Java programming II - Online

Graphical User Interface

GUI

Assignment 2 – Animal Farm GUI

Mandatory

Farid Naisan
University Lecturer
Malmö University, Malmö Sweden
UCSD Extension, CA, USA

The Animal Park - GUI

1 Objectives

The main goals of this assignment are:

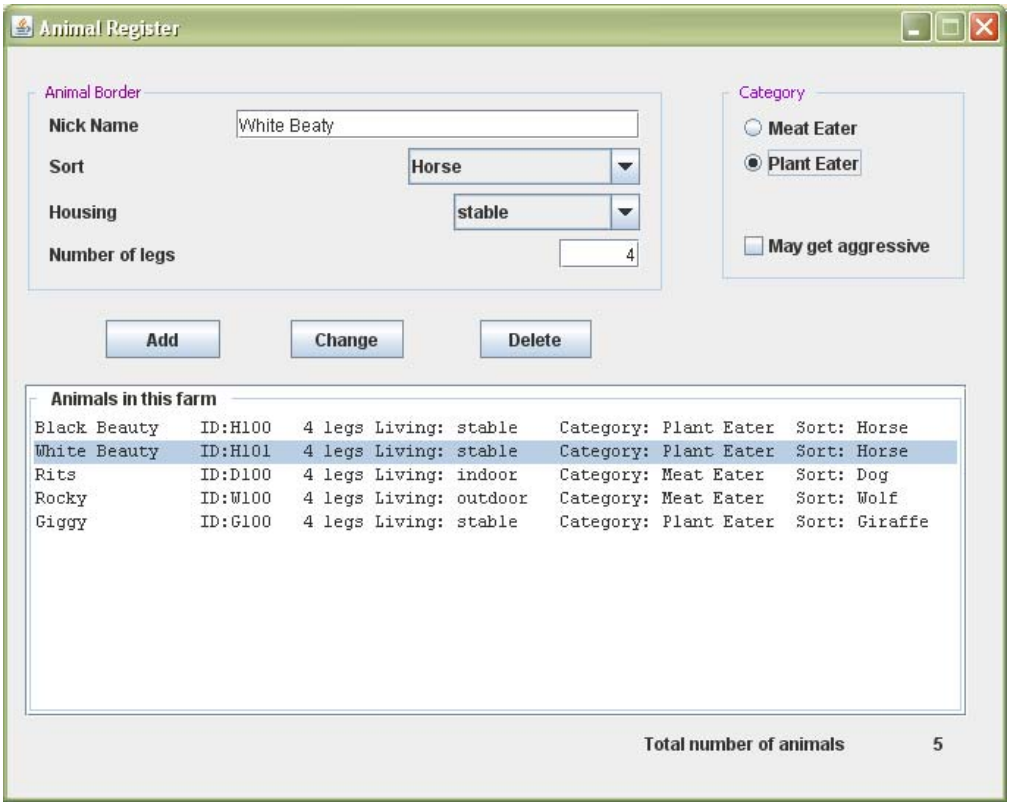
- To design Graphical User Interface (GUI) using the features of Java Swing and a number of commonly used components.
- To get training in event-driven programming.

Today, all programs that interact with users maintain a **G**raphical **U**ser **I**nterface (GUI). Console applications are mainly used for testing and in cases when programs are used behind the scenes. Therefore, you must assume that GUI is an important part of every application. It is also the GUI that users usually base their evaluation of an application on, no matter how much time you spend on the calculations and other parts of the program. From now on, we will be using GUI in all our assignments.

2 Description

In this assignment, your job is to develop Version 2.0 of the Animal Farm from Assignment 1. My suggestion is that you start a new project for Assignment 2, and then copy all files except the UserInterface and the AnimalFarm (starting class) classes to this new project. Begin your work then by creating a class, say **AnimalFarmGUI** that extends **JFrame**. If you are using NetBeans, this is done automatically when you choose to create a new JFrame.

Below is a sample GUI, which needs not to be followed in all details as long as the specifications below are followed. You may bring changes to the GUI, if you have better ideas



As mentioned before, you are not bound to use NetBeans and you may certainly use your favorite IDE or a simple text editor, whatever you feel confortable with.

3 Specification

3.1 All fields (instance variables) in all classes are kept private as in Assignment 1.

3.2 The GUI should include JLabel, JTextBox, JComboBox, JList, JRadioButton, JTextField JCheckBox, JPanel and of course JFrame.

3.3 When using IDEs, compoents get default names like jList1, or jComboBox8, and so on. These are difficult to work with when programming. To make life easier for yourself, give meaningful names to these GUI components. These are usually components that are used for output (textboxes, label that should the total number, the list box, etc). It is also strongly recommended that you begin the component names with a three or four letter prefix. A few recommendations:

Prefix	Component	Example
txtf	JTextField	txtfName
cmb	JComboBox	cmbSort
Lbl	JLabel	lblTotalNumOfAnimals
Lst	JList	lstAnimals
Chk	JCheckBox	chkAggresive
Pnl	JPanel	pnlAnimalInfo

Those components that only get their values at design time (such as headings) need not to be renamed and may hold their default values.

3.4 The total number of animals registered in the program should be shown (as in the bottom of the figure).

3.5 All input and output components should be clear of design text (jLabel1, etc) at the program start. Your GUI may certainly begin with some test or default values.

3.6 Write a method **initGUI** (in the GUI lass) in which you do all the initialization of the GUI. This method should be called from the GUI class' constructor. The initializations in turn should be coded in smaller methods.

3.7 Write also a **updateGUI** method responsible for updating the GUI whenever a change is done, for example when a new animal is added to the register or removed from the register, the JList and the total number of animals should be automatically updated.

3.8 The GUI class should only include code that has to do with the user interactions. All other tasks should be done by other classes. The GUI class should use the services of other classes. It should not do any manipulation of data when it can be done by other classes.

3.9 The GUI components should only be used for presentation of data. .All data must be saved in the related class (animal data should be saved in Animal objects, and so on).

3.10 The IDNumber is provided internally by the **AnimalManager** class.

3.11 It is sufficient that the **Add** and **Delete** buttons work in this version. The following functionalities and features are intended to be completed in next version (Assignment 3):

3.11.1 The change button.

3.11.2 The checkbox showing if the animal can be dangerous for the staff.

3.11.3 Animal food (test of the method ***eat***)

3.11.4 Control and validity of user input

3.12 Be very organized and object-oriented in your solution. Test your application in every step before moving on.

4 Some Guidelines:

4.1 When coding the method **updateGUI** to update changes, it is quite practical to clear the contents of the output components first, and then fill the updated data again. When a new animal is added or removed, the contents of the JList-component should be cleared wholly and then it should be refilled with up-to-date data from the ArrayList (via the instance of the AnimalManager).

4.2 Do not work with JList directly; create an instance of **DefaultListModel** (make sure that you import javax.swing.*), fill it with new data and pass it to the **model** method of JList.

4.3 When filling the ComboBox for Housing, make use of the enum **HousingType**.:

When you find the instructions here insufficient or unclear, make your own judgment. If you have questions or unsure about things, use the module's forum.

A couple of advices from me in designing your GUI:

1. ***Keep your GUI simple; have the "dumb" user in mind.***
2. ***Follow the standards (File-View…Help when designing menus)***

Good Luck!

Programming is fun. Never give up. Ask for help!

Farid Naisan,