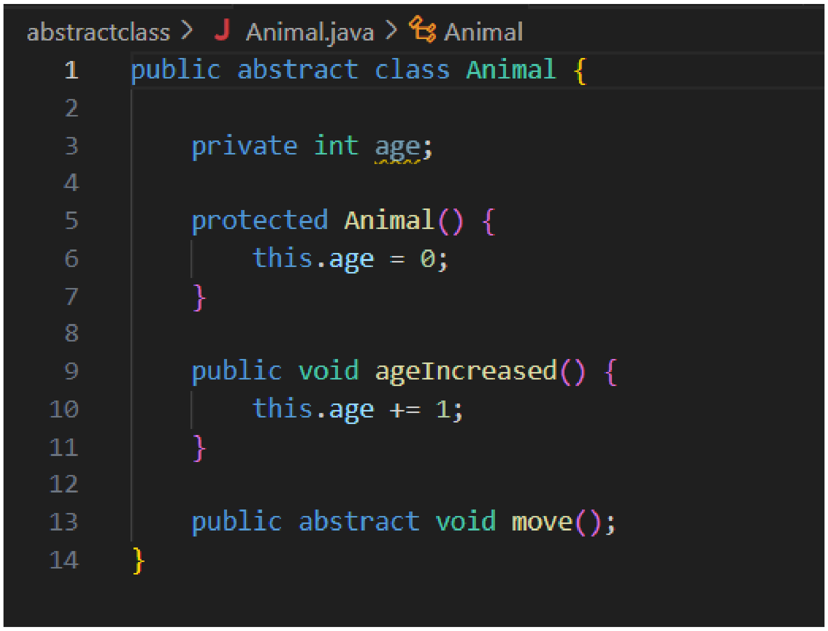
Jobsheet 09 – Abstract Class and Interface

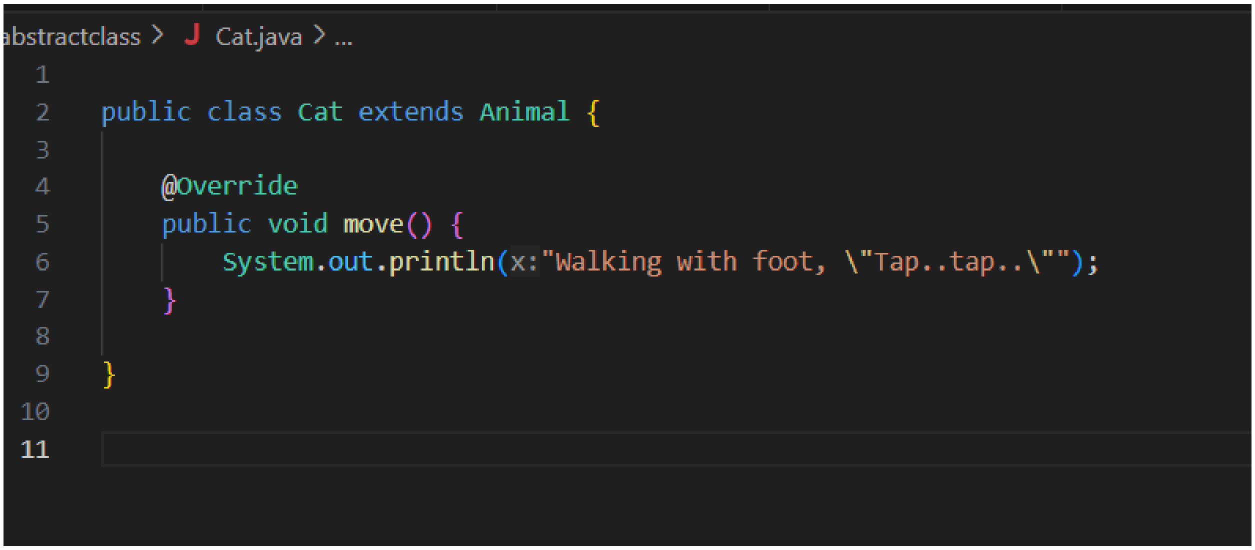
Erwan Majid 08 2i

Link github: https://github.com/Majid5654/Semester-3/tree/Main/JAVA%20OOP/jobsheet%209

