**JAVA 2 FINAL NOTES**

1. **GUIs:**

**GUIExample.java**

|  |
| --- |
| //Importing all our components  import javax.swing.JFrame; //our lihgtwieght main frame  import javax.swing.JLabel; // our lightweight label  import javax.swing.JPanel;  import javax.swing.JTextField;  import javax.swing.JComboBox;  import javax.swing.JCheckBox;  import javax.swing.JRadioButton;  import javax.swing.ButtonGroup; //it is not JBUttonGroup because its not a component  import javax.swing.JTextArea;  import javax.swing.JButton;  import java.awt.GridLayout;  import java.awt.BorderLayout; //its in the awt because the layout does not need to be lightweight, so we dont take it from swing  import java.awt.Color;  import javax.swing.ImageIcon;  //do not include action listener if you will not use it  import java.awt.event.ActionListener;  import java.awt.event.ActionEvent;  import javax.swing.JOptionPane;  public class GUIExample extends JFrame implements ActionListener{    //We declare our components (as variables)  private JLabel fullnameLabel,stateLabel,hobbyLabel,blankLabel, maleLabel, femaleLabel;  private ImageIcon maleIcon, femaleIcon;  private JTextField fnTextField;  private JPanel northPanel,westPanel,westPanel2,centerPanel,southPanel;  private JComboBox stateDropbox;  private JCheckBox musicBox, movieBox;  private JRadioButton maleRadioButton, femaleRadioButton;  private ButtonGroup genderGroup;  private JTextArea commentArea;  private JButton submitButton, cancelButton;  public GUIExample(){    super("My awesome GUI"); //setTitle("My java GUI")//System.out.println("hello");        //\*\*\*Putting the Label component and TextField to a north Panel    northPanel = new JPanel();  //Styling  northPanel.setBackground(Color.BLUE);  //Label component  fullnameLabel = new JLabel("Enter Full Name:");  fullnameLabel.setForeground(Color.WHITE);  add(fullnameLabel,BorderLayout.NORTH);    //TextFieldComponent  fnTextField = new JTextField(5);    northPanel.add(fullnameLabel);  northPanel.add(fnTextField);      westPanel = new JPanel();  westPanel.setLayout(new GridLayout(3,2));  //StateLabel  stateLabel = new JLabel("State");  //Dropbox  String states [] = {"","DC","MD","VA"};  stateDropbox = new JComboBox(states);  //HobbyLabel  hobbyLabel = new JLabel("Hobbies:");  hobbyLabel.setToolTipText("Tips here!");  //Checkboxes  blankLabel = new JLabel("");  musicBox = new JCheckBox("Music");  movieBox = new JCheckBox("Movies");  westPanel.add(stateLabel);  westPanel.add(stateDropbox);  westPanel.add(hobbyLabel);  westPanel.add(blankLabel);  westPanel.add(musicBox);  westPanel.add(movieBox);    westPanel2 = new JPanel();  westPanel2.add(westPanel);    //Center  //RadioButtons  maleRadioButton = new JRadioButton("");  femaleRadioButton = new JRadioButton("");  genderGroup = new ButtonGroup();  genderGroup.add(maleRadioButton);  genderGroup.add(femaleRadioButton);  maleIcon = new ImageIcon("male.gif");  femaleIcon = new ImageIcon("female.gif");  maleLabel = new JLabel(maleIcon);  femaleLabel = new JLabel(femaleIcon);  centerPanel = new JPanel();  centerPanel.add(maleLabel);  centerPanel.add(maleRadioButton);  centerPanel.add(femaleLabel);  centerPanel.add(femaleRadioButton);    //CommentArea  commentArea = new JTextArea(5,10);      //South  southPanel = new JPanel();  submitButton = new JButton("Submit");  submitButton.addActionListener(this);  cancelButton = new JButton("Cancel");  cancelButton.setBackground(Color.RED);  cancelButton.addActionListener(this);  southPanel.add(submitButton);  southPanel.add(cancelButton);  //\*\*\*Adding our panels to our JFrame    add(northPanel,BorderLayout.NORTH);  add(commentArea,BorderLayout.EAST);  add(centerPanel,BorderLayout.CENTER);  add(southPanel,BorderLayout.SOUTH);  add(westPanel2,BorderLayout.WEST);      //Our mainframe settings  setSize(600,500);  setVisible(true);  setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);  }    public void actionPerformed(ActionEvent event){    if(event.getSource() == submitButton){    String fullname = fnTextField.getText();  String comment = commentArea.getText();  String state = stateDropbox.getSelectedItem().toString();    String movie = "No";  String music = "No";  if(movieBox.isSelected()){  movie = "Yes";  }else{  movie = "No";  }    if(musicBox.isSelected()){  music = "Yes";  }else{  music = "No";  }      String gender = "";  if(maleRadioButton.isSelected()){  gender = "male";  }  if(femaleRadioButton.isSelected()){  gender = "female";    }        String output = "Full name: "+ fullname +  "\nComment: " + comment +  "\n State: " + state +  "\n Movie? " + movie +  "\n Music? " + music +  "\n Gender: " + gender;      JOptionPane.showMessageDialog(null,output);        }else if(event.getSource() == cancelButton){  fnTextField.setText("");  commentArea.setText("");  stateDropbox.setSelectedIndex(0);  musicBox.setSelected(false);  movieBox.setSelected(false);  genderGroup.clearSelection();  }      }    public static void main(String [] args) {  GUIExample app = new GUIExample();    }      } |

