

Mục tiêu

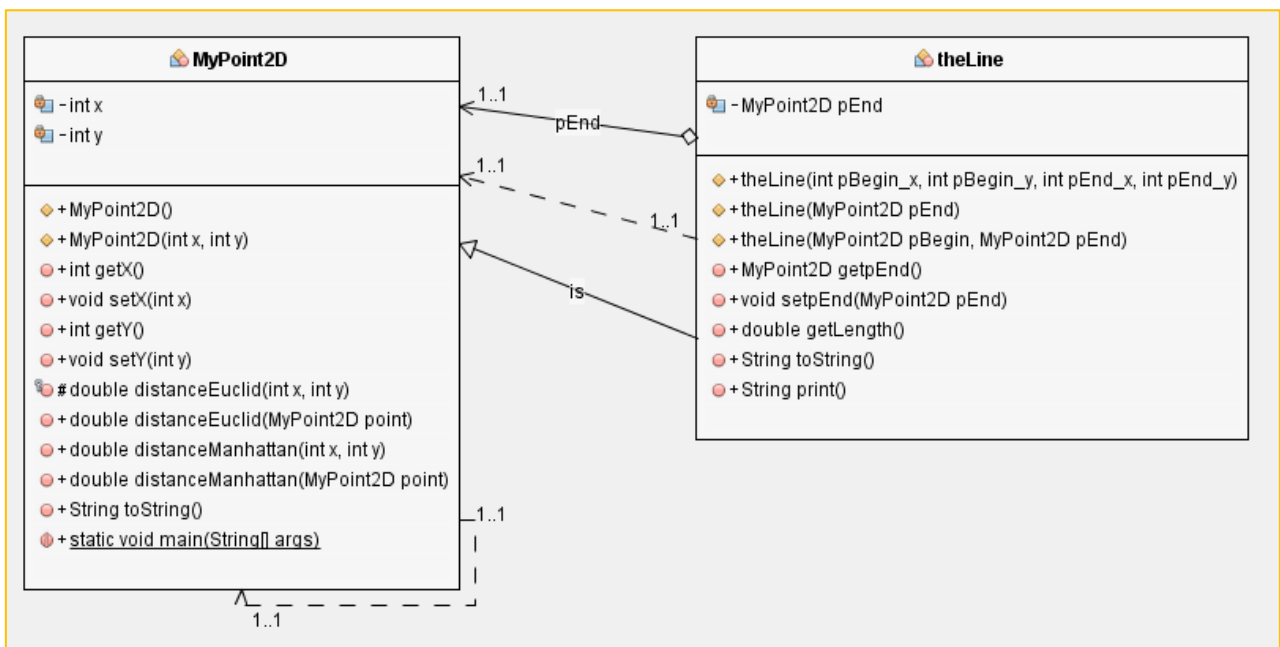
- Trình bày được khái niệm về tính thừa kế và đa hình.
- Tìm hiểu các từ khóa protected.
- Cài đặt các phương thức và nạp chồng (overload), ghi đè (override) phương thức.
- Các lớp có quan hệ IS-A.

Bài tập

Bài Nội dung

1. Thực hiện tạo một class được kế thừa từ một class (Thiết kế 2 class có quan hệ IS-A).

Thiết kế class theLine (đoạn thẳng) được kế thừa từ class MyPoint2D gồm các thành phần dữ liệu và phương thức theo lược đồ UML sau:



1. Thành phần dữ liệu gồm các biến (instance variable)

- pEnd: điểm kết thúc

2. Các phương thức (method)

- 2.1 Phương thức khởi tạo:

- `public theLine(int pBegin_x, int pBegin_y, int pEnd_x, int pEnd_y)`

- Các phương thức nạp chồng (overloaded)

- `public theLine(MyPoint2D pBegin, MyPoint2D pEnd)`

- 2.2 Các phương thức get và set

2.3 Các phương thức khác

- `public double getLength():` tính độ dài của đoạn thẳng theo khoảng cách Euclid.

