9주차 실습 보고서

강의명	객체지향프로그래밍및실습
담당교수	류기열
학과 학년	소프트웨어학과 2학년
학번	202220209
작성자	이욱준

2025.10.30.

□ 1

Car.Java

JAVA > 9 > Car.java > Car

```java
public class Car extends Vehicle
{
    public static final double BASE_VOLUME = 2000.0;  // 기준 배기량
    //private static int distanceRate = 50;         // 거리 요율, 삭제

    // private String no; 삭제
    // private int departure; 삭제
    // private int destination; 삭제
    private int volume;                 // 차량 배기량

    public Car(String no, int departure, int destination, int volume)
    {
        super(no, departure, destination);
        this.volume = volume;
    }
    @Override
    public int calcTollFee()                // 통행료 계산
    {
        double volumeRate = volume / BASE_VOLUME;
        // int distance = destination - departure; 삭제 및 수정
        int distance = getDestination() - getDeparture();
        // int toll = (int) ( distance * distanceRate * volumeRate ); 삭제 및 수정
        int toll = (int) (distance * getDistanceRate() * volumeRate);
        return toll;
    }
    // public String getNo() { return no; } 삭제
    // public int getDeparture() { return departure; } 삭제
    // public int getDestination() { return destination; } 삭제
    public int getVolume() { return volume; }
    // public static void setDistanceRate(int d) { distanceRate = d; } 삭제
    // public static int getDistanceRate() { return distanceRate; } 삭제
}
```

Bus.Java

JAVA > 9 > Bus.java > ...

```java
public class Bus extends Vehicle
{
    public static final double BASE_PASSENGER = 20.0;   // 기준 승객수
    //private static int distanceRate=50;           // 거리 요율, 삭제
    // private String no; 삭제
    // private int departure; 삭제
    // private int destination; 삭제
    private int passengers;             // 승객수

    public Bus(String no, int departure, int destination, int passengers)
    {
        super(no, departure, destination);
        this.passengers = passengers;
    }

    @Override
      public int calcTollFee()                  // 통행료 계산
      {
        double passengerRate = passengers / BASE_PASSENGER;
        // int distance = destination - departure; 삭제 및 수정
        int distance = getDestination() - getDeparture();
        // int toll = (int) (distance * distanceRate * passengerRate); 삭제 및 수정
        int toll = (int) (distance * getDistanceRate() * passengerRate);
        return toll;
      }
    //   public String getNo() { return no; } 삭제
    //   public int getDeparture() { return departure; } 삭제
    //   public int getDestination() { return destination; } 삭제
    public int getPassengers() { return passengers; }
    //   public static void setDistanceRate(int d) { distanceRate = d; } 삭제
    //   public static int getDistanceRate() { return distanceRate; } 삭제
}
```

Vehicle.Java

JAVA > 9 > Vehicle.java > Vehicle

```java
public abstract class Vehicle
{
    private int distanceRate = 50;
    private String no;
    private int departure;
    private int destination;

    public Vehicle(String no, int departure, int destination){
        this.no = no;
        this.departure = departure;
        this.destination = destination;
    }

    public abstract int calcTollFee();
    public String getNo() { return no; }
    public int getDeparture() { return departure; }
    public int getDestination() { return destination; }
    public void setDistanceRate(int d) { distanceRate = d; }
    public int getDistanceRate() { return distanceRate; }
}
```

실행결과

JAVA > 9 > Vehicle.java > Vehicle

```java
public abstract class Vehicle
{
    private int distanceRate = 50;
    private String no;
    private int departure;
    private int destination;

    public Vehicle(String no, int departure, int destination){
        this.no = no;
        this.departure = departure;
        this.destination = destination;
    }

    public abstract int calcTollFee();
    public String getNo() { return no; }
    public int getDeparture() { return departure; }
    public int getDestination() { return destination; }
    public void setDistanceRate(int d) { distanceRate = d; }
    public int getDistanceRate() { return distanceRate; }
}
```

