

Elementos de Selección y Objetos

Hasta el momento se han utilizado diversos elementos de selección (*listas, checkboxes y radio buttons*) en donde al elegir uno se selecciona como valor la etiqueta de la opción. Esto no es conveniente ya que es deseable que aparezca un nombre en la opción, pero al seleccionarlo, se haga a partir de una clave.

Se comenzará trabajando con el proyecto SW06 el cuál ya tiene la interfaz y las clases necesarias para comenzar a implementar las nuevas funcionalidades.

El funcionamiento es el siguiente:

- Todos los elementos estarán ocultos (excepto la lista de licenciaturas)
- Al seleccionar una licenciatura, se cargarán los alumnos de esa licenciatura
- Al seleccionar un alumno se mostrarán dos Radio Buttons para seleccionar si merece o no un apoyo y el botón para registrarlo
- En caso de seleccionar que SI lo merece, se habilitarán los Check Boxes necesarios para seleccionar el apoyo correspondiente
- Al presionar el botón de Registrar se mostrará en pantalla:
- Clave de la Licenciatura, Matrícula del alumno, Si merece o no apoyo y en su caso, clave(s) del apoyo(s)

Llenando los Elementos de Selección con Objetos

Primero se llenarán las listas de selección (*Combo boxes*), ésta vez con objetos que no sean de tipo String. Primero se vaciarán ambas listas y posteriormente se llenará la lista de licenciaturas con objetos de la clase Licenciatura.

FrmSeleccionesObjetos.java

```
public FrmSeleccionesObjetos() {  
    initComponents();  
    agruparSelecciones();  
    agruparChkBoxes();  
    cmbListaAlumnos.removeAllItems();  
    cmbListaLicenciaturas.removeAllItems();  
}
```

Una vez limpia la lista, primero hay que indicar que estará formada por objetos tipo Licenciatura y no objetos String como es su generación inicial.

Para esto, hay que seleccionarla en el modo **Design** y en el panel de propiedades, seleccionar la pestaña **Code**, posteriormente buscar la propiedad **TypeParameters**, dar clic sobre los tres puntos (...) y en lugar de **<String>**, escribir **<Licenciatura>** (que es el nombre de la clase a partir de la cuál se crearán objetos para llenar la lista de selección) y dar clic en **OK**.

Si se observa el código generado automáticamente, se verá que el parámetro ha cambiado. Posiblemente se deba importar el paquete y la clase.

```
private javax.swing.JComboBox<Licenciatura> cmbListaLicenciaturas;
```

Será necesario cambiar también el modelo (*model*) dejando limpio el campo de valores en la pestaña *Properties*.

Se cuenta con una clase con un método que carga de un archivo las licenciaturas con el formato *clave,nombre*, se invocará esta clase para llenar la lista con objetos distintos a un *String*.

FrmSeleccionesObjetos.java

```
public void llenarListaLicenciaturas(){
    AdministrarLicenciaturas administrar = new AdministrarLicenciaturas();
    LinkedList<Licenciatura>lista = administrar.listaLicenciaturas();
    for(int i=0;i<lista.size();i++){
        Licenciatura licenciatura = lista.get(i);
        cmbListaLicenciaturas.addItem(licenciatura);
    }
}
```

Éste método se invoca en el método constructor de la Forma

FrmSeleccionesObjetos.java

```
public FrmSeleccionesObjetos() {
    initComponents();
    agruparSelecciones();
    agruparChkBoxes();
    cmbListaAlumnos.removeAllItems();
    cmbListaLicenciaturas.removeAllItems();
    llenarListaLicenciaturas();
}
```

Si bien se sigue utilizando el método *addItem* para agregar elementos, la configuración establecida en la pestaña *Codes* permite agregar objetos de tipo *Licenciatura*. El valor que aparece en la lista en la interfaz, se toma de los especificado en el método *toString()*.

Leyendo Objetos y Llenando Alumnos

El siguiente paso es obtener la clave (no el nombre) de la licenciatura seleccionada y a partir de esa cargar la lista de alumnos que la cursar.

