Java Programming Laboratory

Interfaces

Exercise 1. The following Person class implements the “Comparable” interface. Write a controller class which stores a set of Person objects in an array and sorts these according to age.

```
class Person implements Comparable
{
  private String firstName;
  private String lastName;
  private int age;

  public Person(String firstName,String  lastName, int age)
  {
    this.firstName = firstName;
    this.lastName=lastName;
    this.age=age;
  }
  public String getFirstName() {return firstName;  }
  public String getLastName() { return lastName;  }
  public int getAge() { return age;  }

  public int compareTo(Object anotherPerson) throws
ClassCastException {

    if (!(anotherPerson instanceof Person))throw new
ClassCastException("A Person object expected.");

    int anotherPersonAge = ((Person) anotherPerson).getAge();

    return this.age - anotherPersonAge;
  }
}
```

Exercise 2. The government has decided that vehicle tax is to be calculated based on the age of the vehicle. Car tax is to be calculated as £25 times the age of the car in years. Motor Bike tax is to be calculated as £15 times the age of the bike in years. Write an interface called “Vehicle” which has two methods as shown below.

```
public int TaxValue();
public String VehicleType();
```

Write two classes called Car and MotorBike which implement the Vehicle interface. The Car and MotorBike classes should have constructors which initialise the vehicle age in years. Write a controller to test your Car and MotorBike classes for a car of age 6 years and a motorbike of age 5 years displaying the vehicle type and tax to be paid.

Exercise 3. Add Javadoc comments to the interface and classes developed for Exercise 2 and using Eclipse generate API documentation in HTML format from the Javadoc comments in your source code.

Dr Alan Crispin
27-11-10