



## Lab 4. Chapter6

## 실습1> Pizza 클래스 만들기(Pizza.java, Circle.java, PizzaTest.java)

- 원을 나타내는 Circle 클래스를 상속받아서 피자를 나타내는 Pizza 클래스를 작성해 보자.

```
class Circle {  
    protected int radius;  
    public Circle(int r) { radius = r; }  
}
```

- 테스트 프로그램을 실행했을 때 아래와 같은 결과가 출력되도록 하라.

```
public class PizzaTest{  
    public static void main(String args[]) {  
        Pizza obj = new Pizza("Pepperoni", 20);  
        System.out.println(obj);  
    }  
}
```

피자의 종류: Pepperoni, 피자의 크기: 20

# 실습2> Magazine 만들기(MyBook.java, Magazine.java)

---

## ■ MyBook 클래스

- 속성 정보: 제목, 페이지수, 저자,  
책의 개수(count-클래스 변수(static-정적 변수)로 정의)
- 하나 이상의 생성자를 정의
  - 책이 생성될 때마다 책의 수(count)를 증가
- 각 필드에 대하여 접근자와 설정자 메소드를 작성
  - 책의 개수에 대해서는 접근자 메소드만 클래스 메소드로 작성

## ■ Magazine 클래스

- MyBook 클래스의 하위 클래스로 작성
  - 추가 속성 정보: 발매일 정보
  - 한 개 이상의 생성자를 작성
    - 생성자 내에서 부모 생성자를 호출
  - 각 필드에 대하여 접근자 메소드와 설정자 메소드를 작성
- 실행을 위한 main 메소드를 포함하는 추가 클래스는 필요 없이 위에서 열거한 클래스만 작성하면 됨. (코드만 캡처해서 제출하면 됩니다.)

## 실습3> ColorPoint 클래스 만들기(Point.java, ColorPoint.java, ColorPointTest.java)

- Point 클래스를 상속받아 색을 가진 점을 나타내는 ColorPoint 클래스를 작성하시오. 다음 main 메소드를 포함하고 실행 결과와 같이 출력되게 하라.

```
class Point{
    private int x, y;
    public Point(int x, int y) {
        this.x = x;
        this.y = y;
    }
    public int getX() { return x;}
    public int getY() {return y;}
    public void setXY(int x, int y) {
        this.x = x;
        this.y = y;
    }
}
```

```
public class ColorPointTest {
    public static void main(String[] args) {
        ColorPoint cp = new ColorPoint(5, 5, "YELLOW");
        cp.setXY(10, 20);
        cp.setColor("RED");
        System.out.println(cp.toString()+ "입니다.");
    }
}
```

```
<terminated> ColorPointTest [Java Applet]
RED색의 (10, 20)의 점입니다.
```

# 실습4> 학생 만들기(Human.java, Student.java, StudentTest.java)

---

## ■ Human 클래스

- 속성
  - 이름(name), 나이(age)
- 생성자
  - 매개변수로 이름과 나이를 받아와서 멤버 변수 값 초기화

```
public Human(String name, int age){  
    // 매개 변수의 값으로 필드를 초기화  
}
```

- 각 필드(이름, 나이)에 대하여 접근자와 변경자 메소드를 작성
- toString() 작성 (Object 클래스의 toString() 메소드 재정의)
  - 객체의 현재 값들을 문자열로 반환

```
public String toString(){  
    // 객체의 현재 값들을 연결하여 문자열로 반환  
}
```

# 실습4> 학생 만들기(Human.java, Student.java, StudentTest.java)

---

## ■ Student 클래스

- Human 클래스를 상속
- 추가속성 정의
  - 전공(major), 학번(sID)
- 생성자
  - 매개변수로 이름, 나이, 전공, 학번을 받아와서 초기화
    - 부모 클래스의 생성자 호출 (이름, 나이)
    - 매개변수로 받은 값으로 멤버 변수의 전공과 학번을 초기화
- 필드(전공)에 대하여 접근자와 설정자 메소드를 작성
- toString() 작성 (Object 클래스의 toString() 메소드 재정의)
  - 부모의 toString()을 호출하여 받은 문자열 값에 추가로 전공 값 연결하여 문자열 반환

