

EJERCICIO GUIADO. JAVA: POO. USAR CLASES PROPIAS EN UN PROYECTO

Aplicación de una Clase Propia

Una vez programada una Clase de creación propia, podremos usar esta en uno de nuestros proyectos para crear objetos de dicha clase. En este ejercicio guiado veremos un ejemplo sobre cómo usar de forma práctica la clase *SalaCine* en un proyecto java hecho con NetBeans.

EJERCICIO GUIADO

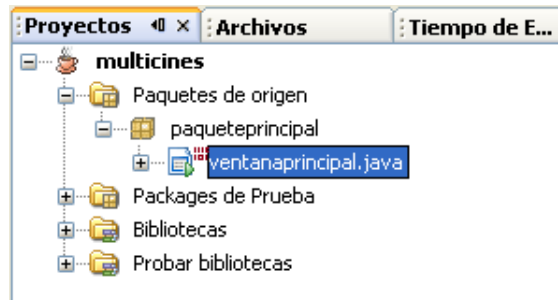
El Multicines Avenida nos encarga un programa para gestionar el funcionamiento de las dos salas con las que cuenta este pequeño multicines.

La primera de las salas es la "sala central". En ella se proyectan los estrenos. Es una sala grande, con 500 localidades. El precio para ver una película en esta sala es de 8.00 euros.

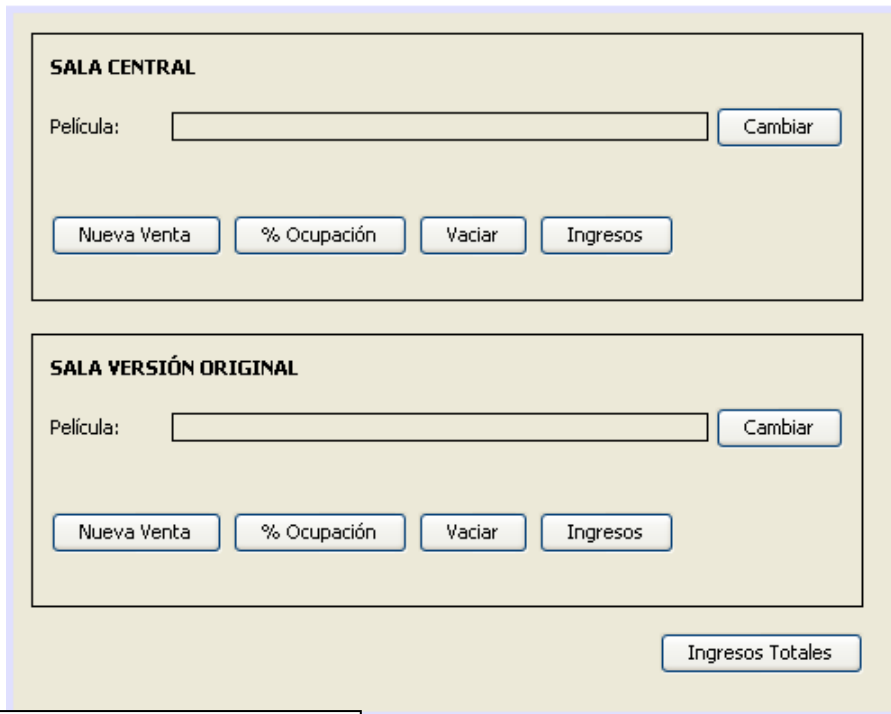
La segunda de las salas es la "sala de versiones originales". En esta sala se proyectan películas clásicas en versión original. Es una sala con un aforo más pequeño, concretamente, de 80 localidades. El precio para ver una película en esta sala es de 6.00 euros.

Se creará un programa que nos permita controlar la entrada de personas en las salas y podremos saber en todo momento el porcentaje de ocupación de cada sala y los ingresos obtenidos.

