

# Object Modeling

경북대학교  
소프트웨어융합과  
배희호 교수

# Object Modeling

## ■ Object Modeling의 기본 개념

### ■ Object

- 현실 세계에서 독립적으로 존재하는 사물
- Software에서 Class(클래스)로 표현되며, Attribute와 Behavior(동작)를 가짐
- 예) 자동차(Car), 고객(Customer), 상품(Product)

### ■ Class

- Object를 만들기 위한 설계도
- 하나의 Class로 여러 Object 생성 가능

### ■ Attribute(속성)

- Object가 가지는 Data(변수)
- 예) 자동차 → 색상, 브랜드, 연식

### ■ Method(메소드)

- Object가 수행할 수 있는 행동(기능)
- 예) 자동차 → 주행(), 정지(), 속도 증가()



Futuristic Innovator

京福大學校  
KYUNGBOK UNIVERSITY

# Object Modeling 예제

- Object Oriented Programming
- 1단계 : 현실 세계의 개념을 분석 (과일 장수와 고객)



- 먼저 System에서 필요한 개념을 파악
  - 필요한 개념
    - 과일 장수(FruitSeller)
    - 고객(Customer)
    - 상품(Product) : 사과, 오렌지

# Object Modeling 예제

- 2 단계 : Object(Class) 도출
  - 각 개념을 Object(Class)로 변환

현실 세계 개념	Object (Class)
과일 장수	FruitSeller
고객	Customer
상품	Product

# Object Modeling 예제

- 3단계 : Attribute와 Method 정의
  - Object를 이루는 것은 Attribute(상태)와 Behavior(행위)
    - 예) 상품 Object의 표현

과일 장수는 과일을 팝니다. **행위**  
과일 장수는 사과 20개, 오렌지 10개를 보유하고 있습니다. **상태**  
사과는 개당 1000원이고, 오렌지는 개당 500원이다.  
과일 장수의 과일 현재 금액은 50,000원입니다. **상태**

## ■ 상품의 Attribute 표현

- ✓ 상품명 : String name;
- ✓ 가격 : int price;
- ✓ 개수 : int count;

# Object Modeling 예제

- 3단계 : Attribute와 Method 정의
  - Object를 이루는 것은 Attribute(상태)와 Behavior(행위)
    - 예) 과일 장수 Object의 표현

과일 장수는 과일을 팝니다. **행위**  
과일 장수는 사과 20개, 오렌지 10개를 보유하고 있습니다. **상태**  
사과는 개당 1000원이고, 오렌지는 개당 500원이다.  
과일 장수의 과일 현재 금액은 50,000원입니다. **상태**

## ■ 과일 장수의 Attribute 표현

- ✓ 보유하고 상품 : Product[] products;
- ✓ 현재 금액 : int myMoney;

# Object Modeling 예제

- Object를 이루는 것은 Attribute(상태)와 Behavior(행위)
  - 과일 장수의 판매 행위 표현 (Method)

```
public void sale(Product product, int count) {  
    boolean flag = false;  
    int i = 0;  
    for (; i < products.length; i++) {  
        if (product.name.equals(products[i].name)) {  
            flag = true;  
            break;  
        }  
    }  
    if (flag) {  
        products[i].count -= count;  
        int num = count * product.price;  
        myMoney += num;  
    }  
}
```

# Object Modeling 예제

- 3단계 : Attribute와 Method 정의
  - Object를 이루는 것은 Attribute(상태)와 Behavior(행위)
    - 예) 고객의 Object의 표현

나는 현재 5000원을 가지고 있으며, **상태**  
과일 장수에게 사과 2개를 샀다. **행위**

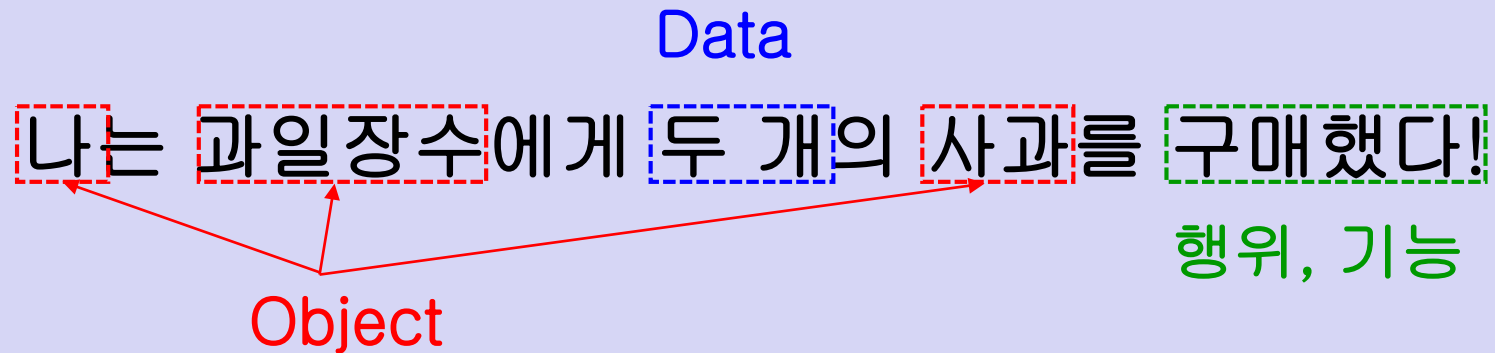
## ■ 고객의 Attribute 표현

- ✓ 현재 금액 : `int myMoney;`
- ✓ 구입한 과일 : `Product product;`



# Object Modeling 예제

- Object
  - 사전적 의미 : 물건 또는 대상



객체지향 프로그래밍에서는 나, 과일장수, 사과라는 Object를 등장시켜서 두 개의 사과 구매라는 행위를 실체화 함

# Object Modeling 예제

- Object를 이루는 것은 Attribute(상태)와 Behavior(행위)
  - 고객의 구입 행위 표현 (Method)

```
public void buy(FruitSeller seller, Product product, int count) {  
    this.product.name = product.name;  
    this.product.count = count;  
    seller.sale(product, count);  
    myMoney -= (count * product.price);  
}
```

# Object Modeling 예제

## ■ '상품' Class 정의

```
public class Product {  
    public String name;  
    public int price;  
    public int count;  
  
    public Product() {  
    }  
  
    public Product(String name, int price, int count) {  
        this.name = name;  
        this.price = price;  
        this.count = count;  
    }  
    @Override  
    public String toString() {  
        return "Product{" +  
            "name='" + name + 'W' + ", price=" + price + ", count=" +  
count  
            + '}';  
    }  
}
```

# Object Modeling 예제

## ■ ‘과일 장수’ Class 정의

```
public class FruitSeller {  
    private Product[] products;  
    private int myMoney;  
  
    public FruitSeller(Product[] products, int myMoney) {  
        this.products = products;  
        this.myMoney = myMoney;  
    }  
  
    public void sale(Product product, int count) {  
        boolean flag = false;  
        int i = 0;  
        for (; i < products.length; i++) {  
            if (product.name.equals(products[i].name)) {  
                flag = true;  
                break;  
            }  
        }  
    }  
}
```

