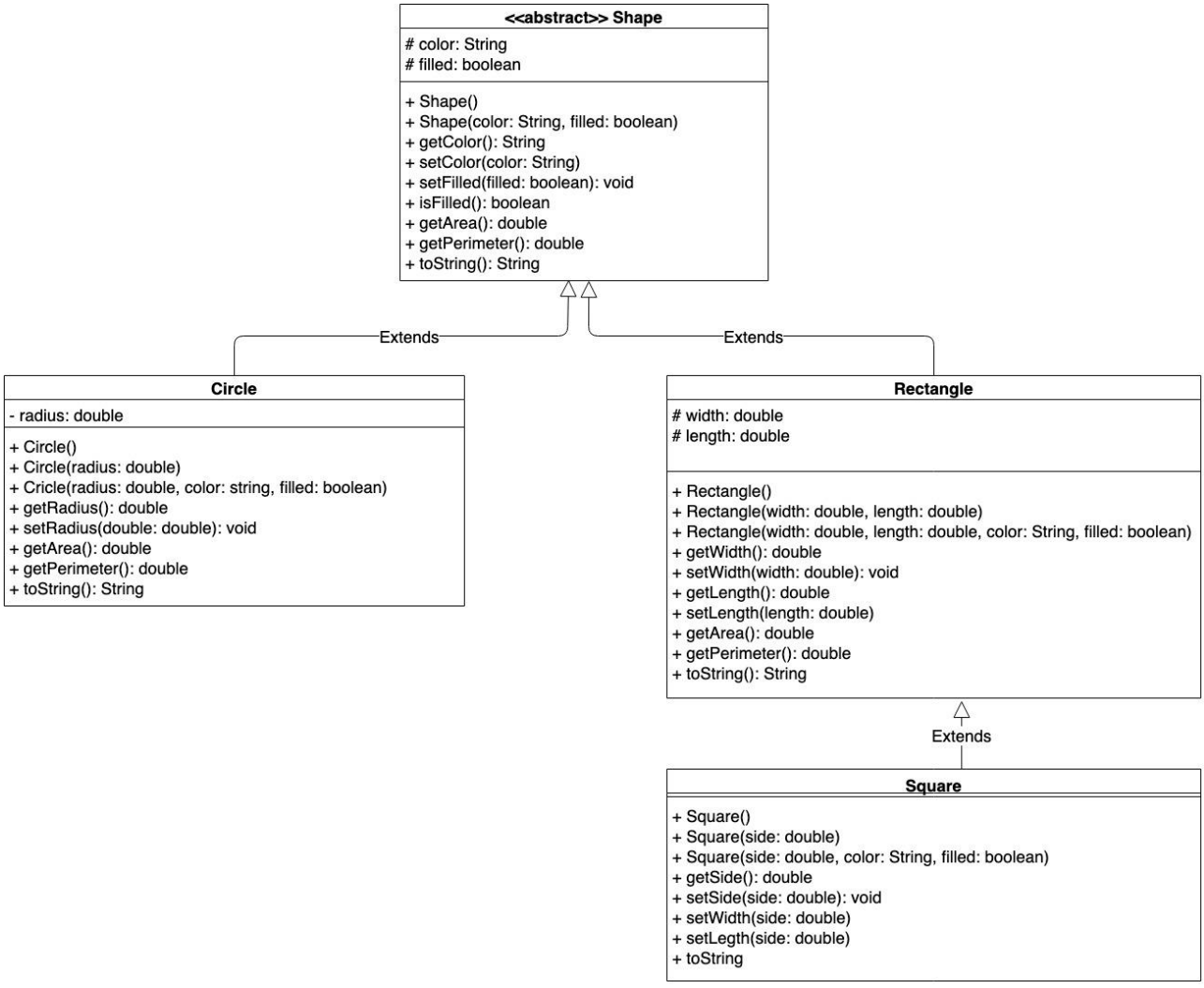
Inheritance (T.T)

Dùng phương pháp lập trình hướng đối tượng, các em hãy giải quyết các bài tập sau đây:

Bài 1: (5 điểm)