1. Crear un nuevo proyecto. Este proyecto se llamará *multicines*, contendrá un paquete llamado *paquetepincipal* y este a su vez contendrá un JFrame llamado *ventanaprincipal*. El aspecto de la estructura del proyecto al crearlo será la siguiente:

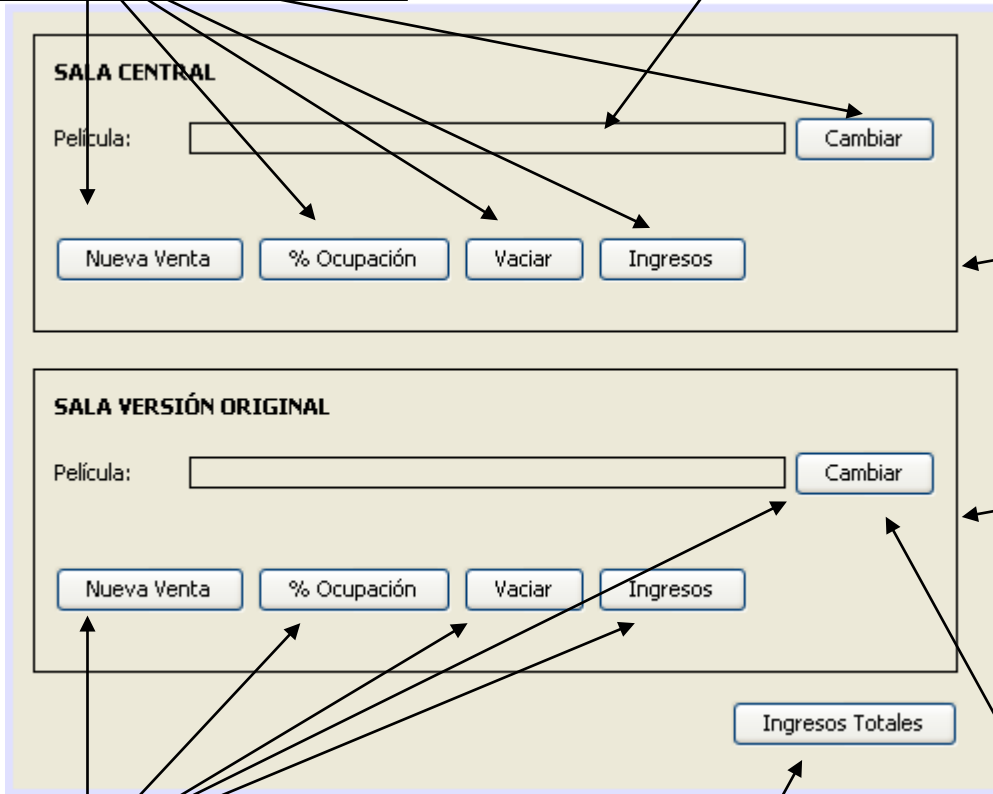


2. Diseñar la ventana principal de forma que quede parecida a la siguiente. Usa la ventana de diseño para hacer el diseño del formulario. Los nombres de los componentes se indica a continuación.



Botones:
 btnCentralNueva
 btnCentralOcupacion
 btnCentralVaciar
 btnCentralIngresos
 btnCentralCambiar

Etiqueta:
 etiCentralPelícula



Panel:
 panelCentral

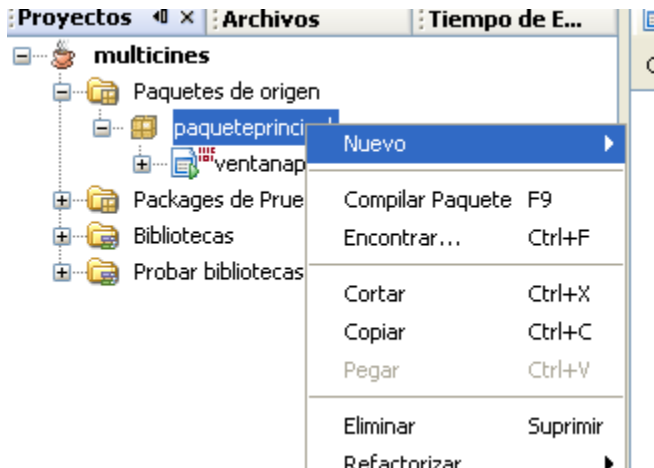
Panel:
 panelVOriginal

Botones:
 btnOriginalNueva
 btnOriginalOcupacion
 btnOriginalVaciar
 btnOriginalIngresos
 btnOriginalCambiar

