**CSC 239 Session 7**

**Putting It All Together (Part 2)**

**// Emprec.java**

**//package pkg;**

**import java.text.DecimalFormat;**

**import java.util.Date;**

**public class Emprec extends HoursRate implements Emprec\_Iface{**

**//define properties**

**private int record;**

**private String jobTitle;**

**private String name;**

**private int age;**

**private int dependents;**

**private char gender;**

**private Date date;**

**//constructor**

**public Emprec( int record, String jobTitle, String name, int age, int dependents, char gender, double hours, double rate)**

**{**

**super(hours, rate);**

**this.record=record;**

**this.jobTitle=jobTitle;**

**this.name=name;**

**this.age=age;**

**this.dependents=dependents;**

**this.gender = gender;**

**date = new Date(); //initialize a date**

**}**

**public Date getDate(){**

**return date;**

**}**

**//interface method implementation**

**public double calc\_gross\_pay()**

**{**

**return ( getHours() \* getRate() );**

**}// calc\_gross\_pay**

**//interface method implementation**

**public double calc\_fed\_tax(double hours,double rate)**

**{**

**double yearly\_income;**

**yearly\_income = hours \* rate \* TOTAL\_WEEKS; //52 - weeks per year**

**if (yearly\_income < YEAR\_INCOME\_S) return (hours \* rate \* TAX\_RATE\_ONE);**

**else if ( yearly\_income < YEAR\_INCOME\_M) return (hours \* rate\* TAX\_RATE\_TWO);**

**else return (hours \* rate \* TAX\_RATE\_THREE);**

**}// calc\_fed\_tax**

**//interface method implementation**

**public double calc\_state\_tax(double hours, double rate)**

**{**

**return (hours \* rate \* STATE\_TAX\_RATE);**

**}// calc\_state\_tax**

**public String toString()**

**{**

**DecimalFormat twoDigits = new DecimalFormat( "0.00" );**

**DecimalFormat dollar = new DecimalFormat( "$.00" );**

**return**

**("\n Record: "+ this.record +**

**"\n Job title: "+ this.jobTitle +**

**"\n Name: "+ this.name +**

**"\n Age: "+ this.age +**

**"\n Dependents: " + this.dependents +**

**"\n Gender: " + this.gender +**

**"\n Hours: " + twoDigits.format(getHours()) +**

**"\n Rate: " + dollar.format(getRate()) +**

**"\n Gross Pay: " + dollar.format(calc\_gross\_pay()) +**

**"\n Federal Tax: " + dollar.format(calc\_fed\_tax(getHours(), getRate())) +**

**"\n State Tax: " + dollar.format(calc\_state\_tax(getHours(), getRate())));**

**}//toString**

**}**

**//Emprec\_Iface.java**

**//package pkg;**

**//create an interface**

**public interface Emprec\_Iface{**

**//define and initialize taxRate**

**public double TAX\_RATE\_ONE = 0.28;**

**public double TAX\_RATE\_TWO = 0.32;**

**public double TAX\_RATE\_THREE = 0.38;**

**public double YEAR\_INCOME\_S = 30000.00;**

**public double YEAR\_INCOME\_M = 50000.00;**

**public double STATE\_TAX\_RATE = 0.0561;**

**public int TOTAL\_WEEKS = 52;**

**public double calc\_gross\_pay();**

**public double calc\_fed\_tax(double hours,double rate);**

**public double