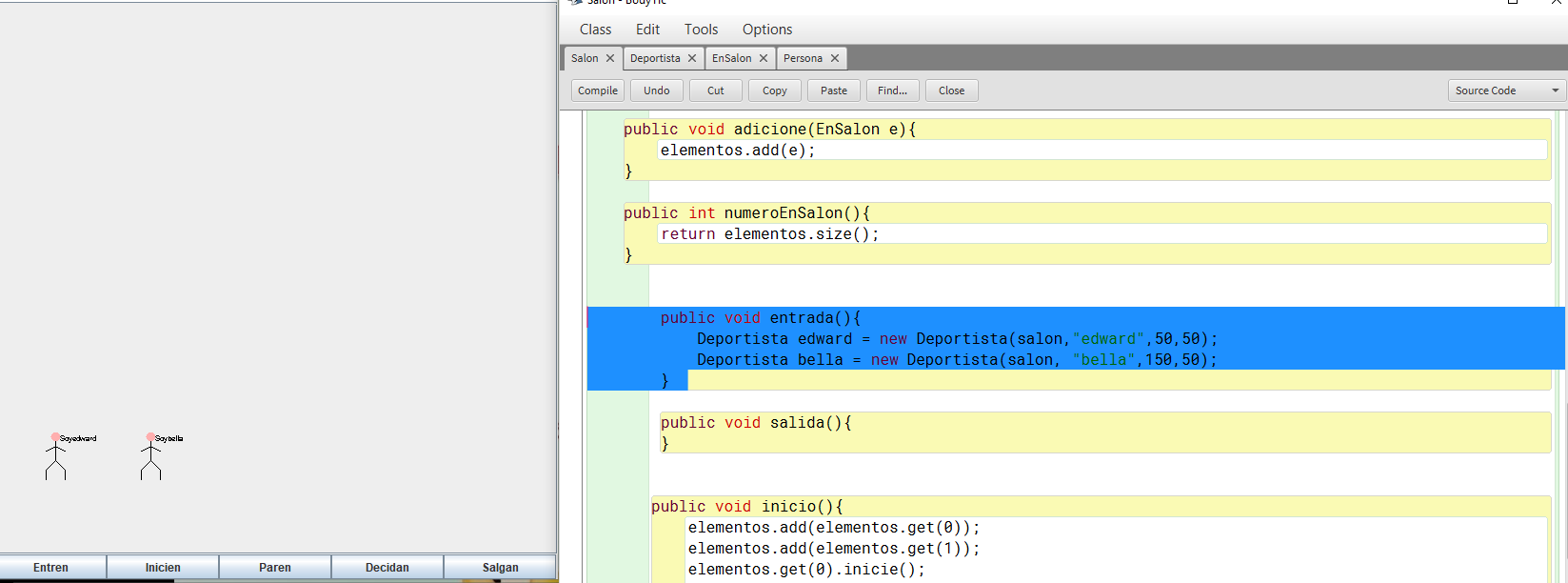
No sabe hacer nada por los default. Nada. Nda.

1. Considerando lo anterior, un Deportista ¿de qué color es? ¿cómo se mueve? ¿cómo decide? ¿cómo inicia ? Justifiquen sus respuestas.

Color.BLACK. Se mueve dependidendo de la dirección que se le indique con su posición en X y Y.Decide si la dirección es hacia: N,E,S Y O.