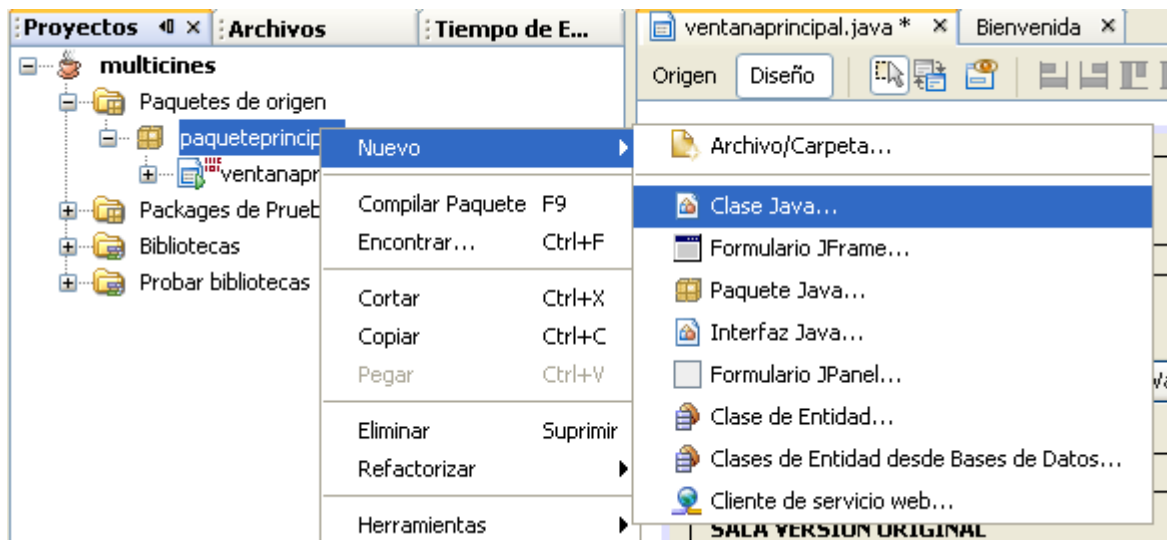
Botón
 btnIngresosTotales

Etiqueta:
 etiOriginalPelícula

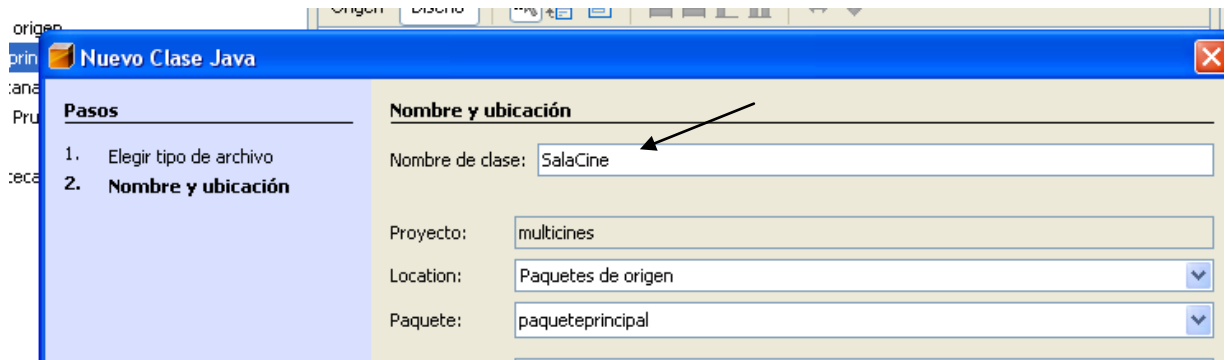
3. Antes de empezar la programación de cada botón del programa, incluiremos en el proyecto la clase *SalaCine*, para así poder generar objetos del tipo “sala de cine” y aprovecharlos en nuestro programa.
4. Para añadir un fichero de clase propio en un proyecto Java con NetBeans, debes hacer clic con el botón derecho sobre el paquete donde quieras introducir la clase. En nuestro caso, haremos clic con el botón derecho sobre el *paqueteprincipal*, que es el único que tenemos:



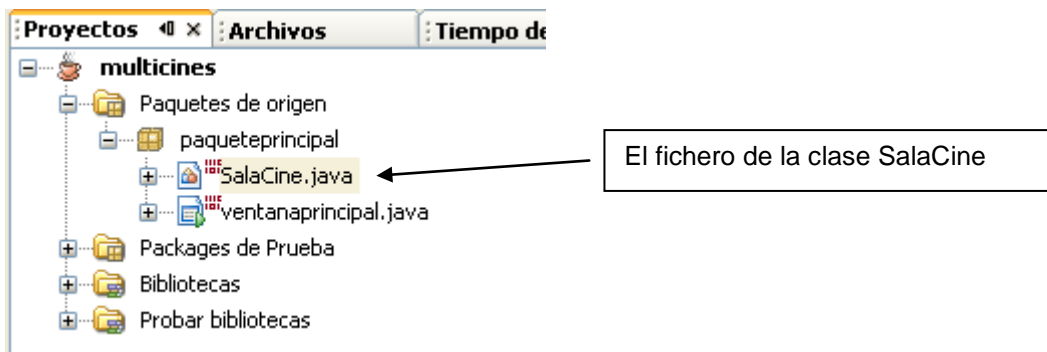
5. Elegiremos la opción Nuevo / Clase Java...



