**TASK DESCRIPTION**

This is a program of a staff details program that reads data from text file and displays according to their job duty. This program allows user to search for existing staff by full time staff, part time staff, and casual staff and also can display all staff members.

1. To perform this program, two class files are created. One file being abstract staff class file having fields’ staff name, staff id and staff position and has sub classes for full-time staff, part-time staff and casual staff. Methods like get and set methods are defined for all three fields including name, id and position. Another method to string (displays string representation of staff details) is defined.
2. Then, subclasses for abstract class has been created namely full-time staff, part-time staff and casual staff. Full-time staff class has get and set method for research area and to string method for its string representation. Part-time staff class has get and set method for time fraction and to string method for its string representation. Casual staff class has get and set method for hours and to string method for its string representation.
3. Another class called as test student has been created to follow the other requirements of program by creating different objects of subclasses of abstract class staff.
4. All the existing data is collected from text file and stored in array list according to staff duties by calling respective staff objects.
5. Then system displays what user wants to do out of search for staff with full time duty, staff with part time duty, staff with casual job, all staff member details and quit option.
6. If user chooses 1 option then it allows user to display staff member details with full time duty.
7. If user chooses 2 option then it allows user to display staff member details with part time duty.
8. If user chooses 3 option then it allows user to display staff member details with casual duty.
9. If user chooses 4 option then it allows user to display all staff member details.
10. If user chooses 5 option or typed other than listed options then programs ends.

**TASK OUTPUT**

|  |  |
| --- | --- |
| **Test Data** | **Screenshot** |
| User choose option 1  Displays all the staff details with full time duty |  |
| User choose option 2  Displays all the staff details with part time duty |  |
| User choose option 3  Displays all the staff details with casual duty |  |
| User choose option 4  Displays all the staff details |  |
| User choose option 5  Quits the program |  |

**TASK2 – CODE**

**Staff Class**

**public** **abstract** **class** Staff {

// declaring data fields to store staff name, staff id and position

**protected** String staffName;

**protected** String staffId;

**protected** String position;

// declaring no-arg/default constructor

**public** Staff()

{

}

// creating get method to get staff name

**public** String getName()

{

**return** staffName;

}

// creating set method to change staff name

**public** **void** setName(String newStaffName)

{

staffName = newStaffName;

}

// creating get method to get staff id

**public** String getId()

{

**return** staffId;

}

// creating set method to change staff id

**public** **void** setId(String newId)

{

staffId = newId;

}

// creating get method to get position

**public** String getPosition()

{

**return** position;

}

// creating set method to change position

**public** **void** setPosition(String newPosition)

{

position = newPosition;

}

// creating a method to represents string format of new staff, staff id and staff position

**public** String toString()

{

String newStaffId = getId();

String newStaffName = getName();

String newPosition = getPosition();

**return** "Staff id is " + newStaffId + ". Staff name is " + newStaffName + ". Staff position is " + newPosition + ".";

}

}

**class** FullTimeStaff **extends** Staff

{

**protected** String researchArea;

// declaring no-arg/default constructor

**public** FullTimeStaff()

{

}

// creating get method to get research area

**public** String getResearchArea()

{

**return** researchArea;

}

// creating set method to change research area

**public** **void** setResearchArea(String newResearchArea)

{

researchArea = newResearchArea;

}

// creating a method to represents string format of research area

**public** String toString()

{

String newResearchArea = getResearchArea();

**return** "Staff duty is full time. " + **super**.toString() + " Research area is " + newResearchArea + ".";

}

}

**class** PartTimeStaff **extends** Staff

{

**private** String timeFraction;

// declaring no-arg/default constructor

**public** PartTimeStaff()

{

}

// creating get method to get time fraction

**public** String getTimeFraction()

{

**return** timeFraction;

}

// creating set method to change time fraction

**public** **void** setTimeFraction(String newTimeFraction)

{

timeFraction = newTimeFraction;

}

// creating a method to represents string format of time fraction

**public** String toString()

{

String newTimeFraction = getTimeFraction();

**return** "Staff duty is part time. " + **super**.toString() + " Time fraction is " + newTimeFraction + ".";

}

}

**class** CasualStaff **extends** Staff

{

**protected** String hours;

// declaring no-arg/default constructor

**public** CasualStaff()

{

}

// creating get method to get hours

**public** String getHours()

{

**return** hours;

}

// creating set method to change hours

**public** **void** setHours(String newHours)

{

hours = newHours;

}

// creating a method to represents string format of hours

**public** String toString()

{

String newHours = getHours();

**return** "Staff duty is casual. " + **super**.toString() + " Hours are " + newHours + ".";

}

}

**TestStaff Class**

**import** java.io.File;

**import** java.io.FileNotFoundException;

**import** java.util.ArrayList;

**import** java.util.Scanner;

**public** **class** TestStaff {

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

{

// creating objects for all class

FullTimeStaff fullTimeStaff = **new** FullTimeStaff();

PartTimeStaff partTimeStaff = **new** PartTimeStaff();

CasualStaff casualStaff = **new** CasualStaff();

