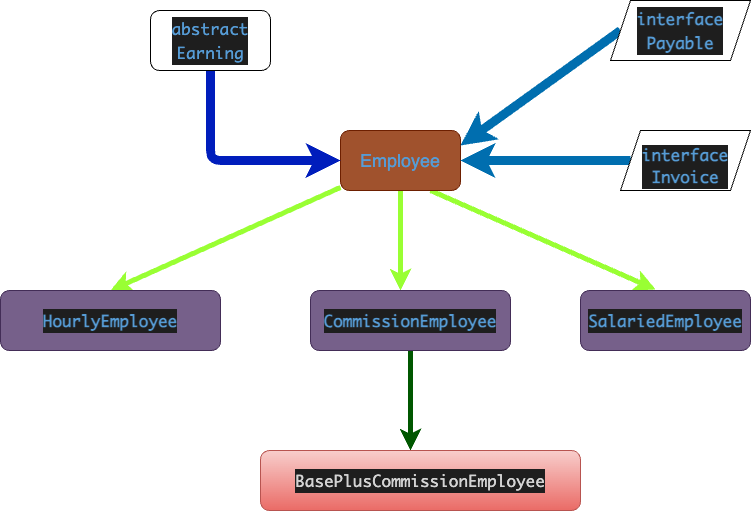
114年智慧行動商務系統開發班第1期

行動裝置APP程式設計學習評量

|  |  |  |  |
| --- | --- | --- | --- |
| **評量編號** | 2-6 | | |
| **評量主題** | 物件導向程式設計 | **實施時間** | 2025/5/5 |
| **評量形式** | 作業紀錄 | **評量日期** | 2025/5/5 |
| **評量目的** | 評定學員是否能以Java程式語言撰寫相關程式碼，並能實作Java類別、繼承、界面、多型、抽象方法等物件導向技術。 | | |
| **評量說明** | **任務一：**基本物件導向類別(Class)及界面(Interface)設計，包含屬性(Property)、方法(Method)等類別成員實作，並具備基礎錯誤參數檢查的建構子(Constructor)與服務方法功能。  **任務二：**設計一個類別階層，其架構按題目要求進行程式撰寫，並進行繼承(Inherit)、介面(Interface)等物件導向特性之實作，且具備基礎錯誤參數檢查的建構式與服務方法功能。  **任務三：**設計一個具備程式進入點Main()功能的類別，利用題目指定的內容實作多型(Polymorphism)，處理各種子類別。 | | |
| **提交要求** | 繳交時請以FTP上傳**相關程式檔案**與**作業文件檔**各一份，**相關程式檔案**請另放置在「完成結果檔」資料夾內，最外層資料夾名稱請用**兩碼學號與姓名**，例如1號朱OO，資料夾請命名為「01朱OO」。 | | |

座號：24 姓名：鍾惟安

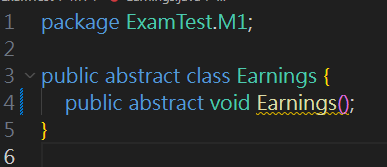
**系統架構圖**



**任務一：設計一個抽象(abstract) 類別「Earning」：**

1. 需要建立一個抽象(abstract)方法為「earnings」

結果畫面截圖：



程式碼：

package ExamTest.M1;

public abstract class Earnings {

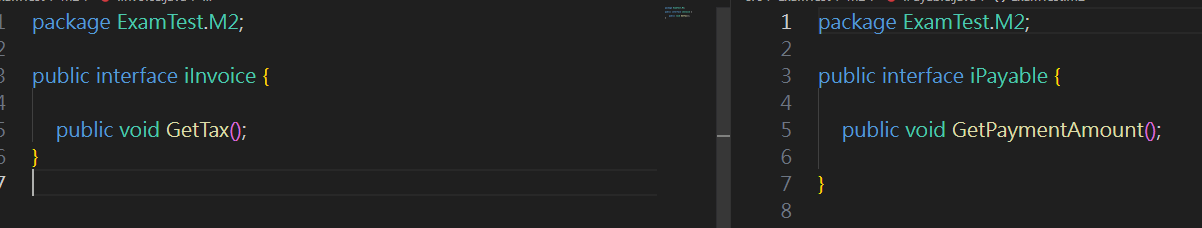
    public abstract void Earnings();