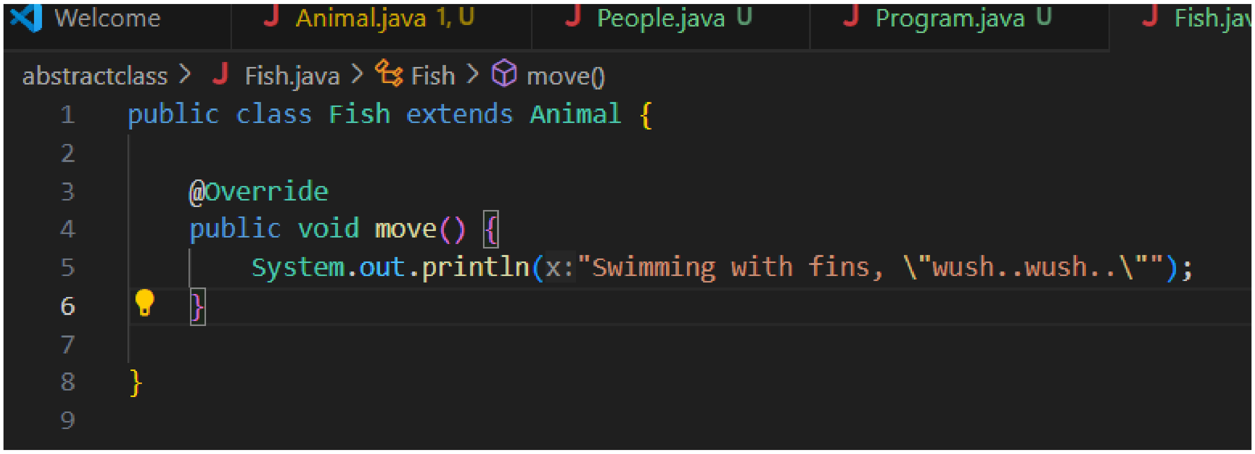
- Experiment 1: Abstract Class



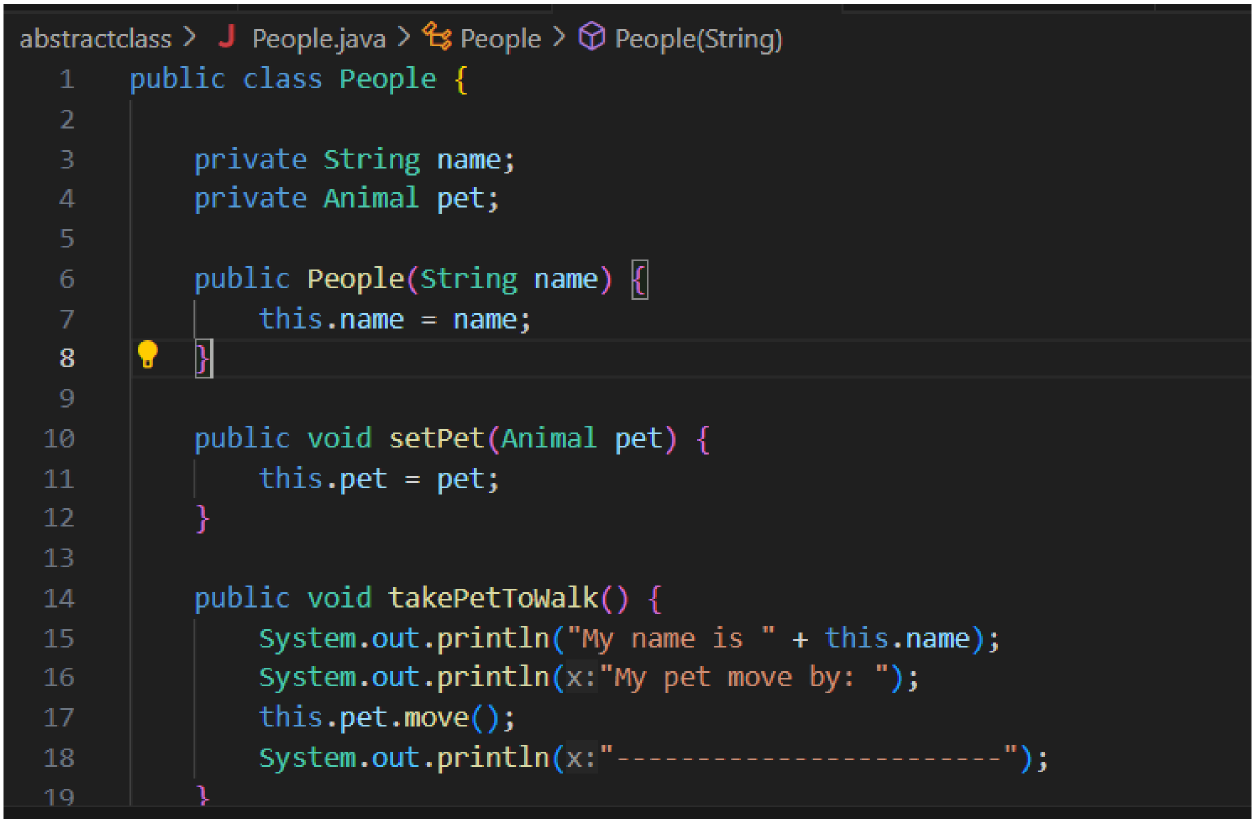
```java
public abstract class Animal {

    private int age;

    protected Animal() {
        this.age = 0;
    }

    public void ageIncreased() {
        this.age += 1;
    }

    public abstract void move();
}
```



