Jesús Serrano Pérez
24/01/2022

0. IMPORTANT. Abans de començar les activitats de la sessió 2, fer una còpia del work space tal com va quedar en acabar la sessió 1.

1. Seguir els passos indicats al tutorial fins arribar a tenir l'AddressApp en l'estat que es mostra al final. Enregistrar en un diari cada pas que es dona, els problemes que es troben i la manera en què se solucionen. Incloure-hi captures de pantalla comentades de cada pas i del resultat final.

2. Consultar http://docs.oracle.com/javase/8/javafx/properties-binding-tutorial/binding.htm.

• Explicar quin avantatge aporta la utilització de propietats (Properties) per representar els atributs de les classes del model.

La utilització de propietats permet encapsular el codi, fent-lo més robust i segur, a l'hora de també fer-lo més accesible mitjançant getters i setters. D'aquesta manera es pot protegir el atribut original al no tenir un accés directe.

• Què és una classe envolvent o wrapper ?

Una classe envolvent o wrapper es una clase que envolcalla una altra funció, aillant aquesta de la resta de codi i protegint-la. També incorpora funcions per poder accedir al contingut en cas de ser necessari.

• Explicar la funció dels tres mètodes que es generen per a cada atribut.

Per cada atribut es genera un getter per recuperar el valor del atribut, un setter per assignar un nou valor a l'atribut, i finalment un altre getter que en comptes de retornar el valor, retorna un objecte igual a l'atribut original.

• Quines són les convencions que se segueixen en la nomenclatura dels mètodes de les propietats (Properties)?

```
getNomPropietat();
setNomPropietat();
nomPropietatProperty();
```

En cas de setters i getters, primer es posa set/get i a continuació el nom de la propietat mitjançant cammelcase. En el cas de Property, s'empra la mateixa lógica, però Property es troba al final del nom del mètode (a diferencia dels setters i getters).

3. Consultar http://docs.oracle.com/javase/8/javafx/collections-tutorial/collections.htm.

• Descriure les noves classes de Collection introduïdes per JavaFX (javafx.collections).

· **FXCollections**: A utility class that consists of static methods that are one-to-one copies of java.util.Collections methods
· **ListChangeListener.Change**: Represents a change made to an ObservableList
· **MapChangeListener.Change**: Represents a change made to an ObservableMap

• Comparar-les amb les utilitzades normalment en el desenvolupament en JavaSE (java.util).

Utilitzar qualsevol clase de Collection, implica també utilitzar el seu equivalent a java.util, ja que es un wrapper d'aquest i el que fa es afegir listeners en cas de que canvii de valor. Per tant, la diferencia consisteix en que les Collections de JavaFX son una extensió de Java.util i implementen listeners.

Jesús Serrano Pérez
24/01/2022

• Explicar com es combinen els dos tipus de col·leccions en l'exemple del tutorial d'Oracle. Comparar amb la manera en què es crea la ObservableList en el codi del exercici de classe.

Es combinen utilitzant la col·lecció de Java-util dins de la col·lecció de JavaFX.

```
    // Use Java Collections to create the List.
    Map<String,String> map = new HashMap<String,String>();

    // Now add observability by wrapping it with ObservableList.
ObservableMap<String,String> observableMap = FXCollections.observableMap(map);
    observableMap.addListener(new MapChangeListener() {
        @Override
        public void onChanged(MapChangeListener.Change change) {
            System.out.println("Detected a change! ");
        }
    });

    // Changes to the observableMap WILL be reported.
    observableMap.put("key 1","value 1");
    System.out.println("Size: "+observableMap.size());

    // Changes to the underlying map will NOT be reported.
    map.put("key 2","value 2");
    System.out.println("Size: "+observableMap.size());
```

• Creus que també s'aplica el concepte de classe envolvent o wrapper ?

Sí.

4. Indicar en quin pas de l'exercici s'estableix l'associació entre la descripció fxml de la interface i els objectes Java corresponents. Descriure com es fa.

Això es fa casi al final, cal anar objecte per objecte, anant a Hierarchy > Code > fx:id. En aquest punt s'especifica la ID del objecte i s'associa aquest objecte amb la seva descripció.

5. Mostrar quin és l'efecte sobre el(s) fitxer(s) fxml d'aquesta associació. (Comparar el(s) fitxer(s) fxml en l'estat final de la sesió 1 i l'estat final de la sessió 2). Creus que es imprescindible utilitzar el SceneBuilder ?

He treballat sobre la mateixa versió, així que no puc comparar, però sembla bastant similar i no crec que l'SceneBuilder sigui estrictament imprescindible, encara que agilitza molt les cosses i això ajuda molt.