# Object Modeling 예제

## ■ ‘과일 장수’ Class 정의

```
if (flag) {  
    products[i].count -= count;  
    int num = count * product.price;  
    myMoney += num;  
}  
}  
  
public void showSaleResult() {  
    for (int i = 0; i < products.length; i++)  
        System.out.println(products[i]);  
    System.out.println("과일 장수 금액 : " + myMoney);  
}  
}
```

# Object Modeling 예제

## ■ 과일 구매자 관점에서의 나(me) Class 표현!

```
public class Customer {  
    private int myMoney;  
    private Product product;  
  
    public Customer(int myMoney) {  
        this.myMoney = myMoney;  
        product = new Product();  
    }  
  
    public void buy(FruitSeller seller, Product product, int count) {  
        this.product.name = product.name;  
        this.product.count = count;  
        seller.sale(product, count);  
        myMoney -= (count * product.price);  
    }  
}
```

# Object Modeling 예제

## ■ 과일 구매자 관점에서의 나(me) Class 표현!

```
public void showBuyResult() {  
    System.out.println("고객 현재 잔액 : " + myMoney);  
    System.out.println("고객이 구입한 과일\n" +  
        String.format("%s 개수 : %d\n", product.name, product.count));  
}  
}
```

# Object Modeling 예제

## ■ 과일 장수 Simulation

```
public static void main(String[] args) {  
    Product[] products = new Product[]{  
        new Product("사과", 1000, 20),  
        new Product("배", 500, 10)};  
    FruitSeller seller = new FruitSeller(products, 50000);  
    seller.showSaleResult();  
    Customer buyer = new Customer(5000);  
    buyer.buy(seller, products[0], 2);  
    buyer.showBuyResult();  
    seller.showSaleResult();  
}
```

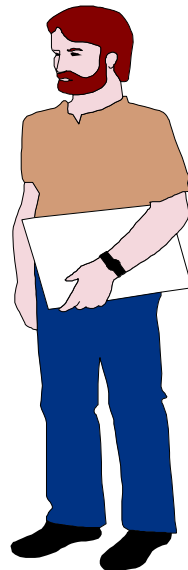
Message 전달은 두 Object 간의 대화 방법  
위 예제에서의 buy() 메소드가 의미하는 바는  
“아저씨 사과 2개 주세요!”



# Class 문제 1

- 다음 Object에 대한 설명에 맞는 Class를 만들고 값을 출력해 보세요.

*나이는 30살, 이름은 홍길동이라는 남자가 있습니다.  
이 남자는 결혼을 했고 자식이 셋 있습니다*



홍길동

# Class 문제 1

- 관심사항을 파악하자 (Data)
  - 명사 (속성)
- Class를 만들어보자
  - Member 변수
- People 클래스를 정의

People
-name : String -age : int -gender : char -isMarried : boolean -children : int
+toString() : String

# Class 문제 1

## ■ People.JAVA

```
public class People {  
    private String name;  
    private int age;  
    private char gender;  
    private boolean isMarried;  
    private int children;  
  
    public People(String name, int age, char gender, boolean isMarried,  
                                                           int children) {  
  
        this.name = name;  
        this.age = age;  
        this.gender = gender;  
        this.isMarried = isMarried;  
        this.children = children;  
    }  
}
```

# Class 문제 1

## ■ People.JAVA

@Override

```
public String toString() {  
    String result = "";  
    result = "이름 : " + name;  
    result += "\n나이 : " + age;  
    result += "\n성별 : " + gender + "자";  
    result += "\n결혼 여부 : " + (isMarried ? "기혼" : "미혼");  
    result += "\n자녀 수 : " + children + '명';  
    return result;  
}
```

# Class 문제 1

## ■ Main.JAVA

```
public static void main(String[] args) {  
    People people = new People("홍길동", 40, '남', true, 3);  
  
    System.out.println(people);  
}
```

# Class 문제 2

- 다음 Object에 대한 설명에 맞는 Class를 만들고 값을 출력해 보세요.

김철수는 키가 183Cm이고, 몸무게는 64.3Kg이고,  
나이는 21살 입니다

# Class 문제 2

- 관심사항을 파악하자 (Data)
  - 명사 (속성)
- Class를 만들어보자
  - Member 변수
- Body 클래스를 정의

Body
-name : String -height : int -weight : float -age : int
+toString() : String

# Class 문제 2

## ■ Body.JAVA

```
public class Body {  
    private String name;  
    int age;  
    protected int height;  
    private float weight;  
  
    public Body(String name, int age, int height, float weight) {  
        this.name = name;  
        this.age = age;  
        this.height = height;  
        this.weight = weight;  
    }  
  
    public void setWeight(float weight){  
        this.weight = weight;  
    }  
}
```



Futuristic Innovator

京福大學校  
KYUNGBOK UNIVERSITY



# Class 문제 2

## ■ Body.JAVA

```
public float getWeight(){  
    return weight;  
}
```

@Override

```
public String toString() {  
    String result = "";  
    result = "이름 : " + name;  
    result += "\n나이 : " + age;  
    result += "\n키 : " + height + "Cm";  
    result += "\n몸무게 : " + weight + "Kg";  
  
    return result;  
}
```

# Class 문제 2

## ■ Main.JAVA

```
public static void main(String[] args) {  
    Body kim = new Body("김철수", 21, 183, 64.3f);  
  
    kim.setWeight(64.7f);  
    kim.age++;  
  
    System.out.print(kim);  
}
```



Futuristic Innovator

京福大學校  
KYUNGBOK UNIVERSITY

# Class 문제 3

- 해당 내용을 Class와 Object를 활용하여 생성하여 보자

나이(age)가 27살, 이름(name)이 '이순신'이라는 남자가 있다. 이 남자는 클럽(club)에 참여 하였고, 클럽 인원(clubMember)은 5명이다.

People
-name : String -age : int -gender : char -club : boolean -clubMember : int
+toString() : String

# Class 문제 3

```
public class People {  
    public int age;           // 나이  
    public String name;      // 이름  
    public boolean club;     // 클럽 참여 여부  
    public int clubMember;   // 클럽 인원  
  
    public void peoplePrint() { // 객체 정보 출력  
        System.out.println("나이: " + age);  
        System.out.println("이름: " + name);  
        System.out.println("클럽 참여 여부: " + club);  
        System.out.println("클럽인원:" + clubMember);  
    }  
}
```

# Class 문제 3

```
public static void main(String[] args) {  
    People people = new People();    //객체 생성  
    people.age = 27;  
    people.name = "peemang";  
    people.club = true;  
    people.clubMember = 5;  
  
    people.peoplePrint();            //객체 정보 출력  
}
```

# Class 문제 4

- 다음과 같이 출력하는 Program을 만들어보자

*나의(홍길동) 자동차는 2001년식 현대자동차에서  
생산한 Grandure로 현재 23,000Km를 운행했습니다.*

- Car 클래스를 작성하여라.
- CarModel 클래스를 작성하여라.