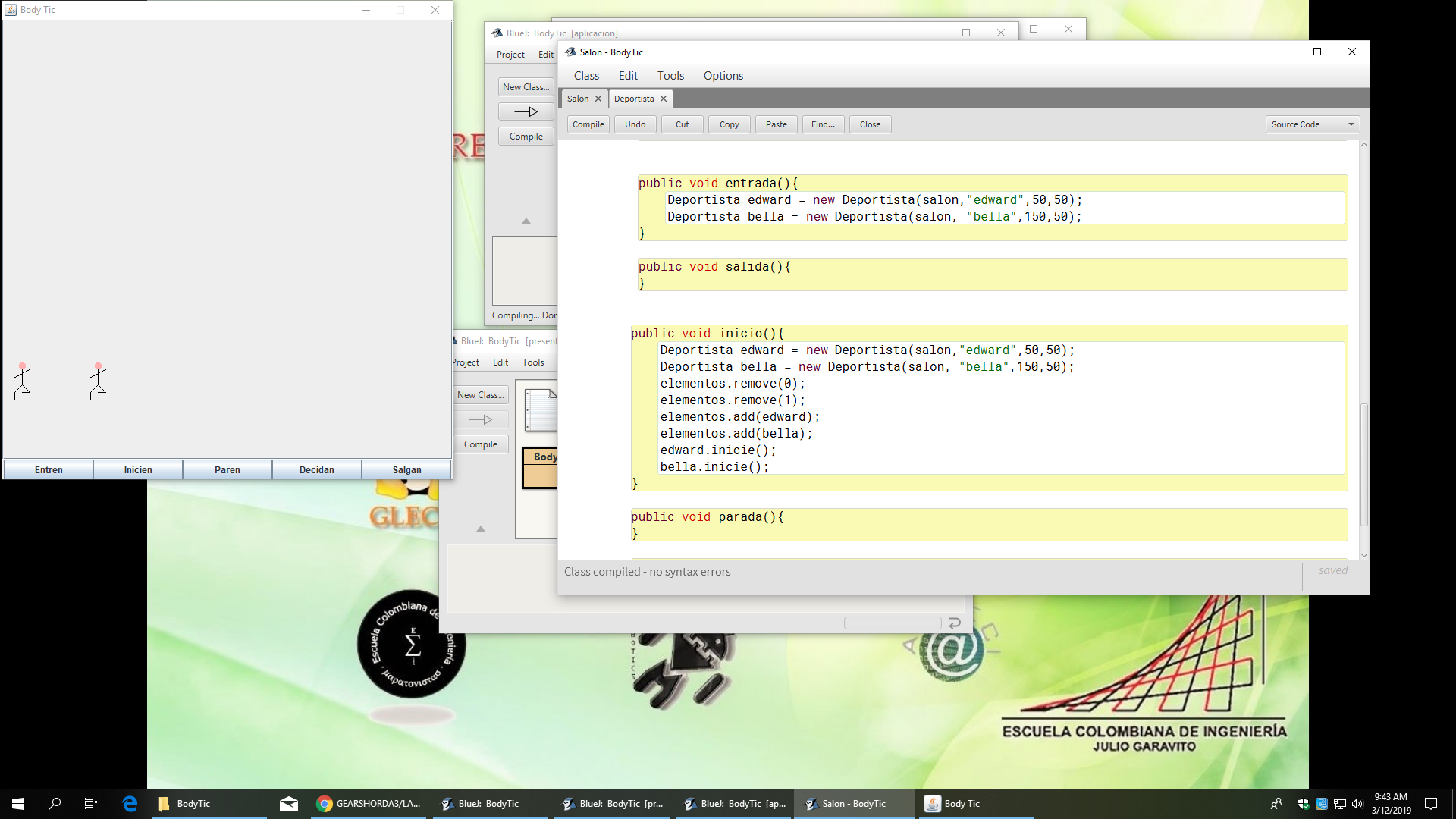
1. Construyan el método que atiende el botón Entren: el método llamado entrada() de la clase Salon. En este método creen dos deportistas edward y bella. Ejecuten el programa, ¿Cómo quedan os deportistas? Capturen una pantalla significativa.

Quedan quietos con una cabeza y extremidades junto con sus respectivos nombres.