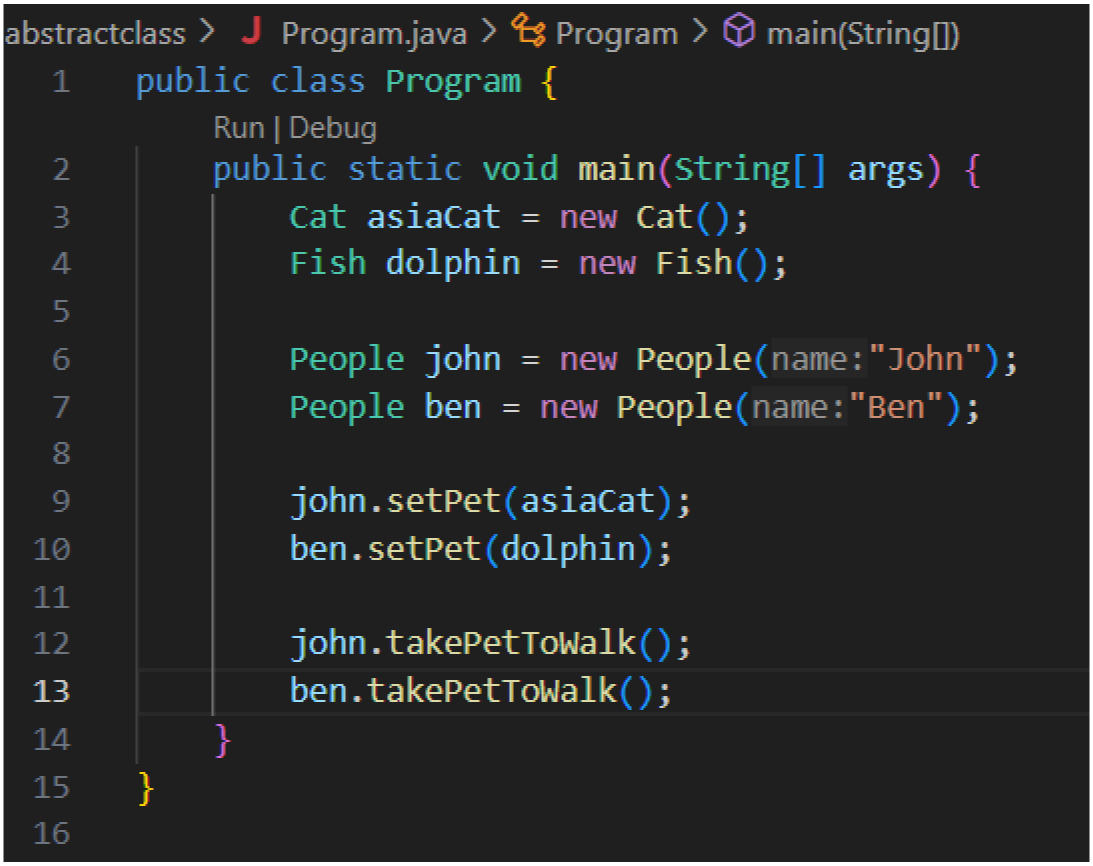
```java

public class Cat extends Animal {

    @Override
    public void move() {
        System.out.println(x:"Walking with foot, \"Tap..tap..\"");
    }

}

```



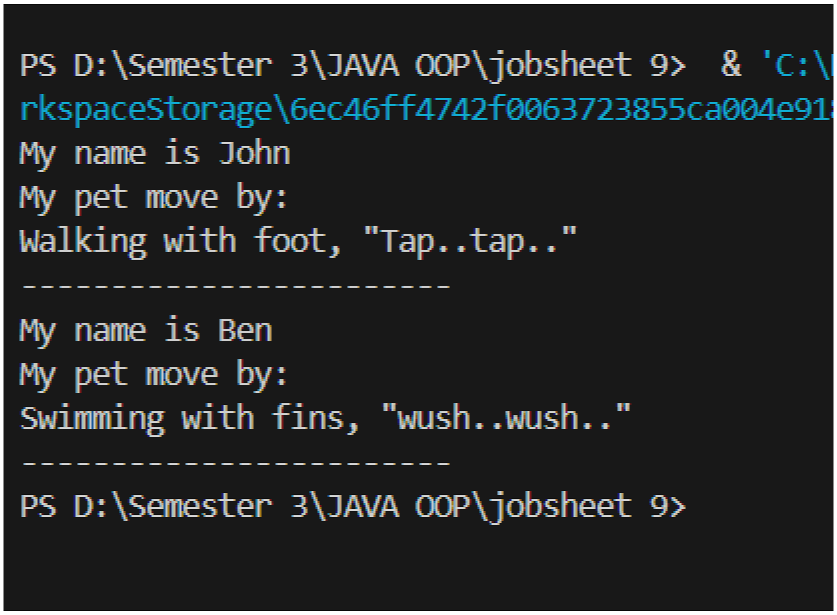
```java
public class Fish extends Animal {

    @Override
    public void move() {
        System.out.println(x:"Swimming with fins, \"wush..wush..\"");
    }

}

```



```java
public class People {

    private String name;
    private Animal pet;

    public People(String name) {
        this.name = name;
    }

    public void setPet(Animal pet) {
        this.pet = pet;
    }

    public void takePetToWalk() {
        System.out.println("My name is " + this.name);
        System.out.println(x:"My pet move by: ");
        this.pet.move();
        System.out.println(x:"------------------------");
    }
```



```java
public class Program {
    public static void main(String[] args) {
        Cat asiaCat = new Cat();
        Fish dolphin = new Fish();

        People john = new People(name:"John");
        People ben = new People(name:"Ben");

        john.setPet(asiaCat);
        ben.setPet(dolphin);

        john.takePetToWalk();
        ben.takePetToWalk();
    }
}
```