Car
-owner : String -CarModel model : CarModel -mileage : int
+toString() : String

CarModel
-modelName : String -company : String -year : int
+toString() : String

# Class 문제 4

## ■ CarModel.JAVA

```
public class CarModel {  
    private String modelName;  
    private String company;  
    private int year;  
  
    public CarModel(String modelName, String company, int year) {  
        this.modelName = modelName;  
        this.company = company;  
        this.year = year;  
    }  
  
    @Override  
    public String toString() {  
        return "모델명 : " + modelName + "Wn" +  
            "제조사 : " + company + "Wn" +  
            "년식 : " + year;  
    }  
}
```

# Class 문제 4

## ■ Car.JAVA

```
public class Car {  
    private String owner;  
    CarModel model;  
    private int mileage;  
  
    public Car(String owner, CarModel model, int mileage) {  
        this.owner = owner;  
        this.model = model;  
        this.mileage = mileage;  
    }  
  
    @Override  
    public String toString() {  
        return "소유자 : " + owner + "\n" +  
            model.toString() + "\n" +  
            "운행거리 : " + String.format("%,d Km", mileage);  
    }  
}
```



# Class 문제 4

## ■ Main.JAVA

```
public static void main(String[] args) {  
    Car myCar = new Car("홍길동",  
                        new CarModel("Grandure", "현대자동차", 2001), 23000);  
  
    System.out.println(myCar);  
}
```

# Class 문제 5

- 다음의 Box Object를 정의할 Box 클래스를 작성해보자.



- Attribute(속성)와 Behavior(행동)을 구분해보자.
  - Attribute
    - Box의 크기 (가로, 세로, 높이)
    - Box의 색상
    - Box의 문구
  - Behavior
    - 부피를 구하다

# Class 문제 5

- Class 생성
  - Class 이름은 File 이름과 같음
  - Class 이름은 대문자로 시작함
  - Attribute 정의
    - Attribute 이름은 소문자로 시작함
- Data의 속성만 선언된 Class 예

```
class Box {  
    private int width;    // attribute(속성)을 Field로 구현  
    private int height;  
    private int depth;  
}
```

# Class 문제 5

## ■ 생성자

- Object를 생성하는 용도로 사용함
- 생성자의 이름은 Class 이름과 같음
- Class를 만들면 자동으로 Default 생성자가 만들어짐
- 생성자를 정의하면 Default 생성자는 없어짐

```
class Box {  
    private int width;  
    private int height;  
    private int depth;  
  
    public Box() {    // default 생성자  
    }  
    public Box(int width, int height, int depth) {    // 생성자  
        this.width = width;  
        this.height = height;  
        this.depth = depth;  
    }  
}
```

# Class 문제 5

- Setter 정의(변경 Method)
  - Attribute 값 설정
  - 일반적으로 Method 이름이 'set~'로 시작함

```
class Box {  
    private int width;  
    private int height;  
    private int depth;  
  
    public void setWidth(int width) {  
        this.width = width;  
    }  
  
    public void setHeight(int height) {  
        this.height = height;  
    }  
}
```

# Class 문제 5

- Getter 정의 (접근 Method)
  - Attribute 값 얻어 오기
  - 일반적으로 메소드 이름이 'get~'로 시작함

```
class Box {  
    private int width;  
    private int height;  
    private int depth;  
  
    public int getWidth(){  
        return this.width;  
    }  
  
    public int getHeight(){  
        return this.height;  
    }  
}
```

# Class 문제 5

- Custom Method 정의
  - Method 이름은 소문자로 시작
  - Method의 반환 Data 형을 고려할 것
  - Data의 Attribute와 Method를 가진 Class 선언 예

```
class Box {  
    private int width;  
    private int height;  
    private int depth;  
  
    public int volume() {  
        int volume;  
        volume = width * height * depth;  
  
        return volume;  
    }  
}
```

# Class 문제 5

- toString( ) Method
  - Object를 String Type으로 변환시켜주는 Method
  - 상속 시 상위 Class의 Method 교체

```
@Override
public String toString() {
    return "Box{" +
        "width=" + width +
        ", height=" + height +
        ", depth=" + depth +
        '}';
}
```



# Class 문제 5

- Object는 **new** Keyword를 이용하여 생성
  - new는 Object 생성자 호출

- Object 생성 과정

1. Object에 대한 Reference 변수 선언
2. Object 생성

```
public static void main (String[] args) {  
    Box brownBox;           // Reference 변수 brownBox 선언  
    brownBox = new Box();   // Box Object 생성  
}
```

```
public static void main (String[] args) {  
    Box brownBox = new Box( );  
}
```

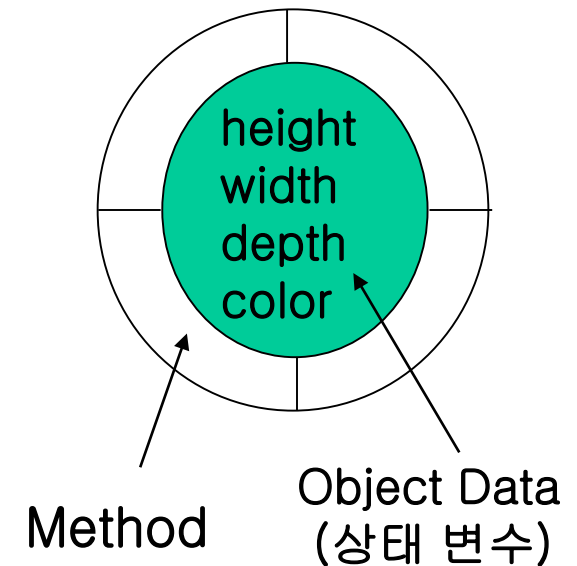
# Class 문제 6

■ 다음의 Box 클래스를 정의해보자



- ✓ Attribute
  - ✓ 높이, 깊이, 넓이, 색상
- ✓ Method
  - ✓ 상자의 크기를 설정하다
  - ✓ 상자의 색상을 설정하다
  - ✓ 상자의 각 속성을 반환하다
  - ✓ 상자의 정보를 반환하다

Box Instance



# Class 문제 6

- Attribute 정의
  - Class가 가지는 Data(Field)
  - Member Field의 접근 제한자는 private로

```
class Box {  
    private int width;  
    private int height;  
    private int depth;  
    private String color;  
}
```

Class 이름은  
대문자로 작성

# Class 문제 6

## ■ 생성자는 Class 이름과 같음

```
class Box {  
    .....  
    public Box() {  
    }  
    public Box(int width, int height, int depth) {  
        this.width = width;  
        this.height = height;  
        this.depth = depth;  
    }  
    public Box(int width, int height, int depth, String color) {  
        this.width = width;  
        this.height = height;  
        this.depth = depth;  
        this.color = color;  
    }  
}
```

# Class 문제 6

## ■ Setter 정의

```
class Box {  
    .....  
  
    public void setWidth(int width) {  
        this.width = width;  
    }  
  
    public void setDepth(int depth) {  
        this.depth = depth;  
    }  
  
    public void setColor(String color) {  
        this.color = color;  
    }  
}
```