6. Entonces tendremos que darle un nombre a nuestro fichero de clase. Recuerda que el nombre debe ser el mismo que el de la clase que queremos programar. En nuestro caso, *SalaCine*. Luego se pulsa el botón *Terminar*.



7. Si vuelves a observar el esquema del proyecto, verás que dentro del paquete principal ha aparecido un nuevo elemento, llamado *SalaCine*. Este es el fichero donde programaremos nuestra clase *SalaCine*.



8. Si haces doble clic sobre el fichero *SalaCine.java*, podrás ver en el centro de la ventana de NetBeans el código de dicha clase. NetBeans añade cierto código a la clase para ahorrarnos trabajo, por eso ya tendremos programado la instrucción inicial de la clase junto con el comienzo del *constructor* y algunos comentarios que se añaden automáticamente:

```
/*
 * SalaCine.java
 *
 * Created on 2 de agosto de 2007, 12:14
 *
 * To change this template, choose Tools | Template Manager
 * and open the template in the editor.
 */

package paqueteprincipal;

/**
 *
 * @author didact
 */
public class SalaCine {

    /** Creates a new instance of SalaCine */
    public SalaCine() {

    }

}
```

Esta instrucción, generada por NetBeans, indica que la clase *SalaCine* está programada dentro del *paqueteprincipal*

Inicio de la clase

Preparación del constructor

9. Ahora ya solo tendremos que completar la programación de la clase, añadiendo las propiedades, completando el constructor, y añadiendo los distintos métodos que programamos en la hoja anterior. Debes modificar el fichero *SalaCine* para que quede así:

```

/*
 * SalaCine.java
 *
 * Created on 2 de agosto de 2007, 12:14
 *
 * To change this template, choose Tools | Template Manager
 * and open the template in the editor.
 */

package paqueteprincipal;

/**
 *
 * @author didact
 */
public class SalaCine {

    //propiedades
    int Aforo;
    int Ocupadas;
    String Pelicula;
    double Entrada;

    /** Creates a new instance of SalaCine */
    //Constructor
    public SalaCine() {
        Aforo = 100;
        Ocupadas = 0;
        Pelicula = "";
        Entrada = 5.0;
    }

    //Métodos set

    //Método setAforo
    public void setAforo(int afo) {
        Aforo = afo;
    }

    //Método setOcupadas
    public void setOcupadas(int ocu) {
        Ocupadas = ocu;
    }

    //Método setPelicula
    public void setPelicula(String peli) {
        Pelicula = peli;
    }

    //Método setEntrada
    public void setEntrada(double entra) {
        Entrada = entra;
    }

    //Método setLibres
    public void setLibres(int lib) {
        int ocu;
        ocu = Aforo - lib;
        Ocupadas = ocu;
    }
}

```

```

//Métodos get

//Método getAforo
public int getAforo() {
    return Aforo;
}

//Método getOcupadas
public int getOcupadas() {
    return Ocupadas;
}

//Método getPelicula
public String getPelicula() {
    return Pelicula;
}

//Método getEntrada
public double getEntrada() {
    return Entrada;
}

//Método getLibres
public int getLibres() {
    int lib;
    lib = Aforo - Ocupadas;
    return lib;
}

//Método getPorcentaje
public double getPorcentaje() {
    double por;
    por = (double) Ocupadas / (double) Aforo * 100.0;
    return por;
}

//Método getIngresos
public double getIngresos() {
    double ingre;
    ingre = Ocupadas * Entrada;
    return ingre;
}

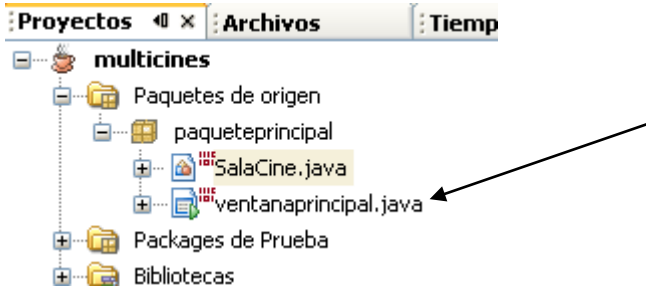
//Métodos de orden

//Método Vaciar
public void Vaciar() {
    Ocupadas = 0;
    Pelicula = "";
}

//Método entraUno
public void entraUno() {
    Ocupadas++;
}
}