Hãy xây dựng một lớp trừu tượng có tên là Shape và các lớp con cụ thể kế thừa từ lớp trên có tên là Circle, Rectangle và Square theo sơ đồ cây thừa kế như sau:



Trong bài tập này, Shape là một abstract class có những thông tin sau:

- Hai thuộc tính color, filled có tầm vực truy xuất là protected (protected được ký hiệu là #), các biến protected có thể truy cập bởi các class con của nó.
- Các phương thức getter() và setter()
- getArea(): tính diện tích, tương tự cho các hình khác

- getPerimeter() : Tính chu vi, và phương thức toString() in ra thông tin của các thể hiện (đối tượng) tương ứng của class (Ví dụ: "Shape[color=red, filled=true]")

Hình vuông (Square) là một dạng đặc biệt của hình chữ nhật (Rectangle), có chiều dài bằng với chiều rộng, cho nên lớp Square sẽ kế thừa từ lớp Rectangle. Trong Square, ta có thể sử dụng lại các phương thức getter() và setter() của Rectangle với width và height bằng nhau.

Bài làm:

```
package lab_assignment6;

public class Bai1{
	public static void main(String[] args) {
		Shape shape1 = new Circle(10, "Xanh Luc", true); // Downcasting
		Shape shape2 = new Rectangle(10, 20, "Do", false); // Downcasting
		Shape shape3 = new Square(30, "Xanh Da Troi", true); // Downcasting

		System.out.println("Hình thứ 1: " + shape1.toString());
		System.out.println("Hình thứ 2: " + shape2.toString());
		System.out.println("Hình thứ 3: " + shape3.toString());
	}
}


abstract class Shape {

    protected String color;
    protected boolean filled;

    public Shape(){
       this.color = "";
       this.filled = false;
    }

    public Shape(String color, boolean filled) {
       this.color = color;
       this.filled = filled;
    }

    public String getColor() {
       return this.color;
    }

    public void setColor(String color){
       if (color != "") this.color = color;
    }


    public void setFilled(boolean filled){
       this.filled = filled;
    }

    public boolean isFilled() {
       return this.filled;
    }
```

```
    abstract public double getArea();

    abstract public double getPerimeter();

    public String toString() {
       return "Shape[color=" + this.color + ",filled=" + this.filled + "]";
    }
}

class Circle extends Shape {

      private double radius;

    public Circle() {
      super();
      this.radius = 0;
    }

    public Circle(double radius) {
      super();
      this.radius = radius;
    }

    public Circle(double radius, String color, boolean filled) {
      super(color, filled);
      this.radius = radius;
    }

    public double getRadius() {
      return this.radius;
    }

    public void setRadius(double radius) {
      if (radius >= 0) this.radius = radius;
    }

    @Override
    public double getArea() {
      return this.radius * this.radius * 3.14;
    }

    @Override
    public double getPerimeter() {
      return this.radius * 2 * 3.14;
    }

    public String toString() {
      return "Circle["+super.toString()+",radius="+this.radius+"]";
    }
}


class Rectangle extends Shape {

      protected double width;
      protected double length;

    public Rectangle(){
      super();
```

```
        this.width = 0;
        this.length = 0;
    }

    public Rectangle(double length, double width){
        super();
        this.width = width;
        this.length = length;
    }

    public Rectangle(double length, double width, String color, boolean filled){
        super(color, filled);
        this.length = length;
        this.width = width;
    }

    public double getWidth() {
        return this.width;
    }

    public void setWidth(double width) {
        if (width >= 0) this.width = width;
    }

    public double getLength() {
        return this.length;
    }

    public void setLength(double length) {
        if (length >= 0) this.length = length;
    }

    @Override
    public double getArea() { // tính diện tích
        return this.width * this.length;
    }

    @Override
    public double getPerimeter() {
        return (this.width + this.length) * 2;
    }

    public String toString() {
        return "Rectangle["+super.toString()+",width="
+this.width+",length="+this.length+"]";
    }
}

class Square extends Rectangle {

    public Square() {
        super();
    }

    public Square(double side) {
        super(side, side);
    }

    public Square(double side, String color, boolean filled) {
```

```
        super(side, side, color, filled);
    }

    public double getSide() {
        //return super.getWidth();
        return super.getLength();
    }

    public void setSide(double side) {
        if (side >= 0) {
                super.setWidth(side);
                super.setLength(side);
        }
    }

    @Override
    public void setWidth(double side) {
        if (side >= 0) setSide(side);
    }

    @Override
    public void setLength(double side) {
        if (side >= 0) setSide(side);
    }

    public String toString() {
        return "Square[" + super.toString() + "]";
    }
}
```

