BBM102 Programming Assignment 2
Özge Kökyay

April 2020

1   Problem Definition

I was expected to develop a system to monitor payroll of Personnel in a University by using the inheritance mechanism that is an important property of object-oriented programming. I mostly use the concept of inheritance and polymorphism, relationships among classes by using object references, control of multiple instances of classes in Java.

2   Solution Approach

The informations of the personnel and the working hours are executed and txt files for every personnel are created in PersonnelManager class.

I created Personnel class as a base class to represent sub-classes of every personnel type, then created the fields and methods to calculate the salaries by generating super-class and sub-class relationships between personnel classes.

I derived FacultyMember and Officer classes from ResearchAssistant class and Chief class from Worker class.

3 UML Diagram

Main `<<Java Class>>` (assignment2)
- Main()
- main(String[]):void

PersonnelManager `<<Java Class>>` (assignment2)
- personnel: String[]
- monitor: String[]
- PersonnelManager(String,String)
- readFile(String):String[]
- personnelInformation():void
- getInformation():void
- writeFile(String,String):void

HoursForWeek `<<Java Class>>` (assignment2)
- registerNumber: String
- week1: int
- week2: int
- week3: int
- week4: int
- HoursForWeek(String,int,int,int,int)
- getRegister():String
- setRegisterNumber(String):void
- getWeek1():int
- setWeek1(int):void
- getWeek2():int
- setWeek2(int):void
- getWeek3():int
- setWeek3(int):void
- getWeek4():int
- setWeek4(int):void

Personnel `<<Java Class>>` (assignment2)
- name: String
- registerNumber: String
- position: String
- yearOfStart: int
- Personnel(String,String,String,int,HoursForWeek)
- earnings():int
- getSeverancePay():int
- getName():String
- getSurname():String
- getRegisterNumber():String
- getPosition():String
- getYearOfStart():int
- toString():String

+information 0..*
+hours 0..*
+hours 0..1

PartTimeEmployee `<<Java Class>>` (assignment2)
- hourPay: int
- PartTimeEmployee(String,String,String,int,HoursForWeek)
- hourOfWork():int
- earnings():int
- toString():String

ResearchAssistant `<<Java Class>>` (assignment2)
- baseSalary: int
- ssBenefits: int
- ResearchAssistant(String,String,String,int,HoursForWeek)
- getBaseSalary():int
- getSsBenefits():int
- earnings():int
- toString():String

Worker `<<Java Class>>` (assignment2)
- dayOfWork: int
- dayPay: int
- overWorkPay: int
- maxOverWork: int
- Worker(String,String,String,int,HoursForWeek)
- overWorkSalary():int
- salary():int
- earnings():int
- toString():String
- getMaxOverWork():int
- getDayOfWork():int
- getDayPay():int
- getOverWorkPay():int

Security `<<Java Class>>` (assignment2)
- foodMoney: int
- transMoney: int
- hourPay: int
- dayWork: int
- Security(String,String,String,int,HoursForWeek)
- hourOfWork():int
- earnings():int
- toString():String

Officer `<<Java Class>>` (assignment2)
- overWorkHour: int
- ssBenefits: int
- Officer(String,String,String,int,HoursForWeek)
- getSsBenefits():int
- overWorkSalary():int
- earnings():int

FacultyMember `<<Java Class>>` (assignment2)
- ssBenefits: int
- FacultyMember(String,String,String,int,HoursForWeek)
- getSsBenefits():int
- addCourseFee():int
- earnings():int

Chief `<<Java Class>>` (assignment2)
- dayPay: int
- overWorkPay: int
- maxOverWork: int
- Chief(String,String,String,int,HoursForWeek)
- getDayPay():int
- getMaxOverWork():int
- getOverWorkPay():int