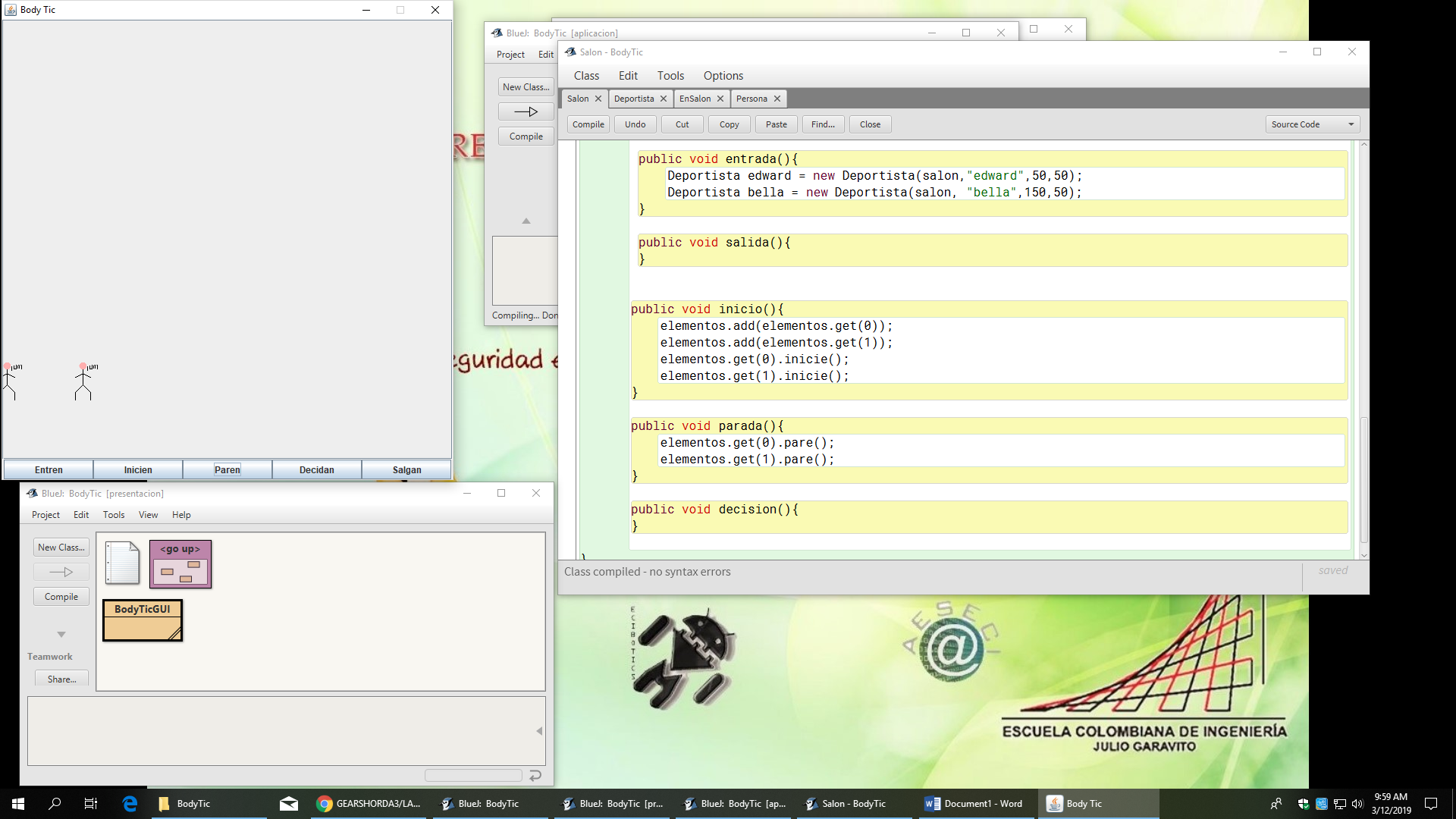
1. Construyan el método que atiende el botón Inicien: el método llamado inicio() de la clase Salon. Ejecuten el programa Ejecute el programa y pídales que entren e inicien. ¿Cómo se comportan los deportistas? Capturen una pantalla significativa.

