Employee Management System

Problem Statement

Write a java program to implement “Employee Management System”.

Include at least three from the given topics as per requirement of the program: Inheritance, Overriding Methods, Polymorphism, Abstract Classes, Nested Classes, Interfaces, Lambda Expressions, Exceptional Handling and I/O Fundamentals.

Solution

In this code I have used the concept of Classes and Objects to solve the problem. I have created a menu for the user which consists of 5 different features.

1. To add an employee
2. To view details of an employee
3. To remove an employee
4. To update info of an employee
5. To exit the Portal.

For each of these features I have created a separate class to maintain the decorum of the code. Also I have use the concept of Switch Cases to automate the task and make it more useful and user friendly. User need to just enter the numbers as per given and it will perform the task.

CODE

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Importing Essential Libraries \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

import java.util.\*;

import java.io.\*;

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* MENU OF EMS \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

class MainMenu

{

public void menu()

{

System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

System.out.println("\t EMPLOYEE MANAGEMENT SYSTEM");

System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

System.out.println("\t\t\t--------------------");

System.out.println("\t\t\t ~$ Ritwik Raj");

System.out.println("\t\t\t--------------------");

System.out.println("\n\nPress 1 : To Add an Employee Details");

System.out.println("Press 2 : To See an Employee Details ");

System.out.println("Press 3 : To Remove an Employee");

System.out.println("Press 4 : To Update Employee Details");

System.out.println("Press 5 : To Exit the EMS Portal");

}

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* To add details of Employee \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

class Employee\_Add

{

public void createFile()

{

Scanner sc=new Scanner(System.in);

EmployDetail emp=new EmployDetail();

emp.getInfo();

try{

File f1=new File("file"+emp.employ\_id+".txt");

if(f1.createNewFile()){

FileWriter myWriter = new FileWriter("file"+emp.employ\_id+".txt");

myWriter.write("Employee ID:"+emp.employ\_id+"\n"+"Employee Name :"+emp.name+"\n"+

"Father's Name :"+emp.father\_name+"\n"+"Employee Contact :"+emp.employ\_contact+"\n"+

"Email Information :"+emp.email+"\n"+"Employee position :"+emp.position+"\n"+

"Employee Salary :"+emp.employ\_salary);

myWriter.close();

System.out.println("\nEmployee has been Added :)\n");

System.out.print("\nPress Enter to Continue...");

sc.nextLine();

}

else {

System.out.println("\nEmployee already exists :(");

System.out.print("\nPress Enter to Continue...");

sc.nextLine();

}

}

catch(Exception e){System.out.println(e);}

}

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Taking Employee Details \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

class EmployDetail

{

String name;

String father\_name;

String email;

String position;

String employ\_id;

String employ\_salary;

String employ\_contact;

public void getInfo()

{

Scanner sc=new Scanner(System.in);

System.out.print("Enter Employee's name --------: ");

name=sc.nextLine();

System.out.print("Enter Employee's Father name -: ");

father\_name=sc.nextLine();

System.out.print("Enter Employee's ID ----------: ");

employ\_id=sc.nextLine();

System.out.print("Enter Employee's Email ID ----: ");

email=sc.nextLine();

System.out.print("Enter Employee's Position ----: ");

position=sc.nextLine();

System.out.print("Enter Employee contact Info --: ");

employ\_contact=sc.nextLine();

System.out.print("Enter Employee's Salary ------: ");

employ\_salary=sc.nextLine();

}

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* To Show details of Employee \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

class Employee\_Show

{

public void viewFile(String s) throws Exception

{

File file = new File("file"+s+".txt");

Scanner sc = new Scanner(file);

while (sc.hasNextLine())

{

System.out.println(sc.nextLine());

}

}

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* To Remove Employee \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

class Employee\_Remove

{

public void removeFile(String ID)

{

File file = new File("file"+ID+".txt");

if(file.exists())

{

if(file.delete());

{

System.out.println("\nEmployee has been removed Successfully");

}

}

else

{

System.out.println("\nEmployee does not exists :( ");

}

}

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* To Update details of Employee \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

class Employee\_Update

{

public void updateFile(String s,String o,String n) throws IOException

{

File file = new File("file"+s+".txt");

Scanner sc = new Scanner(file);

String fileContext="";

while (sc.hasNextLine())

{

fileContext =fileContext+"\n"+sc.nextLine();

}

FileWriter myWriter = new FileWriter("file"+s+".txt");

fileContext = fileContext.replaceAll(o,n);

myWriter.write(fileContext);

myWriter.close();

}

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* To Exit from the EMS Portal \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

class CodeExit

{

public void out()

{

System.out.println("\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

System.out.println("$ cat Thank You For Using my Software :) ");

System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

System.out.println("\t\t/~ <0d3d by Ritwik Raj\n");

System.exit(0);

}

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Main Class \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

class EmployManagementSystem

{

public static void main(String arv[])

{

/\*\* To clear the output Screen \*\*/

System.out.print("\033[H\033[2J");

Scanner sc=new Scanner(System.in);

Employee\_Show epv =new Employee\_Show();

int i=0;

/\*\*\* Callining Main menu Class function \*\*\*\*/

MainMenu obj1 = new MainMenu();

obj1.menu();

/\*\*\* Initialising loop for Menu Choices \*\*\*/

while(i<6)

{

System.out.print("\nPlease Enter choice :");

i=Integer.parseInt(sc.nextLine());

/\*\* Switch Case Statements \*\*/

switch(i)

{

case 1:

{

/\*\* Creating class's object and calling Function using that object \*\*/

Employee\_Add ep =new Employee\_Add();

ep.createFile();

System.out.print("\033[H\033[2J");

obj1.menu();

break;

}

case 2:

{

System.out.print("\nPlease Enter Employee's ID :");

String s=sc.nextLine();

try

{

epv.viewFile(s);}

catch(Exception e){System.out.println(e);}

System.out.print("\nPress Enter to Continue...");

sc.nextLine();

System.out.print("\033[H\033[2J");

obj1.menu();

break;

}

case 3:

{

System.out.print("\nPlease Enter Employee's ID :");

String s=sc.nextLine();

Employee\_Remove epr =new Employee\_Remove();

epr.removeFile(s);

System.out.print("\nPress Enter to Continue...");

sc.nextLine();

System.out.print("\033[H\033[2J");

obj1.menu();

break;

}

case 4:

{

System.out.print("\nPlease Enter Employee's ID :");

String I=sc.nextLine();

try

{

epv.viewFile(I);

}

catch(Exception e)

{

System.out.println(e);

}

Employee\_Update epu = new Employee\_Update();

System.out.print("Please Enter the detail you want to Update :");

System.out.print("\nFor Example :\n");

System.out.println("If you want to Change the Name, then Enter Current Name and Press Enter. Then write the new Name then Press Enter. It will Update the Name.\n");

String s=sc.nextLine();

System.out.print("Please Enter the Updated Info :");

String n=sc.nextLine();

try

{

epu.updateFile(I,s,n);

System.out.print("\nPress Enter to Continue...");

sc.nextLine();

System.out.print("\033[H\033[2J");

obj1.menu();

break;

}

catch(IOException e)

{

System.out.println(e);

}

}

case 5:

{

CodeExit obj = new CodeExit();

obj.out();

}

}

}

}

}

A screenshot of a cell phone

Description automatically generatedOUTPUT SCREENSHOT