  1 /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
  2 NAME: TONNY MUTUMBA  
  3 ENVIRONMENT: JGRASP  
  4 QUOTE: TO PITY DISTRESS IS BUT HUMAN; TO RETIEVE IT IS GODLIKE.  
  5       BY HORACE MANN.   
  6   
  7 This program Calculate payroll data using classes.  
  8 Employee.java  
  9 EmployeeParameters  
 10 EmployeeParameters.txt    
 11 \*\*\*\*\*\*\*\*/  
 12 import java.util.Scanner;  
 13 import java.text.DecimalFormat;  
 14 import java.io.\*;  
 15 public class Tonny\_Mutumba\_10 {  
 16    static Toolkit tools = new Toolkit();  
 17    static int maxEmployees = 30;  
 18   
 19    public static void main(String [] args) throws IOException{  
 20         
 21       File inputFile = new File("Tonny\_Mutumba\_10\_Input.txt");  
 22       File outputFileName = new File("Tonny\_Mutumba\_10\_Output.txt");  
 23       PrintWriter outputFile = new PrintWriter(outputFileName);  
 24       Scanner input = new Scanner ("inputFile");  
 25   
 26       EmployeeParameters params = new EmployeeParameters();  
 27         
 28       int maxEmployees = params.maxEmployees;  
 29       Employee [] empl = new Employee[maxEmployees];  
 30         
 31       //declarations  
 32       int nRead = 0;  
 33       int results = 0;        
 34         
 35       //getEmployeeParameters();  
 36       params.getEmployeeParameters();  
 37       params.displayEmployeeParameters();  
 38       double savingsRate = params.savingsRate;  
 39       double iraRate = params.iraRate;  
 40       double taxRate = params.federalWithholdingRate   
 41                         + params.stateWithholdingRate;  
 42         
 43       //method call                    
 44       nRead = getInfo(empl, input);                    
 45       printHeading(outputFile);  
 46       calGrossPay(empl, nRead);  
 47       calData(empl, nRead, taxRate, iraRate, savingsRate, params);  
 48       printInfo(empl, nRead, outputFile);  
 49       calTotals(empl, nRead, outputFile);  
 50       System.out.println("\nNumber of rows Read: " + nRead);  
 51       //input.close();  
 52         
 53       //displayInfo("input", empl, nRead);  
 54         
 55       //sorting  
 56       results = tools.selectionSortArrayOfClass(empl, nRead, "Name");  
 57       //displayInfo("Name", empl, nRead);  
 58         
 59       results = tools.selectionSortArrayOfClass(empl, nRead, "Gross Pay");  
 60       //displayInfo("Gross Pay", empl, nRead);  
 61       outputFile.close();  
 62       System.exit(0);  
 63    }  
 64      
 65    //\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
 66    public static int getInfo(Employee [] empl, Scanner input){  
 67       int i = 0;  
 68       //Read input  
 69       while (input.hasNext() && (i < empl.length)){  
 70          empl[i] = new Employee();  
 71          empl[i].hoursWorked = input.nextDouble();  
 72          empl[i].payRate = input.nextDouble();  
 73          empl[i].name = input.nextLine().trim();  
 74          //empl[i].grossPay = empl[i].hoursWorked \* empl[i].payRate;  
 75          i++;  
 76       }  
 77       return i;  
 78    }  
 79    public static void displayInfo(String printOrder, Employee[] empl, int nRead){   
 80       final String dollar = "#,##0.00";  
 81         
 82       System.out.println("\nPrint in " + printOrder.toLowerCase() + "order\n");  
 83         
 84       for(int i = 0; i < nRead; i++){  
 85          System.out.println(tools.padString(empl[i].name, 20) + "" +

