

객체지향프로그래밍 – 6주차 실습활동지

성명: _____ 학과: _____ 학번: _____

※ 본 실습활동지에 대한 보고서나 코드를 작성함에 있어 교재나 강의노트를 제외한 다른 자료로부터 일부 또는 전체를 복사하였습니까? 예() 아니오()

제출방식: 별도의 답안지 문서에 각 문제에 대하여 작성한 답안이나 작성한 코드 및 그 실행 결과를 복사해 넣는다. 답안지 문서명은 "report_주차_학번" (예, report_6_201811111)로 만든다. 답안지는 pdf형식으로 저장하여 과제게시판(Bb)에 차주 월요일 23시59분까지 제출한다.

I. Objectives

1. Java 언어에서의 object composition 관계에 대해 학습한다.
2. Java 언어에서의 method(or constructor) overloading 방법에 대해 학습한다.
3. Java 언어에서의 final variable의 용도에 대해 학습한다.
4. Java 언어에서의 static variable 및 static methods에 대해 공부한다.

II. Exercises

1. 아래 코드를 (Employee.java 및 Date.java)를 이용하여 물음에 답하시오. (15점)

- 필요한 library import할 것

```
// Employee.java
public class Employee
{
    private final String name;
    private double salary;
    private final Date hireDay;

    public Employee(String n, double s, int year, int month,
        int day)
    {
        name = n;
        salary = s;
        hireDay = new Date(year, month, day);
    }

    public String getName() { return name; }
    public double getSalary() { return salary; }
    public Date getHireDay() { return hireDay; }

    public void raiseSalary(double byPercent)
    {
        double raise = salary * byPercent / 100;
        salary += raise;
    }
}
```

```

} // end of Employee

// Date.java
public class Date
{
    private int year;
    private int month;
    private int day;
    public Date(int yy, int mm, int dd)
    {
        year=yy; month=mm; day=dd;
    }
    public int getYear() { return year; }
    public int getMonth() { return month; }
    public int getDay() { return day; }
    public void setDate(Date d)
    {
        year = d.getYear();
        month = d.getMonth();
        day = d.getDay();
    }
    public String toString()
    { return year+"."+month+"."+day; }
}

```

가) Employee 클래스를 테스트 하는 프로그램이다. 아래 코드를 수행한 결과가 무엇이고 어떻게 나왔는지 설명하시오. 입력데이터로 각각 2020, 4, 13을 제공한다. [수행결과 및 설명 포함] (1점)

```

public class EmployeeTest
{
    public static void main(String[] args)
    {
        Scanner in = new Scanner(System.in);
        int yy = in.nextInt();
        int mm = in.nextInt();
        int dd = in.nextInt();
        Employee e = new Employee("Hong", 100000.0, yy, mm, dd); // (A)
        printEmployee(e);
    }
    public static void printEmployee(Employee e)
    {
        System.out.println("Name: "+e.getName());
        System.out.println("Salary: "+e.getSalary());
        System.out.println("Hireday: "+e.getHireDay().toString());
    }
}

```

- 나) (가) 코드의 (A) 부분을 아래와 같이 변경하고자 한다. 이러한 방식으로 Employee객체를 생성하기 위해 Employee 클래스에 알맞은 constructor를 추가하시오. 수행 결과는? **[constructor 코드 및 수행 결과 포함] (3점)**

```
Date d = new Date(yy, mm, dd);  
Employee e = new Employee("Hong", 100000.0, d);
```

- 다) (나) 코드에서 yy, mm, dd를 잘못된 날짜 가령, **2021, 2, 29**(존재하지 않는 날짜)를 입력하여 수행해 보자. 어떤 일이 발생하는가? 입력 데이터(날짜)의 유효성을 체크하기 위해 Date 클래스의 constructor를 수정하시오. 윤년을 검사하기 위해 java.util.GregorianCalendar class의 isLeapYear() method를 이용하시오. 유효하지 않은 데이터라면 어떻게 처리할 수 있는지 설명하시오. **[수행결과 및 수정한 코드부분, 설명 포함] (3점)**

- 라) (다)의 문제를 해결하기 위해 유효성 체크를 하는 static method(가령, isValid())를 Date 클래스에 추가하고, Date 객체를 생성하기 전에 입력데이터의 유효성을 체크한 후 유효한 데이터라면 Date 객체를 생성하고, 유효하지 않은 데이터라면 유효한 데이터가 입력될 때까지 다시 입력 받아 Date객체를 생성하도록 (B) 부분을 적절하게 채우시오. 정상적인 데이터와 유효하지 않은 데이터 두 가지에 대해 수행하시오. ((가)의 데이터 입력부분을 포함하여 수정함)
[수정한 코드부분, 수행결과 포함] (3점)

```
(B)  
Date d = new Date(yy, mm, dd);  
Employee e = new Employee("Hong", 100000.0, d);  
  
// in Date class  
public static boolean isValid(int yy, int mm, int dd) { ... }
```

- 마) Employee클래스의 instance variable 중 name과 hireDay의 선언에서 final은 왜 사용하는지 설명하시오. **[설명 포함] (1점)**

- 바) (가) 코드 (A) 부분을 다음과 같이 변경하여 수행하여 그 결과를 본 후 (마)의 관점에서 어떤 문제가 발생했는지 또 그 이유가 무엇인지 설명하시오. **[수행결과 및 설명 포함] (2점)**

```
Date d = new Date(yy, mm, dd); // 2020, 4, 13으로 입력  
Employee e = new Employee("Hong", 100000.0, d);  
printEmployee(e);  
Date d2 = e.getHireday();  
d2.setDate(new Date(2020, 3, 15));
```

- 사) Date class에서 getYear(), getMonth(), getDay() 메소드를 모두 제거한다면 setDate() 메소드를 어떻게 수정할 수 있겠는가? 수정한 후 (바)의 코드를 수행해보시오. 수정할 수 있는 이론적 근거에 대해 설명하시오. **[setDate() 코드 및 수행결과, 근거에 대한 설명 포함] (2점)**