calc\_state\_tax(double hours, double rate);**

**}**

**//HoursRate.java**

**//package pkg;**

**//create class that hold data from hours and rate fields**

**public class HoursRate{**

**private double hours;**

**private double rate;**

**//default constructor**

**public HoursRate(){**

**this.hours = 0.00;**

**this.rate = 0.00;**

**}**

**//constructor**

**public HoursRate(double hours, double rate){**

**this.hours = hours;**

**this.rate = rate;**

**}**

**public HoursRate(String hours, String rate){**

**try{**

**this.hours = Double.valueOf(hours).doubleValue();**

**this.rate = Double.valueOf(rate).doubleValue();**

**}catch(NumberFormatException errmsg){**

**System.out.println("Invalid format " + errmsg);**

**this.hours = 0.0;**

**this.rate = 0.0;**

**}**

**}**

**//define accessor functions**

**public double getHours(){**

**return this.hours;**

**}**

**public double getRate(){**

**return this.rate;**

**}**

**}**

**//MakePayrollFile.java**

**import java.io.\*;**

**import java.awt.\*;**

**import java.awt.event.\*;**

**public class MakePayrollFile extends Frame //Frame class reperesents top-level winow with a title and border**

**implements ActionListener { //ActionListener interface - receive action events**

**// text components that allow for editing of single line of text**

**private TextField RecNumberField, NameField, DependentsField,**

**AgeField, HoursField,RateField, JobTitleField, GenderField;**

**private Button enter, // send record to file**

**done; // quit program**

**// Application other pieces**

**private DataOutputStream output;**

**public MakePayrollFile() //constructor**

**{**

**super( "Payroll File - Write" );**

**// Open the file**

**try {**

**output = new DataOutputStream(**

**new FileOutputStream( "payroll.txt" ) );**

**}**

**catch ( IOException e ) {**

**System.err.println( "File not opened properly\n" +**

**e.toString() );**

**System.exit( 1 );**

**}**

**setSize( 400, 400 ); // set frame size**

**setLayout( new GridLayout( 9, 2 ) ); // create layout with 7 rows and 2 columns**

**// create the components of the Frame**

**add( new Label( " Record Number" ) );**

**RecNumberField = new TextField();**

**add( RecNumberField );**

**add( new Label( " Job Title" ) ); //add new text field**

**JobTitleField = new TextField();**

**add( JobTitleField );**

**add( new Label( " Name" ) );**

**NameField = new TextField( );**

**add( NameField );**

**add( new Label( " Age" ) );**

**AgeField = new TextField( );**

**add( AgeField );**

**add(new Label(" Dependents "));**

**DependentsField = new TextField();**

**add(DependentsField);**

**add(new Label(" Gender "));**

**GenderField = new TextField();**

**add(GenderField);**

**add( new Label( " Hours" ) );**

**HoursField = new TextField( );**

**add( HoursField );**

**add( new Label( " Rate" ) );**

**RateField = new TextField( );**

**add( RateField );**

**enter = new Button( " Enter" );**

