



Lab 0626. Chapter4

실습 1> Person 클래스 작성 (Person.java, PersonTest.java)

- Person 클래스는 다음과 같은 필드와 메소드를 가진다.

구분	이름	설명
필드	name	이름(private)
	mobile	핸드폰 번호(private)
	office	직장 전화번호(private)
	email	이메일 주소(private)
메소드	Person(n, m, o, e)	생성자 메소드
	toString()	Person의 정보를 문자열로 반환하는 메소드
	setName(), getName(), ...	각 필드에 대한 접근자와 설정자 메소드

모든 필드는 private 으로 설정, 생성자 및 메소드는 public으로 설정

실습 1> Person 클래스 작성

```
class Person {
    // 필드 작성

    // 생성자 작성

    @Override
    public String toString() {
        return "_____"; // 출력 결과 참고해서 문자열 구성하기
    }
    // 각 필드에 대해 접근자와 설정자 메소드 작성
}

public class PersonTest {
    public static void main(String[] args) {
        Person obj = new Person("Kim", "01012345678", "027104567", "a@b.c");
        System.out.println(obj);
    }
}
```

```
<terminated> PersonTest [Java Application] C:\Program Files\Java\jdk-17.0.1\bin\javaw.exe
Person [name=Kim, mobile=01012345678, office=027104567, email=a@b.c]
```

실습 2> Song 클래스 작성 (Song.java, SongTest.java)

- Song 클래스는 다음과 같은 필드를 가진다.

구분	이름	설명
필드	title	노래의 제목
	artist	가수
	length	곡의 길이(단위: 초)

- 다음 테스트 프로그램을 실행하기 위해 필요한 생성자 메소드들과 toString()를 작성하십시오.

- this()도 사용해보자.

모든 필드는 private 으로 설정,
메소드는 public으로 설정

```
class Song{  
    String title;  
    String artist;  
    int length;  
    // 생성자 여러개 중복 정의  
    // this() 호출 활용  
    // toString 메소드 재정의  
}
```

실습 2>

```
public class SongTest    {
    public static void main(String[] args)    {
        Song s1 = new Song("Outward Bound", "Nana", 180);
        Song s2 = new Song("Jambalya", "Carpenters");
        Song s3 = new Song("Yesterday");
        Song s4 = new Song();
        System.out.println(s1);
        System.out.println(s2);
        System.out.println(s3);
        System.out.println(s4);
    }
}
```

```
<terminated> SongTest [Java Application] C:\Program Files\Java
Song [제목:Outward Bound, 가수:Nana, 곡의 길이:180]
Song [제목:Jambalya, 가수:Carpenters, 곡의 길이:0]
Song [제목:Yesterday, 가수:정보 없음, 곡의 길이:0]
Song [제목:제목 없음, 가수:정보 없음, 곡의 길이:0]
```

실습 3> Movie 클래스 작성 (Movie.java, MovieTest.java)

- 영화 Movie 클래스를 정의하여 보자. Movie 클래스는 영화 제목, 감독, 발표된 연도, 평점의 필드를 가진다.
- 필요한 생성자를 정의하고, 영화의 모든 정보를 문자열로 요약하여 반환하는 toString() 메소드를 작성하시오.

```
class Movie {  
    // 필드 선언  
  
    // 생성자 정의  
  
    // toString 메소드 작성  
}
```

모든 필드는 private 으로 설정,
생성자와 메소드는 public으로 설정

실습 3>

```
import java.util.Scanner;
public class MovieTest {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("제목: ");
        String s = sc.nextLine();
        System.out.print("감독: ");
        String d = sc.nextLine();
        System.out.print("연도: ");
        int year = sc.nextInt();
        System.out.print("평점: ");
        double rating = sc.nextDouble();
        Movie m= new Movie(s, rating, d, year);
        System.out.println(m);
    }
}
```

<terminated> MovieTest [Java Application] C:\Program Files\Java\jdk-17.0

제목: 공조2

감독: 이석훈

연도: 2022

평점: 8.7

Movie [title=공조2, rating=8.7, director=이석훈, year=2022]

실습 4> Student 클래스 작성

- 이름, 학번, 나이를 사용자로부터 입력 받아서 학생 객체를 생성하고 생성 결과를 출력하는 프로그램을 작성하시오.
 - 학생은 이름(name)과 학번(studentID), 나이(age) 필드를 가진다.
 - 모든 필드는 private으로 선언한다.
 - 필요한 생성자 메소드 작성
 - 모든 필드에 대하여 접근자, 설정자 메소드 작성 (사용하지 않더라도 작성할 것)
 - 학생의 정보를 문자열로 반환하는 toString() 작성

모든 필드는 private 으로 설정,
생성자와 메소드는 public으로 설정

```
class Student{  
    // 필드 선언  
  
    // 생성자 정의  
  
    // 각 필드에 대한 접근자, 생성자 메소드 작성  
  
    // toString 메소드 작성  
}
```


실습 4>

```
public class StudentTest    {
    public static void main(String[] args)    {
        Scanner sc = new Scanner(System.in);
        System.out.print("학생의 이름: ");
        String name = sc.next();
        System.out.print("학생의 학번: ");
        String id = sc.next();
        System.out.print("학생의 나이: ");
        int age = sc.nextInt();

        Student s1 = new Student(name, id, age);
        System.out.println(s1);
    }
}
```

```
<terminated> StudentTest [Java Application] C:\W
학생의 이름: 이숙명
학생의 학번: 12345
학생의 나이: 21
Student [이름=이숙명, 학번=12345, 나이=21]
```

