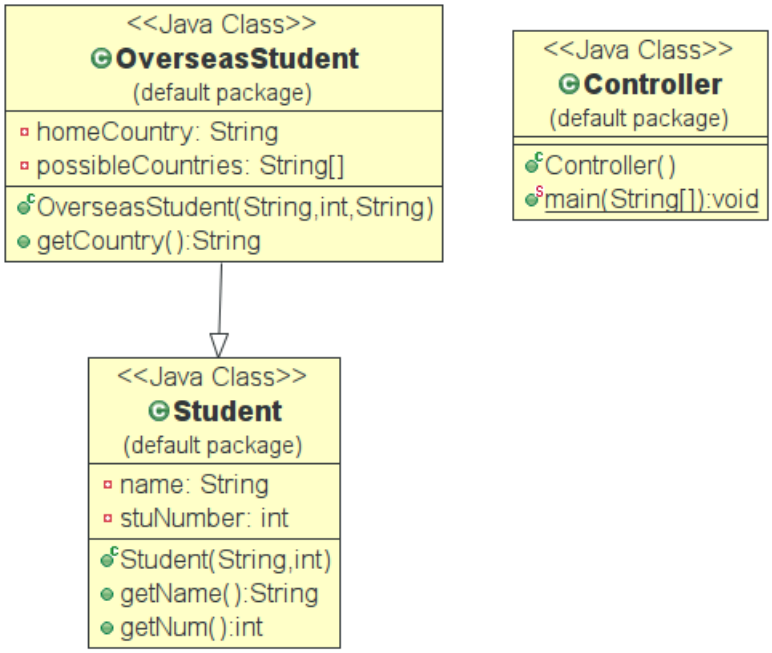
Java Programming Laboratory

Inheritance and Polymorphism

Exercise 1. Implement a subclass called OverseasStudent that extends a Student class as defined by the UML diagram below.



Develop a Controller class which creates home and overseas students to test these classes as shown below.

```
Student homeStud1 = new Student("Alan",1);
System.out.println(homeStud1.getName() + " number " + homeStud1.getNum());

Student homeStud2 = new Student("Jenny",2);
System.out.println(homeStud2.getName() + " number " + homeStud2.getNum());

Student homeStud3 = new Student("Jane",0);
System.out.println(homeStud3.getName() + " number " + homeStud3.getNum());

Student homeStud4 = new Student("James",10001);
System.out.println(homeStud4.getName() + " number " + homeStud4.getNum());

OverseasStudent overStud1 = new OverseasStudent("Pierre",1235,"France");
System.out.print(overStud1.getName() + " number " + overStud1.getNum() + " ");
System.out.println("Country " + overStud1.getCountry());

OverseasStudent overStud2 = new OverseasStudent("Klaus",1234,"Germany");
System.out.print(overStud2.getName() + " number " + overStud2.getNum() + " ");
System.out.println("Country " + overStud2.getCountry());
```

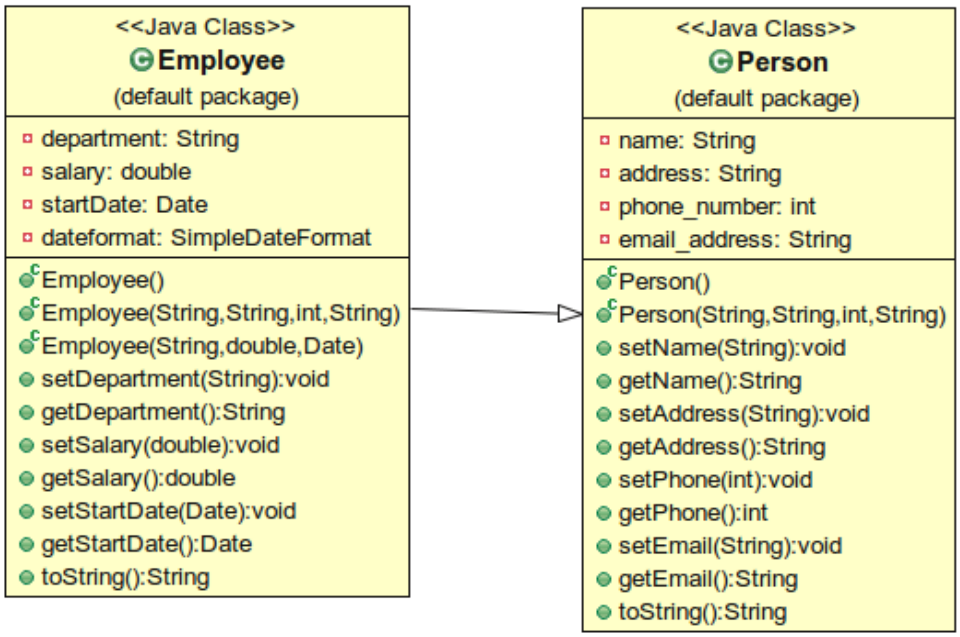
Add error checking to the Student class to display an error message if a student is created with a number outside the range between 1 and 10000. Also add error checking code to the OverseasStudent class to determine if overseas students come from the following set of countries: Spain, Nigeria, France and Germany. You can do

this by creating a String array of possible countries (more can be added) as shown below.