Primero hay que indicar que el elemento *cmbListaAlumnos* se llenará con objetos tipo *Alumno*. Nuevamente se debe seleccionar la lista en el modo *Design* y en el panel de propiedades, seleccionar la pestaña *Code*, posteriormente buscar la propiedad *TypeParameters*, dar clic sobre los tres puntos (...) y en lugar de *<String>*, escribir *<Alumno>* (que es el nombre de la clase a partir de la cuál se crearán

objetos para llenar la lista) y dar clic en **OK**.

Como en el caso anterior, será necesario cambiar también el modelo (**model**) dejando limpio el campo de valores en la pestaña **Properties**.

Para leer los datos de la lista de licenciaturas, se agregará un evento del tipo **ActionPerformed** al elemento **cmbListaLicenciaturas**

FrmSeleccionesObjetos.java

```
private void cmbListaLicenciaturasActionPerformed(java.awt.event.ActionEvent evt) {  
    Licenciatura licenciatura = (Licenciatura)cmbListaLicenciaturas.getSelectedItem();  
    System.out.println(licenciatura.getClave());  
    llenarListaAlumnos(licenciatura.getClave());  
}
```

Aquí se recupera un objeto tipo **Licenciatura** y se manda a llenar la lista de alumnos que cursan la licenciatura seleccionada a partir de la llave, no del nombre.

```
public void llenarListaAlumnos(String clave){  
    AdministrarAlumnos administrar = new AdministrarAlumnos();  
    LinkedList<Alumno>lista = administrar.listaPorCarrera(clave);  
    cmbListaAlumnos.removeAllItems();  
    for(int i=0;i<lista.size();i++){  
        Alumno alumno = lista.get(i);  
        cmbListaAlumnos.addItem(alumno);  
    }  
}
```

Llenando las Opciones (radio buttons) con Objetos

Se creará un método para llenar los **radio buttons** a partir de un objeto de tipo **Opcion**.

FrmSeleccionesObjetos.java

```
public void llenarBotonesMerece(){  
    Opcion merece = new Opcion("MERECE","Merecedor de Apoyos");  
    Opcion noMerece = new Opcion("NO_MERECE","No es Merecedor de Apoyos");  
  
    btnMerece.setSelected(true);  
    btnMerece.setText(merece.getValor());  
    btnMerece.putClientProperty("llave", merece);  
  
    btnNoMerece.setText(noMerece.getValor());  
    btnNoMerece.putClientProperty("llave", noMerece);  
}
```

En este caso se utiliza el método *setText* para asignar el valor que aparecerá en la interfaz, pero con el método *putClientProperty* se indica que habrá un parámetro (*llave*) el cuál identificará a un determinado objeto (*merece* y *noMerece* de tipo *Opcion*).

Leyendo el Valor de un Radio Button

El método de lectura no varía mucho del utilizado solamente con *String*. Primero se creará un botón para capturar (de momento) solamente el dato del Radio Button.

FrmSeleccionesObjetos.java

```
private void btnRegistrarActionPerformed(java.awt.event.ActionEvent evt) {  
    Opcion opcion = seleccionarApoyo();  
    System.out.println(opcion.getLlave());  
}
```

El método para recuperar la opción seleccionada, en este caso un objeto de tipo *Opcion*

FrmSeleccionesObjetos.java

```
public Opcion seleccionarApoyo(){  
    Enumeration<AbstractButton> btnGrp = btnGrpApoyo.getElements();  
    Opcion resultado = new Opcion("a","b");  
    while(btnGrp.hasMoreElements()){  
        JRadioButton btnAux = (JRadioButton)btnGrp.nextElement();  
        if(btnAux.isSelected()){  
            resultado = (Opcion)btnAux.getClientProperty("llave");  
        }  
    }  
    return resultado;  
}
```

En este caso se utiliza la instrucción *getClientProperty* junto con el valor de la llave especificado al momento del llenado de los Radio Buttons (*llave*)

Llenando las Opciones (check boxes) con Objetos

Se llenarán los *checkboxes* también con objetos de tipo *Opcion* (podría ser otro tipo de objeto).