}

**任務二：設計界面(interface)「Payable」、「Invoice」：**

1. 需要為「Payable」建立「getPaymentAmount」方法
2. 需要為「Invoice」建立「getTax」方法

結果畫面截圖：



程式碼：

package ExamTest.M2;

public interface iInvoice {

    public void GetTax();

}

package ExamTest.M2;

public interface iPayable {

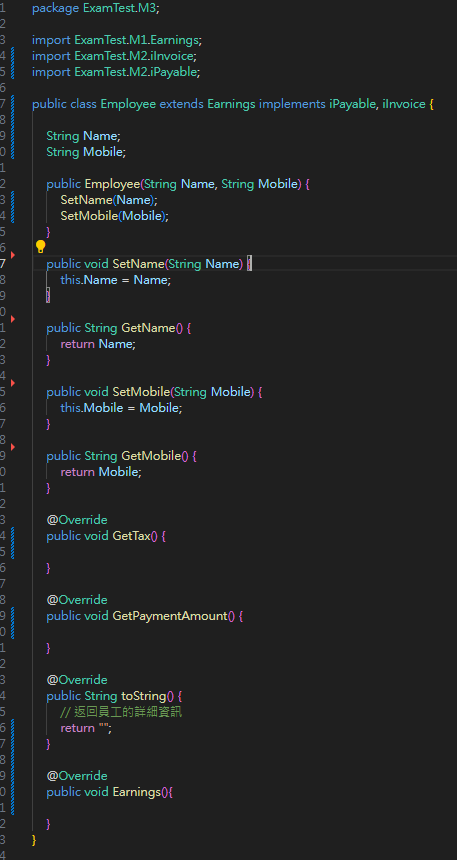
    public void GetPaymentAmount();

}

**任務三：設計一個類別「Employee」繼承 「Earning」，引用「Payable」、「Invoice」界面**

1. 需要建立「name」、「mobile」屬性以及屬性的取得、設定方法
2. 需要建立「Employee」方法，代表類別建構式，且可初始化設定「name」、「mobile」
3. 需要覆寫「toString」方法
4. 需要覆寫「getTax」方法
5. 需要覆寫「earnings」方法
6. 需要覆寫「getPaymentAmount」方**法**

結果畫面截圖：



程式碼：

package ExamTest.M3;

import ExamTest.M1.Earnings;

import ExamTest.M2.iInvoice;

import ExamTest.M2.iPayable;

public class Employee extends Earnings implements iPayable, iInvoice {

    String Name;

    String Mobile;

    public Employee(String Name, String Mobile) {

        SetName(Name);

        SetMobile(Mobile);

    }

    public void SetName(String Name) {

        this.Name = Name;

    }

    public String GetName() {

        return Name;

    }

    public void SetMobile(String Mobile) {

        this.Mobile = Mobile;

    }

    public String GetMobile() {

        return Mobile;

    }

    @Override

    public void GetTax() {

    }

    @Override

    public void GetPaymentAmount() {

    }

    @Override

    public String toString() {

        // 返回員工的詳細資訊

        return "";

