SWING + JDBC. Exercise 1

GUI supported by the previous classes:

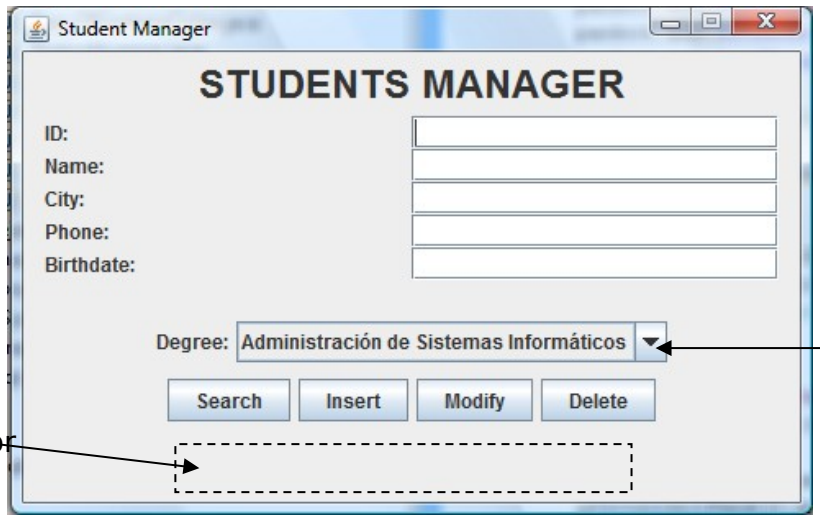
Make the next changes in the StudentManager class:

- None of the methods of the StudentManager class propagates exceptions.
- *removeStudent*(), *insertStudent*(), *modifyStudent*() methods: If they cannot accomplish their task, they will return a String that indicates “the reason why”.
- *searchStudent*() method: we bring it back to its original state (when it returned a Student)
- We will add some more methods:
  - searchDegree( ): that receives a degree identifier and returns an object with the corresponding Degree.
  - allDegrees( ): that returns an array with all the existing degrees (Degree objects) in the database.
  - finish( ): that closes the connection with the database.

Design the VisualManager class:

It is important to consider that this class/window interacts with the user through its GUI, but it never interacts directly with the database (it does not send queries).
In order to achieve this, this class’s main attribute is an instance of your *StudentManager* class.

Label (empty at the beginning), that displays success/error messages.



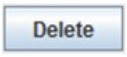
Instantiated over an array of Degree objects, with all the degrees of the database

When the Search button is pressed, it looks for a student with the id typed in the textbox. Once found, his data are loaded in the remaining components, the JComboBox included.
The JLabel displays the success or the failure reason for the search operation.

When the Insert button is clicked, a new student must be inserted into the database, with the data typed in the input fields. The JLabel must show a success message or the reason for the insertion failure.

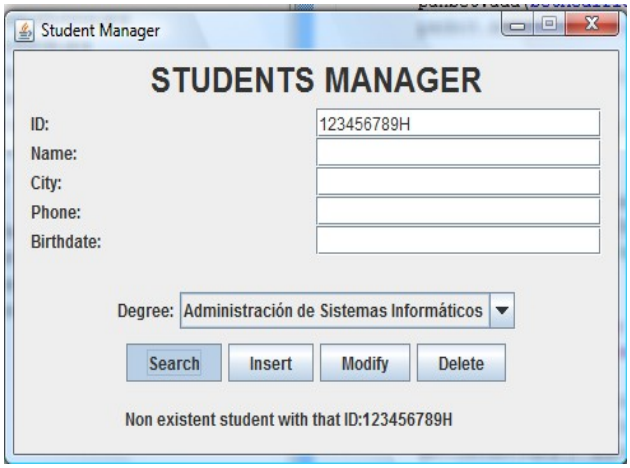
When clicking the 

button, the student with the ID indicated in the upper text field will be modified in the database with the data of the remaining field. The JLabel must show a success message or the reason for the updating failure.

