Java™

스마트인재개발원
Smart Human Resources Development

이 도 연 연구원

학습목표

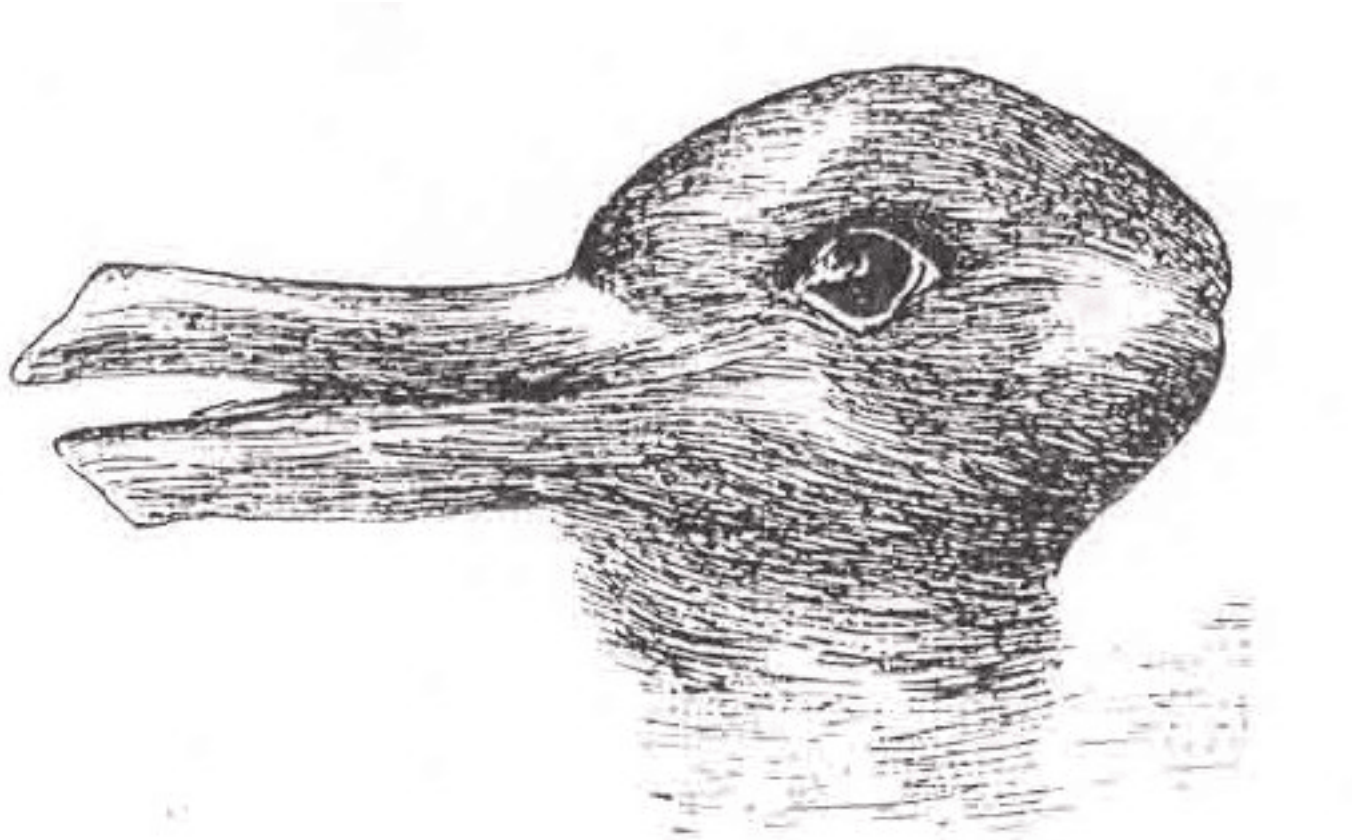
1. OOP의 개념과 필요성을 설명할 수 있다.

2. OOP의 특징에 대해 알아본다.

3. 객체를 생성하고 사용할 수 있다.

