

객체지향프로그래밍 - 6주차 실습 답안지

성명: 조우연

학과: 수학과

학번: 201921195

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

가) EmployeeTest.java의 실행과 결과, 결과설명

(EmployeeTest.java는 실습활동지의 코드와 동일하게 따른다.)

-EmployeeTest.java실행결과
(입력데이터를 각각 2020,4,13으로 함)

```
Console
<terminated> EmployeeTest
2020
4
13
Name: Hong
Salary: 100000.0
Hireday: 2020:4:13
```

EmployeeTest.java의 main 메소드에서 'in'변수를 이용해 사용자에게 값을 차례로 입력받아 각각 정수형 변수 yy, mm, dd에 저장한다. Employee클래스로 정의된 e는 Employee constructor인 public Employee(String n, double s, int year, int month, int day)를 이용해 e의 name, salary의 정보를 다음과 같이 저장할 수 있다.("Hong", 100000.0). 여기서 hireDay는 Date 클래스로 정의된 변수이므로 다시 Date의 constructor인 public Date(int yy, int mm, int dd)를 이용해 hireDay

의 year, month, day 정보를 각각(2020, 4, 13)으로 저장할 수 있다. 따라서 e에 Name, Salary, HireDay의 정보를 각각 constructor를 이용해 할당해줄 수 있고, Employee클래스의 메소드인 getName(), getSalary(), getHireDay()를 통해 e의 각각의 Name, Salary, HireDay의 값을 얻을 수 있다. 이를 출력하는 static method인 printEmployee(Employee e)를 이용해 e의 employee정보를 출력했고, 그 결과가 위의 이미지와 같다. (단, getSalary()는 Date타입의 값을 반환하기 때문에 또다시 Date클래스의 toString()메소드를 호출해 hireDate의 정보를 문자열로 출력할 수 있게끔 만든다.)

나) (가)에서 실행한 코드의 (A)부분을 실습노트에 주어진 대로 변경 후, Employee객체를 생성하기 위해 Employee클래스에 새로운 constructor를 추가 후, 수행

- (A)를 수정 후 수행결과


```
Console
<terminated> EmployeeTest
2020
4
13
Name: Hong
Salary: 100000.0
Hireday: 2020:4:13
```

=> (가)와 같은 결과를 출력한다.

- 추가된 Employee constructor

```
public Employee(String n, double s, Date d)
{
    name = n;
    salary = s;
    hireDay = d;
}
```

다) (나)코드에서 존재하지 않는 날짜를 입력해 수행한 결과, 날짜의 유효성을 체크하기 위해 Date 클래스의 constructor를 수정한 코드 및 유효하지 않은 데이터를 처리하기 위한 방법 설명

- 잘못된 날짜(2021,2,29)를 입력해 수행한 결과	- 수정된 Date클래스의 constructor
 <p>-> 잘못된 날짜임에도 불구하고 Hireday에 그대로 사용한다.</p>	<pre> public Date (int yy, int mm, int dd) { boolean TF; GregorianCalendar calendar = new GregorianCalendar(yy,mm,dd); TF = calendar.isLeapYear(yy); if (dd>31 dd<1) return; if (mm>12 mm<1) return; switch(mm) { case 4 : if(dd>30) return; case 6 : if(dd>30) return; case 9 : if(dd>30) return; case 11 : if(dd>30) return; case 2 : if(TF==false && dd>28) return; else if(TF==true && dd>29) return; } year=yy; month=mm; day=dd; } </pre> <p>-> 입력된 mm(월)이 1~12범위 안에 있지 않거나, dd(일)이 유효하지 않은 범위에 있으면 year, month, day의 값을 default값으로 ضبط하도록 만들었다.</p>

- 유효하지 않은 데이터라면 어떻게 처리할 수 있는지 설명
<p>위의 수정한 constructor는 유효성을 체크해서 유효하지 않으면 default값을 주도록 수정했다. 하지만 이는 main()에서 다시 Hireday의 값이 default값인지 확인해야하는 등의 번거로움과 입력받은 yy,mm,dd가 유효하지 않음에도 계속해서 constructor를 실행한다는 면에서 효율적이지 못하므로, 일반적으로 Date클래스에 유효성을 체크하는 static 메소드를 만들어 constructor를 실행하기 전에 입력받은 yy,mm,dd를 main()에서 유효성을 체크하고, 유효하지 않다면 다시 yy,mm,dd를 입력받을 수 있도록 하는 것이 좋은 방법일 것이다.</p>

라) 유효성 체크를 하는 static method를 Date클래스에 추가하고, 실습(라)에서의 요구사항에 맞게 (B)를 수정 후 프로그램을 수행한 결과

