Employee Work Hours and Salary Calculation System
Lab Report

Date: 16-12-2024

Course: Object Oriented programming Lab

Instructor: Propa Punam

Student Name: Asma UL Hosna

Student ID: 0432410005101002

Section : 2A1

Batch: 55th

Semester: Autumn 2024

Abstract :

This report describes the development and implementation of an Employee Work Hours and Salary Calculation System designed to calculate the salaries of employees based on their working hours or fixed salary. The system allows for both full-time and part-time employee management through a graphical user interface (GUI), providing an easy-to-use platform for employees to enter their details and receive calculated salary outputs. The application was built using Java programming, utilizing object-oriented principles such as inheritance and polymorphism, and integrated with a GUI for user interaction

Objectives:

- To design and implement an Employee Work Hours System using Java.
- To demonstrate the use of inheritance, polymorphism, and GUI components in Java programming.
- To create a functional application that can calculate the salary of full-time and part-time employees based on input data.
- To enhance user experience with a responsive graphical interface that

Equipment and Components

- **Hardware**:
  - Computer with Java Development Kit (JDK) installed.
  - Basic hardware such as keyboard and mouse for data input.
- **Software**:
  - Java SE Development Kit (JDK) version [JDK 23].
  - Integrated Development Environment (IDE) NetBeans.
  - GUI components from Java Swing library.
- **Libraries/Tools**:
  - Java Swing for graphical user interface components.
  - Java.util package for handling basic operations (e.g., exceptions).
  - Java.awt package for creating graphical components and layout management.

Theory :

The Employee Work Hours System is based on basic object-oriented principles in Java, including inheritance and polymorphism. The design follows the following concepts:

- Inheritance: The FullTimeEmployee and PartTimeEmployee classes inherit from Employee. This allows for code reuse and extends the functionality of the base class by adding specific attributes (e.g., salary for full-time employees and hours and rate for part-time employees).
- Polymorphism: The method calculateSalary() is overridden in both the FullTimeEmployee and PartTimeEmployee classes, allowing each subclass to provide its own implementation for salary calculation.
- GUI Development: Java Swing is used to create the graphical interface, which allows for input from users and displays the calculated salary dynamically.
- **Interface:** which allows for input from users and displays the calculated salary dynamically.

The system uses a simple approach to employee salary calculations. Full-time employees have a fixed salary, while part-time employees are compensated based on the number of hours worked and the hourly rate.

Methodology :

The methodology to implement the Employee Work Hours System is as follows:

1. **Class Design**: Create an Employee base class with attributes name and employeeId. Implement a method calculateSalary() that returns 0.0 in the base class.
2. **Subclass Implementation**: Implement FullTimeEmployee and PartTimeEmployee classes that inherit from Employee. The subclasses should override the calculateSalary() method to return the correct salary based on the employee type.
3. **User Interface Design**: Use Java Swing to design a user interface with input fields for employee name, ID, type (Full-Time/Part-Time), hours worked, and hourly rate (for part-time employees). Add a "Calculate" button to compute the salary.
4. **Salary Calculation**: Implement logic to calculate salaries for both full-time and part-time employees. For full-time employees, a fixed salary is used, while for part-time employees, the salary is calculated based on hours worked and the hourly rate.
5. **Error Handling**: Implement error handling to validate user inputs (e.g., checking for empty fields or invalid numeric values).

CODE: 1

```
package com.mycompany.employeeworkhours;
public class Employee {
    protected String name;
    protected int employeeId;
 public Employee(String name, int employeeId) {
        this.name = name;
        this.employeeId = employeeId;
    }
 public double calculateSalary() {
        return 0.0;
    }
}
```

CODE: 2

```
package com.mycompany.employeeworkhours;
public class FullTimeEmployee extends Employee {
    private double salary;

    public FullTimeEmployee(String name, int
employeeId, double salary) {
        super(name, employeeId);
        this.salary = salary;
    }
@Override
    public double calculateSalary()
return salary;
 }
```

CODE: 3

```
package
com.mycompany.employeeworkhours;
public class PartTimeEmployee extends
Employee {
    private int hoursWorked;
    private double hourlyRate;

    public PartTimeEmployee(String name, int
employeeId, int hoursWorked, double
hourlyRate) {
        super(name, employeeId);
        this.hoursWorked = hoursWorked;
        this.hourlyRate = hourlyRate;
    }

    @Override
    public double calculateSalary() {
        return hoursWorked * hourlyRate;
    }
}
```