PROBLEMS 41 OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
lwj@lwj-code:~/workspace/JAVA/9$ java VehicleTest
c1111 : 15000
c2222 : 10000
b3333 : 20000
```

About Polymorphism

위 코드에서 Car class와 Bus class는 calcTollFee()라는 공통된 메소드를 비롯하여 많은 코드들이 중복되어 사용되었다. 이를 Vehicle이라는 두 클래스를 sub로 갖는 super class를 만들어 inherit해줌으로서 calcTollFee()를 반드시 구현해야 하게끔하고, 각 하위 클래스들이 해당 calcTollFee()를 용도에 맞게 추가 구현함으로서 Polymorphism이 드러나고 있다. 이는 abstract class에서 제시하는 클래스의 구성을 하위 클래스들이 갖고 감으로서 전반적인 클래스들의 구성에 대한 파악이 쉬워지는 장점이 있다.

□ 2

Vehicle.java

JAVA > 9 > Vehicle.java > Vehicle > TollComparator > compare(Vehicle, Vehicle)

```
3   public abstract class Vehicle implements Comparable<Vehicle>
22
23      public int compareTo(Vehicle other){
24          return this.no.compareTo(other.getNo());
25      }
26
27      static class TollComparator implements Comparator<Vehicle> {
28          @Override
29          public int compare(Vehicle v1, Vehicle v2){
30              return v1.calcTollFee() - v2.calcTollFee();
31          }
32      };
33  }
```

(가)

JAVA > 9 > VehicleSortTest.java > VehicleSortTest > main(String[])

```
5   public class VehicleSortTest
7       public static void main(String[] args)
9           Vehicle[] vs = new Vehicle[3];
10          vs[0] = new Car("c1111", 0, 200, 3000);  // 3000은 volume
11          vs[1] = new Car("c2222", 100, 300, 2000);   // 2000은 volume
12          vs[2] = new Bus(no:"b3333", departure:200, destination:400, passengers:40
13
14          Arrays.sort(vs);
15          //Arrays.sort(vs, new Vehicle.TollComparator());
16
17          for(Vehicle c: vs)
18          {
19              System.out.printf(format:"%s : %d%n", c.getNo(), c.calcTollFee());
20          }
21      }
22  }
23
```

PROBLEMS 44 OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
lwj@lwj-code:~/workspace/JAVA/9$ java VehicleSortTest
b3333 : 20000
c1111 : 15000
c2222 : 10000
```

(나)

Bus.java | Car.java | Vehicle.java | VehicleSortTest.java 4

JAVA > 9 > VehicleSortTest.java > VehicleSortTest > main(String[])

```
5   public class VehicleSortTest
7       public static void main(String[] args)
9       {
10          Vehicle[] vs = new Vehicle[3];
11          vs[0] = new Car("c1111", 0, 200, 3000);  // 3000은 volume
12          vs[1] = new Car("c2222", 100, 300, 2000);   // 2000은 volume
13          vs[2] = new Bus(no:"b3333", departure:200, destination:400, passengers:40);
14
15          //Arrays.sort(vs);
16          Arrays.sort(vs, new Vehicle.TollComparator());
17
18          for(Vehicle c: vs)
19          {
20              System.out.printf(format:"%s : %d%n", c.getNo(), c.calcTollFee());
21          }
22      }
23  }
```

PROBLEMS 44 | OUTPUT | DEBUG CONSOLE | TERMINAL | PORTS

```
c2222 : 10000
c1111 : 15000
b3333 : 20000
```

☐ 3

Loan.java

JAVA > 9 > p3 > J Loan.java > ...

```java
public class Loan implements Payable{

    private final String name;
    private double principal;
    private double interest;

    public Loan(String name, double principal){
        this.name = name;
        this.principal = principal;
    }

    public String getName() {return name;}
    public double getPrincipal() {return principal;}
    public void setPrincipal(double principal) {this.principal = principal;}
    public double getInterest() {return interest;}
    public void setInterest(double interest) {this.interest = interest;}

    @Override
    public double getPaymentAmount()
    {
        return principal + interest;
    }

    @Override
    public String toString(){
        return "loan: " + this.name + "\nprincipal: " + this.principal + "\ninterest: " + this.interest;
    }
}
```

실행결과

```
lwj@lwj-code:~/workspace/JAVA/9/p3$ java PayableTest
Invoices and Employees processed polymorphically:

invoice: 01234 seat
quantity: 2
price per item:$375.00
payment due: $750.00

invoice: 56789 tire
quantity: 4
price per item:$79.95
payment due: $319.80

salaried employee: John Smith
social security number: 111-11-1111
weekly salary: $800.00
payment due: $800.00

salaried employee: Lisa Barnes
social security number: 888-88-8888
weekly salary: $1,200.00
payment due: $1,200.00

loan: 청년도약
principal: 1000.0
interest: 100.0
payment due: $1,100.00
```