FrmSeleccionesObjetos.java

```
public FrmSeleccionesObjetos() {  
    initComponents();  
    agruparSelecciones();  
    agruparChkBoxes();  
    cmbListaAlumnos.removeAllItems();  
    cmbListaLicenciaturas.removeAllItems();  
}
```

```

llenarListaLicenciaturas();
llenarBotonesMerece();
llenarCheckBoxes();
}

```

La implementación del método es la siguiente:

FrmSeleccionesObjetos.java

```

public void llenarCheckBoxes(){
    Opcion opcionA = new Opcion("PRONABE","Beca Gubernamental Pronabe");
    Opcion opcionB = new Opcion("UAM","Beca Institucional UAM");
    Opcion opcionC = new Opcion("CDMX","Beca de Apoyo Ciudad de México");

    chkCDMX.putClientProperty("llave", opcionC);
    chkUam.putClientProperty("llave", opcionB);
    chkPronabe.putClientProperty("llave", opcionA);

    chkCDMX.setText(opcionA.getValor());
    chkUam.setText(opcionB.getValor());
    chkPronabe.setText(opcionC.getValor());
}

```

También se utiliza el método *putClientProperty* para indicar que a una llave le corresponde un determinado objeto.

Leyendo las Selecciones de los *CheckBoxes*

La lectura es similar a cuando se utilizan solamente objetos de tipo *String*. La lectura también se asociará a presionar el botón de Registrar.

FrmSeleccionesObjetos.java

```

private void btnRegistrarActionPerformed(java.awt.event.ActionEvent evt) {
    Opcion opcion = seleccionarApoyo();
    System.out.println(opcion.getLlave());

    for(int i= 0; i<listaChkBoxes.size();i++){
        JCheckBox aux = listaChkBoxes.get(i);
        if(aux.isSelected()){
            opcion = (Opcion)aux.getClientProperty("llave");
            System.out.println(opcion.getLlave());
        }
    }
}

```

Se recupera un objeto nuevamente con el método *getClientProperty* a través de la llave con la que se llenaron los objetos (llave).

Completando la Funcionalidad

Con esto ya se tiene completo el llenado y selección de elementos de selección (*combo boxes*, *radio buttons* y *checkboxes*) utilizando objetos. Ahora solo se completará el ejercicio para tener la funcionalidad requerida al inicio de la práctica.

Cuando se carga la aplicación, automáticamente se selecciona una licenciatura, por eso aparece la lista de alumnos. Lo que se hará es seleccionar un elemento que no muestre la lista de alumnos hasta que se seleccione como tal el nombre de la licenciatura.

Primero se Agrega esta opción en la generación de listas

AdministrarLicenciaturas.java

```
public LinkedList<Licenciatura> listaLicenciaturas() {
    String cadenaLeida = "";
    FileReader fr;
    LinkedList <Licenciatura> listaLicenciaturas = new LinkedList();
    try {
        fr = new FileReader("licenciaturas.dat");
        BufferedReader archivoLectura = new BufferedReader(fr);
        cadenaLeida = archivoLectura.readLine();
        while (cadenaLeida != null) {
            Licenciatura licenciatura = new Licenciatura();
            StringTokenizer st = new StringTokenizer(cadenaLeida, ",");
            licenciatura.setClave(st.nextToken());
            licenciatura.setNombre(st.nextToken());
            listaLicenciaturas.add(licenciatura);
            cadenaLeida = archivoLectura.readLine();
        }
        archivoLectura.close();
    } catch (FileNotFoundException e) {
        System.out.println("No se pudo encontrar el archivo");
        e.printStackTrace();
    } catch (IOException e) {
        System.out.println("No se pudo leer del archivo");
        e.printStackTrace();
    }
    Licenciatura licenciaturaNula = new Licenciatura();
    licenciaturaNula.setClave("NULA");
    licenciaturaNula.setNombre("Selecciona la Licenciatura");
    listaLicenciaturas.addFirst(licenciaturaNula);
    return listaLicenciaturas;
}
```

Con esto ya solo se cargan alumnos en caso de seleccionar alguna Licenciatura, ahora se ocultará la lista de alumnos y se mostrará solamente cuando se seleccione una licenciatura.