Code 4

```
package com.mycompany.employeeworkhours;



import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

public class WelcomePage extends JFrame {

    public WelcomePage() {
        // Frame setup
        setTitle("Welcome to Employee Work Hours System");
        setSize(400, 300);
        setLayout(new BorderLayout());
        setDefaultCloseOperation(EXIT_ON_CLOSE);

        // Set Background Color
        getContentPane().setBackground(new Color(60, 179, 113)); //
Light green background

        // Create a JPanel to hold all the text and button, centered
vertically and horizontally
        JPanel centerPanel = new JPanel();
        centerPanel.setLayout(new BoxLayout(centerPanel,
BoxLayout.Y_AXIS));
        centerPanel.setOpaque(false); // Transparent panel to show
background color

        // Welcome Label
        JLabel lblWelcome = new JLabel("Welcome to Employee Work
Hours System");
        lblWelcome.setFont(new Font("Arial", Font.BOLD, 16));

lblWelcome.setAlignmentX(Component.CENTER_ALIGNMENT);
        lblWelcome.setForeground(Color.WHITE); // White text color
        centerPanel.add(lblWelcome);

        // Spacer
        centerPanel.add(Box.createVerticalStrut(20));

        // Description Label
        JLabel lblDescription = new JLabel("This system helps calculate
employee salaries");
        lblDescription.setFont(new Font("Arial", Font.PLAIN, 16));

lblDescription.setAlignmentX(Component.CENTER_ALIGNMENT);
        lblDescription.setForeground(Color.WHITE); // White text color
        centerPanel.add(lblDescription);

        // Spacer
        centerPanel.add(Box.createVerticalStrut(30));

        // Enter Button
        JButton btnEnter = new JButton("Enter System");
        btnEnter.setBackground(new Color(255, 215, 0)); // Gold button
background
        btnEnter.setForeground(Color.BLACK); // Black button text
        btnEnter.setFocusPainted(false); // Remove focus border
        btnEnter.setAlignmentX(Component.CENTER_ALIGNMENT);
        centerPanel.add(btnEnter);

        // Add Action Listener for Button
        btnEnter.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                // Open MainFrame (salary calculator)
```