    }

    @Override

    public void Earnings(){

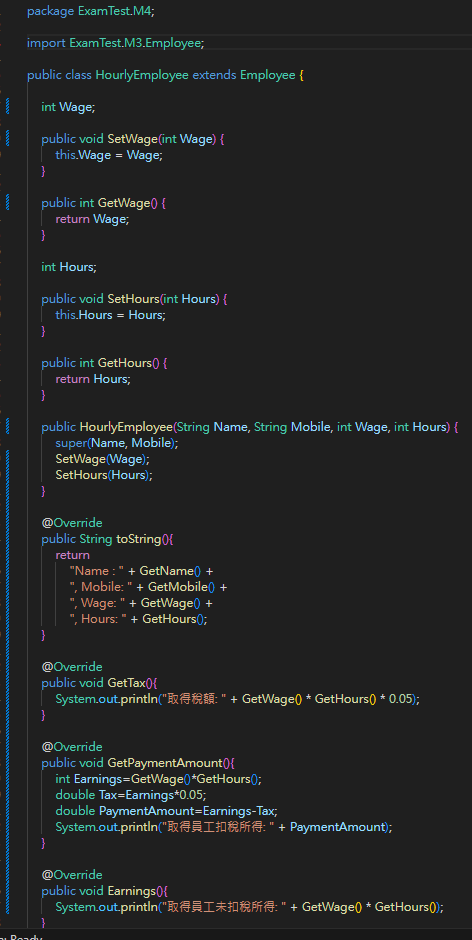
    }

}

**任務四：設計一個類別「HourlyEmployee」，繼承「Employee」：**

1. 需要建立「wage」 、「hours」屬性以及屬性的取得、設定服務方法
2. 需要建立「HourlyEmployee」方法，代表類別建構式 ，且可初始化設定「name」、「mobile」、「wage」、「hours」
3. 請覆寫「toString」方法，提供顯示時薪員工各項基本資料「name」、「mobile」的功能
4. 需要覆寫「getTax」方法，代表取得稅額的功能(wage \* hours \* 0.05)
5. 需要覆寫「earnings」方法，代表取得員工未扣稅所得(wage \* hours)
6. 需要覆寫「getPaymentAmount」方法，代表取得員工扣稅所得功能(wage \* hours) - (wage \* hours \* 0.05)
7. 各種屬性實體變數於操作屬性寫入時需實作數値合理範圍的檢査功能

結果畫面截圖：



程式碼：

package ExamTest.M4;

import ExamTest.M3.Employee;

public class HourlyEmployee extends Employee {

    int Wage;

    public void SetWage(int Wage) {

        this.Wage = Wage;

    }

    public int GetWage() {

        return Wage;

    }

    int Hours;

    public void SetHours(int Hours) {

        this.Hours = Hours;

    }

    public int GetHours() {

        return Hours;

    }

    public HourlyEmployee(String Name, String Mobile, int Wage, int Hours) {

        super(Name, Mobile);

        SetWage(Wage);

        SetHours(Hours);

    }

    @Override

    public String toString(){

        return

            "Name : " + GetName() +

            ", Mobile: " + GetMobile() +

            ", Wage: " + GetWage() +

            ", Hours: " + GetHours();

    }

    @Override

    public void GetTax(){

        System.out.println("取得稅額: " + GetWage() \* GetHours() \* 0.05);

    }

    @Override

    public void GetPaymentAmount(){

        int Earnings=GetWage()\*GetHours();

        double Tax=Earnings\*0.05;

        double PaymentAmount=Earnings-Tax;

        System.out.println("取得員工扣稅所得: " + PaymentAmount);

    }

    @Override

    public void Earnings(){

        System.out.println("取得員工未扣稅所得: " + GetWage() \* GetHours());

