

5주차 실습 보고서



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

2025.10. 2.

□ 1

```
29     // 자동차 정보와 현재 시간을 초기화한다.
30     public static void initialize()
31     {
32         //해당부분작성
33         carList = new Car[3];
34         carList[0] = new Car(no:"1111", speed:80, position:0);
35         carList[1] = new Car(no:"2222", speed:100, position:0);
36         carList[2] = new Car(no:"3333", speed:120, position:0);
37
38         currentTime = new Time(hh:0,mm:0);
39
40         in = new Scanner(System.in);
41     }
```

□ 2

```
73     // 주어진 시간(min) 만큼 이동한 자동차 위치를 조정한다
74     public static void moveAllCars(int min)
75     {
76         //해당부분작성
77         for(Car car : carList){
78             car.move(min);
79             if(car.getPosition() > 500){
80                 car.setPosition(position:500);
81             }
82         }
83     }
```

Car.java에 getter/setter를 생성

```
J HCTS.java 2 J Time.java 1 X J Car.java X
JAVA > 5 > A > J Car.java > Car
1 public class Car
6
7     public Car(String no, int speed, double position){
8         this.no =no;
9         this.speed=speed;
10        this.position = position;
11    }
12
13    public String getNo() {return no;}
14    public int getSpeed() {return speed;}
15    public double getPosition() {return position;}
16
17    public void setNo(String no) {
18        this.no = no;
19    }
20    public void setSpeed(int speed) {
21        this.speed = speed;
22    }
23    public void setPosition(double position) {
24        this.position = position;
25    }
26
```

□ 3

```
19     public int calculateDifference(Time t)
20     {
21         //해당부분작성
22         return (this.hour-t.hour)*60 + this.minute - t.minute;
23     }
24 }
```

□ 4

```
51 // 현재 시간 설정 명령('t') 처리
52 public static void handleSettingTime()
53 {
54     //해당부분작성
55     Time t = new Time(in.nextInt(), in.nextInt());
56     int min = t.calculateDifference(currentTime);
57
58     setCurrentTime(t);
59     moveAllCars(min);
60
61 }
```

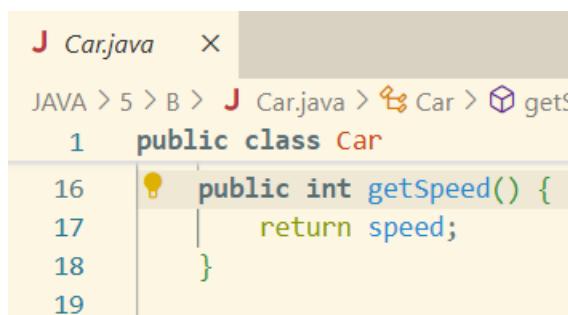
□ 5

```
63     // 차량 정보 출력 명령('s') 처리
64     public static void handleLocatingCars()
65     {
66         //해당부분작성
67         for(Car car:carList){
68             System.out.println(car.toString());
69         }
70     }
```

□ 7

```
18 < public int calcTollFee()
19 {
20     //해당부분작성
21     return 1000 + (int)((this.destination - this.departure) * 50 * car.getSpeed() / 100);
22 }
23
24 }
```

Car.java에 speed에 대한 getter 생성



```
J Car.java  X
JAVA > 5 > B > J Car.java > Car > getS
  1 public class Car
 16   16     public int getSpeed() {
 17       17         return speed;
 18     }
 19
```

□ 8

The screenshot shows a Java development environment with three tabs at the top: Car.java, MovingCar.java, and MovingCarTest.java. The MovingCarTest.java tab is active, displaying the following code:

```
1 public class MovingCarTest{
2     public static void main(String[] args)
3     {
4         //해당부분작성
5         Car car = new Car(no:"c1111", speed:120, position:100);
6         MovingCar mc = new MovingCar(car, dept:0, dest:400);
7         System.out.println(mc.getCar().toString());
8         System.out.println(mc.calcTollFee());
9     }
10 }
```

Below the code editor, there are tabs for PROBLEMS, OUTPUT, DEBUG CONSOLE, TERMINAL, and PORTS. The TERMINAL tab is selected, showing the following terminal output:

- lwj@ljw-code:~/workspace/JAVA/5/B\$ java MovingCarTest
- No=c1111, Speed=120km/h, Position=100.0km
- 25000

□ 코드 전문

Q1~Q6

Car.java

```
public class Car
{
    private String no;
    private int speed;
    private double position;

    public Car(String no, int speed, double position){
        this.no =no;
        this.speed=speed;
        this.position = position;
    }

    public String getNo() {return no;}
    public int getSpeed() {return speed;}
    public double getPosition() {return position;}

    public void setNo(String no) {
        this.no = no;
    }
    public void setSpeed(int speed) {
        this.speed = speed;
    }
    public void setPosition(double position) {
        this.position = position;
    }

    public void move(int min){

        this.position += (double) this.speed *min / 60.0;
    }

    @Override
    public String toString() {

        return "No=" + this.no + ", Speed=" + this.speed + "km/h, Position=" +
this.position +"km";
    }
}
```

```
public int compareTo(Car c){

    if (this.position > c.position){

        return 1;

    }
    else if (this.position < c.position){

        return -1;

    }
    else {

        return 0;
    }
}
```

Time.java

```
public class Time {
    private int hour;
    private int minute;

    public Time(int hh, int mm) {
        this.hour = hh;
        this.minute = mm;
    }

    // Getter 메소드
    public int getHour() {
        return this.hour;
    }

    public int getMinute() {
        return this.minute;
    }
}
```

```
public int calculateDifference(Time t)
{
    //해당부분작성
    return (this.hour-t.hour)*60 + this.minute - t.minute;
}

@Override
public String toString() {
    return String.format("%02d:%02d", this.hour, this.minute);
}

public void add(int hh, int mm) {
    this.minute += mm;
    if (this.minute >= 60) {
        this.hour += this.minute / 60;
        this.minute %= 60;
    }

    this.hour += hh;
    this.hour %= 24;
}

public void add(Time t) {
    this.add(t.getHour(), t.getMinute());
}
}
```

Q7~Q8

MovingCar.java

```
public class MovingCar
{
    Car car;
    int departure;
    int destination;

    public Car getCar(){
        return this.car;
    }

    public MovingCar(Car c, int dept, int dest)
    {
        car = c;
        this.departure = dept;
        this.destination = dest;
    }

    public int calcTollFee()
    {
        //해당부분작성
        return 1000 + (int)((this.destination - this.departure) * 50 *
car.getSpeed() / 100);
    }
}
```

MovingCarTest.java

```
public class MovingCarTest{
    public static void main(String[] args)
    {
        //해당부분작성
        Car car = new Car("c1111", 120, 100);
        MovingCar mc = new MovingCar(car, 0, 400);
        System.out.println(mc.getCar().toString());
        System.out.println(mc.calcTollFee());
    }
}
```