# Class 문제 6

## ■ Getter 정의

```
class Box {  
    .....  
    public int getWidth(){  
        return this.width;  
    }  
  
    public int getHeight(){  
        return this.height;  
    }  
    public int getDepth(){  
        return this.depth;  
    }  
    public String getColor(){  
        return this.color;  
    }  
}
```

# Class 문제 6

## ■ Custom Method 정의

```
public void setSize(int width, int height, int depth) {  
    this.width = width;  
    this.height = height;  
    this.depth = depth;  
}
```

```
public int volume() {  
    int test = width * height * depth;  
    return test;  
}
```

# Class 문제 6

## ■ toString() Method

@Override

```
public String toString() {  
    return "Box{" +  
        "width=" + width +  
        ", height=" + height +  
        ", depth=" + depth +  
        ", color='" + color + 'W' +  
        "'";  
}
```



# Class 문제 6

## ■ Object 생성 및 사용

```
public static void main(String[] args) {  
    Box brown = new Box();  
    Box white;  
  
    brown.setWidth(20);  
    brown.setHeight(40);  
    brown.setDepth(15);  
    brown.setColor("Brown");  
  
    white = new Box(10, 20, 30, "White");  
  
    System.out.printf(" 첫번째 박스 " + brown);  
    System.out.printf(" 첫번째 박스의 부피는 %,d 입니다.\n", brown.volume());  
    System.out.printf(" 첫번째 박스 " + white);  
    System.out.printf(" 두번째 박스의 부피는 %,d 입니다.\n", white.volume());  
}
```

# Class 문제 7

- 반지름(radius)과 이름(name)을 가지고, 면적을 구하는 Circle 클래스를 작성하고, Circle 클래스의 Object를 생성하여 보자.



# Class 문제 7

- Data는 Field로 Information은 Method로 구현

```
public class Circle {  
    int radius;    // 원의 반지름을 저장하는 Member 변수  
    String name;   // 원의 이름을 저장하는 Member 변수  
  
    public double area() {    // Member Method  
        return 3.14 * radius * radius;  
    }  
}
```

# Class 문제 7

## ■ Default Constructor

- 매개 변수 없고, 아무 작업 없이 단순 생성자
- 기본 생성자라고도 불림

```
public class Circle {  
    int radius;    // 원의 반지름을 저장하는 Member 변수  
    String name;   // 원의 이름을 저장하는 Member 변수  
  
    public Circle() { }    // 기본 생성자  
  
    public double area() {    // Member Method  
        return 3.14 * radius * radius;  
    }  
}
```

# Class 문제 7

- Circle 클래스를 이용하여 Object 생성
  - Reference 변수 선언

```
Circle pizza;
```

- Object 생성
  - **new** 연산자와 생성자 이용

```
pizza = new Circle();
```

- Object Member 접근
  - 점(.) 연산자 이용

```
pizza.radius = 10;  
area = pizza.area();
```

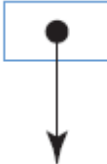
# Class 문제 7

```
public static void main(String[] args) {  
    Circle pizza;                // Circle 객체 선언  
    pizza = new Circle();        // Circle 객체 생성  
  
    pizza.radius = 10;           // 피자의 반지름을 10으로 설정  
    pizza.name = "자바피자";     // 피자의 이름 설정  
    double area = pizza.area();  // 피자의 면적 알아내기  
    System.out.println(pizza.name + "의 면적은 " + area);  
  
    Circle donut = new Circle(); // Circle 객체 생성  
    donut.radius = 2;            // 도넛의 반지름을 2로 설정  
    donut.name = "자바도넛";     // 도넛의 이름 설정  
    area = donut.area();         // 도넛의 면적 알아내기  
    System.out.println(donut.name + "의 면적은 " + area);  
}
```

자바피자의 면적은 314.0  
자바도넛의 면적은 12.56

# Class 문제 7

pizza

A small blue box containing a black dot, with an arrow pointing down to the class object below.

radius


10

name

"자바피자"

```
public double getArea() {  
    return 3.14*radius*radius;  
}
```

donut

A small blue box containing a black dot, with an arrow pointing down to the class object below.

radius

2


name

"자바도넛"


```
public double getArea() {  
    return 3.14*radius*radius;  
}
```

# Class 문제 7

(1) Circle pizza;

pizza 

(2) pizza = new Circle();


pizza 

Circle 타입의 객체

radius   
name   
getArea() { ... }

객체 메모리  
할당 및  
객체 생성

(3) pizza.radius = 10;

pizza 

radius   
name   
getArea() { ... }

radius 값 변경

(4) pizza.name = "자바피자";

pizza 

radius   
name   
getArea() { ... }

name 값 변경

(5) double area = pizza.getArea();

pizza 

radius   
name

area

getArea() {  
    return 3.14\*radius\*radius;  
}

getArea()  
메소드 실행



Futuristic Innovator

京福大學校  
KYUNGBOK UNIVERSITY



# Class 문제 8

- 쇼핑몰에 주문이 들어왔다. 주문 내용을 구현할 수 있는 Class를 만들고 Instance로 생성한 후 해당 내용을 입력 받아 아래와 같은 형식으로 출력하는 Program을 작성해보자

주문 번호 : 202201234

주문자 아이디 : bae1234

주문 날짜 : 2022-03-24

주문자 이름 : 홍길동

주문 상품 번호 : PD-34-ABC

배송 주소 : 경기도 남양주시 진접읍 경복대로

주문 번호 : 202201234

주문자 아이디 : bae1234

주문 날짜 : 2022-03-24

주문자 이름 : 홍길동

주문 상품 정보 : PD-34-ABC

배송 주소 : 경기도 남양주시 진접읍 경복대로

# Class 문제 8

## ■ Shopping 클래스

Shopping
<ul style="list-style-type: none"><li>-orderNumber : String</li><li>-id : String</li><li>-date : String</li><li>-name : String</li><li>-productNumbe : String</li><li>-address : String</li></ul>
<ul style="list-style-type: none"><li>+Setter/Getter</li><li>+toString() : String</li></ul>

# Class 문제 8

## ■ Shopping.JAVA

```
public class Shopping {  
    private String orderNumber;    //주문 번호  
    private String id;            //주문자 ID  
    private String date;          //주문 날짜  
    private String name;          //주문자 이름  
    private String productNumber; //주문 상품 번호  
    private String address;       //배송 주소  
  
    public void setOrderNumber(String orderNumber) {  
        this.orderNumber = orderNumber;  
    }  
    public void setAddress(String address) {  
        this.address = address;  
    }  
  
    public void setDate(String date) {  
        this.date = date;  
    }  
}
```

# Class 문제 8

## ■ Shopping.JAVA

```
public void setName(String name) {
    this.name = name;
}
public void setId(String id) {
    this.id = id;
}
public void setProductNumber(String productNumber) {
    this.productNumber = productNumber;
}
@Override
public String toString() {
    return "주문 번호 : " + orderNumber + "\n" +
        "주문자 아이디 : " + id + "\n" +
        "주문 날짜 : " + date + "\n" +
        "주문자 이름 : " + name + "\n" +
        "주문 상품 정보 : " + productNumber + "\n" +
        "배송 주소 : " + address;
}
}
```