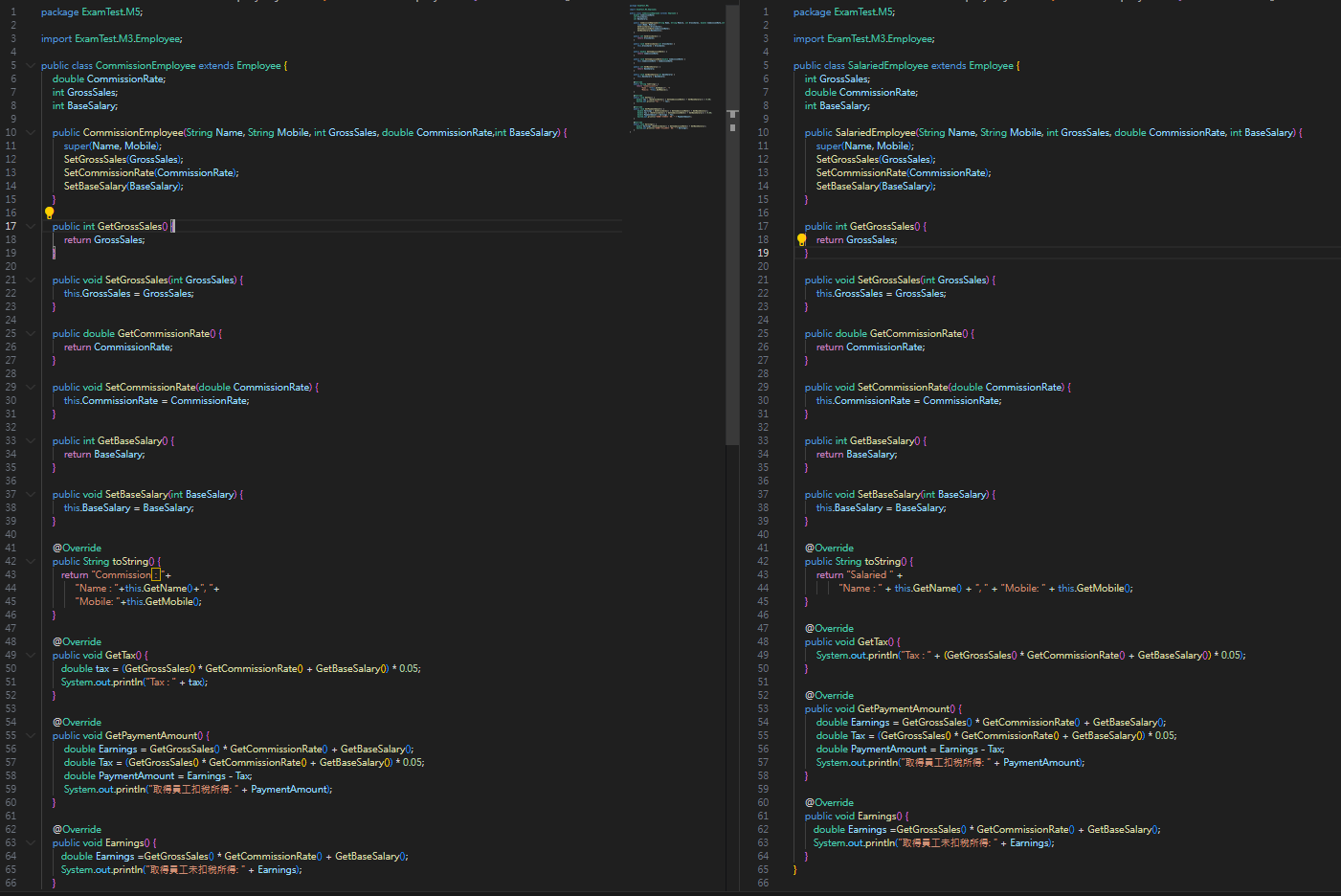
    }

}

**任務五：設計一個類別階層，分別為「CommissionEmployee」、「SalariedEmployee」2個類別，繼承「Employee」：**

1. 需要分別建立 「grossSales」、「commissionRate」、「baseSalary」屬性，以及屬性的取得、設定方法
2. 需要分別建立 「CommissionEmployee」、「SalariedEmployee」方法，代表類別建構式，且可初始化設定「name」、「mobile」、「grossSales」、「commissionRate」、「baseSalary」
3. 需要分別為 「CommissionEmployee」、「SalariedEmployee」覆寫「toString」方法，提供顯示時薪員工各項基本資料「name」、「mobile」的功能
4. 需要分別為 「CommissionEmployee」、「SalariedEmployee」覆寫「getTax」方法，代表取得稅額的功能((grossSales \* commissionRate + baseSalary) \* 0.05)
5. 需要分別為 「CommissionEmployee」、「SalariedEmployee」覆寫「earnings」方法，代表取得員工未扣稅所得(grossSales \* commissionRate + baseSalary)
6. 需要分別為 「CommissionEmployee」、「SalariedEmployee」覆寫「getPaymentAmount」方法，代表取得員工扣稅所得功能(grossSales \* commissionRate + baseSalary) - ((grossSales \* commissionRate + baseSalary) \* 0.05)
7. 各種屬性實體孌數於操作屬性寫入時需實作數值合理範圍的檢查功能

結果畫面截圖：



程式碼：

package ExamTest.M5;

import ExamTest.M3.Employee;

public class CommissionEmployee extends Employee {

    double CommissionRate;

    int GrossSales;

    int BaseSalary;

    public CommissionEmployee(String Name, String Mobile, int GrossSales, double CommissionRate,int BaseSalary) {

        super(Name, Mobile);

        SetGrossSales(GrossSales);

        SetCommissionRate(CommissionRate);

        SetBaseSalary(BaseSalary);

    }

    public int GetGrossSales() {

        return GrossSales;

    }

    public void SetGrossSales(int GrossSales) {

        this.GrossSales = GrossSales;

    }

    public double GetCommissionRate() {

        return CommissionRate;

    }

    public void SetCommissionRate(double CommissionRate) {

        this.CommissionRate = CommissionRate;

    }

    public int GetBaseSalary() {

        return BaseSalary;

    }

    public void SetBaseSalary(int BaseSalary) {

        this.BaseSalary = BaseSalary;

    }

    @Override

    public String toString() {

       return "Commission："+

            "Name : "+this.GetName()+", "+

            "Mobile: "+this.GetMobile();

    }

    @Override

    public void GetTax() {

       double tax = (GetGrossSales() \* GetCommissionRate() + GetBaseSalary()) \* 0.05;

       System.out.println("Tax : " + tax);