Java 패러다임(paradigm)

스마트인재개발원
Smart Human Resources Development

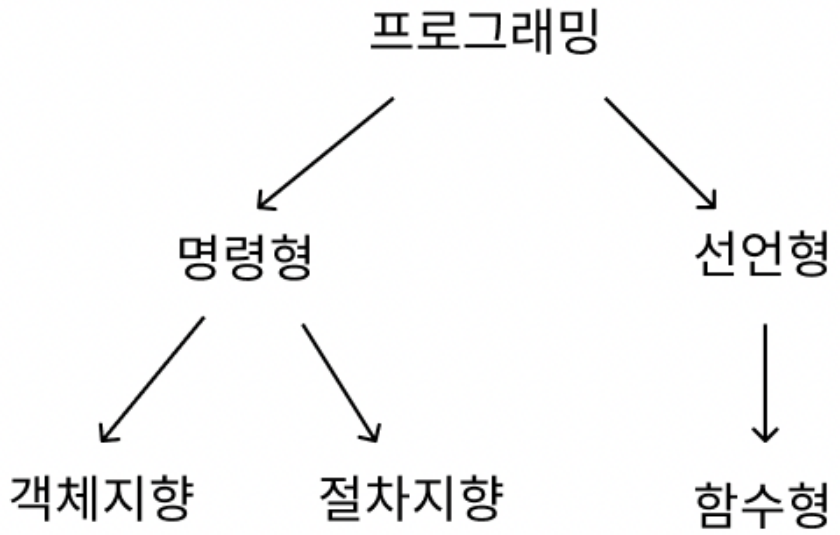


패러다임이란?

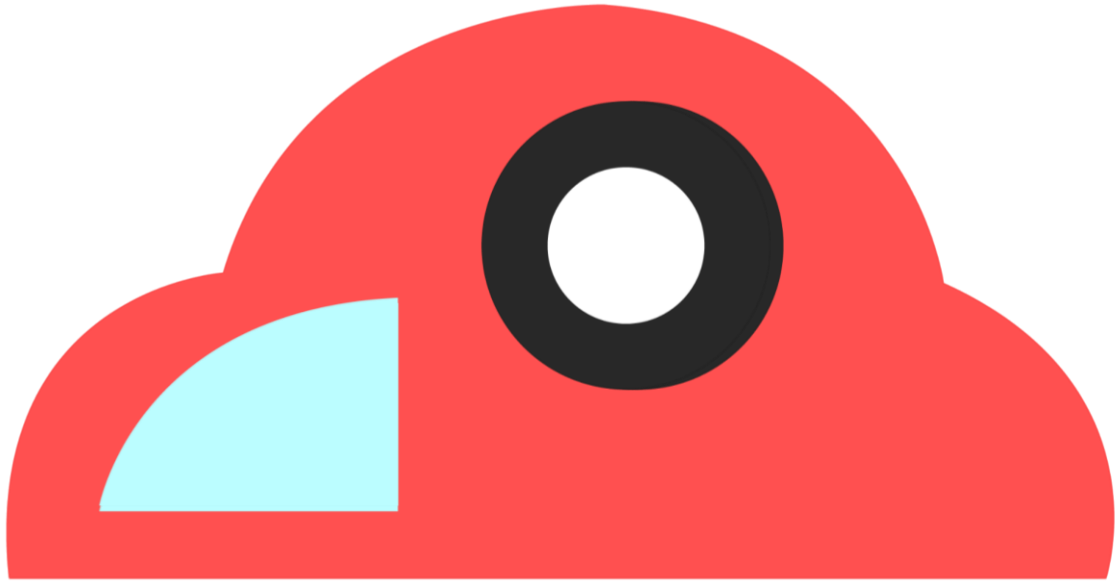
사람들의 견해나 사고를 근본적으로 규정하는 테두리

프로그래밍 방식에도 패러다임이 있다!

프로그래밍을 할 때 가지는 체계, 관점, 틀



객체지향프로그래밍이란?



관련영상:
https://youtu.be/vrhIxBWSJ04

정수형 데이터를 담을 수 있는 num을 선언하세요!

- int num = 0;

사람이라는 나만의 자료형을 만들어볼까?!

사람 데이터 1개를 담을 수 있는 person을 선언하세요!

이름, 나이, 성별… 어떻게 묶지?

Java 객체란?