**GasStation5.java**

A screenshot of a cell phone

Description automatically generated

A screenshot of a social media post

Description automatically generated

A screenshot of a social media post

Description automatically generated

A screenshot of a social media post

Description automatically generated

**AmountException for Gas Station5.java:**

public class AmountException extends Exception{

public AmountException(){

super("Error: Amount is required!");

}

public AmountException(String message){

super(message);

}

}

**2 EXAMPLES OF EXCEPTIONS:**

**A screenshot of a cell phone

Description automatically generated**

**A screenshot of a social media post

Description automatically generated**

**Reading and Writing to Files:**

import 
import 
import 
public 
java. io. File; 
java. io. Fi leNotFoundException ; 
java. util . Scanner; 
class ReadFromFile{ 
public static void main(String C] args){ 
File file 
= new txt"); 
try { 
= new Scanner(file); 
Scanner reader 
// System. out. println(reader. nextLine()); can crash easily 
while(reader . 
System. out . println(reader . nextLine()) ; 
can also do: 
String line 
= reader. nextLine(); 
System. out . println(line); 
reader . close(); 
catch(FileNotFoundException fnfe){ 
fnfe . printStackTrace() ; 

1 
2 
3 
4 
5 
6 
7 
8 
9 
10 
11 
12 
13 
14 
15 
16 
17 
18 
19 
20 
21 
22 
23 
24 
25 
26 
28 
29 
30 
31 
mport 
import 
import 
public 
java. io. File; 
java. io. FileWriter; 
java. io. IOException ; 
class WriteToFile { 
public static void main(String C] args){ 
File file 
= new txt"); 
String sample Txt = 
"Marymount University" ; 
try{ 
FileWriter writer 
= new FileWriter(file); 
// FileWriter writer 
= new FileWriter(file,true); 
writer . hiimaaaaa"); 
writer . write(sampleTxt) ; 
writer. write("\nHappy Galentines dayy"); 
writer . close(); 
catch(IOException ioe){ 
System. out . println(ioe . toString()) ; 
// to keep the old content/ to append text 