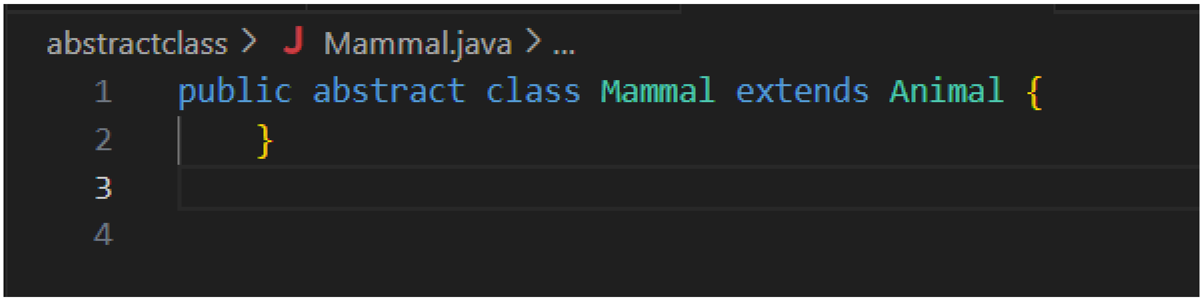


```
PS D:\Semester 3\JAVA OOP\jobsheet 9>  & 'C:\
rkspaceStorage\6ec46ff4742f0063723855ca004e91
My name is John
My pet move by:
Walking with foot, "Tap..tap.."
------------------------
My name is Ben
My pet move by:
Swimming with fins, "wush..wush.."
------------------------
PS D:\Semester 3\JAVA OOP\jobsheet 9>
```

Discussion questions:

Can it be possible if a class that extends an abstract class does not implement abstract methods

in its parent class? Prove it!

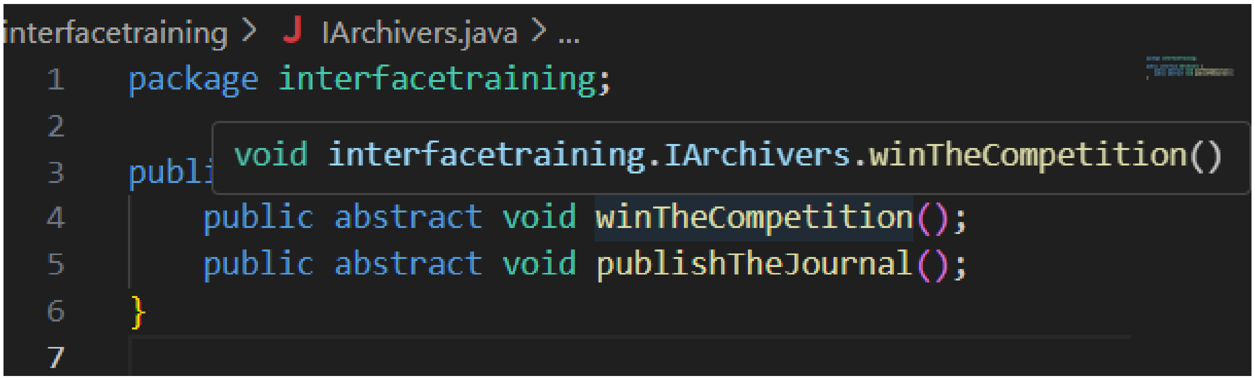


abstractclass > Mammal.java > ...

```
public abstract class Mammal extends Animal {
    }

```

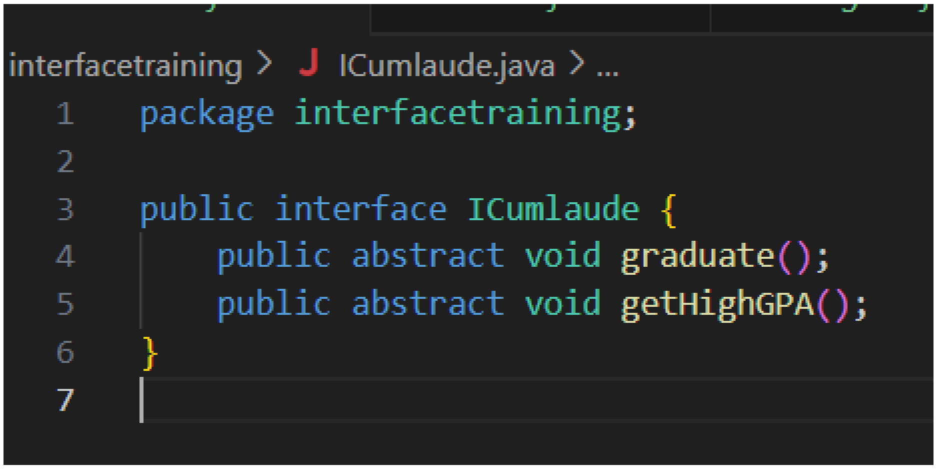
This setup proves that Mammal can extend Animal without implementing the move() method because it's also declared abstract. Only the concrete subclass Cat is required to implement move(). This is the essence of abstract class flexibility in Java.