```

- Ya hemos incluido la clase *SalaCine* en nuestro proyecto. A partir de ahora podremos crear objetos del tipo *SalaCine* y aprovechar sus características en nuestro programa.
- Empezaremos a programar la ventana principal de nuestro programa. Haz doble clic sobre el fichero *ventanaprincipal.java* en la ventana de proyectos para pasar al diseño de la ventana principal:



- Nuestro programa principal, gestiona el funcionamiento de dos salas de cine: la sala central y la sala de versiones originales. Para facilitar nuestro trabajo, sería interesante crear dos objetos globales del tipo *SalaCine*. Uno de ellos representará la sala central, y el otro la sala de versiones originales.
- Para ello, acude a la ventana de código de la ventana principal:



- En la zona de variables globales de la ventana principal declaramos dos objetos del tipo *SalaCine*:

```
public class ventanaprincipal extends javax.swing.JFrame {

    SalaCine central; //sala central
    SalaCine vo;      //sala de versiones originales

    /** Creates new form ventanaprincipal */
    public ventanaprincipal() {
        initComponents();
    }
}
```

Creamos dos objetos "sala de cine"

- Es en el constructor de nuestra ventana principal donde construiremos nuestros objetos de tipo sala. (Recuerda que el constructor de tu ventana principal se suele usar para construir los objetos que vas a usar en tu programa y asignarle las características que tendrán dichos objetos al inicio)

```

public class ventanaprincipal extends javax.swing.JFrame {

    SalaCine central; //sala central
    SalaCine vo;      //sala de versiones originales

    /** Creates new form ventanaprincipal */
    public ventanaprincipal() {
        initComponents();

        central = new SalaCine();
        vo = new SalaCine();
    }
}

```

Construimos los objetos.

16. Recuerda que cuando se construyen objetos, estos tienen por defecto unos valores iniciales. En el caso de los objetos tipo *SalaCine*, sus valores son los siguientes:

Aforo 100, Ocupación 0, Película (cadena vacía) y Entrada 5.0

Será necesario modificar las características de ambas salas ya que no tienen ahora mismo los valores correctos.

Se dijo al comenzar el ejercicio que la sala central tenía las siguientes características:

Aforo: 500 y un precio de entrada de 8 euros.

Por lo tanto, modificamos las características del objeto en el constructor de la siguiente forma:

```

/** Creates new form ventanaprincipal */
public ventanaprincipal() {
    initComponents();

    central = new SalaCine();
    vo = new SalaCine();

    central.setAforo(500);
    central.setEntrada(8.0);
}

```

Modificamos las características del objeto *central* (sala central)

17. En el caso de la sala de versión original sucede lo mismo. Sus características son las siguientes:

Aforo: 80. Precio entrada: 6.0 euros.

Por tanto será necesario modificar el objeto *vo* para asignar estos valores:

```

/** Creates new form ventanaprincipal */
public ventanaprincipal() {
    initComponents();

    central = new SalaCine();
    vo = new SalaCine();

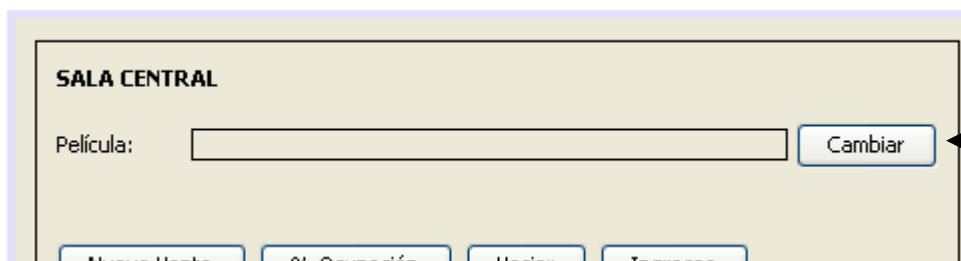
    central.setAforo(500);
    central.setEntrada(8.0);

    vo.setAforo(80);
    vo.setEntrada(6.0);
}

```

Modificamos las características del objeto vo (sala versión original)

18. Ya hemos incluido dos objetos del tipo *SalaCine* en nuestro proyecto. Los hemos declarado en la ventana principal como globales y los hemos construido dándoles las características apropiadas. A partir de ahora, usaremos esos objetos en la programación de cada botón de la ventana.
19. Empezaremos programando el botón *btnCentralCambiar*. El objetivo de este botón es asignar a la sala central un título de película. Ese título se debe mostrar también en la etiqueta *etiCentralPelicula*:



Cambia el título de la película de la sala central.