**enter.addActionListener( this );**

**add( enter );**

**done = new Button( " Done" );**

**done.addActionListener( this );**

**add( done );**

**setVisible( true ); //set visability of frame**

**HoursField.requestFocus(); //set mouse focus in a field Hours**

**}**

**public void actionPerformed( ActionEvent e ) //get mouse events**

**{**

**if ( e.getSource() == enter ) addRecord(); //add new record**

**if ( e.getSource() == done ) { //close a file**

**try {**

**output.close();**

**}**

**catch ( IOException io ) {**

**System.err.println( "File not closed properly\n" + io.toString() );**

**System.exit( 1 );**

**}**

**System.exit( 0 );**

**}**

**}**

**public void addRecord(){**

**//define temporary variables**

**int recNumber = 0;**

**int ageTemp = 0;**

**int depenTemp = 0;**

**Double h,r;**

**char genderTemp;**

**if(!RecNumberField.getText().equals( "" )) { //check if input is not empty**

**try{**

**recNumber = Integer.parseInt( RecNumberField.getText());**

**if ( recNumber > 0 ) { //if account number is more than zero**

**output.writeInt( recNumber ); //write integer RecNumber to a file**

**output.writeUTF( JobTitleField.getText() );**

**output.writeUTF( NameField.getText() );**

**try{**

**ageTemp = Integer.parseInt(AgeField.getText()); //convert str to int**

**output.writeInt(ageTemp); //write age**

**}catch(NumberFormatException nfe){**

**System.err.println("You must enter an integer number" );**

**}**

**try{**

**depenTemp = Integer.parseInt(DependentsField.getText());**

**output.writeInt(depenTemp);**

**}catch(NumberFormatException nfe){**

**System.err.println("You must enter an integer number" );**

**}**

**try{**

**genderTemp = GenderField.getText().charAt(0);**

**output.writeChar(genderTemp);**

**}catch(IndexOutOfBoundsException nfe){**

**System.err.println( "Gender must be entered as a one character" );**

**}**

**try{**

**h = new Double ( HoursField.getText() );**

**output.writeDouble( h.doubleValue() );**

**}catch(NumberFormatException nfe){**

**System.err.println("Number must be entered as a Double");**

**}**

**try{**

**r = new Double ( RateField.getText() );**

**output.writeDouble( r.doubleValue() );**

**}catch(NumberFormatException nfe){**

**System.err.println("Number must be entered as a Double");**

**}**

**}//if statement**

**// clear the TextFields**

**RecNumberField.setText( "" );**

**JobTitleField.setText("");**

**NameField.setText( "" );**

**AgeField.setText("");**

**GenderField.setText("");**

**DependentsField.setText("");**

**HoursField.setText( "" );**

**RateField.setText("");**

**}catch(NumberFormatException nfe){**

**System.err.println("You must enter an integer number" );**

**}catch ( IOException io ) {**

**System.err.println("Error during write to file\n" + io.toString() );**

**System.exit( 1 );**

**}**

**}//if**

**}//addRecord**

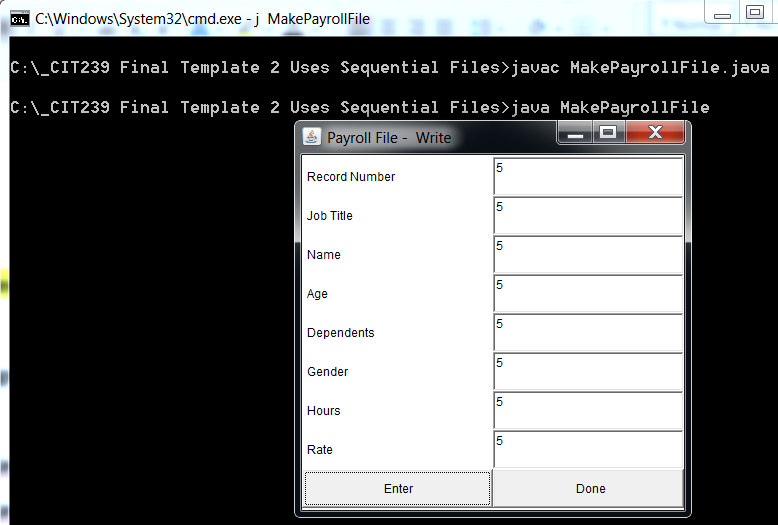
**public static void main( String args[] )**