// creating arraylist to store details of all staff members

ArrayList<String> staffRefList = **new** ArrayList<String>();

// calling scanner method to get input from user

Scanner userInput = **new** Scanner(System.***in***);

// reading text file

File staffFile = **new** File("staff.txt");

// creating scanner object which will be used later to read lines in text file

Scanner getStaffDetails = **new** Scanner(staffFile);

// using while loop to read data from text file line by line

**while**(getStaffDetails.hasNext())

{

// reading line from text file and separating data with space

String[] staffDetails = ((getStaffDetails.nextLine()).trim()).split("\\s+");

// storing first part in each row as staf's duty

String staffDuty = staffDetails[0];

// storing second part in each row as staf's name

String staffName = staffDetails[1];

// storing third part in each row as staf's id

String staffId = staffDetails[2];

// storing forth part in each row as staf's position

String staffPosition = staffDetails[3];

// storing fifth part in each row as staf's other details like working hours, time fraction and research area

String staffOtherDetails = staffDetails[4];

//using if else to see if staff is on which duty out of full, part or casual

**if**(staffDuty.equals("casual"))

{

// if staff is in casual duty then object of casualStaff class is used

// setting name, id, position by calling methods from abstract class Staff using object of casualStaff class

casualStaff.setName(staffName);

casualStaff.setId(staffId);

casualStaff.setPosition(staffPosition);

// setting hours by calling setHours method using object of casualStaff class

casualStaff.setHours(staffOtherDetails);

// adding details to array list of data with format specified for casualStaff by calling toString() method of casualStaff class

staffRefList.add(casualStaff.toString());

}

**else** **if**(staffDuty.equals("full"))

{

// if staff is in full time duty then object of fullTimeStaff class is used

// setting name, id, position by calling methods from abstract class Staff using object of fullTimeStaff class

fullTimeStaff.setName(staffName);

fullTimeStaff.setId(staffId);

fullTimeStaff.setPosition(staffPosition);

// setting research area by calling setResearchArea method using object of fullTimeStaff class

fullTimeStaff.setResearchArea(staffOtherDetails);

// adding details to array list of data with format specified for fullTimeStaff by calling toString() method of fullTimeStaff class

staffRefList.add(fullTimeStaff.toString());

}

**else** **if**(staffDuty.equals("part"))

{

// if staff is in full time duty then object of partTimeStaff class is used

// setting name, id, position by calling methods from abstract class Staff using object of partTimeStaff class

partTimeStaff.setName(staffName);

partTimeStaff.setId(staffId);

partTimeStaff.setPosition(staffPosition);

// setting time fraction by calling setTimeFraction method using object of partTimeStaff class

partTimeStaff.setTimeFraction(staffOtherDetails);

// adding details to array list of data with format specified for partTimeStaff by calling toString() method of partTimeStaff class

staffRefList.add(partTimeStaff.toString());

}

}

**boolean** loop = **true**;

// using while loop to start the program and asks user options until user quit the program manually

**while**(loop)

{

// starts the program by displaying options to user

System.***out***.print("Welcome to staff details program..! \n");

System.***out***.print("1: Display details of only full-time staff \n2: Display details of only part-time staff \n3: Display details of only casual staff \n4: Display details of only all staff \n5: Quit \nChoose option from above: ");

// stores user input in variable

String option = userInput.nextLine();

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

// checking if user chose option 1, that is to display staff with full time duty

**if**(option.equals("1"))

{

String searchCriteria = "full";

System.***out***.println("Following are staff member with full time duty.");

// using for loop to get all data from array list if duty is full time

**for**(String str:staffRefList)

{

**if**(str.contains(searchCriteria))

{

// if found, displays the staff details with full time duty

System.***out***.println(str);

}

**else**

{

**continue**;

}

}

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

}

// checking if user chose option 2, that is to display staff with part time duty

**else** **if**(option.equals("2"))

{

String searchCriteria = "part";

System.***out***.println("Following are staff member with part time duty.");

// using for loop to get all data from array list if duty is part time

**for**(String str:staffRefList)

{

**if**(str.contains(searchCriteria))

{

// if found, displays the staff details with part time duty

System.***out***.println(str);

}

**else**

{

**continue**;

}

}

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

}

// checking if user chose option 3, that is to display staff with casual duty

**else** **if**(option.equals("3"))

{

String searchCriteria = "casual";

System.***out***.println("Following are staff member with casual duty.");

// using for loop to get all data from array list if duty is casual

**for**(String str:staffRefList)

{

**if**(str.contains(searchCriteria))

{

// if found, displays the staff details with casual duty

System.***out***.println(str);

}

**else**

{

**continue**;

}

}

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

}

// checking if user chose option 4, that is to display all the staff members with full details

**else** **if**(option.equals("4"))

{

// displays staff list by using for loop

System.***out***.println("Following are details of all staff members.");

**for**(String str: staffRefList)

{

System.***out***.println(str);

}

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

}

// program ends if user chose option 5 or typed any option other than listed

**else**

{

System.***out***.println("Thank you. Program ends here.");

loop = **false**;

**break**;

}

}

}

}