20. Entra en el *actionPerformed* del botón *btnCentralCambiar* y programa lo siguiente:

```

private void btnCentralCambiarActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO: Agregue su código aquí:

    String titulo;

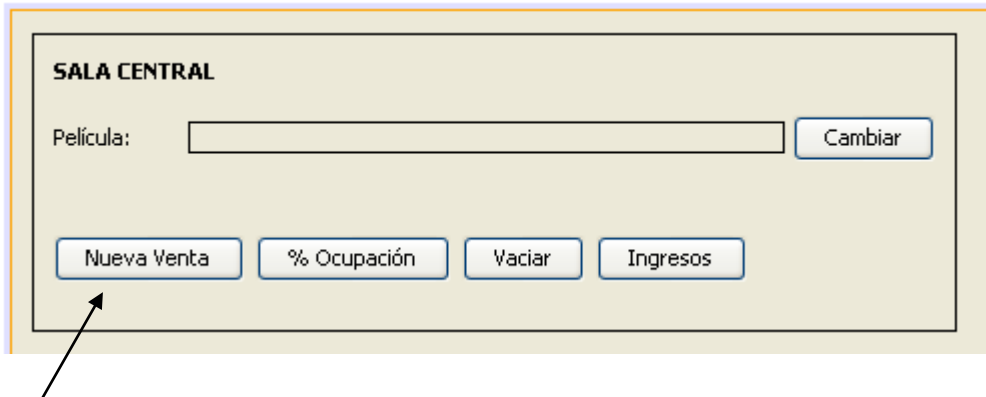
    titulo = JOptionPane.showInputDialog(null, "Introduce el título de la película: ");
    central.setPelicula(titulo);
    etiCentralPelicula.setText(titulo);
}

```

El código de este botón hace lo siguiente:

- Le pide al usuario que introduzca un título para la película, usando un `JOptionPane.showInputDialog`. El título introducido se recoge en una variable de cadena llamada *titulo*.
- Luego se asigna dicho título a la sala central del multicines. Esta línea es interesante, ya que se usa el método `setPelicula` del objeto *central* para asignar el título de la película.
- Y finalmente se coloca el título en la etiqueta para que el usuario pueda verlo.

21. Ejecuta el programa y observa el funcionamiento del botón.
22. Ahora se programará el botón *btnCentralNueva*:



Cada vez que un espectador compre una entrada para la sala central, el usuario del programa deberá pulsar este botón. Internamente, el programa debe decirle al objeto *central* que ha entrado una nueva persona en la sala. Esto se hace con el método *entraUno*.

Así pues, programe el *actionPerformed* del botón *btnCentralNueva* de la siguiente forma:

```
private void btnCentralNuevaActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO: Agregue su código aquí:  
    central.entraUno();  
}
```

Como puedes ver, simplemente se le informa al objeto *central* (sala central) que ha entrado una nueva persona usando el método *entraUno*.

23. Programaremos ahora el botón *btnCentralOcupacion*. Este botón nos mostrará información sobre la sala central. Concretamente, este botón debe mostrar lo siguiente:

- Película que se está proyectando.
- Aforo total de la sala.
- Butacas Ocupadas.
- Butacas Libres.
- Porcentaje de ocupación.

Para mostrar toda esta información, será necesario pedirle al objeto *central* (sala central) que nos de información de la sala. Esto lo conseguiremos usando los distintos métodos *get* del objeto.