    }

    @Override

    public void GetPaymentAmount() {

        double Earnings = GetGrossSales() \* GetCommissionRate() + GetBaseSalary();

        double Tax = (GetGrossSales() \* GetCommissionRate() + GetBaseSalary()) \* 0.05;

        double PaymentAmount = Earnings - Tax;

        System.out.println("取得員工扣稅所得: " + PaymentAmount);

    }

    @Override

    public void Earnings() {

       double Earnings =GetGrossSales() \* GetCommissionRate() + GetBaseSalary();

       System.out.println("取得員工未扣稅所得: " + Earnings);

    }

}

package ExamTest.M5;

import ExamTest.M3.Employee;

public class SalariedEmployee extends Employee {

    int GrossSales;

    double CommissionRate;

    int BaseSalary;

    public SalariedEmployee(String Name, String Mobile, int GrossSales, double CommissionRate, int BaseSalary) {

        super(Name, Mobile);

        SetGrossSales(GrossSales);

        SetCommissionRate(CommissionRate);

        SetBaseSalary(BaseSalary);

    }

    public int GetGrossSales() {

        return GrossSales;

    }

    public void SetGrossSales(int GrossSales) {

        this.GrossSales = GrossSales;

    }

    public double GetCommissionRate() {

        return CommissionRate;

    }

    public void SetCommissionRate(double CommissionRate) {

        this.CommissionRate = CommissionRate;

    }

    public int GetBaseSalary() {

        return BaseSalary;

    }

    public void SetBaseSalary(int BaseSalary) {

        this.BaseSalary = BaseSalary;

    }

    @Override

    public String toString() {

        return "Salaried " +

                "Name : " + this.GetName() + ", " + "Mobile: " + this.GetMobile();

    }

    @Override

    public void GetTax() {

        System.out.println("Tax : " + (GetGrossSales() \* GetCommissionRate() + GetBaseSalary()) \* 0.05);

    }

    @Override

    public void GetPaymentAmount() {

        double Earnings = GetGrossSales() \* GetCommissionRate() + GetBaseSalary();

        double Tax = (GetGrossSales() \* GetCommissionRate() + GetBaseSalary()) \* 0.05;

        double PaymentAmount = Earnings - Tax;

        System.out.println("取得員工扣稅所得: " + PaymentAmount);

    }

    @Override

    public void Earnings() {

       double Earnings =GetGrossSales() \* GetCommissionRate() + GetBaseSalary();

       System.out.println("取得員工未扣稅所得: " + Earnings);

    }

}

**任務六：設計一個類別 「BasePlusCommissionEmployee」，繼承 「CommissionEmployee」:**

1. 將 CommissionEmployee 之獎金比率提高至 30%

結果畫面截圖：



程式碼：

package ExamTest.M6;

import ExamTest.M5.CommissionEmployee;

public class BasePlusCommissionEmployee extends CommissionEmployee {

    public BasePlusCommissionEmployee(String Name, String Mobile, int GrossSales, double CommissionRate, int BaseSalary) {

        super(Name, Mobile, GrossSales, CommissionRate \* 1.3, BaseSalary);

    }

    @Override

    public String toString() {

        return "BasePlusCommission " +

                "Name : " + GetName() + ", " +

                "Mobile: " + GetMobile();