Se mueven a la izquierda y luego a la derecho

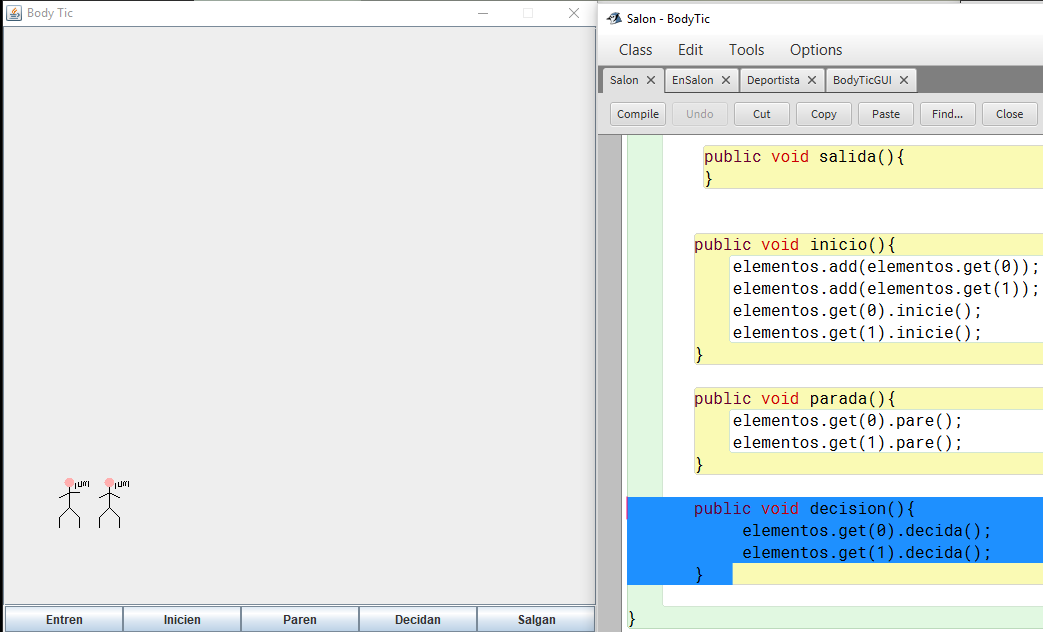


8.Construyan el método que atiende el botón Paren: el método llamado parada() de la clase Salon. Ejecute el programa y pídales que entren, inicien y paren ¿Cómo se comportan los deportistas? Capturen una pantalla significativa.

Entra, Se mueven a la izquierda (quedan en una posicion de movimiento) y luego paran (quedando en la posicion igual a entran)



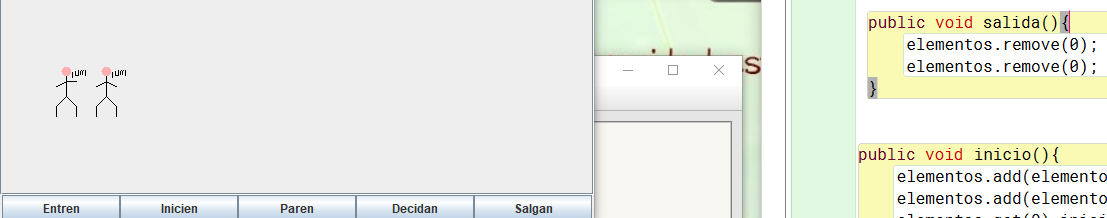
9. Construyan el método que atiende el botón Decidan:el método llamado decision() de la clase Salon. Ejecute el programa y pídales que entren, inicien, decidan y paren ¿Cómo se comportan los deportistas? Capturen una pantalla significativa.

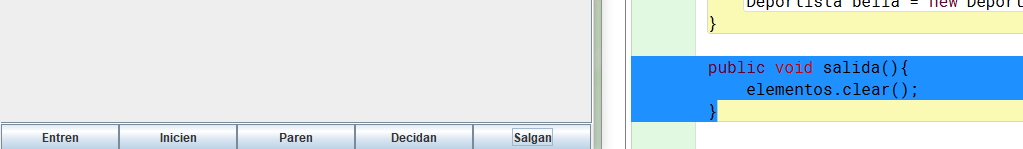


Al desarrollar decidan a los dos deportistas, uno se mueve y el otro se queda quieto ya que se trabaja un aleatorio, después de hacer el paren() como uno tiene un movimiento mas que otro, uno queda con un brazo hacia abajo y el otro si totalmente quieto.

10. Construyan el método que atiende el botón Salgan: el método llamado salida() de la clase Salón. Ejecute el programa y pídales que entren, inicien, decidan, paren y salgan ¿Cómo se comportan los deportistas? Capturen una pantalla significativa.

Como los deportistas son creados al mismo tiempo, después de hacer todas sus otras funciones, cuando se selecciona que salgan los dos desaparecen automáticamente.





Ciclo 2.

Incluyendo a los deportistas avanzados [En lab03.doc y automataasta] (NO OLVIDE BDD - MDD) El objetivo de este punto es recibir en el salón deportistas avanzados. Los deportistas avanzados, vestidos de naranja, se mueven más, sólo paran después de dos órdenes y, cuando les piden que deciden, siempre hacen ejercicio.

