// COSC 1320 Summer 2015

// Name: Adrian

// Programming Assignment 1

// This is my own work; I will not post

package programmingassignment1;

import java.util.Scanner;

import java.io.PrintWriter;

import java.io.FileOutputStream;

import java.io.FileInputStream;

import java.io.FileNotFoundException;

//import java.util.Arrays;

public class Employee {

private static final int MAX\_SIZE = 20;

private static Employee[] empArr = new Employee[MAX\_SIZE];

private static int employeeCtr = 0;

private String name;

private int idNumber;

private char role;

Employee(){

role = 'h';

name = "";

idNumber = 0;

}

Employee(char newRole , String newName , int newID){

role = newRole;

name = newName;

idNumber = newID;

}

public static void addEmployee(Employee obj){

empArr[employeeCtr] = obj;

System.out.println(obj.toString());

employeeCtr++;

}

public static void deleteEmployee(String empName , char empRole){

int h = 0;

for(int i = 0; i < employeeCtr; i++){

Employee curObj = empArr[i];

empArr[i-h] = curObj;

if(empName.equals(curObj.getName()) && curObj.getRole() == empRole){

h = 1;

System.out.println("Delete match");

}

if(h == 1 && i == employeeCtr-1){

employeeCtr--;

System.out.println("hit");

}

if(h == 0 && i == employeeCtr-1){

System.out.println("Employee not found. " + empName + " " + empRole);

}

}

}

public static void displayEmployees(boolean t){

char[] roleArr = {'h' , 'd' , 's' , 'n' , 'a' , 'r' , 'j'};

String[] classArr = {"Hospital Employees" , "Doctors" , "Surgeons" , "Nurses" , "Administrators" , " Receptionists" , " Janitors" };

System.out.println("The hospital has the following employees:");

for(int i = 0; i < roleArr.length; i++){

System.out.println(classArr[i]);

for(int j = 0; j < employeeCtr; j++){

if(empArr[j].getRole() == roleArr[i]){

if(t){

System.out.println(empArr[j].toString(true));

}else{

System.out.println(empArr[j].toString());

}

}

}

}

}

public static void readFile(String file){

Scanner fin = null;

try{

//Attempt to open the file

fin = new Scanner(new FileInputStream(file));

while(fin.hasNextLine()){

if(fin.hasNext()){

char newRole = fin.next().charAt(0);

String newName = fin.next();

int newID = fin.nextInt();

String newSpec;

String newDept;

int ptCnt;

char working;

// reads role and calls corresponding class constructor after reading appropriate data

switch(newRole){

case 'h':

//System.out.println(fin.nextInt());

HospitalEmployee hosEmp = new HospitalEmployee(newRole , newName , newID);

break;

case 'd':

newSpec = fin.next();

Doctor doc = new Doctor(newRole , newName , newID , newSpec);

break;

case 's':

newSpec = fin.next();

working = fin.next().charAt(0);

Surgeon surge = new Surgeon(newRole , newName , newID , newSpec , working);

break;

case 'n':

ptCnt = fin.nextInt();

Nurse nurse = new Nurse(newRole , newName , newID , ptCnt);

break;

case 'a':

newDept = fin.next();

Administrator admin = new Administrator(newRole , newName , newID , newDept);

break;

case 'r':

newDept = fin.next();

working = fin.next().charAt(0);

Receptionist rec = new Receptionist(newRole , newName , newID , newDept , working);

break;

case 'j':

newDept = fin.next();

working = fin.next().charAt(0);

Janitor jan = new Janitor(newRole , newName , newID , newDept , working);

break;

default:

System.out.println("Role not found");

break;

}

}

}

fin.close();

} catch(FileNotFoundException e){

//Executed if the file is not found

System.out.println("File not found.");

System.exit(0);

}

}

public static void writeFile(String file){

PrintWriter fout = null;

try{

//Attempt to open the file

fout = new PrintWriter(new FileOutputStream(file));

System.out.println("Start writing");

for(int i = 0; i < employeeCtr; i++){

//System.out.println(i + ": " + empArr[i].toString());

fout.println(empArr[i].toString(true));

//fout.println("halt");

}

} catch(FileNotFoundException e){

//Executed if the file is not found

System.out.println("File not found.");

System.exit(0);

}

fout.close();

}

public char getRole(){

return role;

}

public String getName(){

return name;

}

public int getID(){

return idNumber;

}

public void setRole(char newRole){

role = newRole;

}

public void setName(String newName){

name = newName;

}

public void setID(int newID){

idNumber = newID;

}

//@Override

public String toString(boolean t){

if(t){

return ("Name: " + this.getName() + " Employee Number: " + this.getID());

}else{

return (this.getRole() + " " + this.getName() + " " + this.getID());

}

}

public boolean equals(Employee obj){

if(obj == null){

return false;

}else if(getClass() != obj.getClass()){

return false;

}else{

Employee otherEmp = (Employee)obj;

return (name.equals(otherEmp.getName()) && role == otherEmp.getRole());

}

}

}