Anthony Meunier

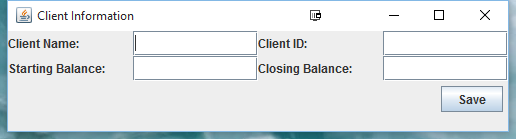
DeVry University

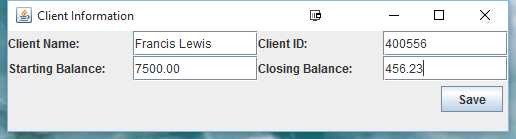
CIS 355A

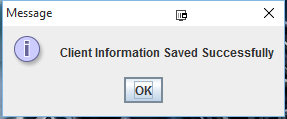
Week 5 iLab

GUI Graphics and File I/O

Step 1: Writing out Client Information







/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Program Name: Client.java

Programmer's Name: Anthony Meunier

Program Description: This class stores information for the client.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

**public** **class** Client {

**private** String ClientName;

**private** String ClientId;

**private** **double** StartingBalance;

**private** **double** ClosingBalance;

**public** Client(String cname,String cid,**double** sbal,**double** cbal){

ClientName = cname;

ClientId = cid;

StartingBalance = sbal;

ClosingBalance = cbal;

}

**public** **void** setClientName(String cname){

ClientName = cname;

}

**public** String getClientName(){

**return** ClientName;

}

**public** **void** setClientId(String cid){

ClientId = cid;

}

**public** String getClientId(){

**return** ClientId;

}

**public** **void** setStartingBalance(**double** bal){

StartingBalance = bal;

}

**public** **double** getStartingBalance(){

**return** StartingBalance;

}

**public** **void** setClosingBalance(**double** bal){

ClosingBalance = bal;

}

**public** **double** getClosingBalance(){

**return** ClosingBalance;

}

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Program Name: StoreClientInformation.java

Programmer's Name: Anthony Meunier

Program Description: This class allows user to enter client information,

and then stores client information in a text file.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

**import** java.awt.Container;

**import** java.awt.Dimension;

**import** java.awt.FlowLayout;

**import** java.awt.GridLayout;

**import** java.awt.event.ActionEvent;

**import** java.awt.event.ActionListener;

**import** java.io.BufferedWriter;

**import** java.io.File;

**import** java.io.FileWriter;

**import** java.io.IOException;

**import** java.text.DecimalFormat;

**import** javax.swing.JButton;

**import** javax.swing.JFrame;

**import** javax.swing.JLabel;

**import** javax.swing.JOptionPane;

**import** javax.swing.JPanel;

**import** javax.swing.JTextField;

**public** **class** StoreClientInformation **implements** ActionListener{

//GUI controls

**static** JFrame *frame* ;

**static** Container *pane*;

**static** JLabel *lblClientName*;

**static** JLabel *lblClientId*;

**static** JLabel *lblStartingBal*;

**static** JLabel *lblClosingBal*;

**static** JTextField *txtClientName*;

**static** JTextField *txtClientId*;

**static** JTextField *txtStartingBal*;

**static** JTextField *txtClosingBal*;

**static** JButton *btnSave*;

**public** StoreClientInformation()

{

*frame* = **new** JFrame("Client Information");

*frame*.setDefaultCloseOperation(JFrame.***EXIT\_ON\_CLOSE***);

//Set up the content pane

*pane* = *frame*.getContentPane();

*pane*.setPreferredSize(**new** Dimension(500,100));

*pane*.setLayout(**new** GridLayout(2,1));

//Panel for input controls

JPanel pnlInput = **new** JPanel(**new** GridLayout(2,4));

//Create the input fields

*lblClientName* = **new** JLabel("Client Name:");

*txtClientName* = **new** JTextField(25);

*lblClientId* = **new** JLabel("Client ID:");

*txtClientId* = **new** JTextField(25);

*lblStartingBal* = **new** JLabel("Starting Balance:");

*txtStartingBal* = **new** JTextField(25);

*lblClosingBal* = **new** JLabel("Closing Balance:");

*txtClosingBal* = **new** JTextField(25);

*btnSave* = **new** JButton("Save");

*btnSave*.addActionListener(**this**);

pnlInput.add(*lblClientName*);

pnlInput.add(*txtClientName*);

pnlInput.add(*lblClientId*);

pnlInput.add(*txtClientId*);

pnlInput.add(*lblStartingBal*);

pnlInput.add(*txtStartingBal*);

pnlInput.add(*lblClosingBal*);

pnlInput.add(*txtClosingBal*);