□ 코드 전문

Car.java

```
public class Car extends Vehicle
{
    public static final double BASE_VOLUME = 2000.0;  // 기준 배기량
    //private static int distanceRate = 50;         // 거리 요율, 삭제

    // private String no; 삭제
    // private int departure; 삭제
    // private int destination; 삭제
    private int volume;                // 차량 배기량

    public Car(String no, int departure, int destination, int volume)
    {
        super(no, departure, destination);
        this.volume = volume;
    }
    @Override
    public int calcTollFee()              // 통행료 계산
    {
         double volumeRate = volume / BASE_VOLUME;
        // int distance = destination - departure; 삭제 및 수정
        int distance = getDestination() - getDeparture();
        // int toll = (int) ( distance * distanceRate * volumeRate ); 삭제 및
수정
        int toll = (int) (distance * getDistanceRate() * volumeRate);
         return toll;
    }
    // public String getNo() { return no; } 삭제
    // public int getDeparture() { return departure; } 삭제
    // public int getDestination() { return destination; } 삭제
    public int getVolume() { return volume; }
    // public static void setDistanceRate(int d) { distanceRate = d; } 삭제
    // public static int getDistanceRate() { return distanceRate; } 삭제
}
```

Bus.java

```java
public class Bus extends Vehicle
{
    public static final double BASE_PASSENGER = 20.0; // 기준 승객수
    //private static int distanceRate=50;     // 거리 요율, 삭제
    // private String no; 삭제
    // private int departure; 삭제
    // private int destination; 삭제
    private int passengers;       // 승객수

    public Bus(String no, int departure, int destination, int passengers)
    {
        super(no, departure, destination);
        this.passengers = passengers;
    }

    @Override
      public int calcTollFee()          // 통행료 계산
      {
         double passengerRate = passengers / BASE_PASSENGER;
        // int distance = destination - departure; 삭제 및 수정
        int distance = getDestination() - getDeparture();
        // int toll = (int) (distance * distanceRate * passengerRate); 삭제 및
수정
        int toll = (int) (distance * getDistanceRate() * passengerRate);
         return toll;
      }
    //   public String getNo() { return no; } 삭제
    //   public int getDeparture() { return departure; } 삭제
    //   public int getDestination() { return destination; } 삭제
    public int getPassengers() { return passengers; }
    //   public static void setDistanceRate(int d) { distanceRate = d; } 삭제
    //   public static int getDistanceRate() { return distanceRate; } 삭제
   }
```

Vehicle.java

```java
import java.util.Comparator;

public abstract class Vehicle implements Comparable<Vehicle>
{
    private int distanceRate = 50;
    private String no;
    private int departure;
    private int destination;

    public Vehicle(String no, int departure, int destination){
        this.no = no;
        this.departure = departure;
        this.destination = destination;
    }

    public abstract int calcTollFee();
    public String getNo() { return no; }
    public int getDeparture() { return departure; }
    public int getDestination() { return destination; }
    public void setDistanceRate(int d) { distanceRate = d; }
    public int getDistanceRate() { return distanceRate; }

    public int compareTo(Vehicle other){
        return this.no.compareTo(other.getNo());
    }

    static class TollComparator implements Comparator<Vehicle> {
        @Override
        public int compare(Vehicle v1, Vehicle v2){
            return v1.calcTollFee() - v2.calcTollFee();
        }
    };
}
```

Loan.java

```
public class Loan implements Payable{

    private final String name;
    private double principal;
    private double interest;

    public Loan(String name, double principal){
        this.name = name;
        this.principal = principal;
    }

    public String getName() {return name;}
    public double getPrincipal() {return principal;}
    public void setPrincipal(double principal) {this.principal = principal;}
    public double getInterest() {return interest;}
    public void setInterest(double interest) {this.interest = interest;}

    @Override
    public double getPaymentAmount()
    {
        return principal + interest;
    }

    @Override
    public String toString(){
        return "loan: " + this.name + "\nprincipal: " + this.principal +
"\ninterest: " + this.interest;
    }
}
```