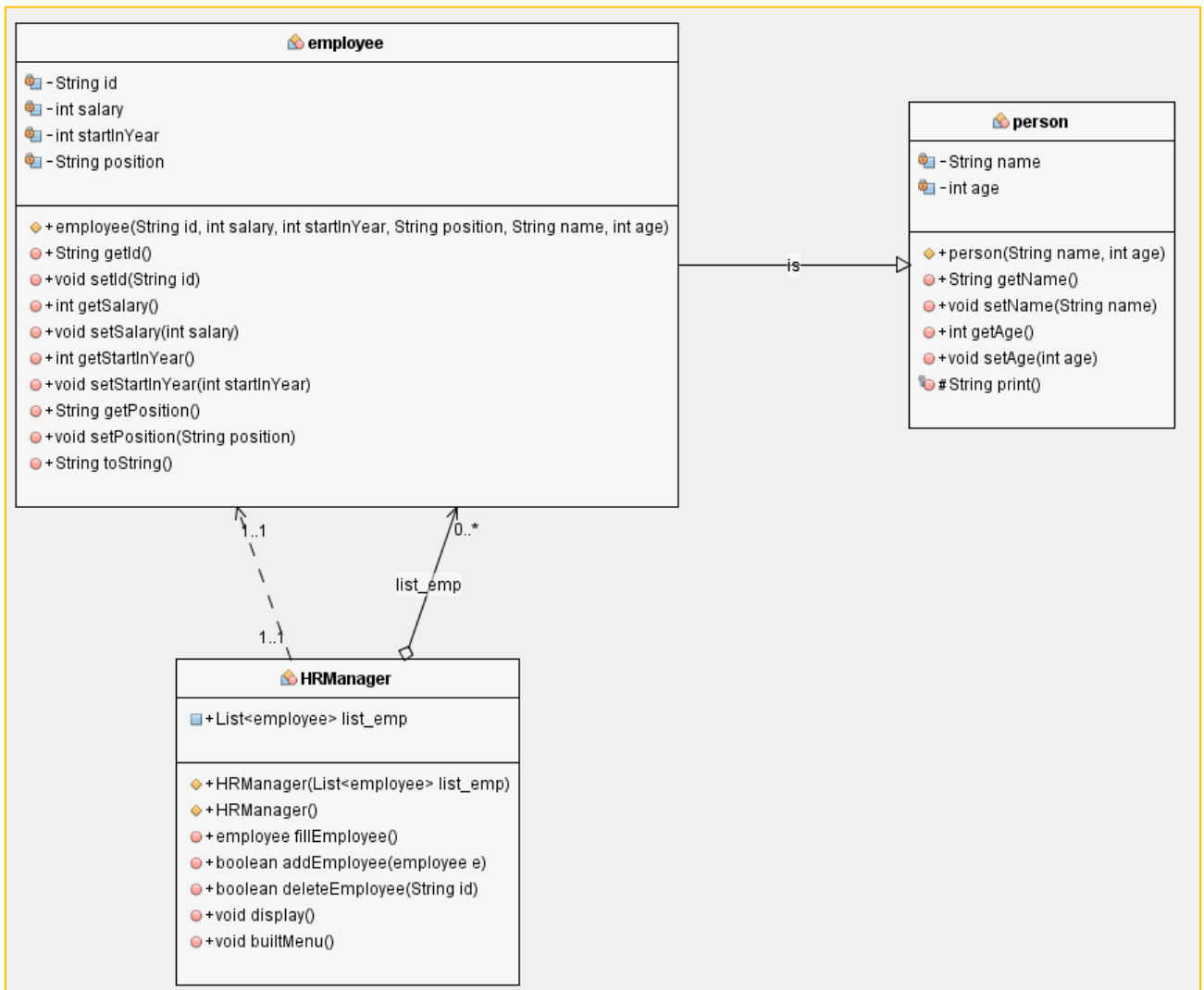
@Override

- `public String toString()` in ra màn hình kết quả theo định dạng

The line begin at point has $x = 0$ $y = 5$ and end at point has $x = 4$ $y = 8$

2. Thiết kế 2 class có quan hệ IS-A, và 1 class thực hiện các công việc về quản lý

Trong bài toán quản lý nhân sự cần: thiết kế class person, class employee kế thừa từ class person, class HRManager thực hiện việc quản lý nhân sự, gồm các thành phần dữ liệu và phương thức theo lược đồ UML sau:



1. Thành phần dữ liệu gồm các biến (instance variable) của class person

- `name:` tên cá nhân
- `age:` tuổi

2. Các phương thức (method)

2.1 Phương thức khởi tạo:

- `public class person` không tham số

Các phương thức nạp chồng (overloaded)

- `public` `person`(`String` `name`, `int` `age`)

2.2 Các phương thức `get` và `set`

2.3 Các phương thức khác

- `protected` `String` `print()` in thông tin cá nhân.

3. Thành phần dữ liệu gồm các biến (instance variable) của class `employee`

- Thừa kế cá thuộc tính của class `person`
- `id`: mã nhân viên
- `salary`: lương
- `startInYear`: năm bắt đầu làm việc
- `position`: vị trí.

4. Các phương thức (method)

4.1 Phương thức khởi tạo:

```
public employee(String id, int salary, int startInYear, String position, String name, int age)
```

4.2 Các phương thức `get` và `set`

4.3 Các phương thức khác

- `public` `String` `print()` in thông tin cá nhân.

5. Thiết kế lớp quản lý nhân sự `HRManager` gồm các phương thức thêm, xóa, sửa, tìm kiếm nhân viên.

-