- 여러가지 데이터들이 모여 있는 복잡한 자료형

- 내가 만드는 나만의 새로운 자료형

- 내가 새롭게 만드는 나만의 도구

ex) Scanner, Random, Math

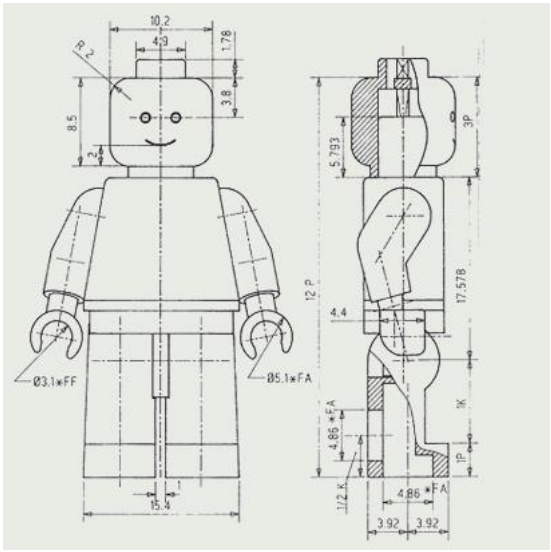
객체를 너무 어렵게
생각하지 말기!

class

object

class

실제 Object를 제작하기 전 설계하는 도면



Object(객체)

클래스를 기반으로 실제로 메모리에 할당되는 것

하나의 클래스로 서로 다른
객체를 생성할 수 있다.

Data

Field (속성)

- 이름
- 키
- 나이
- 성별
- 머리색
- 눈동자색

Logic

Method (행동)

- 걷다
- 먹다
- 자다
- 말하다
- 싸우다
- 생각하다

동일한 설계도면
↓
어떤 값이 들어가냐에 따라
서로 다른 값을 가진 객체 생성 가능!
↓
하나의 설계도면을 가지고
여러 개의 객체 생성 가능

People avatars collection

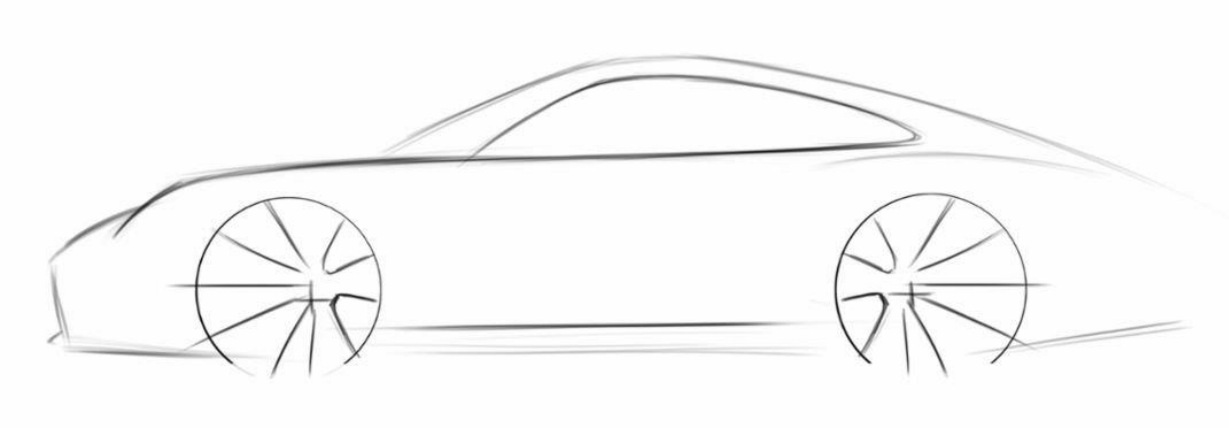
객체지향프로그래밍의 특징

추상화
(Abstract)

캡슐화
(Encapsulation)

상속
(inheritance)