# Class 문제 8

## ■ Main.JAVA

```
public static void main(String[] args) {  
    Scanner keyboard = new Scanner(System.in);  
    Shopping user = new Shopping(); //객체 생성  
    System.out.print("주문 번호 : ");  
    user.setOrderNumber(keyboard.nextLine());  
    System.out.print("주문자 아이디 : ");  
    user.setId(keyboard.nextLine());  
    System.out.print("주문 날짜 : ");  
    user.setDate(keyboard.nextLine());  
    System.out.print("주문자 이름 : ");  
    user.setName(keyboard.nextLine());  
    System.out.print("주문 상품 번호 : ");  
    user.setProductNumber(keyboard.nextLine());  
    System.out.print("배송 주소 : ");  
    user.setAddress(keyboard.nextLine());  
    System.out.println(user);  
}
```

# Class 문제 9

- 나의 Nikon 사진기 판매점을 Program해보자
  - 우리 shop에서 취급하는 Camera는 “Nikon” 제품으로 400,000원이고, 재고수(numberOfStock)는 30개, 팔린 개수(sold)를 50개이다.
  - 이것을 Modeling하여 JAVA의 Class로 표현해보자



# Class 문제 9

- Camera 하나를 표현하는 클래스 Camera를 작성
  - Camera 클래스는 4개의 Field를 가짐
    - String 타입의 name(상품명 이름)
    - int 타입의 price(가격)
    - int 타입의 numberOfStock(재고 수)
    - int 타입의 sold(팔린 개수)
- Shop 클래스
  - Camera 클래스로 만든 Object를 Member로 함
  - 수입금 (int income)
  - Camera를 판매한다

# Class 문제 9

## ■ Camera 클래스

```
public class Camera {  
    private String name;  
    private int price;  
    private int numberOfStock;  
    private int sold;  
  
    public Camera(String name, int price, int numberOfStock, int sold) {  
        this.name = name;  
        this.price = price;  
        this.numberOfStock = numberOfStock;  
        this.sold = sold;  
    }  
  
    public void setNumberOfStock(int numberOfStock) {  
        this.numberOfStock = numberOfStock;  
    }  
}
```



# Class 문제 9

## ■ Camera 클래스

```
public void setSold(int sold) {  
    this.sold = sold;  
}  
  
public int getPrice() {  
    return price;  
}  
  
public int getNumberOfStock() {  
    return numberOfStock;  
}  
  
public int getSold() {  
    return sold;  
}
```

# Class 문제 9

## ■ Camera 클래스

@Override

```
public String toString() {  
    return "이름 : " + name + '\n' +  
        String.format("가격 : %,d원\n", price) +  
        String.format("재고 수량 : %,d 개\n", numberOfStock) +  
        String.format("판매 수량 : %,d 개", sold);  
}
```

# Class 문제 9

## ■ Shop 클래스

```
public class Shop {  
    private Camera camera;  
    private int income;  
  
    public Shop(Camera camera) {  
        this.camera = camera;  
        income = 0;  
    }  
  
    public void sale() {  
        if (camera.getNumberOfStock() == 0)  
            System.err.println("재고가 없습니다");  
        else {  
            camera.setNumberOfStock(camera.getNumberOfStock() - 1);  
            camera.setSold(camera.getSold() + 1);  
            income += camera.getPrice();  
        }  
    }  
}
```

# Class 문제 9

## ■ Shop 클래스

@Override

```
public String toString() {  
    return camera.toString() + "\n" +  
        String.format("수입금 : %,d원\n", income);  
}  
}
```



Futuristic Innovator

京福大學校  
KYUNGBOK UNIVERSITY

# Class 문제 9

## ■ Main.JAVA

```
public static void main(String[] args) {  
    Camera nikon = new Camera("Nikon", 350000, 10, 0);  
    Shop shop = new Shop(nikon);  
  
    shop.sale();  
    shop.sale();  
  
    System.out.println(shop);  
}
```

# Class 문제 10

- 학생이 대중 교통(버스, 지하철)을 이용하여 학교를 오는 것을 Program 해보자
  - 단 버스 요금은 1,200원, 지하철은 1,500원
- ✓ 김철수는 용돈을 10,000원 받아서 100번 버스를 타고 등교하였습니다.
- ✓ 이영희는 용돈으로 12,000원을 받아서 4호선 지하철을 타고 와서, 100번 버스로 환승하여 등교하였습니다.

# Class 문제 10

## 필요한 클래스

Student
-name : String -money : int
+takeOnBus() : void +takeOnSubway() : void +takeOffBus() : void +takeOffSubway() : void +toString() : String

Bus
-busNo : String -passengerCount : int -money : int -passenger : ArrayList
+takeOn() : void +takeOff() : void +toString() : String

Subway
-lineNo : String -passengerCount : int -money : int -passenger : ArrayList
+takeOn() : void +takeOff() : void +toString() : String

# Class 문제 10

## ■ Bus 클래스

```
public class Bus {  
    String busNo;           // 버스번호  
    int passengerCount;     // 승객 수  
    int money;              // 돈  
    ArrayList<String> passenger;  
  
    public Bus(String busNo) {  
        passenger = new ArrayList<>();  
        this.busNo = busNo;  
        money = 0;  
        passengerCount = 0;  
    }  
  
    public void takeOn(String name) { // 승객을 태움  
        passenger.add(name);  
        ++passengerCount;  
        this.money += 1200;  
    }  
}
```



# Class 문제 10

## ■ Bus 클래스

```
public boolean takeOff(String name) {  
    if (passenger.contains(name))  
        return true;  
    else  
        return false;  
}
```

@Override

```
public String toString() {  
    return String.format("busNo : %s, 탑승객 수 : %d 명, 수입금 : %,d 원,  
                           승객명 = %s",  
                           busNo, passengerCount, money, passenger);  
}
```

# Class 문제 10

## ■ Subway 클래스

```
public class Subway {  
    private String lineNo;  
    private int passengerCount;  
    private int money; // 돈  
    private ArrayList<String> passenger;  
  
    public Subway(String lineNo) {  
        this.lineNo = lineNo;  
        passengerCount = 0;  
        money = 0;  
        passenger = new ArrayList<>();  
    }  
  
    public void takeOn(String name) { // 승객을 태움  
        passenger.add(name);  
        passengerCount++;  
        this.money += 1500;  
    }  
}
```

# Class 문제 10

## ■ Subway 클래스

```
public boolean takeOff(String name) {  
    if (passenger.contains(name))  
        return true;  
    else  
        return false;  
}
```

@Override

```
public String toString() {  
    return String.format("%s호선 지하철, 승객 수 : %d 명, 수입금 : %,d 원,  
                           승객명 : %s", lineNo, passengerCount, money, passenger);  
}
```

# Class 문제 10

## ■ Student 클래스

```
public class Student {  
    private String name;  
    private int money;  
  
    public Student(String name, int money) {  
        this.name = name;  
        this.money = money;  
    }  
  
    public void takeOnBus(Bus bus) {  
        bus.takeOn(name);  
        money -= 1200;  
    }  
    public void takeOffBus(Bus bus) {  
        bus.takeOff(name);  
    }  
}
```