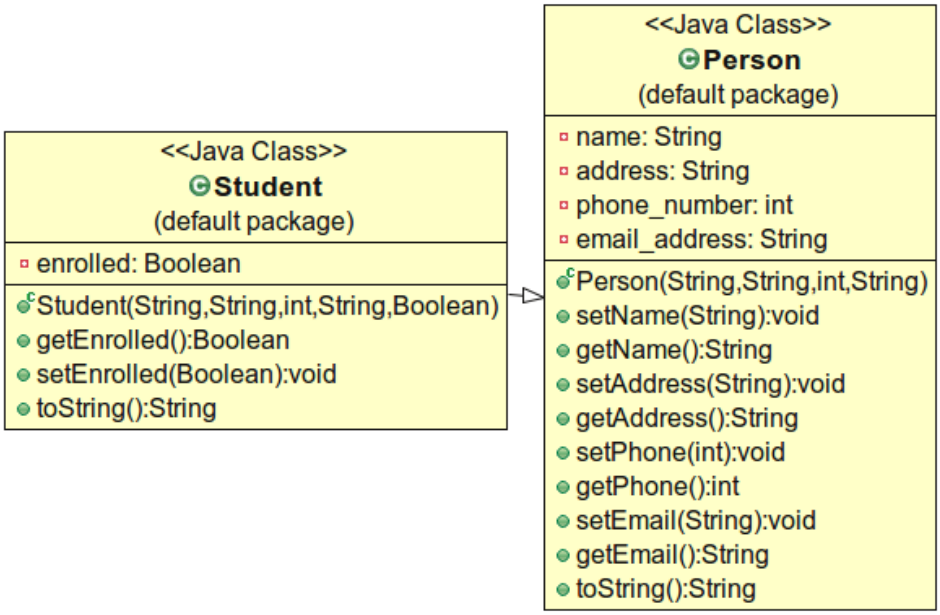
```
private String[] possibleCountries = {"Spain", "Nigeria", "France", "Germany"};
```

Then cycle through the possibleCountries array and test if the country of origin matches a possible country using the String class equals() method.

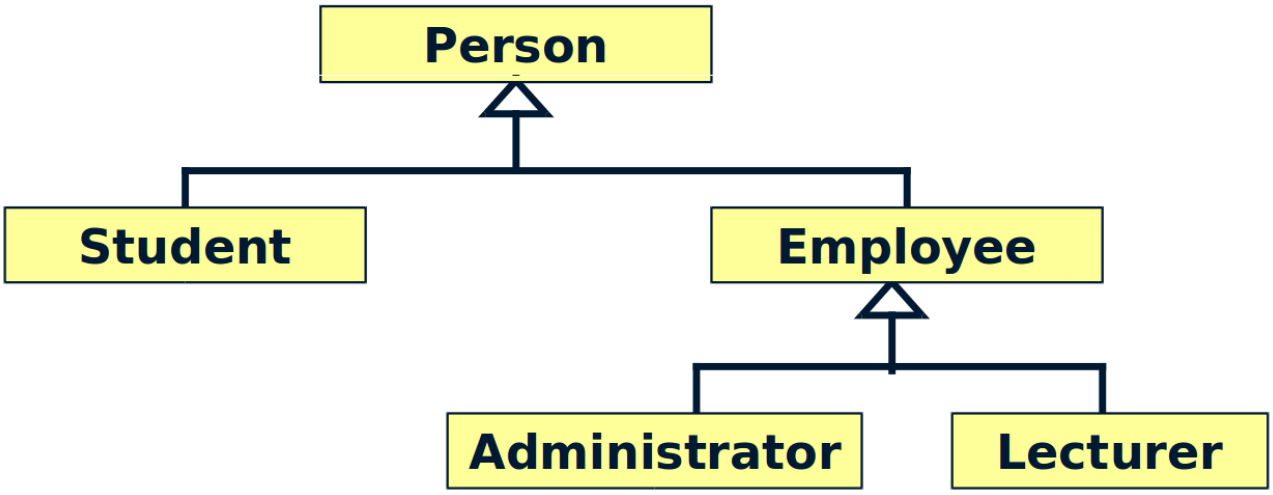
Exercise 2. Implement a subclass called Employee that extends a Person class defined by the UML diagram below. Develop a Controller class which creates two Employee objects and prints details of these to the console screen.