- 유효성체크를 하는 static method 코드	- 수정된 (B)
<pre> public static boolean isValid(int yy, int mm, int dd) { boolean TF; //윤달인지 확인하기 위한 변수 GregorianCalendar calendar = new GregorianCalendar(yy,mm,dd); TF = calendar.isLeapYear(yy); //윤달이면 TF는 true //dd가 1~31사이에 있지 않으면 false반환 if (dd>31 dd<1) return false; //mm이 1~12사이에 있지 않으면 false반환 if (mm>12 mm<1) return false; switch(mm) { //4,6,8,11월은 30일까지 있으므로, dd가 30이 넘으면 false반환 case 4 : if(dd>30) return false; case 6 : if(dd>30) return false; case 9 : if(dd>30) return false; case 11 : if(dd>30) return false; //2월에는 윤달인 경우, dd가 29일을 넘으면 false반환 //윤달이 아닌 경우, dd가 28일을 넘으면 false반환 case 2 : if(TF==false && dd>28) return false; else if(TF==true && dd>29) return false; } //유효한 mm,dd를 가졌다면 true반환 return true; } </pre> <p>-> 윤달 뿐만 아니라, mm(월)이 1~12사이가 아닌 경우, dd(일)이 1~31(30 혹은 28,29)이 아니면 유효하지 않은 것으로 간주</p>	<pre> public static void main(String[] args) { Scanner in = new Scanner(System.in); int yy,mm,dd; while(true) { System.out.println("년 : "); yy = in.nextInt(); System.out.println("월 : "); mm = in.nextInt(); System.out.println("일 : "); dd = in.nextInt(); if(Date.isValid(yy,mm,dd)==true) break; System.out.println("유효하지 않은 날을 입력하셨습니다. \n다시 입력하세요."); } Date d = new Date(yy,mm,dd); Employee e = new Employee("Hong", 100000.0, d); printEmployee(e); } </pre> <p>- 실행결과 (위:유효하지 않은 데이터, 아래:유효한 데이터)</p> <div data-bbox="863 1003 1410 1720"> <pre> Console <terminated> EmployeeTest [Java Applicat 년 : 2021 월 : 2 일 : 29 유효하지 않은 날을 입력하셨습니다. 다시 입력하세요. 년 : 2020 월 : 10 일 : 12 Name: Hong Salary: 100000.0 Hireday: 2020:10:12 </pre> </div>

마) Employee클래스의 instance variable 중 name과 hireDay의 선언에서 왜 final을 사용하였는가?

Employee클래스의 instance variable 중 name과 hireDay는 constructor를 실행한 후, 바뀌지 않을 값이기 때문에 final로 선언하였다. final변수는 한번만 초기화 가능하다. 즉, final 변수에는 한번 값을 할당하고 나면 값을 변경할 수 없다. Employee클래스의 instance variable인 name, hireDay, salary 중에서 salary는 증가, 감소하며 값이 변할 수 있는 변수이지만, name과 hireDay는 바뀔 필요가 없다. 이름과 고용일은 한번 정해지면 그 변수가 사라지기 전까지는 계속 같은 이름과 고용일을 가지기 때문이다. 이렇듯 final변수는 한번 초기화하고 값을 변경할 필요가 없을 때 사용해주다.

바) (가)코드 (A)부분을 실습활동지에 주어진 것처럼 변경하여 수행하고, 그 결과를 (마) 관점에서 어떤 문제가 발생했는지, 그 이유는 무엇인지 설명하시오.

- (A)부분 변경 후 수행결과	- 발생한 문제와 이유
<pre><terminated> EmployeeTest Name: Hong Salary: 100000.0 Hireday: 2020:4:13 Name: Hong Salary: 100000.0 Hireday: 2020:3:15</pre>	<p>Date d2 = e.getHire();를 실행함으로써 d2는 e의 hireDay를 가리키게 된다. d2.setDate(new Date(2020,3,15);를 실행했더니 d2(즉 hireDay)의 year, month, day가 각각 2020, 3, 15로 바뀌었다. hireDay는 final변수이기 때문에 바뀌면 그 값이 변할 수 없는데 값이 변하는 문제가 발생하였다. 이러한 결과는 아마도 Date d2 = e.getHire();을 실행할 때, d2자체가 final변수로 선언되지 않았기 때문에 d2에 변수 타입도 함께 assign 되지 않고, hireDay자체의 값만 전달되었기 때문일 것 같다. 즉, e가 Date d2 = e.getHire();을 실행하면서 (hireDay와 같은 값을 가지지만, 선언된 타입(final의 유무)은 다른)d2를 hireDay대신 가리키고 있고, d2는 값을 변경할 수 있으므로 hireDay의 값이 변경된 것처럼 보이는 것으로 생각된다.</p>

사) Date class를 수정 후, setDate()코드 및 수행결과, 수정할 수 있는 이론적 근거

- setDate() 코드	- 수정할 수 있는 이론적 근거
<pre>public void setDate(Date d) { this.year = d.year; this.month = d.month; this.day = d.day; }</pre>	<p>Date 클래스에 정의된 instance variable들 year, month, day는 모두 private으로 선언되어 있다. 이러한 private 데이터는 클래스 내에서만 접근할 수 있으며, 같은 클래스의 메소드 내에서도 접근할 수 있다. 클래스 내의 메소드는 기본적으로 instance variable을 사용할 때, this.을 붙여 실행된다.(생략가능) (바)에서 d2.setDate(new Date(2020,3,15))를 실행했을 때, 매개변수로 받은 Date 타입의 d, 즉 (바)코드에서는 new Date(2020,3,15)의 year를 d2의 year로, d의 month는 d2의 month로, d의 day는 d2의 day로 assign된다. 이것이 가능한 이유는 앞서 말한 것과 같이, private data는 클래스내의 메소드에서 접근이 가능하기 때문이다. 따라서 this.year, this.month, this.day는 당연히 접근 가능하고, setDate()의 parameter로 받는 값도 Date 클래스 변수이기 때문에, d의 year, month, day도 접근이 가능하므로, setDate() 메소드를 왼쪽 코드와 같이 실행시킬 수 있는 것이다.</p>
<p>- 수정 후 실행결과</p> <pre>Console <terminated> EmployeeTest Name: Hong Salary: 100000.0 Hireday: 2020:4:13 Name: Hong Salary: 100000.0 Hireday: 2020:3:15</pre>	