

­­­

OOP Lab-13 Task

Name: Syed Muhammad Raza Ali

Enrolment: 02-134231-028

Course: OOP Lab

Faculty: Miss Hafsa Munawar

|  |
| --- |
| **Lab13: File Handling** |

Designing and implementing Java programs that deal with:

|  |
| --- |
| 1. Reading Input From Text Files 2. Writing Output To Text Files |

|  |
| --- |
| **Exercises** |

Exercise

Create a system that gets following information of the employees

**Employee name, Employee department, Employee contact, Employee designation, Employee Salary, Employee status** and store it in the file names as **EmplyeeData.txt.** Once the data is inserted properly then fetch the data from the file display the data properly. Update record for sec and third employee and reenter the data in new file named as **UpdatedEmployeeInfo.** Ad display the updated data again.

Code:

Employee Class:

class Employee {

private String name;

private String department;

private String contact;

private String designation;

private String salary;

private String status;

//constructor

public Employee(String name, String department, String contact, String designation, String salary, String status) {

this.name = name;

this.department = department;

this.contact = contact;

this.designation = designation;

this.salary = salary;

this.status = status;

}

//getters

public String getContact() {

return contact;

}

public String getDesignation() {

return designation;

}

public String getDepartment() {

return department;

}

public String getName() {

return name;

}

public String getSalary() {

return salary;

}

public String getStatus() {

return status;

}

//setters

public void setContact(String contact) {

this.contact = contact;

}

public void setDepartment(String department) {

this.department = department;

}

public void setDesignation(String designation) {

this.designation = designation;

}

public void setName(String name) {

this.name = name;

}

public void setSalary(String salary) {

this.salary = salary;

}

public void setStatus(String status) {

this.status = status;

}

//toString method

public String toString() {

return name + "," + department + "," + contact + "," + designation + "," + salary + "," + status;

}

}

Application class:

import java.io.BufferedReader;

import java.io.FileReader;

import java.io.FileWriter;

import java.io.IOException;

import java.io.PrintWriter;

import java.util.ArrayList;

import java.util.List;

public class EmployeeManagement {

public static void main(String[] args) {

// Initial employee data

List<Employee> employees = new ArrayList<>();

employees.add(new Employee("Muhammad Raza", "CS", "1234567890", "Manager", "50000", "Active"));

employees.add(new Employee("Muskan Khan", "CS", "9876543210", "Assistant", "40000", "Active"));

employees.add(new Employee("Ali", "IT", "5555555555", "Developer", "60000", "Active"));

// Write employee data to a file

writeToEmployeeFile("EmployeeData.txt", employees);

// Display initial employee data

System.out.println("Initial Employee Data:");

readEmployeeFile("EmployeeData.txt");

// Update records for the second and third employee

employees.get(1).setSalary("45000"); // Update salary for Jane Smith

employees.get(2).setStatus("Inactive"); // Update status for Alice Johnson

// Write updated employee data to a new file

writeToEmployeeFile("UpdatedEmployeeInfo.txt", employees);

// Display updated employee data

System.out.println("\nUpdated Employee Data:");

readEmployeeFile("UpdatedEmployeeInfo.txt");

}

// Function to write employee data to a file

public static void writeToEmployeeFile(String filename, List<Employee> employees) {

try (PrintWriter writer = new PrintWriter(new FileWriter(filename))) {

for (Employee employee : employees) {

writer.println(employee.toString());

}

} catch (IOException e) {

e.printStackTrace();

}

}

// Function to read employee data from a file

public static void readEmployeeFile(String filename) {

try (BufferedReader reader = new BufferedReader(new FileReader(filename))) {

String line;

while ((line = reader.readLine()) != null) {

System.out.println(line);

}

} catch (IOException e) {

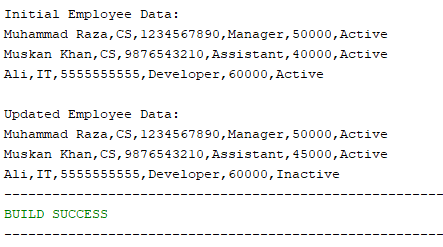
e.printStackTrace();

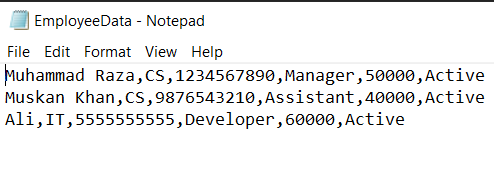
}

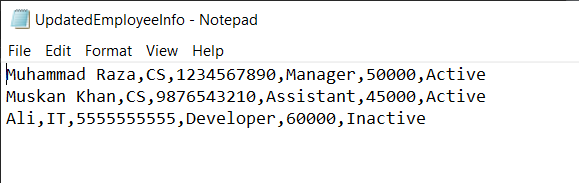
}

}

Output:

On IDE:

In File (Before Updating)

 In File (After Updating)