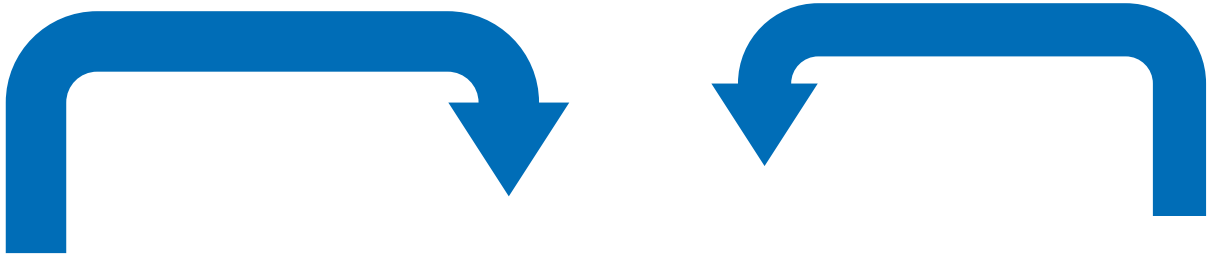
다형성
(polymorphism)



공통된 특징만 뽑아서
밑그림(틀) 제작

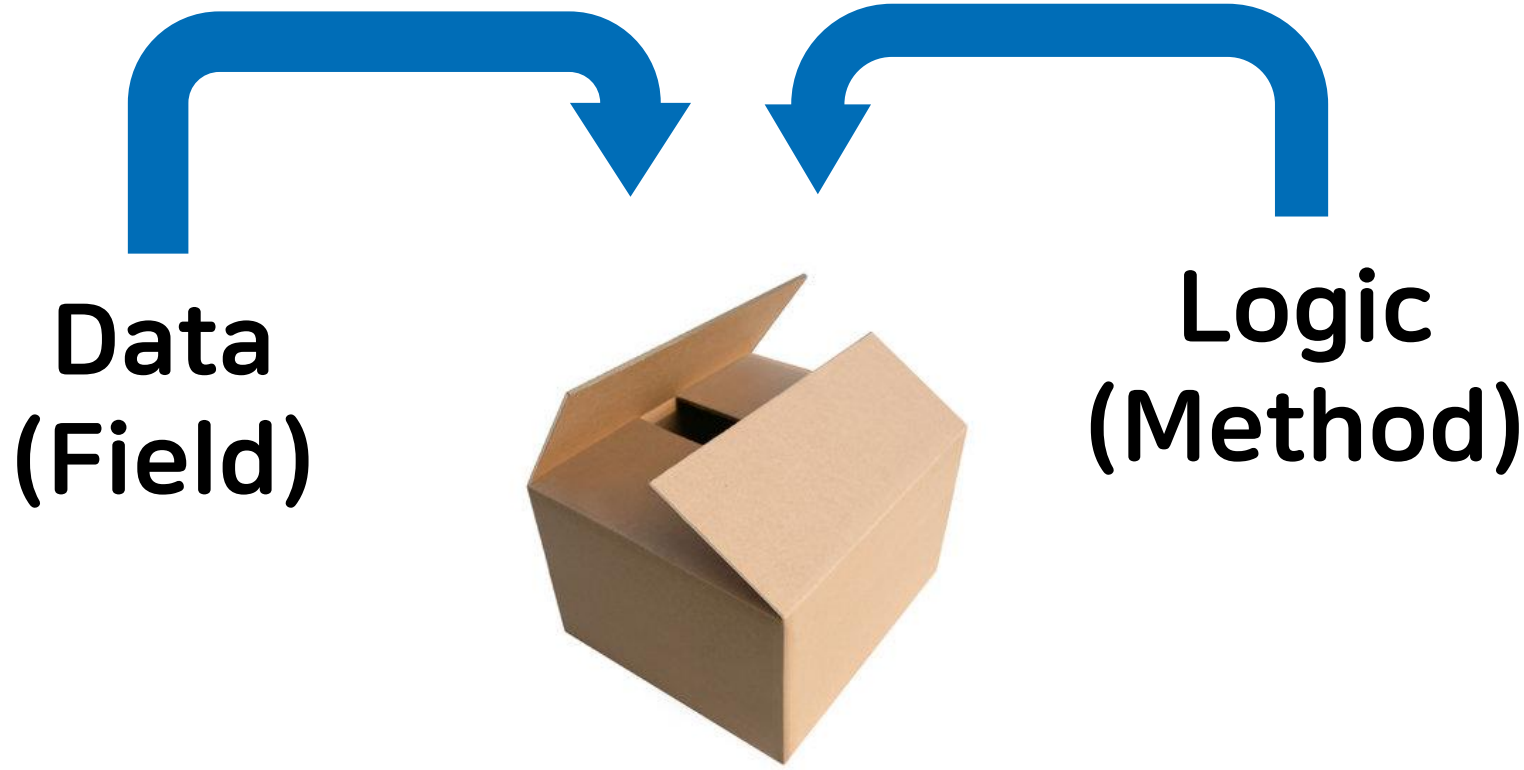
추상화(Abstraction)