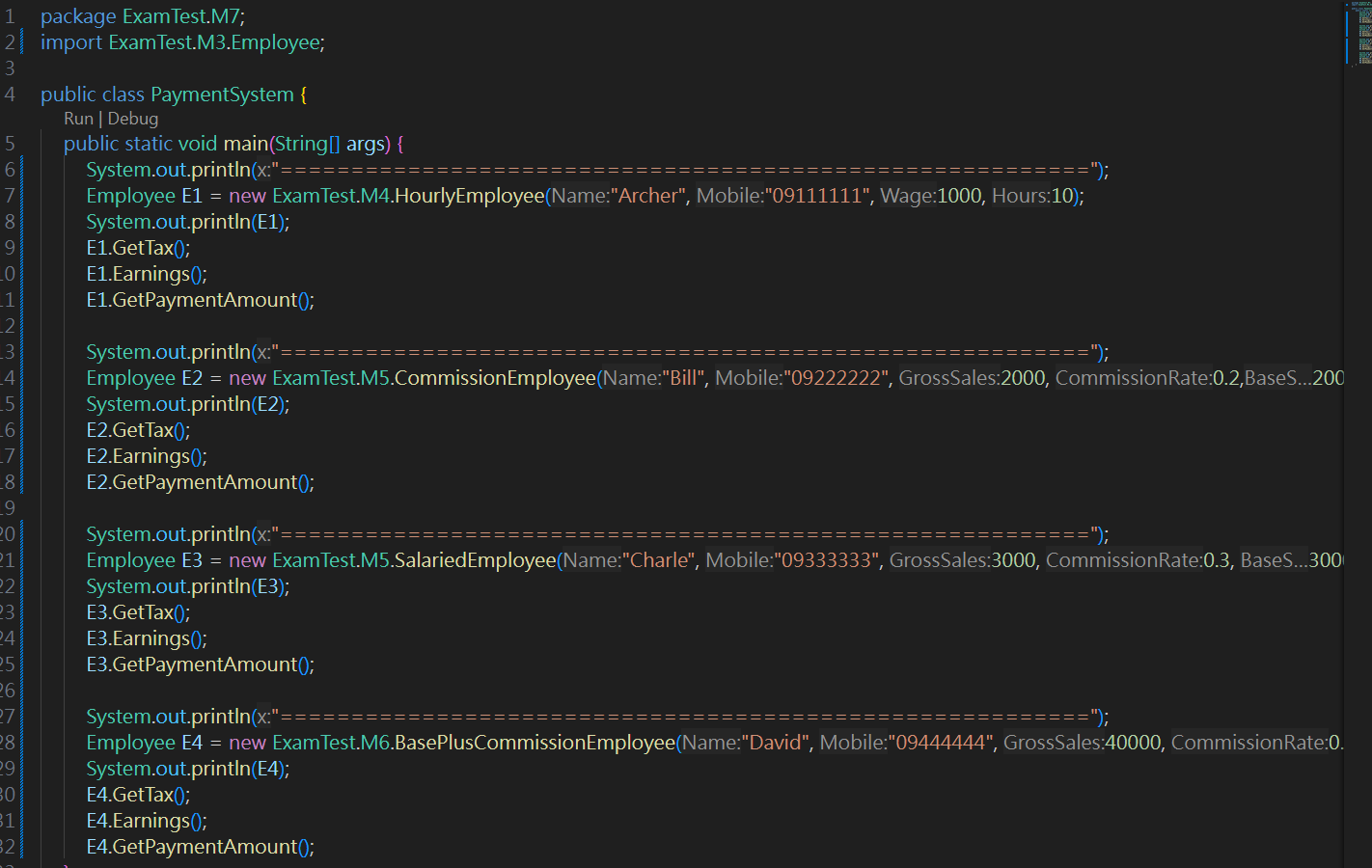
    }

}

**任務七：設計一個類別 「PaymentSystem」，具備程式進入點Main()功能，利用多型印出各個員工的薪資：**

1. 至少需要內含一個 HourlyEmployee 實體物件
2. 至少需要內含一個 SalariedEmployee 實體物件
3. 至少需要內含一個 ComrissionEmployee、一個 BasePlusCommissionEmployee 實體物件
4. 請利用toString、getTax、earnings、getPaymentAmount 方法，進行 Employee 多型呼叫，以輸出所有應付帳款明細項目

結果畫面截圖：



程式碼：

package ExamTest.M7;

import ExamTest.M3.Employee;

public class PaymentSystem {

    public static void main(String[] args) {

        System.out.println("=========================================================");

        Employee E1 = new ExamTest.M4.HourlyEmployee("Archer", "09111111", 1000, 10);

        System.out.println(E1);

        E1.GetTax();

        E1.Earnings();

        E1.GetPaymentAmount();

        System.out.println("=========================================================");

        Employee E2 = new ExamTest.M5.CommissionEmployee("Bill", "09222222", 2000, 0.2,20000);

        System.out.println(E2);

        E2.GetTax();

        E2.Earnings();

        E2.GetPaymentAmount();

        System.out.println("=========================================================");

        Employee E3 = new ExamTest.M5.SalariedEmployee("Charle", "09333333", 3000, 0.3, 30000);

        System.out.println(E3);

        E3.GetTax();

        E3.Earnings();

        E3.GetPaymentAmount();

        System.out.println("=========================================================");

        Employee E4 = new ExamTest.M6.BasePlusCommissionEmployee("David", "09444444", 40000, 0.2, 30000);

        System.out.println(E4);

        E4.GetTax();

        E4.Earnings();

        E4.GetPaymentAmount();

    }

}