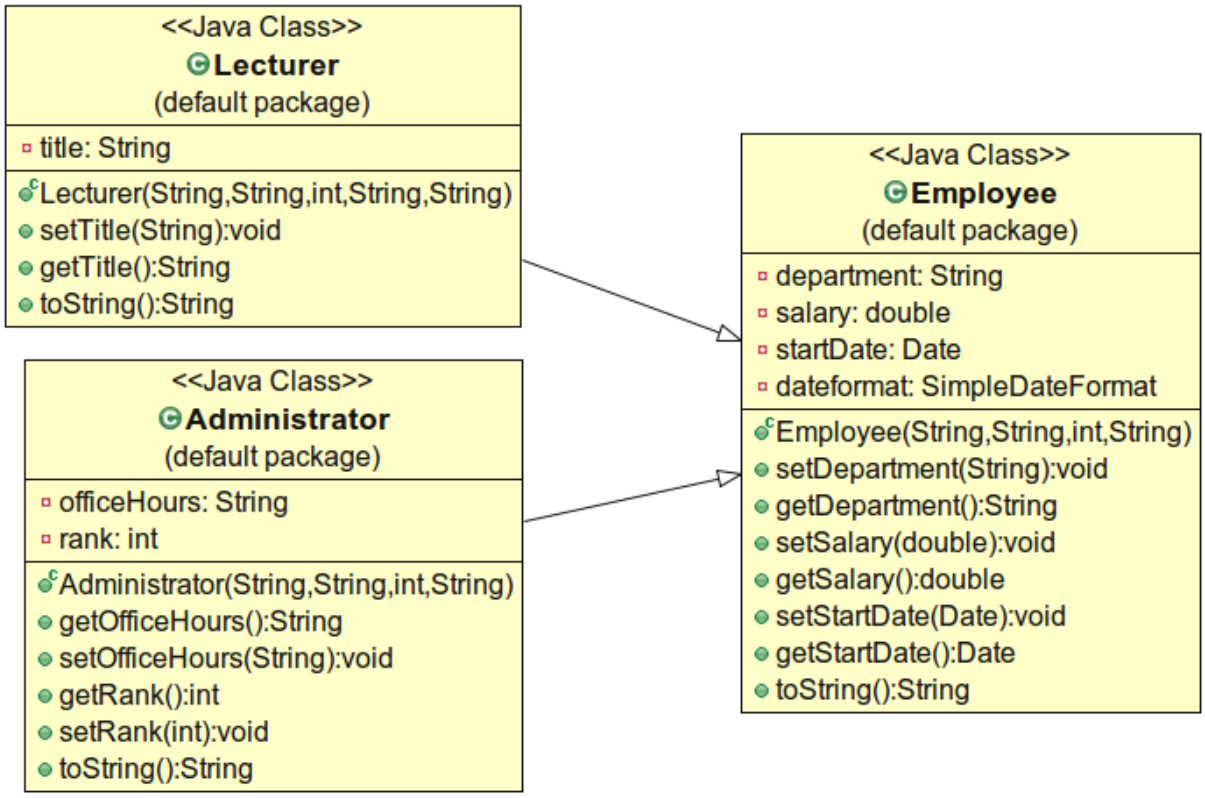


Exercise 3. Implement a subclass called Student that extends a Person class as defined by the UML diagram below. Develop a Controller class which tests the Student class by creating a Student object and printing details to the console screen.

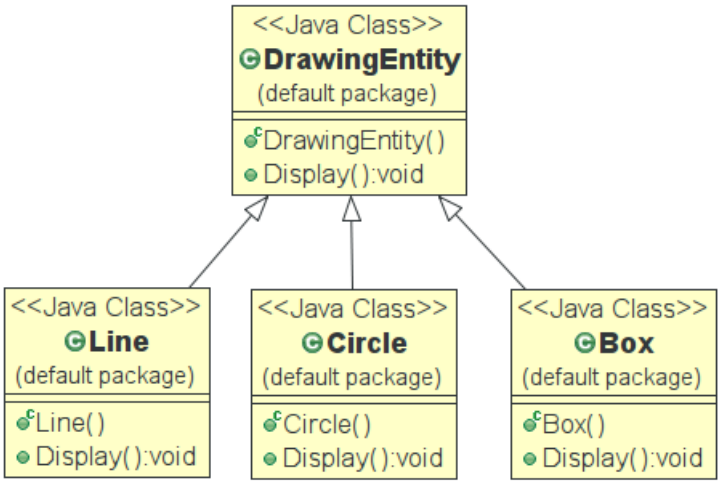


Exercise 4. Building on the classes developed in Exercises 2 and 3 implement the hierarchical inheritance structure shown in the diagram below. Develop a Controller class which creates Lecturer and Administrator objects and prints details to the console screen.





Exercise 5: Extend the polymorphism example discussed in the lecture to include a third class called Box which inherits from DrawingEntity as shown in the UML diagram below.



The Display method of the Box class should print “Box Object”. In the Controller class, add a Box object to the array and check that the run-time type of each object is invoked through polymorphism to execute the correct Display method.

Dr Alan Crispin
14-09-15