- Experiment 2: Interface



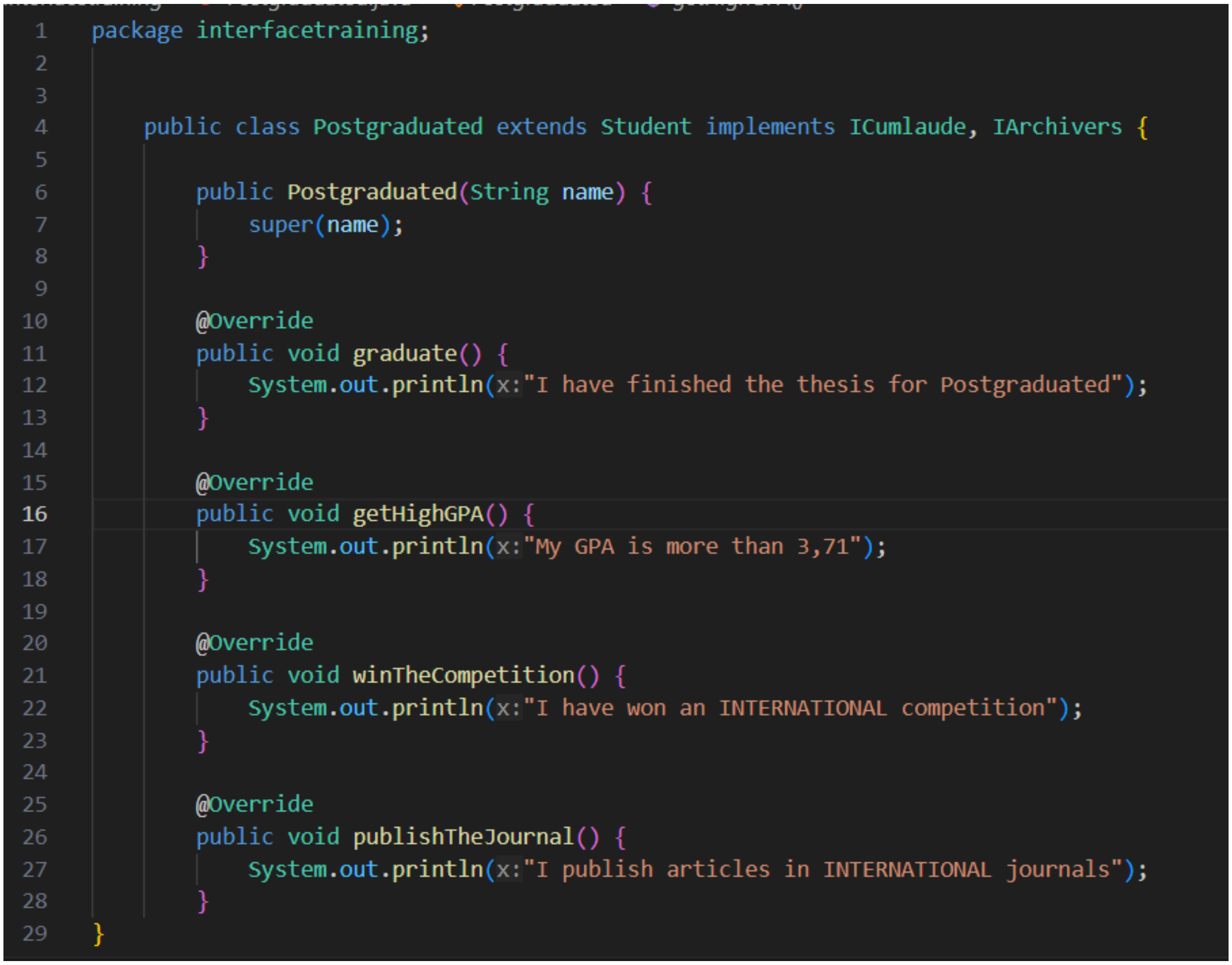
interfacetraining > IArchivers.java > ...

```
package interfacetraining;

publi  void interfacetraining.IArchivers.winTheCompetition()
    public abstract void winTheCompetition();
    public abstract void publishTheJournal();
}

```

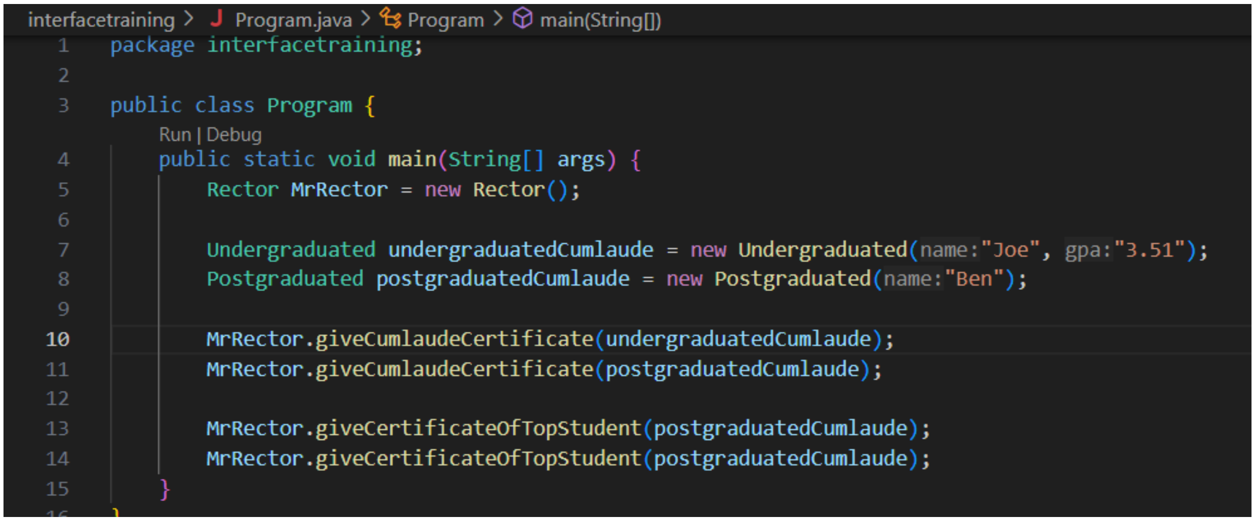


interfacetraining > ICumlaude.java > ...