Estudia atentamente la programación del botón *btnCentralOcupacion*:

```

private void btnCentralOcupacionActionPerformed(java.awt.event.ActionEvent evt) {
// TODO: Agregue su código aquí:
String info; //una variable de texto que contendrá el mensaje a mostrar

info = "Sala Central\n";
info = info + "Película: "+central.getPelicula()+"\n";
info = info + "Aforo: "+central.getAforo()+"\n";
info = info + "Ocupadas: "+central.getOcupadas()+"\n";
info = info + "Libres: "+central.getLibres()+"\n";
info = info + "Porcentaje Ocupacion: "+central.getPorcentaje()+"\n";

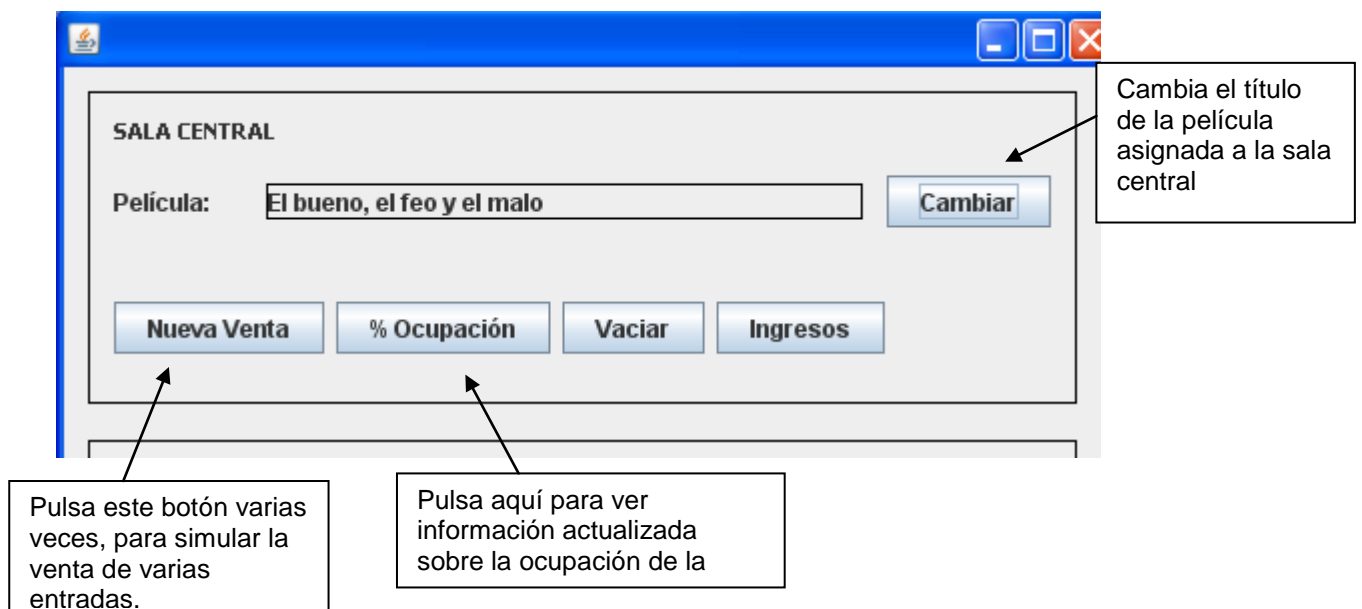
JOptionPane.showMessageDialog(null,info);
}

```

Observa el código:

- Primero se crea una variable de cadena llamada *info* (información)
- A continuación, se van concatenando textos a la variable de cadena. Recuerda que los \n permiten añadir saltos de línea.
- Observa como se va pidiendo información al objeto a través de métodos get y se va concatenando esta información a la variable info.
- Finalmente se presenta el texto de la variable info en un JOptionPane.

24. Ejecuta el programa para observar el funcionamiento de los tres botones que se han programado. Asigna una película a la sala central, luego pulsa varias veces el botón *Nueva Venta* y pulsa el botón *%Ocupación* para ver información sobre la sala:



25. A continuación se programará el botón *Vaciar* de correspondiente a la sala central (*btnCentralVaciar*) Este botón debe situar la ocupación de la sala a cero y quitar la película asignada a la sala central.

Programa en el *actionPerformed* del botón lo siguiente:

```
private void btnCentralVaciarActionPerformed(java.awt.event.ActionEvent evt) {  
// TODO: Agregue su código aquí:  
    central.Vaciar();  
    etiCentralPelicula.setText("");  
}
```

Como puedes observar, el código es muy simple. Solo tengo que ordenar al objeto *central* (sala central) que se vacíe. Debido a que ya no hay película asignada a la sala central, es conveniente también borrar el contenido de la etiqueta que muestra el título de la película.