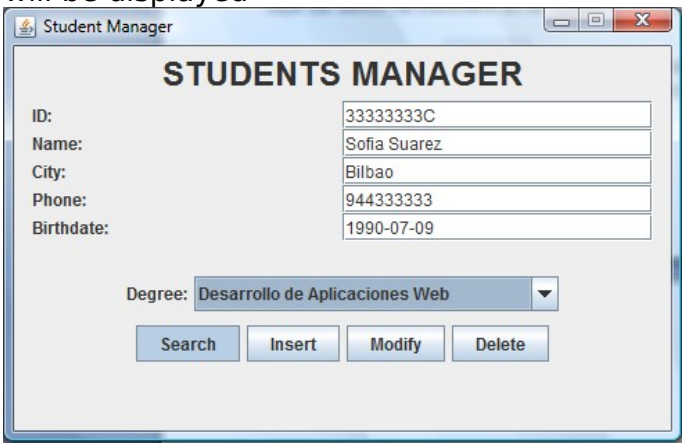
When clicking the 

button, the student with the typed ID must be removed from the database. The JLabel must show a success message or the reason for the removing failure.

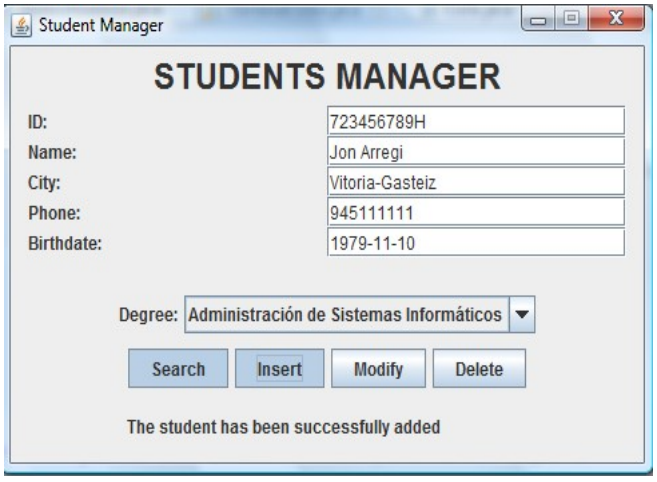
We look for a non-existing student



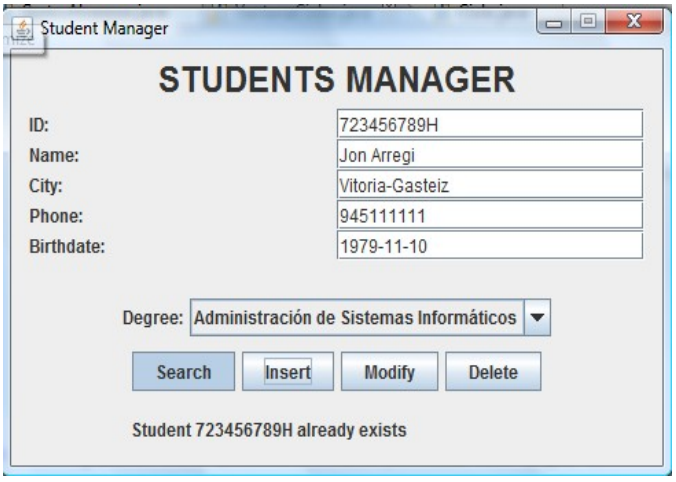
We search an existing student and his data will be displayed



