

객체지향프로그래밍 및 실습 – 4주차 실습활동지(9월25일)

성명: _____ 학과: _____ 학번: _____ 실습반: _____

제출방식: 보고서(파일명: report_4)를 제출한다. 보고서는 pdf 형식으로 저장한다.

제출기한: 9월 26일(금) 23시59분까지 (지연 제출 0점 처리함, 미완성이라도 반드시 제출하길 바람)

I. Objectives

1. Class를 정의하고 object를 생성하여 사용하는 기본 방법을 이해한다.
2. 메소드를 작성하고 사용할 수 있다.
3. Overloaded method를 작성하고 사용할 수 있다.
4. Reference type에 대해 이해하고 parameter passing에 활용할 수 있다.

II. Exercises (15점)

1. 1, 2주차 실습활동지를 참조하여 다음 설명을 읽고 아래 Car 클래스에 필요한 메소드를 추가하고자 한다. 물음에 답하시오.

```
// Car.java
public class Car
{
    private String no;
    private int speed;
    private double position;
}
```

- 1-(가) Car class에 no와 speed, position 값을 매개변수로 전달받아 Car 객체를 초기화하는 constructor를 작성하시오. (1점) [constructor 코드 포함]

```
public Car(String no, int speed, double position) { ... }
```

- 1-(나) Car 객체를 주어진 시간 min (분 단위)만큼 이동시키는 move() 메소드를 구현하시오. (1점) [method 코드 포함]

```
public void move(int min) { ... } // assume min >= 0
```

- 1-(다) Car 객체를 하나의 String 값으로 반환하는 toString() method를 작성하시오. (1점) [method 코드 포함]

```
public String toString() { ... }
```

단, `toString()`이 반환하는 `String`은 아래 형식(예시)과 같다.

No=c1111,Speed=80km/h,Position=120km

- 1-(라) 하나의 `Car` 객체를 매개변수로 전달받아 `this` 객체와 비교하는 `compareTo()` method를 작성하시오. `this` 객체가 크면 1, 작으면 -1, 같으면 0을 반환함. (1점) [method 코드 포함]

```
public int compareTo(Car c) { ... }
```

단, 두 개의 `Car` 객체 `c1`과 `c2`를 비교하는 방법은 아래와 같다.

```
if c1.position > c2.position, then c1 > c2  
if c1.position== c2.position, then c1 == c2
```

2. 1번 문제에서 정의한 `Car` 클래스를 이용하여 다음 각 문항에 적절한 문장들을 순서대로 아래 `main()` 메소드에 채운 후 실행해 보시오.

```
// CarTest.java  
public class CarTest  
{  
    public static void main(String[] args)  
    {  
        Car[] carList = new Car[2];  
        // 여기 채울 것  
    }  
}
```

- 2-(가) 다음 초기값을 이용하여 두 개의 `car` 객체를 생성하여 `carList`에 저장하시오. (1점)[코드]

각 `Car` 객체의 초기값 (`no, speed, position`)은 각각 다음과 같다.
(“c1111”, 80, 0)
(“c2222”, 100, 0)

- 2-(나) (가)에서 정의한 두 개의 `Car` 객체를 각각 60분씩 이동시키는 문장을 작성하시오. 단, `enhanced-for` 루프를 이용할 것. (1점)[코드]

- 2-(다) (나)에서 이동시킨 두 `Car` 객체를 `compareTo()` 메소드를 이용하여 비교하여 더 큰 차를 `toString()` 메소드를 이용하여 출력하는 코드를 추가하시오. 크기가 동일하면 “equal”을 출력한다.

(1점)[코드]

2-(라) (가-다)를 추가한 코드를 실행해 보시오. (1점)[실행결과]

3. 1, 2주차 실습활동지를 참조하여 다음 설명을 읽고 아래 Time 클래스를 완성하고자 한다. 물음에 답 하시오.

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

    public Time(int hh, int mm) // 0<= hh < 24, 0<= mm < 60
    {
        hour = hh; minute = mm;
    }

    public int getHour() { return hour; }
    public int getMinute() { return minute; }
    // 여기에 추가할 것
}
```

- 3-(가) Time 클래스에 Time 객체를 String 값으로 반환하는 `toString()`메소드를 추가하시오. (1점)
[method 코드 포함]

```
public String toString() { ... }
```

반환하는 String 형식은 hh:mm이다. 예시: 13시 30분 => 13:30, 0시 5분 => 00:05
힌트) 지정된 자리수의 정수와 앞자리에 0을 채우기 위해 `String.format()` 메소드 사용

- 3-(나) Time 클래스에 매개변수에 주어진 시간만큼 증가시키는 `add()` 메소드를 다음 두 가지 방식으로 추가하시오. 단, 두 번째 `add()` 메소드는 첫 번째 `add()`메소드를 이용하여 구현할 것.
 $0 \leq \text{minute} \leq 60$, $0 \leq \text{hour} \leq 24$. 가령, 증가한 hour가 25시간이라면 1로 한다. (2점) [method 코드 포함]

```
public void add(int hh, int mm) { ... } // hh를 hour에, mm을 minute에 각각 증가시킬 것
                                         // assume  $0 \leq \text{hh} < 24$ ,  $0 \leq \text{mm} < 60$ 
public void add(Time t) // t 만큼 증가시킬 것
```

4. 3번 문제에서 정의한 Time 클래스를 이용하여 다음 각 문항에 적절한 문장들을 순서대로 아래 main() 메소드에 채운 후 실행해 보시오.

```
// TimeTest.java
public class TimeTest
{
    public static void main(String[] args)
    {

        Time t1 = new Time(23, 30);
        Time t2 = new Time(2, 40);
        // 여기 채울 것
    }
}
```

4-(가) 두 개의 Time 객체를 `toString()` 메소드를 이용하여 각각 출력하시오. (1점)[코드 포함]

4-(나) `toString()` 메소드를 사용하지 않고 (가)번과 동일한 결과가 나오도록 문장을 작성하시오. (1점)
[코드 포함]

4-(다) t1의 시간을 t2만큼 증가시키는 문장을 추가한 후 t1을 출력하는 문장을 추가하시오. (1점)[코드 포함]

4-(라) (가-다)를 추가한 코드를 실행해 보시오. (1점)[실행결과]