

# 자바프로그래밍

## Lab 01 (10점)

1. 다음과 같은 조건의 Employee, EmployeeTest 클래스를 생성하라. (6점)

(1) Employee

① 세 개의 인스턴스 변수

접근지정자는 모두 private

a first name ( type String ), a last name ( type String ), monthly salary ( double )

② 생성자 ( Constructor )

세 개의 매개변수를 통해 전달 받은 값으로 각 인스턴스 변수를 초기화한다.

③ 각 인스턴스 변수에 대한 set, get method ( 즉, setter, getter )

만약, monthly salary가 양수가 아니라면, 그 값을 세팅하지 않는다.

※ setter, getter를 직접 작성하거나 이클립스의 자동 추가기능을 사용할 수도 있음

(인스턴스 변수 선택 > 우측버튼 > source > Generate Getters and Setters...)

(2) EmployeeTest

Employee 클래스의 여러 가지 특성들을 테스트하는 클래스이다.

① 두 개의 Employee 객체를 생성하고 각 객체의 yearly salary를 출력한다.

② 두 객체에 대해 10% 인상된 yearly salary를 출력한다.

```
Problems Javadoc Declaration Console
<terminated> EmployeeTest [Java Application] C:\Program Files\Java\jre1.8.0_73\bin\javaw.exe (2016. 3. 10. 오후 12:54:57)
Employee 1: Bob Jones; Yearly Salary: 34500.00
Employee 2: Susan Baker; Yearly Salary: 37809.00

Increasing employee salaries by 10%
Employee 1: Bob Jones; Yearly Salary: 37950.00
Employee 2: Susan Baker; Yearly Salary: 41589.90
```

2. 위 소스에 대해 각 Employee 객체 생성 시 필요한 초기 값들을 사용자로부터 입력받도록 수정하라.(4점)

```
Problems Javadoc Declaration Console
<terminated> EmployeeTest (1) [Java Application] C:\Program Files\Java\jre1.8.0_73\bin\javaw.exe (2016. 3. 10. 오후 12:57:18)
Enter first name: Gildong
Enter last name: Hong
Enter monthly salary: 100

Enter first name: Nemo
Enter last name: Kim
Enter monthly salary: 150

Employee 1: Gildong Hong; Yearly Salary: 1200.00
Employee 2: Nemo Kim; Yearly Salary: 1800.00

Increasing employee salaries by 10%
Employee 1: Gildong Hong; Yearly Salary: 1320.00
Employee 2: Nemo Kim; Yearly Salary: 1980.00
```

※ 참고: 데이터 입력 오류의 예와 해결방법

아래 case1은 입력오류의 예, case2와 case3은 제대로 된 입력의 예입니다.

```
Scanner input = new Scanner( System.in );
```

```
String firstName;  
String lastName;  
double monthlySalary;
```

```
System.out.print( "Enter first name: " );  
firstName = input.next();  
System.out.print( "Enter last name: " );  
lastName = input.next();  
System.out.print( "Enter monthly salary: " );  
monthlySalary = input.nextDouble(); // 숫자 입력후 치는 엔터키가 버퍼에 \n 문자로 남아 있음  
Employee employee1 = new Employee( firstName, lastName, monthlySalary );
```

```
/* // Case1 : 데이터 입력 오류의 예 - 숫자 입력 후 nextLine()을 사용한 경우  
System.out.print( "Enter first name: " );  
firstName = input.nextLine(); // 버퍼에 있는 '\n' 문자를 한 줄의 문장으로 인식해서 이 문장을 실행해 버림  
System.out.print( "Enter last name: " ); // 그래서, 두 개의 프롬프트가 연속적으로 실행되면서  
lastName = input.next(); // 입력 대기 상태가 됨  
System.out.print( "Enter monthly salary: " );  
monthlySalary = input.nextDouble();  
*/
```

```
/* // Case 2: 데이터 입력 OK - 숫자 입력 후 next()를 사용한 경우  
System.out.print( "Enter first name: " );  
firstName = input.next(); // 버퍼에 있는 '\n' (white character)를 무시하고 새로운 단어를 입력하도록 기다림, 단어만 입력 가능  
System.out.print( "Enter last name: " );  
lastName = input.next();  
System.out.print( "Enter monthly salary: " );  
monthlySalary = input.nextDouble();  
*/
```

```
// Case 3: 데이터 입력 OK - 숫자 입력 후 버퍼에 남아 있는 '\n'을 제거한 후 nextLine()를 사용, 한줄 문장 입력 가능  
//{  
System.out.println();  
System.out.print( "Enter first name: " );  
  
// 키보드 버퍼 clear  
if(input.hasNextLine())  
    input.nextLine();  
  
firstName = input.nextLine(); // 한 줄 문장 입력  
System.out.print( "Enter last name: " );  
lastName = input.next();  
System.out.print( "Enter monthly salary: " );  
monthlySalary = input.nextDouble();  
//}  
  
Employee employee2 = new Employee( firstName, lastName, monthlySalary );
```

■ 제출 형식

워크스페이스 이름 : Lab 01

프로젝트 이름 : 1, 2

- 각 소스파일에 주석처리

“학번 이름”

“본인은 이 소스파일을 다른 사람의 소스를 복사하지 않고 직접 작성하였습니다.”

제출파일 :

(1) 실행화면을 캡처한 보고서의 pdf 파일

(2) 문제별로 디렉토리 만들고 소스파일(.java) 추가한 압축파일

■ 주의

- 소스복사로는 실력향상을 기대할 수 없습니다!!!

- 1차 마감 : 수업일 자정 (만점)

- 2차 마감 : 수업 다음날 자정(50%)