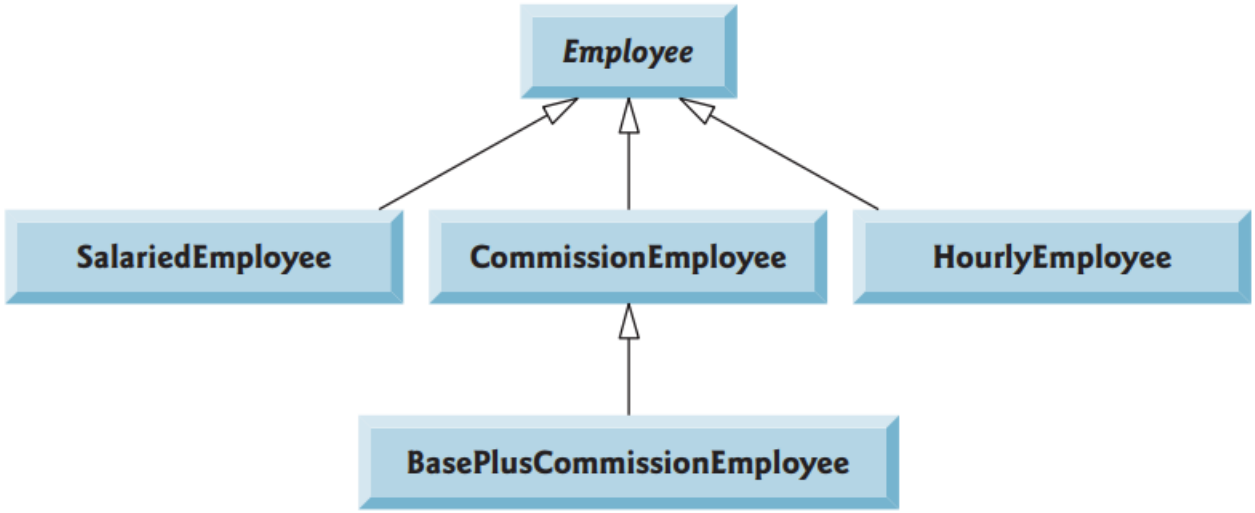
Bài 2: (5 điểm)

Một công ty trả lương cho nhân viên hàng tuần. Các nhân viên của công ty có 4 loại:

- Nhân viên SalariedEmployee được trả một mức lương cố định hàng tuần bất kể số giờ
- Nhân viên HourlyEmployee được trả lương theo giờ và nhận tiền làm thêm giờ (tức là gấp 1,5 lần mức lương theo giờ) cho tất cả các giờ làm việc vượt quá 40 giờ
- Nhân viên CommissionEmployee được nhận một tỷ lệ phần trăm dựa trên doanh thu của họ
- Nhân viên BasePlusCommissionEmployee được nhận lương cơ bản cộng với tỷ lệ phần trăm doanh thu của họ

Đối với thời gian thanh toán hiện tại, công ty quyết định thưởng cho nhân viên hưởng lương bằng cách thêm vào 10% mức lương cơ bản của họ.

Hãy viết một ứng dụng thực hiện việc tính toán bảng lương cho các loại nhân viên nói trên của công ty.