**{**

**new MakePayrollFile();**

**}**

**}**



**//ReadPayrollFile.java**

**// This program reads a file sequentially and displays each**

**// record in text fields.**

**//import pkg.Emprec;**

**import java.io.\*;**

**import java.awt.\*;**

**import java.awt.event.\*;**

**import java.text.DecimalFormat;**

**import java.text.SimpleDateFormat;**

**import java.util.Date;**

**public class ReadPayrollFile extends Frame**

**implements ActionListener {**

**Emprec employee;**

**private TextField RecNumberField, NameField, DependentsField, AgeField, HoursField,**

**RateField, JobTitleField, GenderField, GrossPayField, FedTaxField, StateTaxField, DateField;**

**private Button next, // get next record in file**

**done; // quit program**

**// Application other pieces**

**private DataInputStream input;**

**// Constructor -- initialize the Frame**

**public ReadPayrollFile()**

**{**

**super( "Read Client File" );**

**// Open the file**

**try {**

**//open a file**

**input = new DataInputStream(**

**new FileInputStream( "payroll.txt" ) );**

**}**

**catch ( IOException e ) {**

**System.err.println( "File not opened properly\n" +**

**e.toString() );**

**System.exit( 1 );**

**}**

**setSize( 400, 500 );**

**setLayout( new GridLayout( 13, 2 ) );**

**// add Text Filds**

**add( new Label(" Date"));**

**DateField = new TextField();**

**DateField.setEditable(false);**

**add(DateField);**

**add( new Label( " Record Number" ) );**

**RecNumberField = new TextField();**

**RecNumberField.setEditable( false );**

**add( RecNumberField );**

**add( new Label( " Job Title" ) );**

**JobTitleField = new TextField();**

**JobTitleField.setEditable(false);**

**add( JobTitleField );**

**add( new Label( " Name" ) );**

**NameField = new TextField();**

**NameField.setEditable( false );**

**add( NameField );**

**add( new Label( " Age" ) );**

**AgeField = new TextField( );**

**AgeField.setEditable(false);**

**add( AgeField );**

**add(new Label(" Dependents "));**

**DependentsField = new TextField();**

**DependentsField.setEditable(false);**

**add(DependentsField);**

**add(new Label(" Gender "));**

**GenderField = new TextField();**

**GenderField.setEditable(false);**

**add(GenderField);**

**add( new Label( "Hours" ) );**

**HoursField = new TextField();**

**HoursField.setEditable( false );**

**add( HoursField );**

**add( new Label( "Rate" ) );**

**RateField = new TextField();**

**RateField.setEditable( false );**

**add( RateField );**

**//add new text fields**

**add( new Label("Gross Pay") );**

**GrossPayField = new TextField();**

**GrossPayField.setEditable( false );**

**add(GrossPayField);**

**add( new Label( "Federal Tax" ) );**

**FedTaxField = new TextField();**

**FedTaxField.setEditable( false );**

**add(FedTaxField);**

**add( new Label("State Tax") );**

**StateTaxField = new TextField();**