Insert a new student

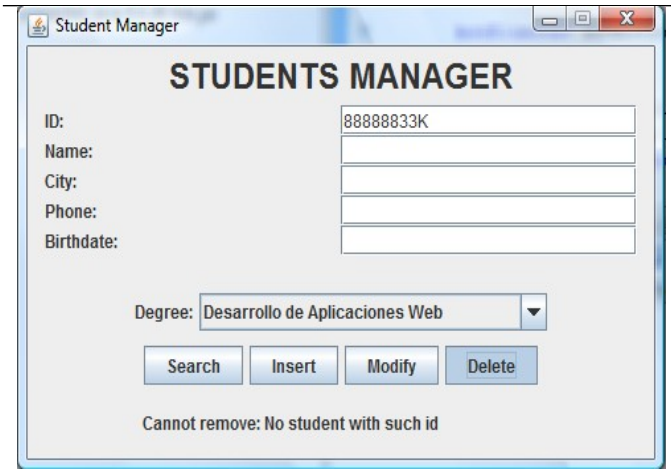


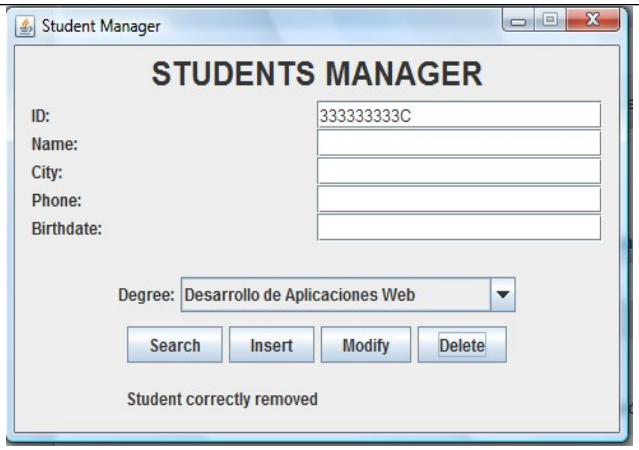
We try to insert the same student again



We try to delete a non-existent student

We delete an existing student


Student Manager
STUDENTS MANAGER
ID:
88888833K
Name:
City:
Phone:
Birthdate:
Degree:
Desarrollo de Aplicaciones Web
Search
Insert
Modify
Delete
Cannot remove: No student with such id


Student Manager
STUDENTS MANAGER
ID:
333333333C
Name:
City:
Phone:
Birthdate:
Degree:
Desarrollo de Aplicaciones Web
Search
Insert
Modify
Delete
Student correctly removed

etc

SWING + JDBC. Exercise 2

Create a MySQL database with the next structures, and load it with some tuples.

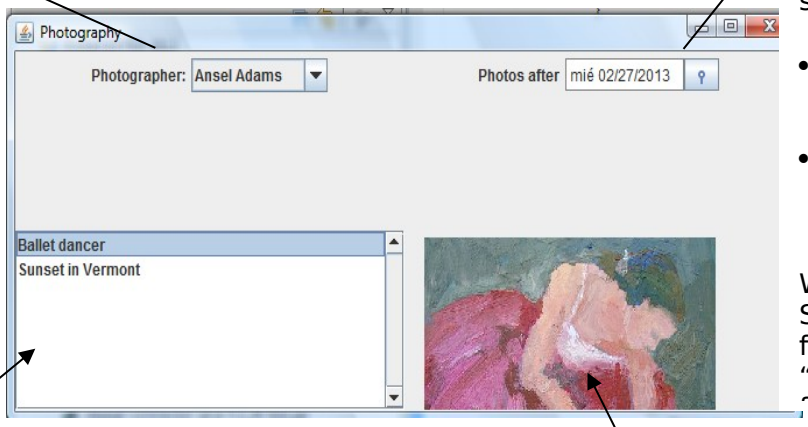
Pictures TABLE	
PictureId	Numeric
Title	Text
Date	Date
File	Text
Visits	Numeric
PhotographerId	Numeric

Photographers TABLE	
PhotographerId	Numeric
Name	Text
Awarded	Boolean

Create the PictureViewer class in Java, which corresponds to the next window:
(You can create a GridLayout with 4 areas)

JComboBox with all the photographers in the database.

It is built from an array of Photographer objects



Component of the **JXDatePicker** class (This class is in the swingx-0.9.2.jar. It is shared in your Y: unit)

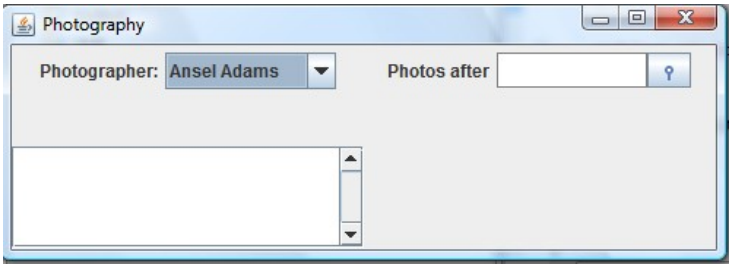
- You instantiate it this way: date=new JXDatePicker();
- For retrieving the selected date: getDate() method

We will format the date with a SimpleDateFormat-type formatter, in order to get a "day-month-year" String in the

JList with the pictures of the Photographer that has been selected on the JComboBox.
It is built over a model with Picture (new class) objects

Label with the selected image

1. At the beginning, the window appears like this (all the photographers loaded in the combobox, and the remaining components empty)
   We need to:
   - Codify a Photographer class
   - Call a method that returns an array of Photographers, with all the DB's Photographers



2. Next, a date must be picked from the JXDatePicker

3. Next, when we select a photographer from the Combo:

    - The list field must be loaded with all the pictures made by that photographer since the date indicated on the JXDatePicker.
      For this purpose, you must create a Picture class .
      You add elements to this list by using a model (DefaultListModel class).