FrmSeleccionesObjetos.java

```
public FrmSeleccionesObjetos() {
    initComponents();
    agruparSelecciones();
    agruparChkBoxes();
    cmbListaAlumnos.removeAllItems();
    cmbListaLicenciaturas.removeAllItems();
    llenarListaLicenciaturas();
    llenarBotonesMerece();
    llenarCheckBoxes();
    ocultarComponentes();
}
```

FrmSeleccionesObjetos.java

```
public void ocultarComponentes(){
    cmbListaAlumnos.setVisible(false);
    chkCDMX.setVisible(false);
    chkPronabe.setVisible(false);
    chkUam.setVisible(false);
    btnMerece.setVisible(false);
    btnNoMerece.setVisible(false);
    lblAlumno.setVisible(false);
    btnRegistrar.setVisible(false);
}
```

Al seleccionar una licenciatura, se mostrará la lista de alumnos y los componentes para seleccionar si tiene derecho a apoyo o no.

FrmSeleccionesObjetos.java

```
private void cmbListaLicenciaturasActionPerformed(java.awt.event.ActionEvent evt) {
    Licenciatura licenciatura = (Licenciatura)cmbListaLicenciaturas.getSelectedItem();

    if(licenciatura.getClave().compareTo("NULA")!=0){
        System.out.println(licenciatura.getClave());
        llenarListaAlumnos(licenciatura.getClave());
        cmbListaAlumnos.setVisible(true);
        btnMerece.setVisible(true);
        btnNoMerece.setVisible(true);
        lblAlumno.setVisible(true);
        btnRegistrar.setVisible(true);
    }
}
```

```

    }else{
        cmbListaAlumnos.setVisible(false);
        ocultarComponentes();
    }
}

```

Al seleccionar la opción de que si se merecen apoyos, se mostrarán los *checkboxes* para seleccionarlos, si no se merecen apoyos, éstos no serán visibles. En este caso se tiene seleccionada la opción de “Merecedor de Apoyos” por lo que los elementos aparecerán al seleccionar la licenciatura. En caso de que se seleccione la otra opción, se ocultarán.

Para eso se agregará un evento del tipo `StateChanged` a cada uno de los `Radio Buttons`

FrmSeleccionesObjetos.java

```

private void btnNoMereceStateChanged(javax.swing.event.ChangeEvent evt) {
    if (btnNoMerece.isSelected()){
        chkCDMX.setVisible(false);
        chkPronabe.setVisible(false);
        chkUam.setVisible(false);
        chkCDMX.setSelected(false);
        chkPronabe.setSelected(false);
        chkUam.setSelected(false);
    }
}

```

FrmSeleccionesObjetos.java

```

private void btnMereceStateChanged(javax.swing.event.ChangeEvent evt) {
    if (btnMerece.isSelected()){
        chkCDMX.setVisible(true);
        chkPronabe.setVisible(true);
        chkUam.setVisible(true);
    }
}

```

Solo resta imprimir la matrícula del alumno al presionar el botón registrar junto con el indicador de si merece o no apoyos y cuáles.

FrmSeleccionesObjetos.java

```

private void btnRegistrarActionPerformed(java.awt.event.ActionEvent evt) {
    Opcion opcion = seleccionarApoyo();

    Alumno alumno = (Alumno)cmbListaAlumnos.getSelectedItem();
    System.out.println(alumno.getMatricula());

    System.out.println(opcion.getLlave());
}

```

```
for(int i= 0; i<listaChkBoxes.size();i++){
    JCheckBox aux = listaChkBoxes.get(i);
    if(aux.isSelected()){
        opcion = (Opcion)aux.getClientProperty("llave");
        System.out.println(opcion.getLlave());
    }
}
}
```

Con esto finaliza la práctica quedando algunas validaciones por realizar, por ejemplo:

- Que si se tiene seleccionada la opción de Merecer Apoyos, se marque al menos una de las opciones.
- Limpiar las opciones seleccionadas (en particular los apoyos) cuando se selecciona una nueva licenciatura o un nuevo alumno