**Database:**

* **Insert Commands: A screenshot of a social media post

  Description automatically generated**

**Select Commands (Something is returned to us):**

A screenshot of a social media post

Description automatically generated

**MULibrary :**

|  |
| --- |
| //Imports -- layout  import javax.swing.JFrame;  import javax.swing.JPanel;  import java.awt.BorderLayout;  import javax.swing.JLabel;  import javax.swing.JTextField;  import javax.swing.JButton;  //Imports -- events  import java.awt.event.ActionListener;  import java.awt.event.ActionEvent;  import javax.swing.JOptionPane;  //Imports -- for database  import java.sql.Connection;  import java.sql.DriverManager;  import java.sql.SQLException;  import java.sql.Statement;  import java.sql.ResultSet;  //For Sockets  import java.net.Socket;  import java.net.ServerSocket;  import java.io.ObjectOutputStream;  import java.io.ObjectInputStream;  import java.net.InetAddress;  import java.net.UnknownHostException;  import java.io.IOException;  import java.util.Scanner;  public class MULibrary extends JFrame implements ActionListener {    private JPanel centerPanel;  private JLabel isbnLabel,titleLabel;  private JTextField isbnField, titleField;  private JButton newBookBtn, searchMUBtn, searchArlBtn;    private Socket socket;  private ServerSocket serverSocket;  private ObjectOutputStream output;              public MULibrary(){  super("MU Library");    //CenterPanel  centerPanel = new JPanel();  isbnLabel = new JLabel("ISBN");  titleLabel = new JLabel("Title");  isbnField = new JTextField(10);  titleField = new JTextField(10);  centerPanel.add(isbnLabel);  centerPanel.add(isbnField);  centerPanel.add(titleLabel);  centerPanel.add(titleField);    newBookBtn = new JButton("Add New Book");  newBookBtn.addActionListener(this);  searchMUBtn = new JButton("Search ISBN - MU Library");  searchMUBtn.addActionListener(this);  searchArlBtn = new JButton("Search ISBN - Arlington Library");  searchArlBtn.addActionListener(this);  centerPanel.add(newBookBtn);  centerPanel.add(searchMUBtn);  centerPanel.add(searchArlBtn);      //Last things added to JFrame:  add(centerPanel,BorderLayout.CENTER);    //JFrame Dimensions:  setSize(500,500);  setVisible(true);  setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);    try{  serverSocket = new ServerSocket(1097,500);  }catch(IOException ioe){  ioe.printStackTrace();  }      }    public void actionPerformed(ActionEvent event){      //\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*DATABASE CONNECTION  String url = "jdbc:mysql://localhost/mulibrary?user=root&password="; //This protocol is for mySQL only.  Connection connection;    try {      // 1. Connecting to the db  Class.forName("com.mysql.jdbc.Driver"); //used to connect to somee packages  connection = DriverManager.getConnection(url);  System.out.print("Database is connected");        // 2. Our actual event handlers now ...    //\*\*\*\*If Add New Book is clicked  if(event.getSource() == newBookBtn){    String isbn = isbnField.getText();  String title = titleField.getText();    //Showing the messageDialog  String output = "The following book is successfully added to MU Library"+  "\n ISBN"+ isbn +  "\n Title:"+ title + "";  JOptionPane.showMessageDialog(null,output);  //Saving to database  try {  //To insert, update, delete, create etc... Any command where you don't need to receive anything in return  String insertStatement = "insert into book values("+ isbn + ","+ "'" + title + "'"+ ")";  System.out.println("\n"+insertStatement);  Statement stmt = connection.createStatement();  stmt.execute(insertStatement);  System.out.print("Inserted Successfully!");  }catch(SQLException sqle){  sqle.printStackTrace(); //note that you can also write this printStackTrace to a text file :)  }    }  //\*\*\*\*If Search MU Library is Clicked  else if(event.getSource() == searchMUBtn){      String isbn = isbnField.getText();    //A select command --> querying the database --> here something will be returned to us  Statement stmt = connection.createStatement();  String query = "select isbn,title from book where isbn =" + isbn + "";  String output = "";  ResultSet result = stmt.executeQuery(query);  while(result.next()){ //will check whether the next row is a null or if there's a value  isbn = result.getString(1);  String title = result.getString(2);  output = "ISBN: " +isbn + "\nTitle:" + title + "\n Available at MU Library";  }    if(output.equals("")){  JOptionPane.showMessageDialog(null, "ISBN:"+ isbn+ " is not available at MU Library");  }else{  JOptionPane.showMessageDialog(null,output);  }    }//\*\*\*\*If Search Arl Library is Clicked --> we connect to the ARLibrary server  else if(event.getSource() == searchArlBtn){    try{        //this doesnt go in a loop. Do not use a loop on the gui        //Sending the isbn to the ARLibrary Server:  socket = new Socket(InetAddress.getByName("localhost"),1098);  output = new ObjectOutputStream(socket.getOutputStream());      Scanner scan = new Scanner(System.in);  String message = isbnField.getText(); //Scanner to type in the message to send      output.writeObject(message); //Ready to send  output.flush();    //Receiving response from ARLibraryServer:      socket = serverSocket.accept();  ObjectInputStream input = new ObjectInputStream(socket.getInputStream()); //Display message received  String message2 = (String) input.readObject();  JOptionPane.showMessageDialog(null,message2);    }  catch(UnknownHostException uhe){  uhe.printStackTrace();  }catch(IOException ioe){  ioe.printStackTrace();  }      }    }catch(ClassNotFoundException cnfe){  cnfe.printStackTrace();  }catch(SQLException sqle){  sqle.printStackTrace();  }    }  public static void main(String [] args) {    MULibrary app = new MULibrary();      }  } |