Mô tả chi tiết từng lớp:

	earnings	toString
Employee	abstract	*firstName lastName* social security number: *SSN*
Salaried-Employee	weeklySalary	salaried employee: *firstName lastName* social security number: *SSN* weekly salary: *weeklySalary*

Hourly-Employee	if (hours <= 40)
	  wage * hours
	else if (hours > 40)
	{
	  40 * wage +
	  (hours - 40) *
	  wage * 1.5
	}
		hourly employee: *firstName lastName*
		social security number: *SSN*
		hourly wage: *wage*; hours worked: *hours*

Commission-Employee	commissionRate * grossSales	commission employee: *firstName lastName* social security number: *SSN* gross sales: *grossSales*; commission rate: *commissionRate*
BasePlus-Commission-Employee	(commissionRate * grossSales) + baseSalary	base salaried commission employee: *firstName lastName* social security number: *SSN* gross sales: *grossSales*; commission rate: *commissionRate*; base salary: *baseSalary*

Bài làm:

```
package lab_assignment6;

public class Bai2 {
	public static void main(String[] args) {
		Employee emp1 = new SalariedEmployee("A", "Micheal", "7384927", 200);
		Employee emp2 = new CommissionEmployee("Hieu", "Tran", "83027294",
10000, (5.0 / 100));
		Employee emp3 = new HourlyEmployee("Nam", "Nguyen", "8394728", 200, 50);
		Employee emp4 = new BasePlusCommissionEmployee("Markendy", "John",
"9384028", 30000, (5.0 / 100), 200);

		System.out.println("Nhân viên 1: " + emp1.toString());
		System.out.println("Nhân viên 2: " + emp2.toString());
		System.out.println("Nhân viên 3: " + emp3.toString());
```

```
            System.out.println("Nhân viên 4: " + emp4.toString());
      }
}

abstract class Employee {
   protected String firstName;
   protected String lastName;
   protected String SSN;

   public Employee(String firstName, String lastName, String SSN) {
         this.firstName = firstName;
         this.lastName = lastName;
         this.SSN = SSN;
   }

   public String getFirstName() {
         return this.firstName;
   }

   public void setFirstName(String firstName) {
         if (firstName != "") this.firstName = firstName;
   }

   public String getLastName() {
         return this.lastName;
   }

   public void setLastName(String lastName) {
         if (lastName != "") this.lastName = lastName;
   }

   public String getSSN() {
         return this.SSN;
   }
   public void setSSN(String SSN) {
         if (SSN != "") this.SSN = SSN;
   }

   abstract public double earnings();

   public String toString() {
         return
"Employee["+"FirstName:"+this.firstName+",LastName:"+this.lastName+",Social security
number:"+this.SSN+"]";
   }
}

class SalariedEmployee extends Employee {
         private int weeklySalary;

         public SalariedEmployee(String firstName, String lastName, String SSN, int
weeklySalary ) {
               super(firstName,lastName,SSN);
               this.weeklySalary = weeklySalary;
         }

         public int getWeeklySalary() {
               return weeklySalary;
         }
```

```
	public void setWeeklySalary(int weeklySalary) {
		if(weeklySalary >= 0)
			this.weeklySalary = weeklySalary;
		else
			System.out.println("Error");
	}

	@Override
	public double earnings() {
		return this.weeklySalary + this.weeklySalary * 10.0 / 100;
	}

	@Override
	public String toString() {
	  return
"SalariedEmployee["+super.toString()+",WeeklySalary:"+this.weeklySalary+",
Earnings:"+this.earnings()+"]";
	}
}

class HourlyEmployee extends Employee {
	private int hourlyWage;
	private int hours;