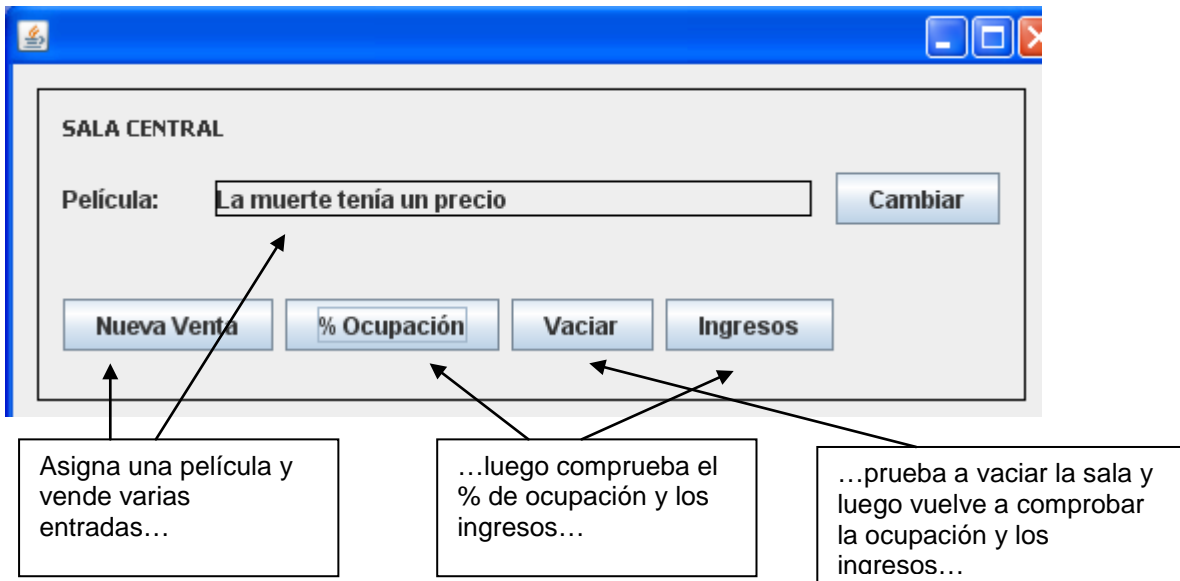
26. Programaremos ahora el botón *Ingresos* del panel de la sala central. Este botón nos debe dar información sobre los ingresos producidos por la sala en el momento en que se pulsa. Esta información la mostraremos en un *JOptionPane*. Programa el *actionPerformed* de este botón como sigue:

```
private void btnCentralIngresosActionPerformed(java.awt.event.ActionEvent evt) {  
// TODO: Agregue su código aquí:  
    String info;  
  
    info = "Ingresos de la sala: "+central.getIngresos()+" euros";  
    JOptionPane.showMessageDialog(null, info);  
}
```

Como puedes ver, la programación es bastante sencilla. Solo hay que pedirle al objeto *central* (sala central) que nos diga los ingresos correspondientes.

Observa la ventaja de usar objetos. No tenemos que hacer ningún cálculo aquí. Solo tenemos que pedirle al objeto que haga el cálculo necesario por nosotros. Dicho de otra forma, hacemos que ellos hagan el trabajo por nosotros.

27. Ejecuta el programa para comprobar el funcionamiento de estos botones. Asigna una película a la sala central, vende algunas entradas, comprueba el porcentaje de ocupación y luego comprueba los ingresos. Vacía la sala central, y vuelve a comprobar el porcentaje de ocupación y los ingresos actuales.



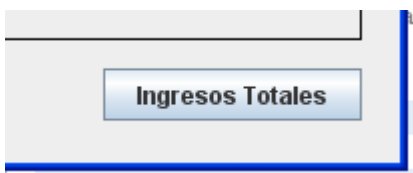
EJERCICIO

El segundo de los paneles hace referencia a la sala de versiones originales. Su funcionamiento es el mismo que el de la sala central. Se pide que programe todos los botones correspondientes a la sala de versiones originales.

La programación es prácticamente igual, solo hay que actuar sobre el objeto *vo* en vez del objeto *central*.

EJERCICIO

El botón *IngresosTotales* de la ventana, debería mostrar en un *JOptionPane* la suma de los ingresos actuales de ambas salas. Programe dicho botón. Solo tiene que pedirle a cada objeto: *central* y *vo* que le de sus ingresos, y sumar ambos.



CONCLUSIÓN

Se pueden añadir a un proyecto Java clases programadas por uno mismo.

Al hacer esto, en el mismo proyecto podemos crear objetos de dichas clases y usarlos para hacer el programa.

En general, un proyecto java está constituido por un grupo de clases, algunas de las cuales serán de creación propia. En el programa usaremos objetos derivados de dichas clases.