# Class 문제 10

## ■ Student 클래스

```
public void takeOnSubway(Subway subway) {  
    subway.takeOn(name);  
    money -= 1500;  
}
```

```
public void takeOffSubway(Subway subway) {  
    subway.takeOff(name);  
}
```

@Override

```
public String toString() {  
    return String.format("%s님의 잔액은 %,d원 입니다.\n", name, money);  
}  
}
```

# Class 문제 10

## ■ Main.JAVA

```
public static void main(String[] args) {  
    Student kim = new Student("김철수", 10000);  
    Student lee = new Student("이영희", 12000);  
    Bus bus100 = new Bus("100");  
    Subway subwayGreen = new Subway("4"); // 2호선  
    kim.takeOnBus(bus100);  
    kim.takeOffBus(bus100);  
    lee.takeOnSubway(subwayGreen);  
    lee.takeOffSubway(subwayGreen);  
    lee.takeOnBus(bus100);  
    lee.takeOffBus(bus100);  
  
    System.out.print(kim);  
    System.out.print(lee);  
    System.out.println(bus100);  
    System.out.println(subwayGreen);  
}
```

# Class 문제 11

- 학생이 대중교통(버스, 지하철, Taxi)을 이용하여 학교를 오는 것을 Program 해보자
  - 단 버스 요금은 1,200원, 지하철은 1,500원
  - Taxi 요금은 meter기 요금

- ✓ 김철수는 100번 버스를 타고 등교하였습니다.
- ✓ 이영희는 4호선 지하철을 타고 와서, 100번 버스로 환승하여 등교하였습니다.
- ✓ 박정민은 늦잠을 자서 4호선 지하철을 타고 와서, 경북 Taxi를 타고 등교하였습니다.

# Class 문제 12

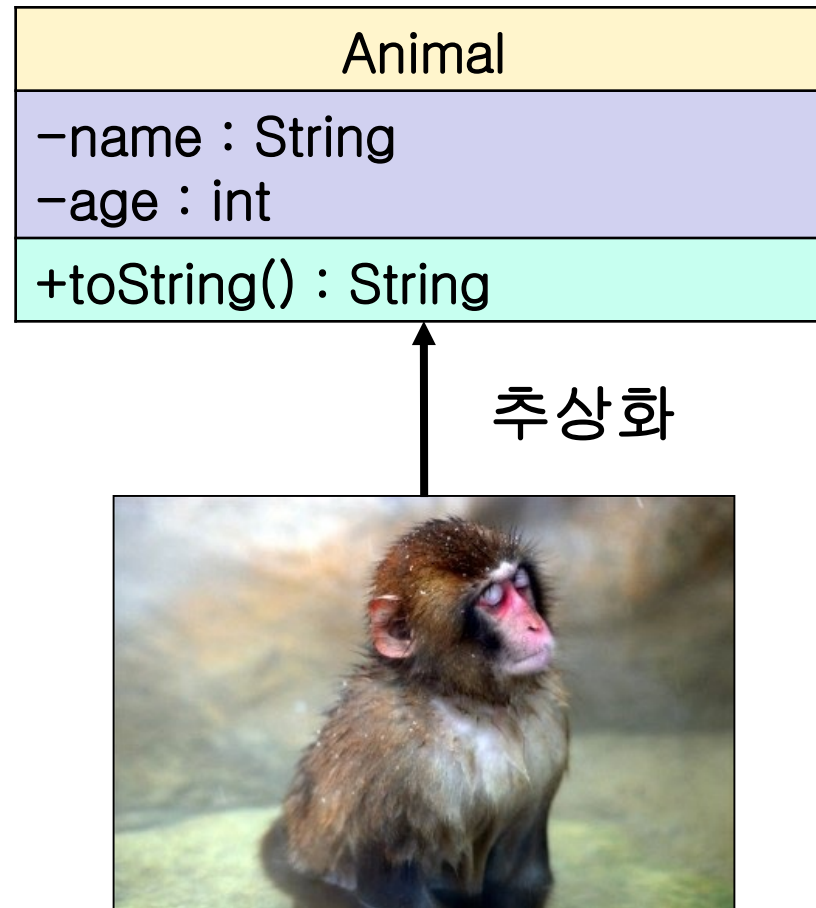
- 동물(Animal)의 이름(name)과 나이(age)의 속성을 갖는 동물 Class를 만들고, 원숭이(Monkey) Object를 만들어 출력하여 보아라





# Class 문제 12

## ■ Class 설계



# Class 문제 12

## ■ Class 작성

//이름과 나이를 속성으로 갖는 Animal 클래스에 설계

```
class Animal {  
    private String name;  
    public int age;  
  
    public Animal(String name, int age) { // 생성자  
        this.name = name;  
        this.age = age;  
    }  
  
    public String toString() { // 메소드  
        return String.format("동물 이름 : %s, 나이 = %d 살\n", name,  
                                age);  
    }  
}
```