	public HourlyEmployee(String firstName, String lastName, String SSN, int
hourlyWage, int hours) {
		super(firstName,lastName,SSN);
		this.hourlyWage = hourlyWage;
		this.hours = hours;
	}

	public int getWage() {
		return hourlyWage;
	}

	public void setWage(int hourlyWage) {
		if(hourlyWage >= 0)
			this.hourlyWage = hourlyWage;
		else
			System.out.println("Error");
	}

	public int getHour() {
		return hours;
	}

	public void setHour(int hours) {
		if(hours >= 0)
			this.hours = hours;
		else
			System.out.println("Error");
	}

	@Override
	public double earnings() {
		if(this.hours <= 40)
			return (this.hourlyWage * this.hours);
		else
```

```
                        return (40 * hourlyWage + (this.hours - 40) * hourlyWage *
1.5);
                }

                @Override
                public String toString() {
                        return
"HourEmployee:"+super.toString()+",Wage:"+this.hourlyWage+",Hour:"+this.hours+",
Earning:" + this.earnings() +"]";
                }
}


class CommissionEmployee extends Employee {
                private int grossSales;
                private double commissionRate;

                public CommissionEmployee(String firstName, String lastName, String SSN, int
grossSales, double commissionRate) {
                        super(firstName,lastName,SSN);
                        this.grossSales = grossSales;
                        this.commissionRate = commissionRate;
                }

                public double getGrossSales() {
                        return this.grossSales;
                }

                public void setGrossSales(int grossSales) {
                        if(grossSales >= 0)
                                this.grossSales = grossSales;
                        else
                                System.out.println("Error");
                }

              public double getCommissionRate() {
                     return this.commissionRate;
                }

                public void setCommissionRate(double commissionRate) {
                        if(commissionRate >= 0)
                                this.commissionRate = commissionRate;
                        else
                                System.out.println("Error");
                }

                @Override
                public double earnings() {
                        return (grossSales * commissionRate);
                }

                @Override
                public String toString() {
                        return
"CommissionEmployee["+super.toString()+"GrossSales:"+this.grossSales+"CommissionRate:
"+this.commissionRate+", Earning:" + this.earnings() +"]";
                }
}
```

```
class BasePlusCommissionEmployee extends CommissionEmployee {
	private double baseSalary;

	public BasePlusCommissionEmployee(String firstName, String lastName, String
SSN, int grossSales, double commissionRate, double baseSalary) {
		super(firstName,lastName,SSN,grossSales,commissionRate);
		this.baseSalary = baseSalary;
	}

	public double getBaseSalary() {
		return this.baseSalary;
	}

	public void setBaseSalary(double baseSalary) {
		if(baseSalary >= 0)
			this.baseSalary = baseSalary;
		else
			System.out.println("Error");
	}

	@Override
	public double earnings() {
		return (super.earnings() + baseSalary + baseSalary * 10.0 / 100);
	}

	@Override
	  public String toString() {
	    return "BasePlusCommissionEmployee["+super.toString()+", Base salary: "
+ this.baseSalary + ", Earnings:"+this.earnings()+"]";
	}
}
```

