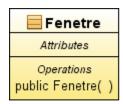
Lucie TEILHAC

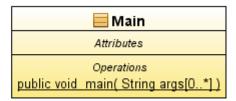
Multithreaded Address Book

Advanced Java

2013-2014

Package Controller





Class Main:

The main class contains a Fenetre variable and is only used to call the Fenetre class.

Class Fenetre:

The Fenetre class contains its configuration and is used to declare a container, (Panel).

Package view

```
🚃 Panel
                                 Attributes
private int NBCONTACTS = 10
private JComboBox list
private JTextArea displaycontact = new JTextArea("")
private JScrollPane scroll = new JScrollPane(displaycontact)
private JComboBox group
private JTextField jfname = new JTextField("Enter the name")
private JTextField jlname = new JTextField("Enter the last name")
private JTextField jaddress = new JTextField("Enter the address")
private JTextField jcaddress = new JTextField("Enter the company address")
private JTextField jpersophone = new JTextField("00")
private JTextField jprophone = new JTextField("00")
private JTextField jhome = new JTextField("www.")
private JTextField iproemail = new JTextField("@ece.fr")
private JTextField jpersoemail = new JTextField("@gmail.com")
private JTextField jpict = new JTextField("inconnu.jpg")
private Image img
private Imagelcon imgicon
private JLabel imagecontact
private JTextField jgroupe = new JTextField("Enter a group")
                                 Operations
public Panel( )
public void paintComponent( Graphics g )
public void actionPerformed( ActionEvent arg0 )
public String[0..*] updateBox( )
```

CLASS PANEL:

The panel class extends the JPanel class.

It's my container and it contains in its attributes all the elements of the frame.

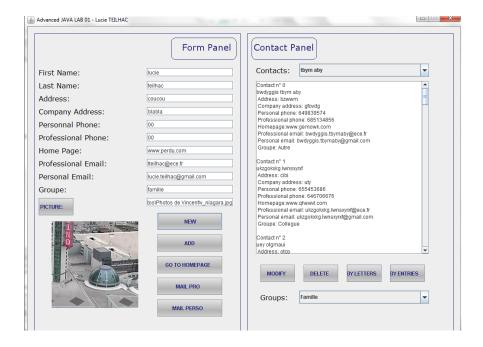
Its constructor set up the view and adds all the components to the container.

The PaintComponent function is used to draw in the background of the panel.

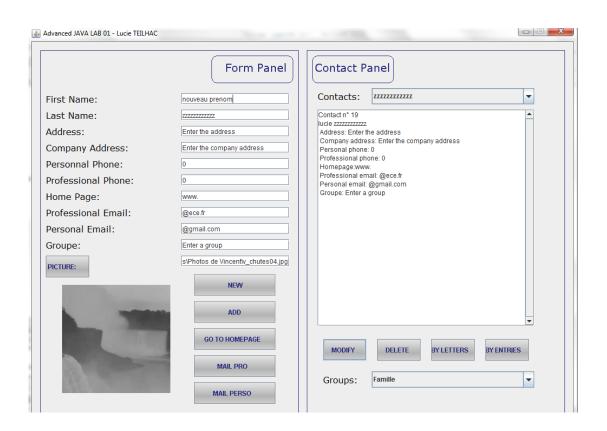
The ActionPerformed function is used to handle all the actions make by the user by clicking on the buttons. It recognize which button or component has been clicked and do the corresponding action: add a contact, modify a contact, delete a contact, order the list by entries, by groups or by letters, send a mail ,visit a website or simply display a chosen contact.

The UpdateBox function is used to full the JComboBox containing the contact's lastname list.

All the buttons aren't enable to use all the time. For example, to create a new contact, you need to click on the "new" button, then to fill the form and finally, click on the "Add" button. The new contact is displayed on the JTextField.



To modify or delete a contact, you first need to select it in the comboBox and then, you can modify it or delete it. You just have to click on the good button, or to complete/change the form value and click on "modify".