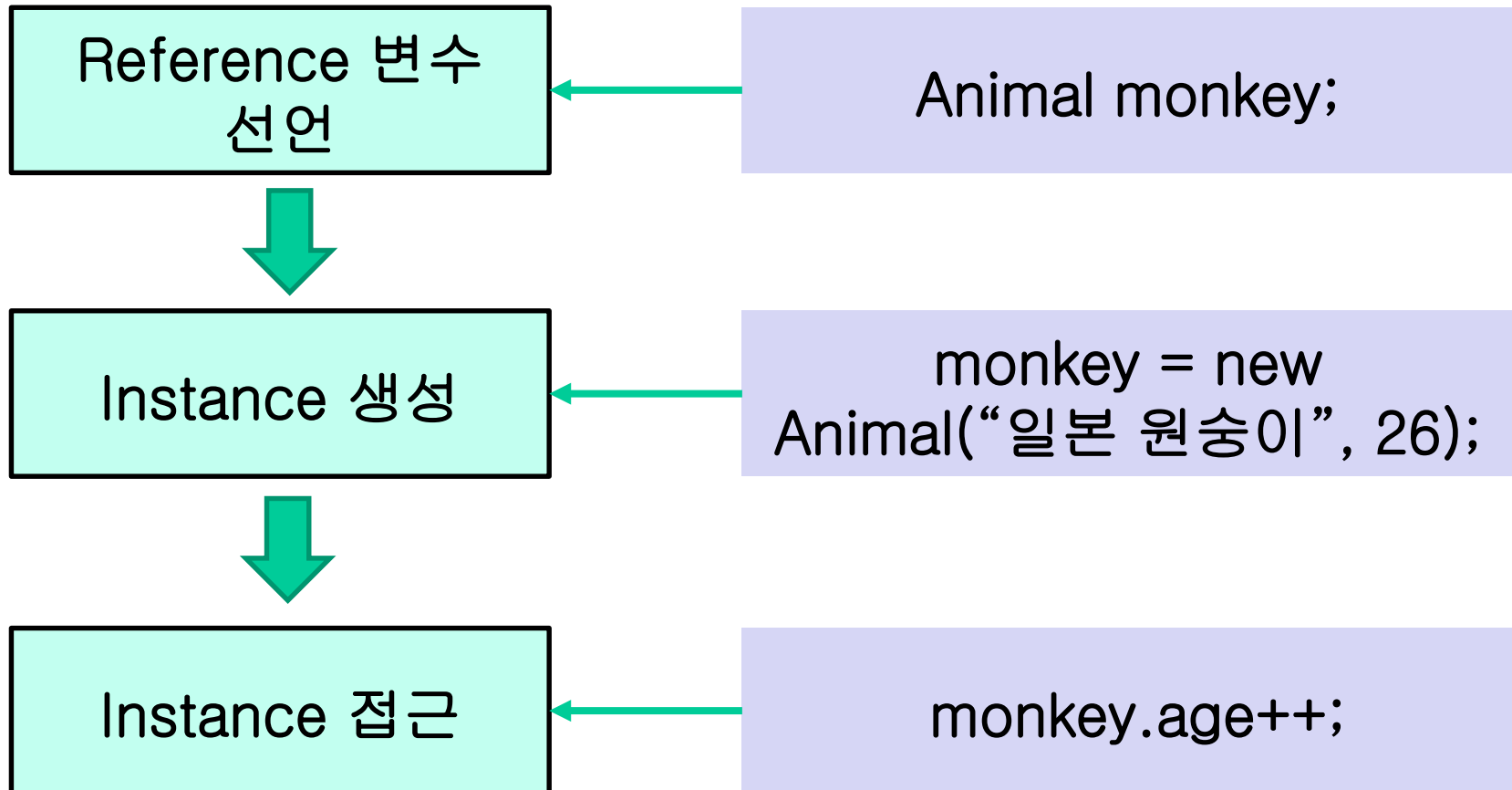
# Class 문제 12

## ■ 원숭이 Object 생성

```
public static void main(String[] args) {  
    Animal monkey; //Reference 변수 선언  
  
    monkey = new Animal("일본 원숭이", 26); //객체 생성  
    monkey.age++;  
    System.out.println(monkey);  
}
```

# Class 문제 12

## ■ Object 생성



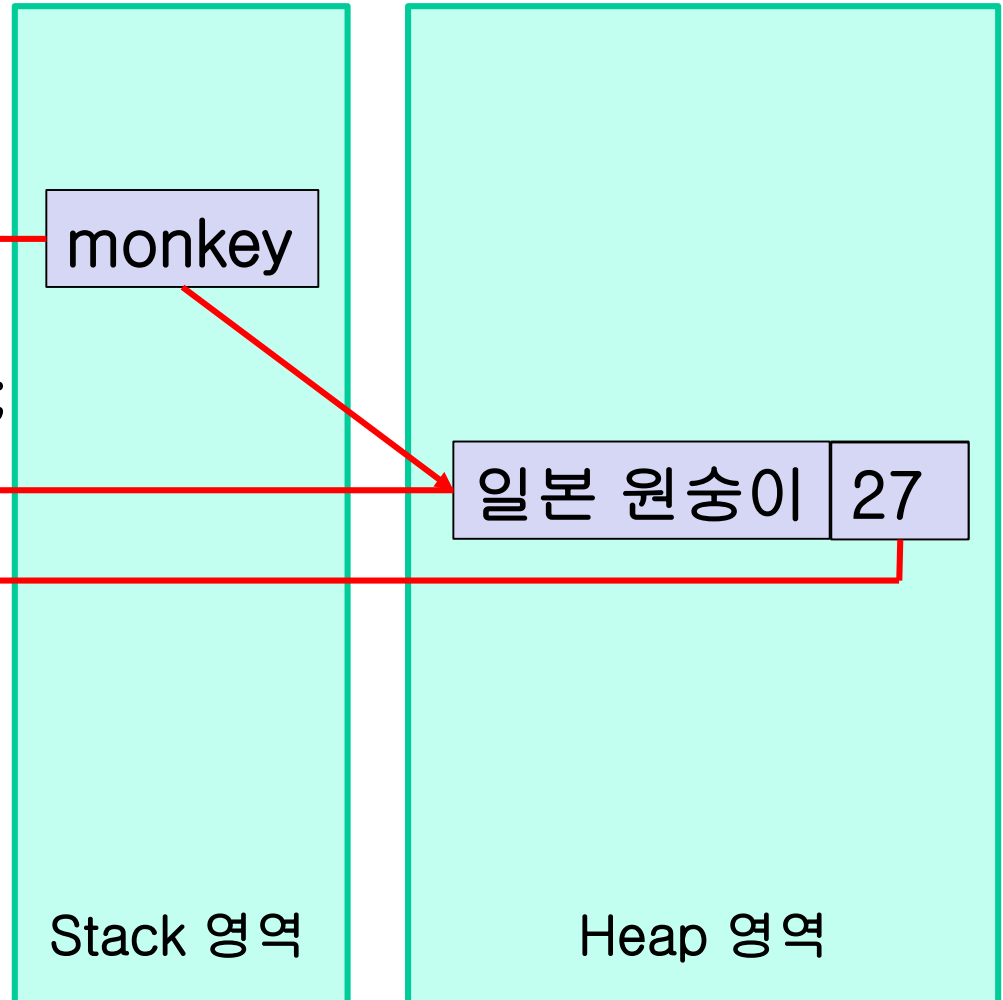
# Class 문제 12

## ■ 원숭이 Object 생성

Animal monkey;

monkey = new  
Animal("일본 원숭이", 26);

monkey.age++;



# Class 문제 13[심화]

- Animal 객체를 하나 더 생성하여 Reference 변수 penguin로 접근하도록 하고 이름은 “황제 펭귄” 나이는 2를 저장한 후 출력하시오.



# Class 문제 13[심화]

## ■ 펭귄 Object 생성

```
public static void main(String[] args) {  
    .....  
    Animal penguin = new Animal("황제 펭귄", 2); //객체 생성  
    System.out.println(penguin);                //객체의 멤버에 저장된 값 출력  
}
```

# Class 문제 13[심화]

```
class Animal {  
    private String name;  
    private int age;  
  
    public Animal() {  
    }  
  
    public Animal(String name, int age) {  
        this.name = name;  
        this.age = age;  
    }  
  
    public void setAge(int age) {           // setter()와 getter()  
        this.age = age;  
    }  
  
    public int getAge(){  
        return age;  
    }  
  
    public toString( ) {  
        return String.format("동물 이름 : %s, 나이 = %d 살\n", name, age);  
    }  
}
```



# Class 문제 13[심화]

## ■ 원숭이 Object 생성

```
public static void main(String[] args) {  
    Animal monkey;                                // Reference 변수 선언  
  
    monkey = new Animal("일본 원숭이", 26); //객체 생성  
    monkey.setAge(monkey.getAge() + 1);  
    System.out.println(monkey);  
}
```

# Class 문제 14

- 집에서 사용하는 Desk Lamp를 클래스로 작성하여 보자

DeskLamp
-isOn : bool
+turnOn() +turnOff()