*pane*.add(pnlInput);

JPanel pnlSave = **new** JPanel(**new** FlowLayout(FlowLayout.***RIGHT***));

pnlSave.add(*btnSave*);

*pane*.add(pnlSave);

//Display the frame

*frame*.pack();

*frame*.setVisible(**true**);

}

**public** **static** **boolean** saveToFile(String clientInfo){

String file\_name = "client.txt";

File file = **new** File(file\_name);

**try** {

**if**(!file.exists()){

file.createNewFile();

}

FileWriter fstream = **new** FileWriter(file\_name,**true**);

BufferedWriter out = **new** BufferedWriter(fstream);

out.write(clientInfo);

out.newLine();

out.close();

**return** **true**;

} **catch** (IOException e) {

e.printStackTrace();

**return** **false**;

}

}

**public** **static** **void** main(String[] args) {

@SuppressWarnings("unused")

StoreClientInformation c = **new** StoreClientInformation();

}

@Override

**public** **void** actionPerformed(ActionEvent arg0) {

String clientName = *txtClientName*.getText();

String clientId = *txtClientId*.getText();

DecimalFormat df = **new** DecimalFormat("#.00");

**double** startBal = Double.*parseDouble*(*txtStartingBal*.getText());

**double** closeBal = Double.*parseDouble*(*txtClosingBal*.getText());

String contents = clientName + "|" + clientId + "|" + df.format(startBal) +"|" + df.format(closeBal) + "\n";

**boolean** isSaved = *saveToFile*(contents);

**if**(isSaved){

JOptionPane.*showMessageDialog*(**null**,"Client Information Saved Successfully");

}

**else**

{

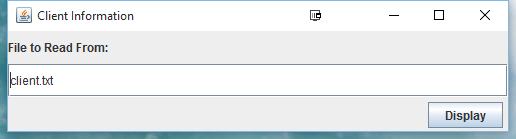
JOptionPane.*showMessageDialog*(**null**,"Client Information Could Not Be Saved");

}

}

}

Step 2: Reading in Client Information





/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Program Name: ReadClientInformation.java

Programmer's Name: Anthony Meunier

Program Description: This class reads information from the text file and displays it.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

**import** java.awt.Container;

**import** java.awt.Dimension;

**import** java.awt.FlowLayout;

**import** java.awt.GridLayout;

**import** java.awt.event.ActionEvent;

**import** java.awt.event.ActionListener;

**import** java.io.BufferedReader;

**import** java.io.FileReader;

**import** java.io.IOException;

**import** java.util.ArrayList;

**import** javax.swing.JButton;

**import** javax.swing.JFrame;

**import** javax.swing.JLabel;

**import** javax.swing.JPanel;

**import** javax.swing.JTextField;

**public** **class** ReadClientInformation **implements** ActionListener {

//GUI controls

**static** JFrame *frame* ;

**static** Container *pane*;

**static** JLabel *lblFileName*;

**static** JTextField *txtFileName*;

**static** JButton *btnDisplay*;

**public** ReadClientInformation(){

*frame* = **new** JFrame("Client Information");

*frame*.setDefaultCloseOperation(JFrame.***EXIT\_ON\_CLOSE***);

//Set up the content pane.

*pane* = *frame*.getContentPane();

*pane*.setPreferredSize(**new** Dimension(500,100));

*pane*.setLayout(**new** GridLayout(3,1));

*lblFileName* = **new** JLabel("File to Read From:");

*txtFileName* = **new** JTextField(25);

*txtFileName*.setText("client.txt");

JPanel pnlDisplay = **new** JPanel(**new** FlowLayout(FlowLayout.***RIGHT***));

*btnDisplay* = **new** JButton("Display");

*pane*.add(*lblFileName*);

*pane*.add(*txtFileName*);

pnlDisplay.add(*btnDisplay*);

*btnDisplay*.addActionListener(**this**);

*pane*.add(pnlDisplay);

//Display the frame

*frame*.pack();

*frame*.setVisible(**true**);

}

**public** **static** **void** main(String[] args) {

@SuppressWarnings("unused")

ReadClientInformation r = **new** ReadClientInformation();

}

@Override

**public** **void** actionPerformed(ActionEvent e) {

ArrayList<Client> clients = readFromFile(*txtFileName*.getText());

System.***out***.println("\t\t\t\tClient Activity Report");