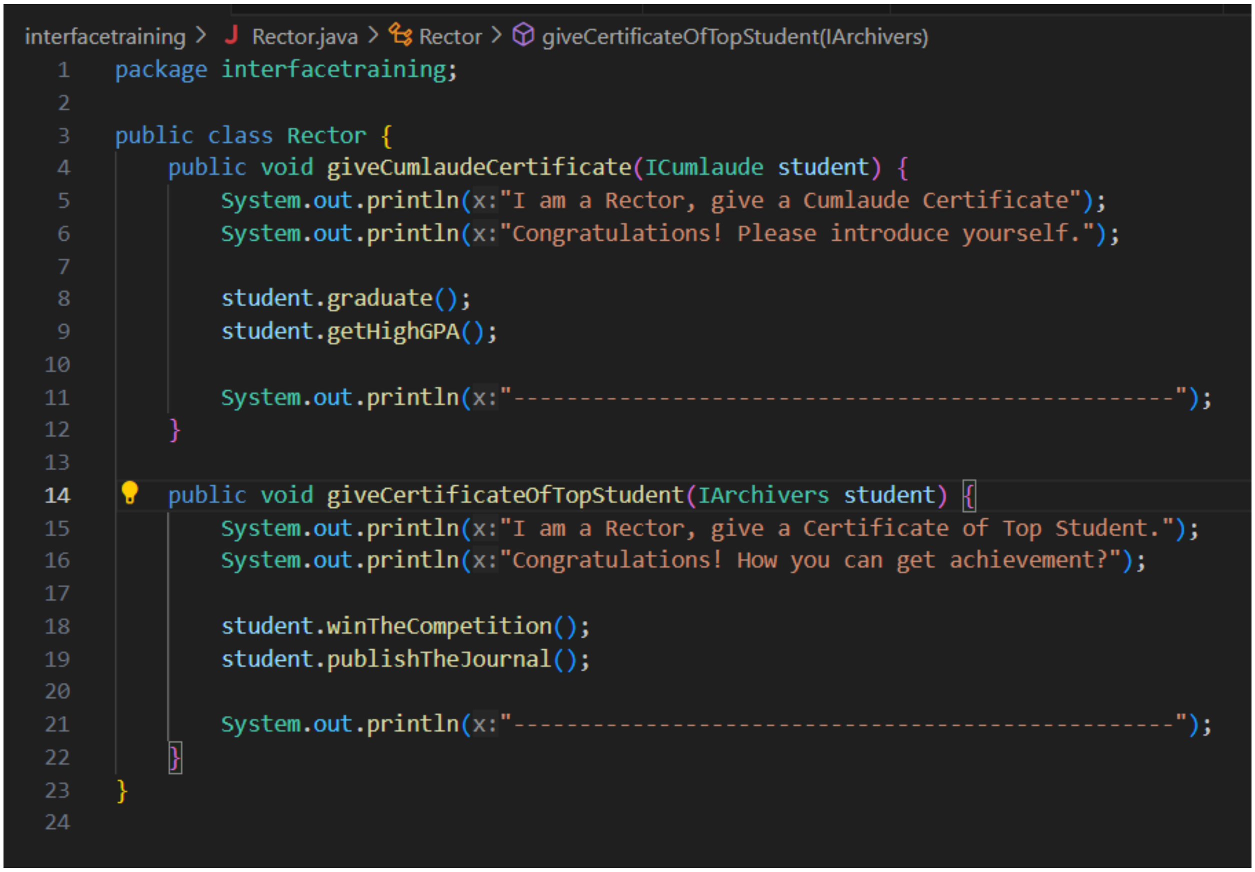
```
package interfacetraining;

public interface ICumlaude {
    public abstract void graduate();
    public abstract void getHighGPA();
}

```



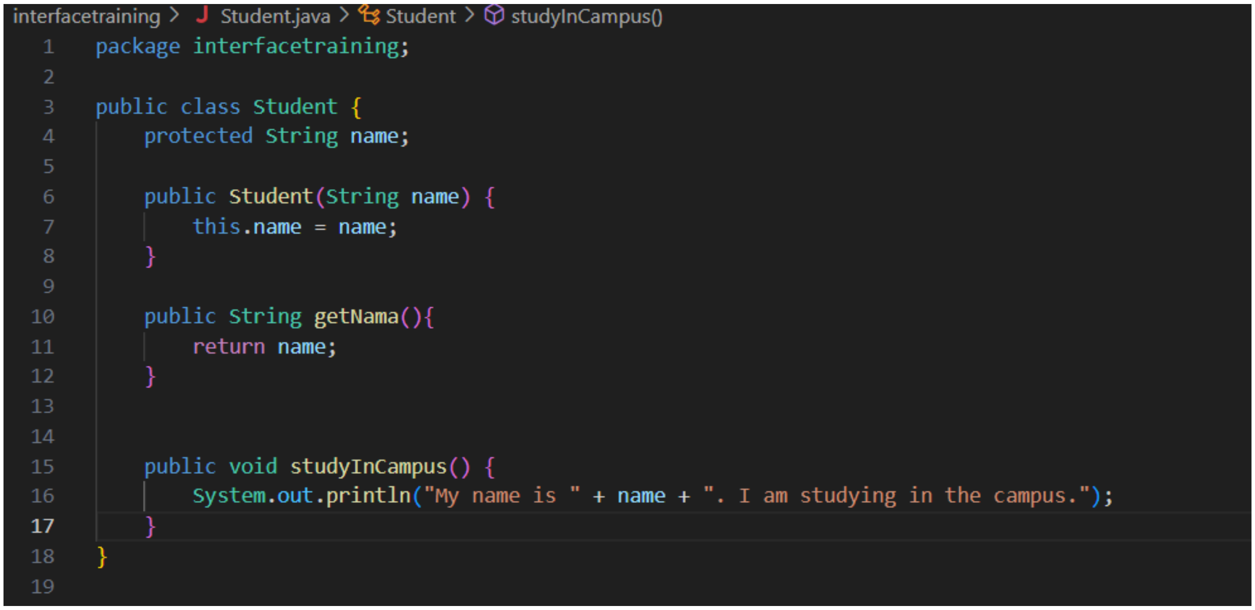
```java
package interfacetraining;


    public class Postgraduated extends Student implements ICumlaude, IArchivers {

        public Postgraduated(String name) {
            super(name);
        }

        @Override
        public void graduate() {
            System.out.println(x:"I have finished the thesis for Postgraduated");
        }

        @Override
        public void getHighGPA() {
            System.out.println(x:"My GPA is more than 3,71");
        }

        @Override
        public void winTheCompetition() {
            System.out.println(x:"I have won an INTERNATIONAL competition");
        }

        @Override
        public void publishTheJournal() {
            System.out.println(x:"I publish articles in INTERNATIONAL journals");
        }
}
```