<u>\$</u>	Advanced JAVA LAB 01 - Lucie TEILHAC	Relay Completion - Monad Mad.	- X
		Form Panel Contact Panel	
	First Name:	nouveau prenom Contacts: ZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZ	_
	Last Name:	Contact n° 19 nouveau prenom zzzzzzzzzzz	
	Address:	Enter the address Address: Enter the address	
	Company Address:	Enter the company address Enter the company address Personal phone: 0	
	Personnal Phone:	Professional phone: 0 Homepage:www.	
	Professional Phone:	Professional email: @ece.fr Personal email: @gmail.com	
	Home Page:	www. Groupe: Enter a group	
	Professional Email:	@ece.fr	
	Personal Email:	@gmail.com	
	Groupe:	Enter a group	
	PICTURE:	s\Photos de Vincent\(\frac{1}{2}\) chutes\(04.jpg\)	

In the same way, you can see a homepage and send an email only if you first select a contact in the combo box. All the functionalities are working.

Package Model



Attributes private String name

Operations public Label(String str.)

Bouton

Attributes

private String name private Image img private Graphics g

Operations

public Bouton(String str)
public void paintComponent(Graphics g)
public void mouseClicked(MouseEvent event)
public void mouseEntered(MouseEvent event)
public void mouseExited(MouseEvent event)
public void mousePressed(MouseEvent event)

public void mouseReleased(MouseEvent event)

```
Public Repertoire

private int action = 0

Attributes

Operations

public Repertoire( int nbContacts )
public String show/Repertoire( )
public string show/Repertoire( )
public void addContact (String nfname, String nlname, String naddress, String ncompanyaddress, int npersonalphone, int nprofessionalphone, String nhomepage, String nproemail, String nperemail, String nphoto, String groupe )
public void supprContact (int id, String nfname, String nlname, String naddress, String ncompanyaddress, int npersonalphone, int nprofessionalphone, String nhomepage, String nproemail, String nphoto, String groupe )
public void ChargeContacts( )
public void EnregContacts( )
public void setAction( int a )
public void setAction( int a )
public void setAction( int a )
```

contactlist 0..* **■** Contact Attributes package int ID private String fname private String Iname private String address private String companyaddress private int personalphone private int professionalphone private String homepage private String proemail private String persoemail private String photo private String groupe private int count = 0 public Contact() public Contact(int nID)
package String randomGroupe() package String randomString() package String randomInt() public void Contact(String nfname, String nlname, String naddress, String ncompanyaddress, int ppersonalphone, int pprofessionalphone, String nhomepage, String nproemail, String nperemail, String nphoto, String groupe) public String showContact(public String getFirstname(public void setFirstname(String firstname)
public String getLastname() public void setLastname(String lastname) public String getAddress() public void setAddress(String address) public String getCompanyaddress() public void setCompanyaddress(String companyaddress) public int getPersonalphone()
public void setPersonalphone(int personalphone) public int getProfessionalphone()
public void setProfessionalphone(int professionalphone) public String getHomepage()
public void setHomepage()
public void setHomepage(String homepage)
public String getProemail()
public void setFroemail()
public string getPeremail() public void setPeremail(String persoemail)
public String getPhoto()
public void setPhoto(String photo) . public int getID() public void setID(int id) package int getCount()
public void setCount() public String getGroupe()
public void setGroupe(String groupe)

CLASS BOUTON:

I create my own JButton class only to customize my buttons.

CLASS LABEL:

I create my own JLabel class only to customize my labels.

CLASS CONTACT:

The contact class contains all the parameters of a contact in its attributes and all the setter and getter to access to them in its operation.

CLASS REPERTOIRE:

The Repertoire class contains an Arraylist of contact. This class processes the contact data thanks to this contact list. This class also implements runnable.

The constructor initialize the Arraylist.

The ShowRepertoire function calls a contact function that returns a string containing all the informations about each contact. This function is used to display the contact list in the view.

The add/delete/modify function is used to respectively add/delete/modify a contact of the Arraylist.

The ChargerContact function and the EnregContact function are used to import and export the contact list in the .csv file. These functions are called by the run function.

Threads:

As I said, the repertoire class implements runnable. That's why there is a run function in it. This function is used to, according to the action required, import or export the contact in the address book. To do this, I used threads and this choice allows the user to keep working on the interface while the address book is saving the changes the user made.

Since I used NetBeans to code this project, I give you a GitHub link to my address book:

https://github.com/LucieT/ADVANCEDJAVA/