3. Thiết kế class `Triangle` (tam giác) theo mô tả một hình tam giác là một đoạn thẳng có thêm 1 điểm thứ 3.

Viết phương thức tính diện tích tam giác.

Thiết kế class tập hợp các tam giác gồm thành phần dữ liệu là một mảng các tam giác (dùng cấu trúc mảng). Viết phương thức tìm tam giác có diện tích lớn nhất, sắp xếp các tam giác theo diện tích tăng dần.

THAM KHẢO

Bài 1.

```
package theLine;  
  
import mypoint2d.MyPoint2D;  
  
/**  
 *  
 * @author dinhkhai  
 */  
public class theLine extends MyPoint2D{  
    private MyPoint2D pEnd;  
  
    public theLine(int pBegin_x, int pBegin_y, int pEnd_x, int pEnd_y) {  
        super(pBegin_x, pBegin_y);  
        this.pEnd=new MyPoint2D(pEnd_x, pEnd_y);  
    }  
}
```

```

public theLine(MyPoint2D pEnd) {
    this.pEnd = pEnd;
}

public theLine(MyPoint2D pBegin, MyPoint2D pEnd) {
    super(pBegin.getX(), pBegin.getY());
    this.pEnd = pEnd;
}
public MyPoint2D getpEnd() {
    return pEnd;
}

public void setpEnd(MyPoint2D pEnd) {
    this.pEnd = pEnd;
}

public double getLength() {
    return super.distanceEuclid(pEnd);
}

public String toString(){
    return "The line begin at " + super.toString() + " and end at "
+this.getpEnd().toString();
}

}

```

output

The line begin at point has x = 0 y = 5 and end at point has x = 4 y = 8

Bài 2.

```

package person;

/**
 *
 * @author dinhkhai
 */
public class person {
    private String name;
    private int age;

    public person(String name, int age) {
        this.name = name;
        this.age = age;
    }

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }
}