1. ¿Cuáles son las adiciones necesarias en el diseño? ¿y los cambios? ¡Hágalos! ¿cuáles métodos se sobre-escriben (overriding)? Ahora escriba el código correspondiente al deportista avanzado.

package aplicacion;

/\*\*

\* Write a description of class SuperDeportista here.

\* \* @author (your name)

\* @version (a version number or a date)

\*/

public class SuperDeportista extends Deportista implements EnSalon

{

private Salon salon;

protected String palabras;

protected int paso;

/\*\*

\* Constructor for objects of class SuperDeportista

\*/

public SuperDeportista(Salon salon,String nombre,int posicionx, int posiciony)

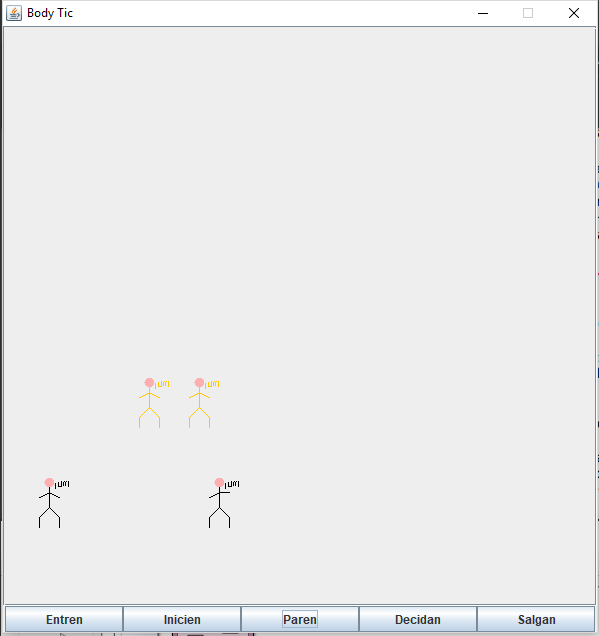
{

super(salon,nombre,posicionx,posiciony, 1);

}

}

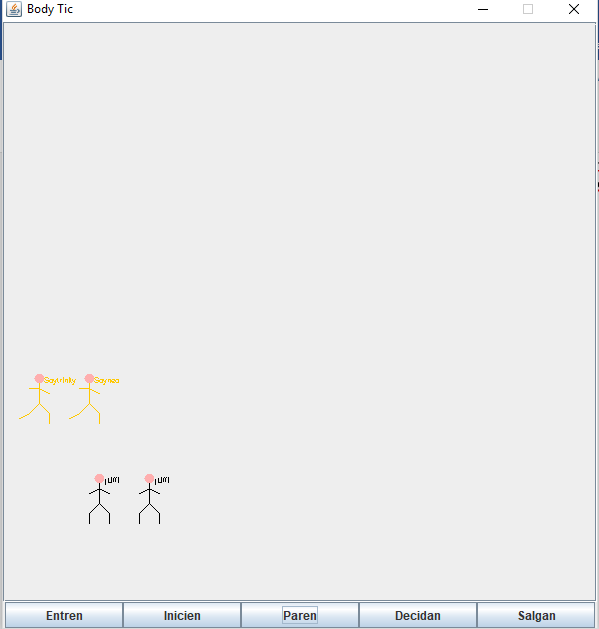
2. Adicionen una pareja de deportistas avanzados en la fila 3, llámenlas neo y trinity, Ejecute el programa y pídales que entren, inicien, decidan y paren. Capturen pantallas significativas. ¿Qué pasa?



3. Ahora, los avanzados quieren sorprender con su resistencia; es decir, sólo son paran muy pocas veces. ¿Qué modificaría para lograr este comportamiento? ¡Hágalo!

En la clase salon en el metodo parada, se coloco una variable conta, que cuenta las iteraciones que se necesitan para parar.

4. Nuevamente ejecute el programa y pídales a todos que entren, inicien, decidan y paren. Capturen una pantalla significativa. ¿Qué pasa?



Los SuperDeportistas no se cansan tan rapido, para que descanses necesitan otra iteracion de paren, minetras que los normales se cansan mas rapido.