## 실습4> 학생 만들기(Human.java, Student.java, StudentTest.java)

### ■ StudentTest 클래스

- main 메소드가 있는 시작 클래스
  - 크기 3의 Human 객체와 Student 객체 배열 생성
  - 사용자로부터 이름과 나이를 입력 받아서 3명의 Human 객체를 생성
  - 사용자로부터 이름, 나이, 전공, 학번을 입력 받아서 3명의 Student 객체 생성
  - 각 객체참조변수 이름으로 출력문 작성

```
출력 결과 [1] Human 입력:춘향 18
           [2] Human 입력:몽룡 21
           [3] Human 입력:사또 50
           [1] Student 입력:미진 21 컴퓨터 112345
           [2] Student 입력:가현 22 경영 123456
           [3] Student 입력:용준 24 경제 143215
이름: 춘향, 나이: 18
이름: 몽룡, 나이: 21
이름: 사또, 나이: 50
[학생 정보]이름: 미진, 나이: 21, 전공:컴퓨터, 학번:112345]
[학생 정보]이름: 가현, 나이: 22, 전공:경영, 학번:123456]
[학생 정보]이름: 용준, 나이: 24, 전공:경제, 학번:143215]
```

# 실습5>

---

- Item 클래스 (Item.java)
  - 속성: 가격, 이름
- Food 클래스: Item 클래스의 하위 클래스로 작성 (Food.java)
  - 생성자 작성
    - 이름과 가격을 매개변수로 받아서 설정
  - toString 메소드 재정의 (출력 결과 참고)
- Book 클래스: Item 클래스의 하위 클래스로 작성 (Book.java)
  - 속성: 저자
  - 생성자 작성
    - 이름, 가격, 저자를 매개변수로 받아서 설정
  - toString 메소드 재정의 (출력 결과 참고)
- Movie 클래스: Item 클래스의 하위 클래스로 작성 (Movie.java)
  - 속성: 감독
  - 생성자 작성
    - 이름, 가격, 감독을 매개변수로 받아서 설정
  - toString 메소드 재정의 (출력 결과 참고)



## 실습 5>

---

### ■ Buyer 클래스(Buyer.java)

- 속성: money
- 생성자 작성
  - money를 매개변수로 받아서 멤버 변수 설정
- buy 메소드 : buy(Item t, int n)
  - Item과 개수를 매개변수로 받아서 출력 결과와 같은 형태로 내용 출력
    - t.toString() + “=>” +n+”개 구매 ”
  - 구매 가능 여부 체크하는 코드 작성 후 money에서 차감
    - 출력 결과 참고해서 적절한 출력 메시지 작성

## 실습 5> BuyerTest 클래스(BuyerTest.java)

### ■ BuyerTest 클래스

- 초기 자금 50000원을 가진 Buyer 생성
- 총 5번의 구매 진행

```
public class BuyerTest {  
    public static void main(String[] args) {  
        Buyer m = new Buyer(50000);  
        m.buy((new Food("비빔밥", 7000)), 3);  
        m.buy((new Food("라면", 5000)), 2);  
        m.buy((new Book("Java Programming", 12000, "이설명")), 1);  
        m.buy((new Movie("부산행", 9000, "연상호")), 1);  
        m.buy((new Food("김밥", 2000)), 3);  
    }  
}
```

## 실습 5> 실행 결과

[Food]비빔밥=>3개 구매

구매 완료 후 잔액: 29000

[Food]라면=>2개 구매

구매 완료 후 잔액: 19000

[Book]Java Programming, 저자:이설희=>1개 구매

구매 완료 후 잔액: 7000

[Movie]부산행, 감독: 연상호=>1개 구매

\*\* 잔액 부족으로 구매 불가능 \*\*

잔액:7000

[Food]김밥=>3개 구매

구매 완료 후 잔액: 1000

각 클래스에서 toString 메소드에서  
출력해야 할 형식은 다음 참고  
(red box 참고)