```

```

    public int getAge() {
        return age;
    }

    public void setAge(int age) {
        this.age = age;
    }

    protected String print(){
        return "Name: " + name + "Age " + age;
    }
}

package employee;
import person.person;

/**
 *
 * @author dinhkhai
 */
public class employee extends person{
    private String id;
    private int salary;
    private int startInYear;
    private String position;

    public employee(String id, int salary, int startInYear, String position, String
name, int age) {
        super(name, age);
        this.id = id;
        this.salary = salary;
        this.startInYear = startInYear;
        this.position = position;
    }

    public String getId() {
        return id;
    }

    public void setId(String id) {
        this.id = id;
    }

    public int getSalary() {
        return salary;
    }

    public void setSalary(int salary) {
        this.salary = salary;
    }

    public int getStartInYear() {
        return startInYear;
    }

    public void setStartInYear(int startInYear) {
        this.startInYear = startInYear;
    }
}

```

```

    }

    public String getPosition() {
        return position;
    }

    public void setPosition(String position) {
        this.position = position;
    }

    public String toString(){
        return super.getName() + " " +super.getAge() + " " + getId() + "
"+getPosition()+ " " +getSalary() + " "+getStartInYear();
    }

}

package hrmanager;

import employee.employee;
import java.util.ArrayList;
import java.util.Iterator;
import java.util.List;
import java.util.Scanner;

/**
 *
 * @author dinhkhai
 */
public class HRManager {

    //co the viet cac phuong thuc duoi dang static
    public List<employee> list_emp;

    public HRManager(List<employee> list_emp) {
        this.list_emp = list_emp;
    }

    public HRManager() {
        list_emp=new ArrayList<>();
    }

    public employee fillEmployee(){
        //co the viet phuong thuc getInfo trong class Employee va goi phuong thuc tai
        day
        Scanner input=new Scanner(System.in);
        System.out.println("Please enter employee informations");
        String name=input.nextLine();
        int age=Integer.parseInt(input.nextLine());
        String id = input.nextLine();
        int salary=Integer.parseInt(input.nextLine());
        int startInYear = Integer.parseInt(input.nextLine());
        String position=input.nextLine();

        employee newEmp=new employee(id, salary, startInYear, position, name, age);
        //co the dua new employee vao danh sach

```

```

        list_emp.add(newEmp);
        return newEmp;
    }

    public boolean addEmployee(employee e){
        if(!list_emp.contains(e.getId())) {
            list_emp.add(e);
            return true;
        } else
            return false;
    }

    public boolean deleteEmployee(String id){
        for (Iterator<employee> it = list_emp.iterator(); it.hasNext();) {
            employee e = it.next();
            if(e.getId().compareToIgnoreCase(id)==0){
                it.remove();
                return true;
            }
        }

        System.out.println("This employee does not exist");
        return false;
    }

    public boolean findEmployee(String name){
        for (Iterator<employee> it = list_emp.iterator(); it.hasNext();) {
            employee emp = it.next();
            if(emp.getName().compareToIgnoreCase(name)==0)
                return true;
        }
        return false;
    }

    public boolean editEmployee(){
        // your code here ! he he
        return true;
    }

    public void display(){
        for (Iterator<employee> it = list_emp.iterator(); it.hasNext();) {
            employee e = it.next();
            System.out.println(e.toString());
        }
    }

    // có thể viết phần menu trong 1 class dành riêng cho các menu
    public void builtMenu(){
        System.out.println("Welcome to HRManager system");
        int choice ;

        Scanner input =new Scanner(System.in);
        choice=Integer.parseInt(input.nextLine())%5;
        switch(choice){
            case 0: System.out.println("Create a new employee");
                //call the function
            case 1: System.out.println("Find an employee");
                //call the function
            case 2: System.out.println("Edit an employee");
                //call the function
            case 3: System.out.println("Delete an employee");

```

```
        //call the function
        case 4: System.out.println("Display a list of employees");
        //call the function
    }
}
}
```

Bài 3.