tools.leftPad(empl[i].payRate, 10, dollar)  
 86                      + " " + tools.leftPad(empl[i].hoursWorked, 10, dollar)  
 87                      + " " + tools.leftPad(empl[i].grossPay, 10, dollar));  
 88       }  
 89    }  
 90      
 91    public static void printHeading(PrintWriter outputFile){  
 92          String line;  
 93          line = "Name                   Gross Pay          Net Pay         "   
 94                + "Wealth          Taxes           Hours            Pay Rate"  
 95                + "\n\_\_\_\_\_\_\_               \_\_\_\_\_\_\_\_\_\_\_        \_\_\_\_\_\_\_\_        "  
 96                + "\_\_\_\_\_\_\_\_\_      \_\_\_\_\_\_\_\_\_\_      \_\_\_\_\_\_\_\_\_\_        "       
 97                + "\_\_\_\_\_\_\_\_\_\_\_\_";  
 98          outputFile.println(line);  
 99          System.out.println(line);        
100    }  
101   
102    public static void calGrossPay(Employee[]empl, int nRead){  
103       for (int i = 0; i < nRead; i++){  
104          if (empl[i].hoursWorked <= 0.0){  
105             empl[i].grossPay = 0.0;  
106          }  
107          else if (empl[i].hoursWorked <= 40.0){  
108             empl[i].grossPay = empl[i].payRate \* empl[i].hoursWorked;  
109          }  
110          else if (40.0 < empl[i].hoursWorked && empl[i].hoursWorked <= 50.0){  
111             empl[i].grossPay = empl[i].payRate \* (3.0 / 2.0) \*   
112                            empl[i].hoursWorked - 20.0;  
113          }  
114          else if (empl[i].hoursWorked > 50.0){  
115             empl[i].grossPay = empl[i].payRate \* (3.0 / 2.0) \*   
116                            empl[i].hoursWorked - 45.0;  
117          }  
118       }  
119     }    
120     public static void calData(Employee[] empl, int nRead,   
121                double taxRate, double iraRate, double savingsRate,  
122                EmployeeParameters params){  
123        //double sumTaxes = 0.0;  
124        for (int i = 0; i < nRead; i ++){  
125          empl[i].adjustedGrossPay = (empl[i].grossPay - empl[i].iraAmount);  
126          empl[i].iraAmount = (empl[i].adjustedGrossPay \* iraRate) / 100;  
127          empl[i].taxAmount = (empl[i].adjustedGrossPay \* taxRate) / 100;  
128          empl[i].netPay = empl[i].adjustedGrossPay - empl[i].taxAmount;  
129          empl[i].savingsAmount = (empl[i].adjustedGrossPay \* savingsRate) / 100;  
130          empl[i].wealth = empl[i].savingsAmount + empl[i].iraAmount;  
131    
132        }          
133     }  
134     public static void calTotals(Employee [] empl, int nRead,   
135                         PrintWriter outputFile){  
136       double totalGrossPay = 0.0;  
137       double totalNetPay = 0.0;  
138       double totalTaxes = 0.0;  
139       double totalHours = 0.0;  
140       double totalWealth = 0.0;  
141       double totalPayRate = 0.0;  
142       double wealth = 0.0;  
143       double averagePayRate = 0.0;  
144       double average = 0.0;  
145         
146       for(int i = 0; i < nRead; i++){  
147          empl[i].wealth = empl[i].savingsAmount + empl[i].iraAmount;  
148          totalWealth += empl[i].wealth;  
149          totalGrossPay += empl[i].grossPay;  
150          totalNetPay += empl[i].netPay;  
151          totalHours += empl[i].hoursWorked;  
152          totalTaxes += empl[i].taxAmount;  
153          totalPayRate += empl[i].payRate;  
154       }  
155       if (nRead > 0){  
156          average = totalPayRate / nRead;  
157       }  
158       System.out.println("                       "   
159                        + tools.leftPad(totalGrossPay, 5, "0.00") +   
160                 "              "+ tools.leftPad(totalNetPay, 5, "0.00") +

"           "  
161                        + tools.leftPad(totalWealth, 5, "0.00") + "           "  
162                        + tools.leftPad(totalTaxes, 5, "0.00") + "            "  
163                        + tools.leftPad(totalHours, 5, "0.00") + "            "  
164                        + tools.leftPad(totalPayRate, 5, "0.00") +

"                ");   
165     }  
166     public static void printInfo(Employee [] empl, int nRead,   
167                         PrintWriter outputFile){  
168       for(int i = 0; i < nRead; i++){  
169          System.out.println(tools.padString(empl[i].name, 22) + " "   
170                           + tools.leftPad(empl[i].grossPay, 6, "0.00")  
171                           + tools.leftPad(empl[i].netPay, 6, "0.00")  
172                           + tools.leftPad(empl[i].wealth, 6, "0.00")  
173                           + tools.leftPad(empl[i].taxAmount, 6, "0.00")  
174                           + tools.leftPad(empl[i].hoursWorked, 6, "0.00")  
175                           + tools.leftPad(empl[i].payRate, 6, "0.00") );  
176       }                      
177     }  
178   }