    - If no date has been selected in the DatePicker, all the pictures by the selected photographer must appear.

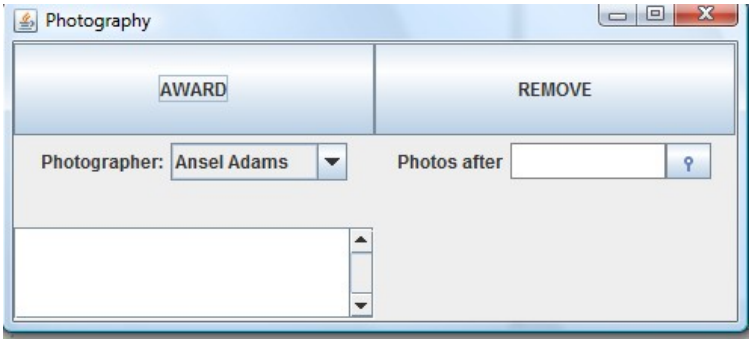
4. When “double clicking” one of the pictures on the list, the corresponding picture shows in the label
   * Remember that the “File” field of the Pictures table of the database contains the path of the pictures within our project.

5. Finally, do the necessary changes so that the field “visits” in the Pictures table, keeps how many times a picture has been displayed. Create a method with this prototype:

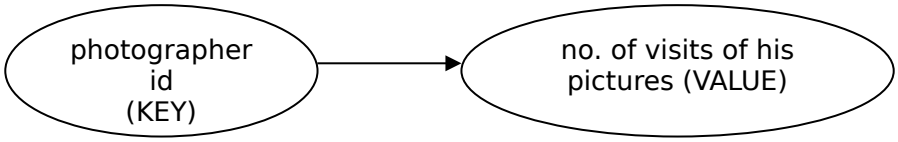
```
void  incrementVisits(Picture p);
```

SWING + JDBC. Exercise 3

Add 2 buttons “AWARD” and “REMOVE” to the previous window (anywhere), in order to practise updates over resultsets.



1º) Create a method createVisitsMap that creates and returns a HashMap with these associations:

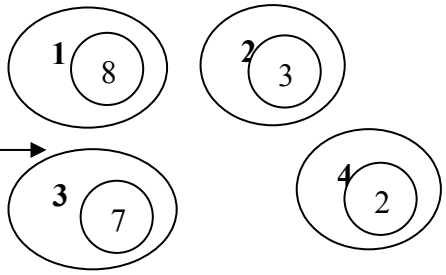


This method navigates through the pictures table, row by row, and creates new key/value pairs or updates the existing ones.

pictures table

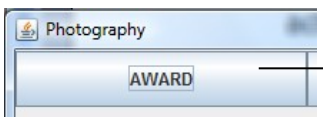
pictid	title	date	file	visits	phtid
1	Bailarina ...	2012-05-08	Ballet.jpg	4	1
2	Casas en playa	2012-05-15	Beach.jpg	2	4
3	Paisaje	2012-05-08	landscape.jpg	6	3
4	Amanecer	2012-05-09	ama2.jpg	3	2
5	Mujer rostro alargado	2012-05-11	Modigliani.jpg	1	3
6	Coche rojo	2007-05-11	Redcar.jpg	4	1

HashMap

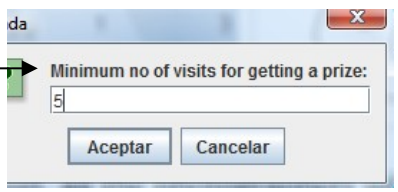


2º) When the AWARD button is pressed, all the photographers with a minimum of N visits to their pictures must be marked as awarded (N is asked in a popup window)
**Don’t use an update sql query*

We clic the button



Ask N



Update Photographers table, **row by row**

phogpherId	name	awarded
1	Ansel Adams	1
2	Morgan Norman	0
3	Annie Leibovitz	1
4	Irving Penn	0

3º) When the REMOVE button is clicked, the program must go through the pictures that have never been displayed and correspond to non-awarded photographers. Each time one of these pictures is found, the user must be prompted whether he wants to delete it or not, and the corresponding actions must be performed.

Finally, all those photographers with no pictures left will also be removed.