Bài 3 (+2 điểm)

Dựa trên kết quả của bài tập 2, hãy viết chương trình tạo một mảng kiểu Employee để nhập và lưu trữ tập hợp các nhân viên của công ty. Sau đó, duyệt qua từng phần tử của mảng và tính tiền thanh toán cho từng nhân viên của công ty.

Bài làm:

```
package lab_assignment6;;
import java.util.ArrayList;
import java.util.Scanner;

public class Bai3 {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        ArrayList<Employee> danhSachNhanVien = new ArrayList<Employee>();
        Employee nhanVienMoi;
        System.out.println("Nhập vào danh sách các nhân viên trong công ty:
");
        int luaChon = 1; // 0 - Dừng, 1 - Tiếp tục nhập
        do {
            System.out.println("Chọn loại nhân viên: ");
            System.out.println("1 - Nhân viên trả lương hàng tuần (Salaried
Employee)");
            System.out.println("2 - Nhân viên trả lương theo giờ và nhận tiền
làm thêm theo giờ (Hourly Employee)");
            System.out.println("3 - Nhân viên nhận tiền hoa hồng (Commission
Employee)");
            System.out.println("4 - Nhân viên nhận lương cơ bản và tiền hoa
hồng (Base Plus Commission Employee)");
            System.out.print("Nhập lựa chọn: ");
            luaChon = scanner.nextInt();
            scanner.nextLine();
            switch (luaChon) {
                case 1:
                    nhanVienMoi = InputSalariedEmployee(scanner);
                    danhSachNhanVien.add(nhanVienMoi);
                break;

                case 2:
                    nhanVienMoi = InputHourlyEmployee(scanner);
                    danhSachNhanVien.add(nhanVienMoi);
                break;

                case 3:
                    nhanVienMoi = InputCommissionEmployee(scanner);
                    danhSachNhanVien.add(nhanVienMoi);
                break;

                case 4:
                    nhanVienMoi = InputBasePlusCommissionEmployee(scanner);
                    danhSachNhanVien.add(nhanVienMoi);
                break;
            }
            System.out.print("Bạn có muốn tiếp tục nhập ? (0 - Dừng, 1 - tiếp
tục): ");
            luaChon = scanner.nextInt();
            System.out.println();
```

```
        } while (luaChon != 0);

        System.out.println("Tính và xuất lương của từng nhân viên: ");
        int count= 0;
        for(Employee nhanVien : danhSachNhanVien) {
            count++;
            System.out.println("Nhân viên " + count + ": " +
nhanVien.toString());
        }
    }

    public static Employee InputSalariedEmployee(Scanner scanner) {
        System.out.print("Nhập tên nhân viên: ");
        String firstName = scanner.nextLine();

        System.out.print("Nhập họ nhân viên: ");
        String lastName = scanner.nextLine();

        System.out.print("Nhập số an sinh xã hội: ");
        String ssn = scanner.nextLine();

        System.out.print("Nhập số lương cơ bản hàng tuần: ");
        int weeklySalary = scanner.nextInt();
        scanner.nextLine();

        Employee nhanVienMoi = new SalariedEmployee(firstName, lastName, ssn,
weeklySalary);

        return nhanVienMoi;
    }

    public static Employee InputCommissionEmployee(Scanner scanner) {
        System.out.print("Nhập tên nhân viên: ");
        String firstName = scanner.nextLine();

        System.out.print("Nhập họ nhân viên: ");
        String lastName = scanner.nextLine();

        System.out.print("Nhập số CMND: ");
        String ssn = scanner.nextLine();

        System.out.print("Nhập doanh số bán hàng của nhân viên: ");
        int grossSales = scanner.nextInt();
        scanner.nextLine();

        System.out.print("Nhập tỉ lệ hoa hồng: ");
        double commissionRate = scanner.nextDouble();
        scanner.nextLine();

        Employee nhanVienMoi = new CommissionEmployee(firstName, lastName,
ssn, grossSales, commissionRate);

        return nhanVienMoi;
    }

    public static Employee InputHourlyEmployee(Scanner scanner) {
        System.out.print("Nhập tên nhân viên: ");
        String firstName = scanner.nextLine();

        System.out.print("Nhập họ nhân viên: ");
        String lastName = scanner.nextLine();
```

```
        System.out.print("Nhập số CMND: ");
        String ssn = scanner.nextLine();

        System.out.print("Nhập chi phí mỗi giờ làm việc của nhân viên: ");
        int hourlyWage = scanner.nextInt();
        scanner.nextLine();

        System.out.print("Nhập số giờ làm việc của nhân viên: ");
        int hours = scanner.nextInt();
        scanner.nextLine();

        Employee nhanVienMoi = new HourlyEmployee(firstName, lastName, ssn,
hourlyWage, hours);

        return nhanVienMoi;
    }

    public static Employee InputBasePlusCommissionEmployee(Scanner scanner) {
        System.out.print("Nhập tên nhân viên: ");
        String firstName = scanner.nextLine();

        System.out.print("Nhập họ nhân viên: ");
        String lastName = scanner.nextLine();

        System.out.print("Nhập số CMND: ");
        String ssn = scanner.nextLine();

        System.out.print("Nhập doanh số bán hàng của nhân viên: ");
        int grossSales = scanner.nextInt();
        scanner.nextLine();

        System.out.print("Nhập tỉ lệ hoa hồng: ");
        double commissionRate = scanner.nextDouble();
        scanner.nextLine();

        System.out.print("Nhập mức lương cơ bản của nhân viên: ");
        double baseSalary = scanner.nextDouble();
        scanner.nextLine();

        Employee nhanVienMoi = new BasePlusCommissionEmployee(firstName,
lastName, ssn, grossSales, commissionRate, baseSalary);

        return nhanVienMoi;
    }
}
```