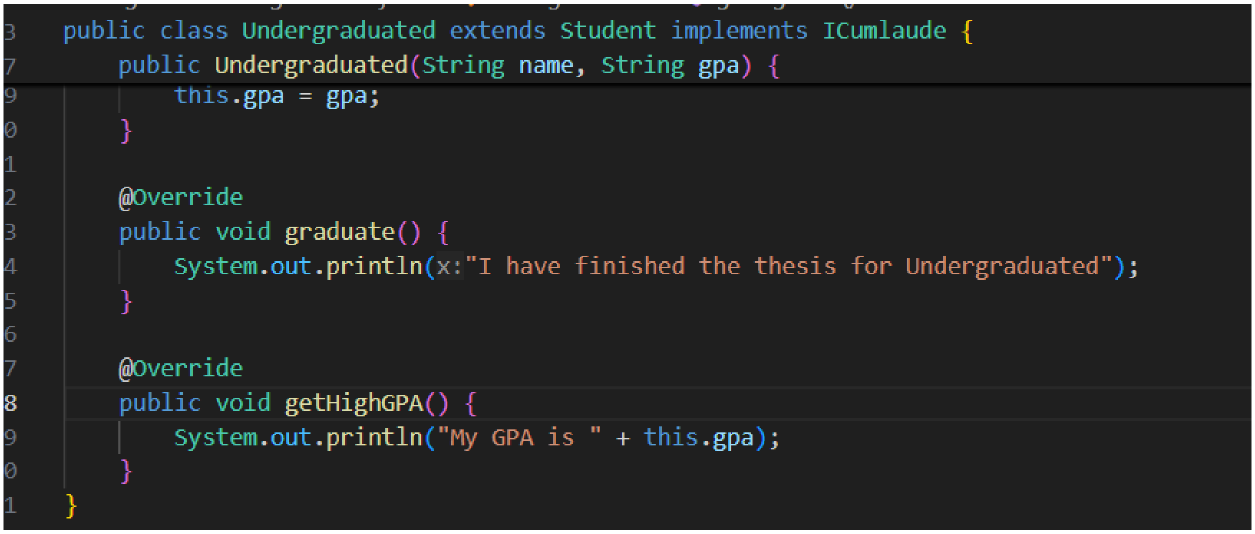
```java
package interfacetraining;

public class Program {
    public static void main(String[] args) {
        Rector MrRector = new Rector();

        Undergraduated undergraduatedCumlaude = new Undergraduated(name:"Joe", gpa:"3.51");
        Postgraduated postgraduatedCumlaude = new Postgraduated(name:"Ben");

        MrRector.giveCumlaudeCertificate(undergraduatedCumlaude);
        MrRector.giveCumlaudeCertificate(postgraduatedCumlaude);

        MrRector.giveCertificateOfTopStudent(postgraduatedCumlaude);
        MrRector.giveCertificateOfTopStudent(postgraduatedCumlaude);
    }
}
```


interfacetraining > Rector.java > Rector > giveCertificateOfTopStudent(IArchivers)

```java
package interfacetraining;

public class Rector {
    public void giveCumlaudeCertificate(ICumlaude student) {
        System.out.println(x:"I am a Rector, give a Cumlaude Certificate");
        System.out.println(x:"Congratulations! Please introduce yourself.");

        student.graduate();
        student.getHighGPA();

        System.out.println(x:"--------------------------------------------------");
    }

    public void giveCertificateOfTopStudent(IArchivers student) {
        System.out.println(x:"I am a Rector, give a Certificate of Top Student.");
        System.out.println(x:"Congratulations! How you can get achievement?");

        student.winTheCompetition();
        student.publishTheJournal();

        System.out.println(x:"--------------------------------------------------");
    }
}

```


interfacetraining > Student.java > Student > studyInCampus()

```java
package interfacetraining;

public class Student {
    protected String name;

    public Student(String name) {
        this.name = name;
    }

    public String getNama(){
        return name;
    }


    public void studyInCampus() {
        System.out.println("My name is " + name + ". I am studying in the campus.");
    }
}

```



```java
public class Undergraduated extends Student implements ICumlaude {
    public Undergraduated(String name, String gpa) {
        this.gpa = gpa;
    }

    @Override
    public void graduate() {
        System.out.println(x:"I have finished the thesis for Undergraduated");
    }

    @Override
    public void getHighGPA() {
        System.out.println("My GPA is " + this.gpa);
    }
}
```

a. Whydidstep 9 cause an error? Explain!