실습 5> MyDate 클래스 작성

- 사용자로부터 연도, 월, 일의 정보를 입력받아서 날짜 형식으로 출력하는 프로그램을 작성하시오.
- MyDate 클래스는 연도, 월, 일 등의 속성을 가지며
 - 생성자와 다음과 같은 날짜 출력 메소드 2개를 가진다.
 - 날짜를 “2012.7.16”과 같이 출력하는 메소드 printDate1()
 - 날짜를 “July 12, 2012”와 같이 출력하는 메소드 printDate2()

```
<terminated> DateTest [Java Appl  
연도: 2012  
월: 7  
일: 3  
2012.7.3  
Jul 3, 2012
```

```
<terminated> DateTest [Ja  
연도: 2022  
월: 9  
일: 22  
2022.9.22  
Sep 22, 2022
```

- 월의 영어 문자열 정보는 문자열 배열 활용

```
{"Jan", "Feb", "Mar", "Apr", "May", "Jun",  
"Jul", "Aug", "Sep", "Oct", "Nov", "Dec"};
```

모든 필드는 private 으로 설정,
메소드는 public으로 설정

```
class MyDate {  
    // 필드 선언  
  
    // 생성자 정의  
  
    // printDate1() 정의  
  
    // printDate2() 정의  
}
```

<terminated> DateTest [Ja

연도: 2022

월: 9

일: 22

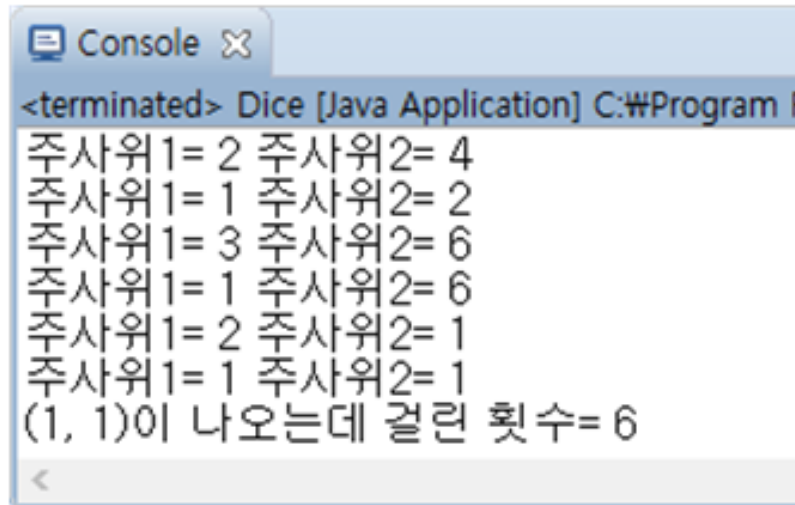
2022.9.22

Sep 22, 2022

```
public class DateTest {  
    public static void main(String[] args) {  
        Scanner sc = new Scanner(System.in);  
        System.out.print("연도: ");  
        int y = sc.nextInt();  
        System.out.print("월: ");  
        int m = sc.nextInt();  
        System.out.print("일: ");  
        int d = sc.nextInt();  
        MyDate date = new MyDate(y, m, d);  
        date.printDate1();  
        date.printDate2();  
    }  
}
```

실습 6. 실습문제-DiceTest.java

- 주사위를 Dice 클래스로 모델링한다. Dice 클래스는 주사위면을 필드로 가지고 있고 roll(), getValue() 메소드를 가지고 있다. 2개의 주사위 객체를 생성하여서 주사위를 반복하여 던진다. 두 주사위 값의 합이 2가 되면 반복을 종료하고 2가 나오는데 걸린 횟수를 화면에 출력한다.



```
Console X
<terminated> Dice [Java Application] C:\Program F
주사위1= 2 주사위2= 4
주사위1= 1 주사위2= 2
주사위1= 3 주사위2= 6
주사위1= 1 주사위2= 6
주사위1= 2 주사위2= 1
주사위1= 1 주사위2= 1
(1, 1)이 나오는데 걸린 횟수= 6
<
```

모든 필드는 private 으로 설정,
생성자, 메소드는 public 또는
default로 설정

```

3 class Dice{
4     private int value;
5     public Dice() {
6         value = 0;
7     }
8     void roll() {
9         // 랜덤함수를 사용하여 1~6사이의 값을 저장
10    }
11    int getValue() {
12        // 주사위 변의 값을 반환
13    }
14 }
15 public class DiceTest {
16     public static void main(String[] args) {
17         // 주사위 2개 객체 생성
18         // 주사위 굴린 횟수를 세는 변수 선언 및 초기화
19         do {
20             // 주사위 2개 각각 굴리는 메소드 호출
21             // 실행 결과와 같이 주사위의 각 면의 값을 출력
22             // 실행 횟수 증가
23         }while( ); // 반복 조건(종료 조건) 작성
24         System.out.println("(1,1)이 나오는데 걸린 횟수 = "+count);
25     }
26 }

```