# Class 문제 14

## ■ 클래스 작성

```
class DeskLamp {  
    private boolean isOn;  
  
    public void turnOn( ) {  
        isOn = true;  
    }  
  
    public void turnOff( ) {  
        isOn = false;  
    }  
  
    public String toString( ) {  
        return “현재 상태는 ” + (isOn ? “켜짐” : “꺼짐”);  
    }  
}
```

# Class 문제 14

## ■ 객체 생성

```
public static void main(String[] args) {  
    DeskLamp myLamp = new DeskLamp( );  
  
    myLamp.turnOn();  
    System.out.println(myLamp);  
    myLamp.turnOff();  
    System.out.println(myLamp);  
}
```

# Class 문제 15

## ■ 날짜 클래스를 작성해보자

Date
-year : int
-month : string
-day : int
+setDate() +printDate()



# Class 문제 15

## ■ Class 생성

```
class Date {  
    private int year;           // 년도  
    private String month;      // 월 이름  
    private int day;           // 일  
  
    public Date(int year, String month, int day) {    // 날짜 초기화  
        this.year = year;  
        this.month = month;  
        this.day = day;  
    }  
}
```

# Class 문제 15

## ■ Class 생성

```
public int getYear() {  
    return year;  
}
```

```
public void setYear(int year) {  
    this.year = year;  
}
```

```
public String getMonth() {  
    return month;  
}
```

```
public void setMonth(String month) {  
    this.month = month;  
}
```

# Class 문제 15

## ■ Class 생성

```
public int getDay() {  
    return day;  
}  
public void setDay(int day) {  
    this.day = day;  
}  
public void setDate(int year, String month, int day) { // 날짜 지정  
    this.year = year;  
    this.month = month;  
    this.day = day;  
}  
public void printDate() {  
    System.out.println(year + "년 " + month + " " + day + "일");  
}  
}
```



# Class 문제 15

## ■ 객체 생성 및 사용

```
public static void main(String[] args) {  
    Date date = new Date(2017, "5월", 15);  
  
    date.printDate( );  
    date.setDate(2022, "3월", 20);  
    date.printDate( );  
    date.setYear(2018);  
    date.printDate( );  
}
```

# Class 문제 16[심화]

## ■ 시간을 나타내는 Time 클래스

Time
<ul style="list-style-type: none"><li>- hour : int</li><li>- minute : int</li><li>- second : int</li></ul>
<ul style="list-style-type: none"><li>+Time()</li><li>+Time(hour, minute, second)</li><li>+setter()</li><li>+getter()</li><li>+toString()</li></ul>

# Class 문제 17

- 직원(Employee)을 나타내는 클래스에서 직원들의 수를 카운트하는 예를 살펴보자. 직원의 수를 정적 변수로 나타낸다

Employee
<ul style="list-style-type: none"><li>- name : String</li><li>- salary : int</li><li>- count : int (static)</li></ul>
<ul style="list-style-type: none"><li>+setter()</li><li>+getter()</li><li>+retire() : void</li></ul>

# Class 문제 17

## ■ Employee.JAVA

```
public class Employee {  
    private String name;  
    private int salary;  
    private static int count = 0;           // Class 변수  
  
    public Employee(String name, int salary) { // 생성자  
        this.name = name;  
        this.salary = salary;  
        count++;           // Class 변수인 count를 증가  
    }  
  
    public String getName() {  
        return name;  
    }  
  
    public int getSalary() {  
        return salary;  
    }  
}
```

# Class 문제 17

## ■ Employee.JAVA

```
public static int getCount() {    // Class Method  
    return count;  
}
```

*// 객체가 소멸될 때 호출*

```
protected void retire() {  
    count--;    // 직원이 하나 줄어드는 것이므로 count를 하나 감소  
}  
}
```

# Class 문제 17

## ■ Employee.JAVA

```
public static void main(String[] args) {  
    Employee employee1 = new Employee("김철수", 35000);  
    Employee employee2 = new Employee("홍길동", 50000);  
    Employee employee3 = new Employee("이대한", 20000);  
  
    System.out.println("현재의 직원수 = " + Employee.getCount());  
}
```

# Class 문제 18(Counter)

- 계수기를 만들어보자
  - 키는 up, reset



# Class 문제 18(Counter)

## ■ Counter.JAVA

```
public class Counter {  
    private int count;  
  
    public Counter() {  
        count = 0;  
    }  
  
    int getCount() {  
        return count;  
    }  
  
    void reset() {  
        count = 0;  
    }  
  
    void up() {  
        count++;  
    }  
}
```



# Class 문제 18(Counter)

## ■ Main.JAVA

```
public static void main (String[] args) {  
    Counter counter = new Counter();  
  
    counter.up();  
    counter.up();  
    counter.up();  
    System.out.println("Counter 값은 : " + counter.GetCount());  
    counter.reset();  
    System.out.println("Counter 값은 : " + counter.GetCount());  
}
```

# Class 문제 19(삼각형 면적)

- Data와 Information을 구분해보자
- 삼각형 클래스 정의
  - 속성
    - 밑변(width) : double
    - 높이(height) : double
  - 메소드
    - 면적을 계산하다(area()) : double
    - Data와 Information을 출력하다(toString()) : void
- 객체 생성 및 활용

# Class 문제 19(삼각형 면적)

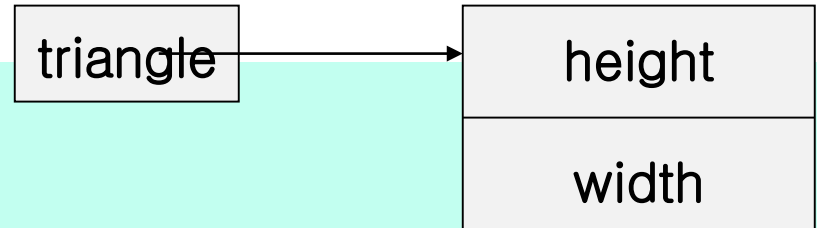
## ■ Triangle.JAVA

```
public class Triangle {  
    private double width;  
    private double height;
```

```
    public Triangle(double width, double height) {  
        this.width = width;  
        this.height = height;  
    }
```

```
    public void setWidth(double width) {  
        this.width = width;  
    }
```

```
    public void setHeight(double height) {  
        this.height = height;  
    }
```



# Class 문제 19(삼각형 면적)

## ■ Triangle.JAVA

```
public double area() {  
    double area = (height * width) / 2.0;  
  
    return area;  
}
```

@Override

```
public String toString() {  
    return String.format("%n 삼각형의 면적 "+  
        "%nWt 밑변 : %.2f CmWnWt 높이 : %.2f CmWn" +  
        "%n 면적 : %.2f Cm2Wn", width, height, area());  
}
```



Futuristic Innovator

京福大學校  
KYUNGBOK UNIVERSITY

# Class 문제 19(삼각형 면적)

## ■ Main.JAVA

```
public static void main(String[] args) {  
    Triangle triangle = new Triangle(3.5, 5);  
  
    triangle.setHeight(6.0);  
    triangle.setWidth(5.6);  
    System.out.println(triangle);  
}
```

삼각형의 면적

밑변 : 5.60 Cm

높이 : 6.00 Cm

면적 : 16.80 Cm<sup>2</sup>