

객체지향프로그래밍및실습 – 1주차 실습활동지 (9월 4일)

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

제출방식: 보고서(파일명: report_1.pdf)를 Bb로 제출한다. 보고서는 반드시 pdf 형식이어야 함.

제출기한: 9월 5일(금) 23:59 (지연 제출 불가)

I. Objectives

1. Java 프로그래밍환경을 설치하고 사용법을 숙지한다(JDK 및 jupyter notebook, VSCode).
2. Java 프로그램의 기본적인 구조와 Java 클래스 파일의 관계에 대해 이해한다.
3. 구조적 프로그래밍 스타일로 작성된 Java 프로그램의 구조를 C언어와 비교하여 학습한다.

II. Exercises - 15점

1. 아래에서 주어진 코드(HCTS.java)는 강의자료(Structured Programming Review)의 Highway Car Tracking System을 간소화하여 Java 언어로 구현한 것이다. 다음 물음에 답하시오. [15점]

가) 프로그램을 jupyter notebook VSCode를 이용하여 실행해보시오. [보고서 포함 안함]

실행 예시:

```
Enter a command: t 1 30
Enter a command: l
car no. = 1111, speed = 80km/h, car position = 120.0km
car no. = 2222, speed = 100km/h, car position = 150.0km
car no. = 3333, speed = 120km/h, car position = 180.0km
Enter a command: t 2 0
Enter a command: l
car no. = 1111, speed = 80km/h, car position = 160.0km
car no. = 2222, speed = 100km/h, car position = 200.0km
car no. = 3333, speed = 120km/h, car position = 240.0km
Enter a command:
```

나) 이 프로그램을 C 언어로 프로그램을 작성한다고 가정하고 어떤 부분이 유사하고 어떤 부분이 다른지, 다음의 관점에서 비교해보시오. 이 코드는 Java 언어의 구조적 프로그래밍 요소를 이용하여 작성한 것임. 두 언어를 일반적으로 비교하지 말고 아래 코드에 나타나는 경우에 대해서만 예를 들어 설명할 것. [아래 테이블을 채워 보고서에 포함-15점]

- 1) 프로그램의 구조
- 2) 함수의 구조 (매개변수, return 타입, 함수 선언, 함수 호출 및 return 문 등)
- 3) 데이터 타입 및 변수 선언 (정수형, 문자열 등)
- 4) 제어문 (if, while 등)

- 5) 배열
- 6) 스트링
- 7) 구조체
- 8) 입출력

	유사성	차이점
프로그램구조		
함수의 구조		
데이터 타입		
제어문		
배열		
스트링		
구조체		
입출력		
기타		

```
// HCTS.java
import java.util.*;

public class HCTS
{
    static Car[] car_list;
    static Time current_time;
    static Scanner in;

    public static void main(String[] args)
    {
        initialize();

        do {
            char cmd = read_command();
            if(cmd=='t')
                handle_setting_time();
            else if(cmd=='l')
                handle_locating_cars();
            else
                System.out.printf("command error: %s%n", cmd);
        } while(true);

    } // end of main()

    // 자동차 정보와 현재 시간을 초기화한다.
    public static void initialize()
    {
        car_list = new Car[3];

        for(int i=0; i<car_list.length; i++)
            car_list[i] = new Car();

        car_list[0].no = 1111;
    }
}
```

```

        car_list[0].speed = 80;
        car_list[0].position = 0;

        car_list[1].no = 2222;
        car_list[1].speed = 100;
        car_list[1].position = 0;

        car_list[2].no = 3333;
        car_list[2].speed = 120;
        car_list[2].position = 0;

        current_time = new Time();
        current_time.hour = 0;
        current_time.minute = 0;

        in = new Scanner(System.in);
    }

    // 명령어를 읽어들인다
    public static char read_command()
    {
        System.out.printf("Enter a command: ");
        String s = in.next();
        return s.charAt(0); // 읽어들인 스트링의 첫번째 문자를 명령어로 인식
    }

    // 현재 시간 설정 명령('t') 처리
    public static void handle_setting_time()
    {
        Time t = new Time();
        t.hour = in.nextInt();
        t.minute = in.nextInt();

        int min = calculate_time_difference(t);
        set_current_time(t);
        move_all_cars(min);
    }

    // 차량 정보 출력 명령('s') 처리
    public static void handle_locating_cars()
    {
        for(int i=0; i<car_list.length; i++)
            print_car_info(car_list[i]);
    }

    // 새로운 current_time을 설정한다
    public static void set_current_time(Time t)
    {
        current_time = t;
    }

```

```

// 시간의 차이(분단위) t - current_time를 계산한다
public static int calculate_time_difference(Time t)
{
    return (t.hour-current_time.hour)*60 + (t.minute-current_time.minute);
}

// 주어진 시간(min) 만큼 이동한 자동차 위치를 조정한다
public static void move_all_cars(int min)
{
    for(int i=0; i<car_list.length; i++)
    {
        car_list[i].position += (double) car_list[i].speed/60*min;
        if(car_list[i].position > 500.0)
            car_list[i].position = 500.0; // 목적지에 도착
    }
}

// 차량 정보(자동차 번호, 및 현재 위치)를 출력한다
public static void print_car_info(Car c)
{
    System.out.printf("car no. = %d, car position = %f%n", c.no, c.position);
}
} // end of CarManager class

class Time
{
    int hour;
    int minute;
}

class Car
{
    int no;
    int speed;
    double position;
}

```