The code causes an error because the giveCumlaudeCertificate method in the Rector class expects an instance of ICumlaude as its parameter. However, in your main method, you are passing ordinaryStudent, which is an instance of the Student class, and Student does not implement the ICumlaude interface.

b. Canthe studyInCampus() method be called from the undergraduatedCumlaude object in the Program class? Why?

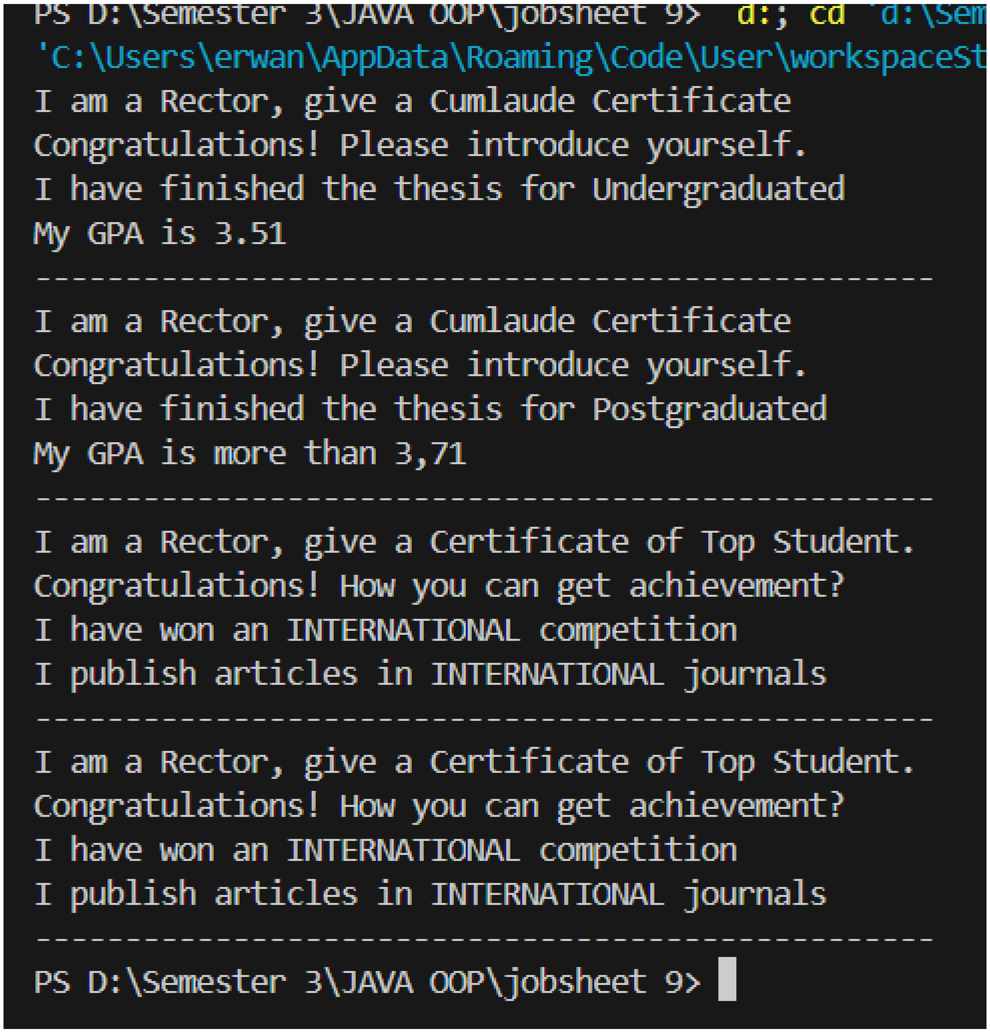
Since undergraduatedCumlaude is an Undergraduated object, and Undergraduated inherits from Student, it has access to all public and protected methods from Student, including studyInCampus(). Therefore, calling undergraduatedCumlaude.studyInCampus() is valid and will work without any issues.

c. Canthe studyInCampus() method be called from the student parameters in the giveCumlaudeCertificate() method in the Rector class? Why?

Directly calling studyInCampus() on student in giveCumlaudeCertificate is not possible because student is of type ICumlaude, which doesn’t have this method.

d. Modify the giveCumlaudeCertificate() method on the Rector class so that the results of the Program class execution become as follows

Experiment 3



```
PS D:\Semester 3\JAVA OOP\jobsheet 9>  d:; cd 'd:\Sem
'C:\Users\erwan\AppData\Roaming\Code\User\workspaceSt
I am a Rector, give a Cumlaude Certificate
Congratulations! Please introduce yourself.
I have finished the thesis for Undergraduated
My GPA is 3.51
--------------------------------------------------
I am a Rector, give a Cumlaude Certificate
Congratulations! Please introduce yourself.
I have finished the thesis for Postgraduated
My GPA is more than 3,71
--------------------------------------------------
I am a Rector, give a Certificate of Top Student.
Congratulations! How you can get achievement?
I have won an INTERNATIONAL competition
I publish articles in INTERNATIONAL journals
--------------------------------------------------
I am a Rector, give a Certificate of Top Student.
Congratulations! How you can get achievement?
I have won an INTERNATIONAL competition
I publish articles in INTERNATIONAL journals
--------------------------------------------------
PS D:\Semester 3\JAVA OOP\jobsheet 9> 
```