**Arlington Server**

|  |
| --- |
| //Imports  //For sockets  import java.net.ServerSocket;  import java.net.Socket;  import java.io.IOException;  import java.io.ObjectInputStream;  import java.io.ObjectOutputStream;  import java.net.InetAddress;  import java.util.Scanner;  //For database  import java.sql.Connection;  import java.sql.DriverManager;  import java.sql.SQLException;  import java.sql.Statement;  import java.sql.ResultSet;  public class ARLibraryServer {      //Declaring some private vars  private ServerSocket serverSocket;  private Socket socket;  private ObjectInputStream input;  public ARLibraryServer(){    //Server to receive inputs from MU LIbrary  System.out.print("Arlington Library Server is running");  Scanner scanner = new Scanner(System.in);    //Initialize the socket and its stuff  try{  serverSocket = new ServerSocket(1098,500); //the port the server socket will be receiving messages on    while(true){ //Use a loop to keep the server running  socket = serverSocket.accept();  input = new ObjectInputStream(socket.getInputStream()); //Display message received  String message = (String) input.readObject();  // message = String.valueOf(message);//converting our message to String    //Now we must connect to db to see if we have the book!!  String url = "jdbc:mysql://localhost/arlibrary?user=root&password=";  try{  Class.forName("com.mysql.jdbc.Driver");  System.out.println("JDBC is found");    Connection connection = DriverManager.getConnection(url);  System.out.println("Database is connected");    Statement stmt = connection.createStatement();  String query = "select \* from book where isbn = " + message + "";  ResultSet result = stmt.executeQuery(query);    String response;  if(result.next()){  response = result.getString(1) + " is available at ARLibrary";  }else{  response = message + " is not available at ARLibrary";  }    //Return results to MU Library app  Socket socket2 = new Socket(InetAddress.getByName("localhost"),1097);  ObjectOutputStream output = new ObjectOutputStream(socket2.getOutputStream());  output.writeObject(response);  output.flush();          }  catch(ClassNotFoundException cnfe){  cnfe.printStackTrace();  }  catch(SQLException sqle){  sqle.printStackTrace();  }    }  }  catch(IOException ioe){  ioe.printStackTrace();  }catch(ClassNotFoundException cnfe){  cnfe.printStackTrace();  }        }      public static void main(String [] args){    ARLibraryServer app = new ARLibraryServer();    }    } |

**Chat User 1**

import java.net.ServerSocket;

import java.net.Socket;

import java.io.IOException;

import java.io.ObjectInputStream;

import java.net.InetAddress;

import java.io.ObjectOutputStream;

import java.util.Scanner;

public class ChatUser1 {

//Declare

private ServerSocket serverSocket;

private Socket socket;

private ObjectInputStream input;

private ObjectOutputStream output;

public ChatUser1() {

System.out.println("Server is running!!!");

Scanner scanner = new Scanner(System.in);

//Initialize

try {

serverSocket = new ServerSocket(1098, 500); //Do NOT ever put this in loop... its only the listening socket that goes in the loop

while(true) {//Use a loop to keep server running

//initialize socket to send a message to client running on port 1097

socket = new Socket(InetAddress.getByName("localhost"),1097);

//Server socket to receive messages from Chat User 1

socket = serverSocket.accept(); //accept connection from client

input = new ObjectInputStream(socket.getInputStream());//Display recieved package / Receive output stream object

String message = (String) input.readObject(); //convert String byte to String

System.out.println("Client says: " + message);//display the message received from client

//Client socket to send messages to Chat User 2

//initialize output stream object

output = new ObjectOutputStream(socket.getOutputStream());

System.out.print("Server says:");

String message2 = scanner.nextLine();

//write buffered output bytes and flush through to the underlying stream

output.writeObject(message2);

output.flush();

System.out.println("Message sent!!");

}

}

catch(IOException ioe) {

ioe.printStackTrace();

}

catch(ClassNotFoundException cnfe) {

cnfe.printStackTrace();

}

}

public static void main(String [] args) {

new ChatUser1();

}

}

**Chat User2:**

import java.net.Socket;

import java.net.ServerSocket;

import java.io.ObjectOutputStream;

import java.net.InetAddress;

import java.net.UnknownHostException;

import java.io.IOException;

import java.util.Scanner;

import java.io.ObjectInputStream;

public class ChatUser2 {

//Declare

private Socket socket; //declare an actual socket

private ObjectOutputStream output; //declare an output stream to send a message

private ObjectInputStream input;

private ServerSocket serverSocket;

public ChatUser2() {

//Initialize

try {

serverSocket = new ServerSocket(1097, 500);

while(true) {

//Client socket to send messages to Chat User 1

socket = new Socket(InetAddress.getByName("localhost"), 1098);

//socket = new Socket("10.8.42.210", 1098);

output = new ObjectOutputStream(socket.getOutputStream()); //carrier

Scanner scan = new Scanner(System.in);

System.out.print("User 2 says:");

String message = scan.nextLine(); //Scanner to type in a message sent to the client (the package being sent)

//write buffered output bytes and flush throught to the underlying stream

output.writeObject(message);

output.flush();

System.out.println("Message sent!!!");

//Server socket to receive messages from Chat User 1

socket = serverSocket.accept(); //accept connection from client

input = new ObjectInputStream(socket.getInputStream());//Display recieved package / Receive output stream object

String message2 = (String) input.readObject(); //convert String byte to String

System.out.println("User 1 says: " + message2);//display the message received from

}

}

catch(UnknownHostException uhe) {

uhe.printStackTrace();

}

catch(IOException ioe) {

ioe.printStackTrace();

}

catch(ClassNotFoundException cnfe){

cnfe.printStackTrace();

}

}

public static void main(String [] args) {

new ChatUser2();

}

}