**StateTaxField.setEditable( false );**

**add(StateTaxField);**

**next = new Button( "Next" );**

**next.addActionListener( this );**

**add( next );**

**done = new Button( "Done" );**

**done.addActionListener( this );**

**add( done );**

**setVisible( true );**

**}**

**public void actionPerformed( ActionEvent e )**

**{**

**if ( e.getSource() == next )**

**readRecord();**

**else**

**closeFile();**

**}**

**public void readRecord() //read record from a file**

**{**

**DecimalFormat twoDigits = new DecimalFormat( "0.00" );**

**DecimalFormat dollar = new DecimalFormat( "$.00" );**

**//define temporary variables**

**int recnumber = 0;**

**int age = 0;**

**int depTemp = 0;**

**String name, jobTitle;**

**double h,r;**

**char gender;**

**// input the values from the file**

**try {**

**//read data from a file**

**recnumber = input.readInt();**

**jobTitle = input.readUTF(); //read one character**

**name = input.readUTF();**

**age = input.readInt();**

**depTemp = input.readInt();**

**gender = input.readChar();**

**h = input.readDouble();**

**r=input.readDouble();**

**//create new instance of Emprec object**

**employee=new Emprec(recnumber, jobTitle, name, age, depTemp, gender, h, r);**

**System.out.println(employee); //toString()**

**//format date and time**

**SimpleDateFormat dateFormat = new SimpleDateFormat("MM/dd/yyyy, hh:mm");**

**Date date = employee.getDate();**

**//show data on fields**

**DateField.setText(String.valueOf(dateFormat.format(date)));**

**RecNumberField.setText( String.valueOf( recnumber ) );**

**JobTitleField.setText(jobTitle);**

**NameField.setText( name );**

**AgeField.setText(String.valueOf(age));**

**DependentsField.setText(String.valueOf(depTemp));**

**GenderField.setText(String.valueOf(gender));**

**HoursField.setText( String.valueOf( twoDigits.format(h) ) );**

**RateField.setText( String.valueOf( dollar.format(r) ) );**

**GrossPayField.setText( String.valueOf( dollar.format(employee.calc\_gross\_pay())));**

**FedTaxField.setText( String.valueOf( dollar.format(employee.calc\_fed\_tax(h, r))));**

**StateTaxField.setText( String.valueOf( dollar.format(employee.calc\_state\_tax(h, r))));**

**}**

**catch ( EOFException eof ) { //end of file exception**

**closeFile();**

**}**

**catch ( IOException e ) {**

**System.err.println( "Error during read from file\n" + e.toString() );**

**System.exit( 1 );**

**}**

**}//end readRecord()**

**private void closeFile()**

**{**

**try {**

**input.close();**

**System.exit( 0 );**

**}**

**catch ( IOException e ) {**

**System.err.println( "Error closing file\n" + e.toString() );**

**System.exit( 1 );**

**}**

**}**

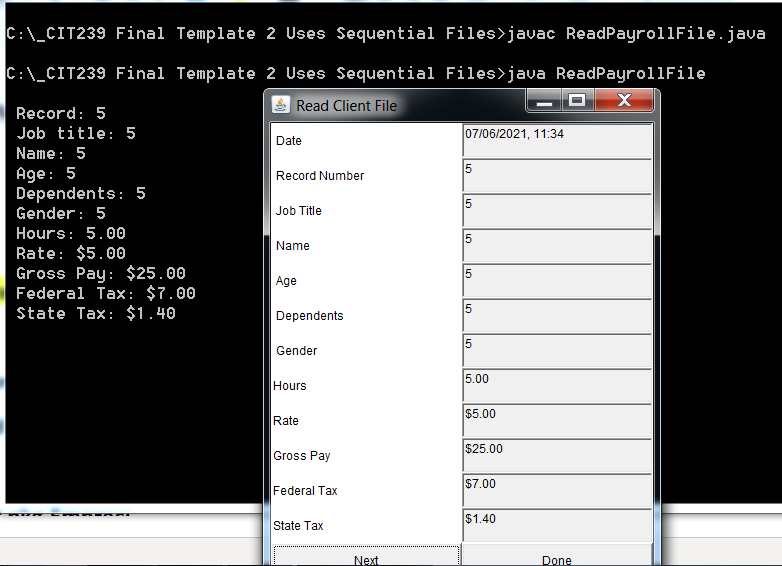
**public static void main( String args[] )**

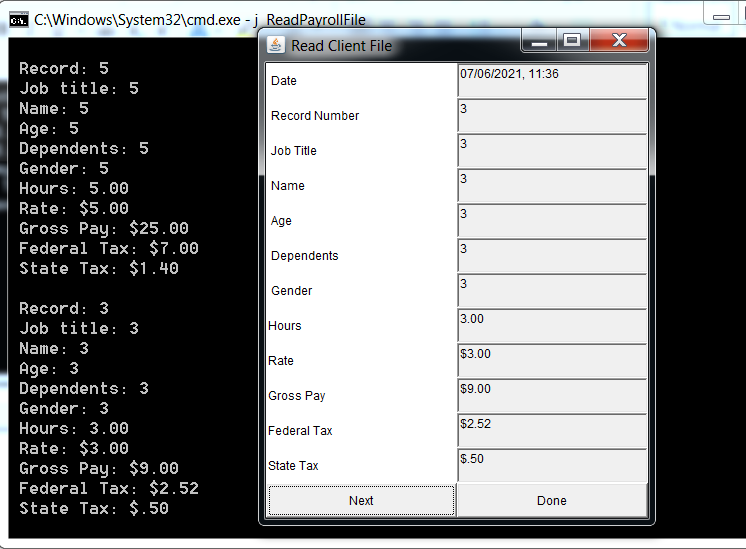
**{**

**new ReadPayrollFile();**

**}**

**}**





C’est Finis !!!!!!