System.***out***.format("%20s%20s%20s%20s","Client Name","Client ID","Starting Balance","Closing Balance");

System.***out***.println();

**for**(**int** i=0;i<clients.size();++i)

{

System.***out***.format("%20s%20s%20.2f%20.2f",clients.get(i).getClientName(), clients.get(i).getClientId(),

clients.get(i).getStartingBalance(),clients.get(i).getClosingBalance());

System.***out***.println();

}

}

**public** ArrayList<Client> readFromFile(String fileName){

ArrayList<Client> clients = **new** ArrayList<Client>();

//Opens the file and display the contents

FileReader freader;

**try** {

freader = **new** FileReader(fileName);

BufferedReader in = **new** BufferedReader(freader);

String clientInfo = "";

**while**((clientInfo = in.readLine()) != **null**)

{

String[] clientattr = clientInfo.split("\\|");

**if**(clientattr.length==4){

Client c = **new** Client(clientattr[0],clientattr[1],Double.*parseDouble*(clientattr[2]),Double.*parseDouble*(clientattr[3]));

clients.add(c);

}

}

in.close();

} **catch** (IOException e) {

e.printStackTrace();

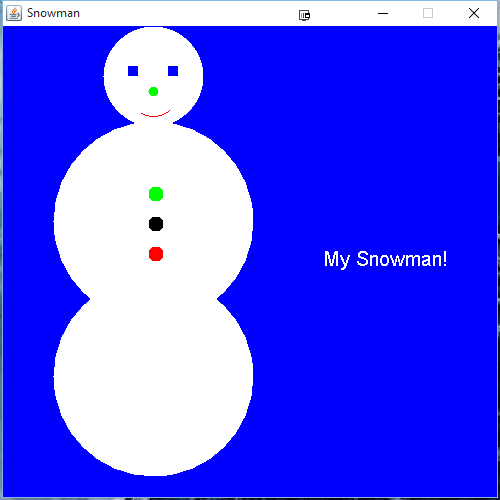
}

**return** clients;

}

}

Step 3: Snowman!



/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Program Name: Snowman.java

Programmer's Name: Anthony Meunier

Program Description: This class draws and displays a snowman.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

**import** java.awt.Color;

**import** java.awt.Font;

**import** java.awt.Graphics;

**import** javax.swing.JFrame;

**import** javax.swing.JPanel;

**public** **class** Snowman **extends** JPanel{

/\*\*

\*

\*/

**private** **static** **final** **long** ***serialVersionUID*** = 1L;

**private** JFrame frame = **new** JFrame("Snowman"); //Frame object for the display

**public** Snowman(){

frame.add(**this**);

frame.setSize(500, 500);

frame.setResizable(**false**);

frame.setVisible(**true**);

}

/\*\*

\* paintComponent method

\* **@param** g Graphics object

\*/

**public** **void** paintComponent(Graphics g)

{

**super**.paintComponent(g); //Base class method clears background

g.setColor(Color.***blue***);

g.fillRect(0, 0, getWidth(), getHeight());

g.setColor(Color.***white***);

g.fillOval(100,0, 100, 100);

g.fillOval(50,95, 200, 200);

g.fillOval(50,250, 200, 200);

g.translate(125, 10);

g.translate(0, 70);

g.setColor(Color.***BLUE***);

g.fillRect(0, -40, 10, 10);

g.translate(40, 0);

g.setColor(Color.***BLUE***);

g.fillRect(0, -40, 10, 10);

g.translate(-20, 20);

g.setColor(Color.***GREEN***);

g.fillOval(0,-40, 10, 10);

g.setColor(Color.***red***);

g.translate(-20, -20);

g.drawArc(0, -40, 50,50, 240, 70);

g.translate(20, 100);

g.setColor(Color.***green***);

g.fillOval(0,-20, 15, 15);

g.translate(0, 30);

g.setColor(Color.***black***);

g.fillOval(0,-20, 15, 15);

g.translate(0, 30);

g.setColor(Color.***red***);

g.fillOval(0,-20, 15, 15);

g.translate(100, -100);

**int** fontSize = 20;

g.setFont(**new** Font("Arial", Font.***PLAIN***, fontSize));

g.setColor(Color.***white***);

g.drawString("My Snowman!", 75, 100);

}

**public** **static** **void** main(String[] args) {

// **TODO** Auto-generated method stub

@SuppressWarnings("unused")

Snowman s = **new** Snowman();

}

}