Code 5

```
package com.mycompany.employeeworkhours;



import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

public class MainFrame extends JFrame {
    private JLabel lblName, lblId, lblType, lblHours, lblRate,
lblSalary, lblSalaryOutput;
    private JTextField txtName, txtId, txtHours, txtRate;
    private JComboBox<String> cmbType;
    private JButton btnCalculate;

    public MainFrame() {
        // Frame setup
        setTitle("Employee Salary Calculator");
        setSize(400, 350);
        setLayout(null); // Using null layout for manual
positioning
        setDefaultCloseOperation(EXIT_ON_CLOSE);
        getContentPane().setBackground(new Color(128, 0,
128)); // Set background color (purple)

        // Create Labels and Fields
        setupComponents();

        // Action Listener for Calculate Button
        btnCalculate.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                try {
                    String name = txtName.getText();
                    int id = Integer.parseInt(txtId.getText());
                    String type = (String)
cmbType.getSelectedItem();
                    double salary = 0.0;

                    if (type.equals("Full-Time")) {
                        salary = 5000.00; // Fixed salary for full-time
                    } else if (type.equals("Part-Time")) {
                        int hours =
Integer.parseInt(txtHours.getText());
                        double rate =
Double.parseDouble(txtRate.getText());
                        salary = hours * rate;
                    }

                    lblSalaryOutput.setText(String.format("%.2f",
salary)); // Format salary to 2 decimal points
                } catch (NumberFormatException ex) {
                    JOptionPane.showMessageDialog(null, "Please
enter valid numeric values for ID, Hours, and Rate.",
"Input Error", JOptionPane.ERROR_MESSAGE);
                }
            }
        });
    }

    private void setupComponents() {
        // Name
        lblName = new JLabel("Name:");
        lblName.setBounds(20, 20, 100, 25);
        lblName.setFont(new Font("Arial", Font.PLAIN, 12));
        lblName.setForeground(Color.WHITE); // White text
color
```

```
            new MainFrame().setVisible(true);
            dispose(); // Close the welcome page
        }
    });

    // Add the panel to the frame
    add(centerPanel, BorderLayout.CENTER);
  }

  public static void main(String[] args) {
    new WelcomePage().setVisible(true);
  }
}
```

```
    add(lblName);
    txtName = new JTextField();
    txtName.setBounds(120, 20, 200, 25);
    add(txtName);

    // Employee ID
    lblId = new JLabel("Employee ID:");
    lblId.setBounds(20, 60, 100, 25);
    lblId.setFont(new Font("Arial", Font.PLAIN, 12));
    lblId.setForeground(Color.WHITE); // White text color
    add(lblId);
    txtId = new JTextField();
    txtId.setBounds(120, 60, 200, 25);
    add(txtId);

    // Employee Type
    lblType = new JLabel("Type:");
    lblType.setBounds(20, 100, 100, 25);
    lblType.setFont(new Font("Arial", Font.PLAIN, 12));
    lblType.setForeground(Color.WHITE); // White text
color
    add(lblType);
    cmbType = new JComboBox<>(new String[]{"Full-
Time", "Part-Time"});
    cmbType.setBounds(120, 100, 200, 25);
    add(cmbType);

    // Hours Worked
    lblHours = new JLabel("Hours:");
    lblHours.setBounds(20, 140, 100, 25);
    lblHours.setFont(new Font("Arial", Font.PLAIN, 12));
    lblHours.setForeground(Color.WHITE); // White text
color
    add(lblHours);
    txtHours = new JTextField();
    txtHours.setBounds(120, 140, 200, 25);
    add(txtHours);

    // Hourly Rate
    lblRate = new JLabel("Hourly Rate:");
    lblRate.setBounds(20, 180, 100, 25);
    lblRate.setFont(new Font("Arial", Font.PLAIN, 12));
    lblRate.setForeground(Color.WHITE); // White text
color
    add(lblRate);
    txtRate = new JTextField();
    txtRate.setBounds(120, 180, 200, 25);
    add(txtRate);

    // Calculate Button
    btnCalculate = new JButton("Calculate");
    btnCalculate.setBounds(120, 220, 100, 30);
    btnCalculate.setBackground(new Color(255, 215, 0)); //
Gold button background
    btnCalculate.setForeground(Color.BLACK); // Black
button text
    btnCalculate.setFocusPainted(false); // Remove focus
border
    add(btnCalculate);

    // Salary Output
    lblSalary = new JLabel("Salary:");
    lblSalary.setBounds(20, 260, 100, 25);
    lblSalary.setFont(new Font("Arial", Font.PLAIN, 12));
    lblSalary.setForeground(Color.WHITE); // White text
color
    add(lblSalary);
    lblSalaryOutput = new JLabel("0.0");
    lblSalaryOutput.setBounds(120, 260, 200, 25);
```

```
        lblSalaryOutput.setFont(new Font("Arial",
Font.BOLD, 14));
        lblSalaryOutput.setForeground(Color.YELLOW); //
Yellow output text
        add(lblSalaryOutput);
    }

    public static void main(String[] args) {
        new MainFrame().setVisible(true);
    }
}
```

Observations:

During the development of the Employee Work Hours System, the following observations were made:

1. The GUI components worked as expected, allowing for clear input of employee data.
2. The application correctly calculates salaries for both full-time and part-time employees.
3. Error handling mechanisms effectively prevented invalid input from disrupting the program.

Results:

- Functionality: The system successfully calculates salaries for both full-time and part-time employees based on user input.
- Performance: The application runs efficiently without delays, even with multiple salary calculations.
- User Interface: The graphical user interface is intuitive and easy to use, with clear labels and buttons for user interaction.

Discussion and Analysis:

The project meets the primary goal of calculating employee salaries based on their employment type. The use of object-oriented programming principles such as inheritance and polymorphism allowed for an extensible design, where new employee types or salary calculation methods could be added in the future without significant changes to the existing code.

One limitation of the current implementation is that the salary for full-time employees is fixed. A possible enhancement could include adding features like performance bonuses or benefits to the full-time salary calculation**.**

Conclusion :

The Employee Work Hours System was successfully implemented using Java. The project demonstrates key concepts of object-oriented programming, including inheritance and polymorphism. The system is fully functional, providing an easy-to-use interface for calculating employee salaries. Future enhancements could include adding features like employee records management, performance-based salaries, and saving employee data to a file or database.

References:

Mr. Al-Imtiaz

Associate Professor & Head
Ph. D. Fellow, Bangladesh University of Engineering and Technology (BUET)
M.Sc. in CSE, East West University (EWU)
B.Sc in CSE (Gold Medalist), University of Information Technology & Sciences (UITS)

Office :+8809678008487, Ext-403

Cell :+8801715135517

Email :al.imtiaz@uits.edu.bd
head.cse@uits.edu.bd

Propa Punam

Lecturer
B. Sc. (Engg.) in CSE, Khulna University of Engineering & Technology (KUET)

Office :+8809678008487, Ext-403

Cell :+8801641162080

Email :propa.punam@uits.edu.bd