- 객체에서 공통된 속성과 행위를 추출하는 기법
- 상세한 정보는 무시하고 필요한 정보들만 간추려서 구성
- 코드 상에서 구현(로직)부분을 제외한 오직 선언 부분만을 설계



장난감

설명서


Data
(Field)
Logic
(Method)
class

개발자

내부 보지마!
내부 열면 AS 안해준다?

사용자

내부에 흥미 없어!
Tv 프로가 보이면 OK~

음량조절이나 채널변경
같은 처리만 공개

인터페이스

공개된 처리만 사용

Scanner

next()
nextLine()
nextInt()
nextFloat()

Random

nextInt()

Arrays

sort()
binarySearch()

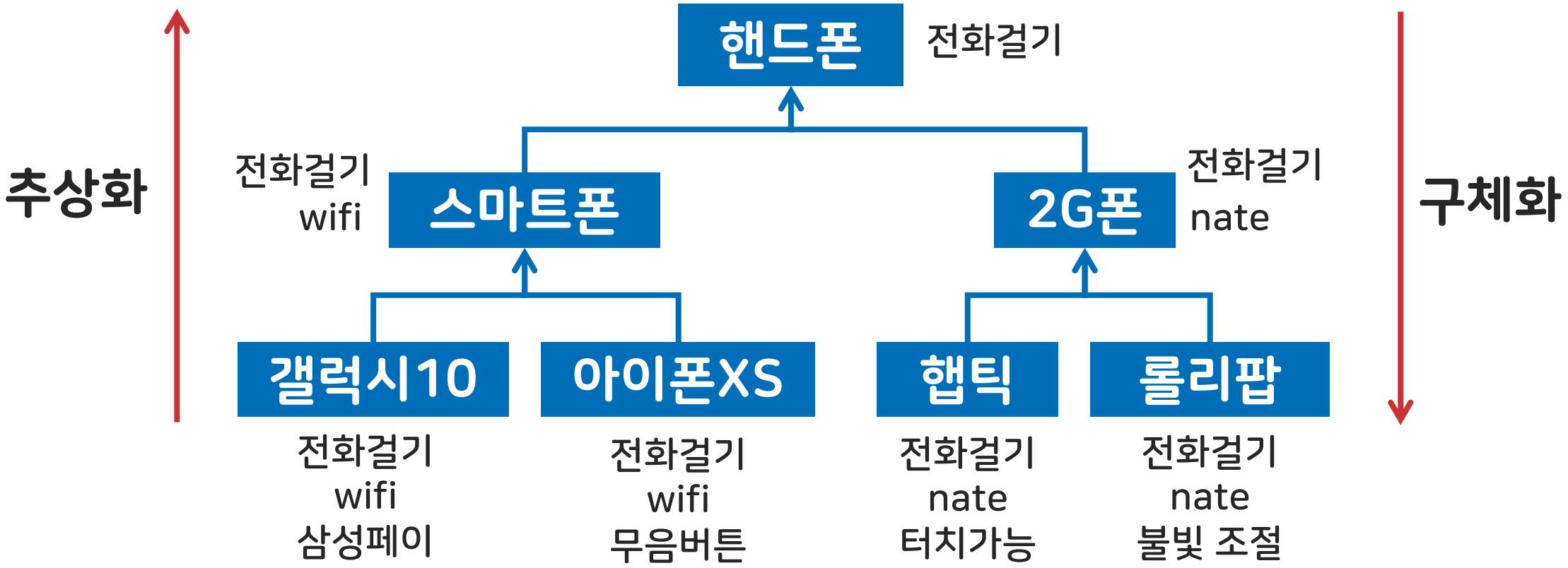
Math

pow()
round()
cos()
min()
max()

지금까지 내부 로직을 명확하게 모르더라도
공개된 메소드명을 통해 사용함!

캡슐화(Encapsulation)

- 관련된 필드(속성)와 메소드(기능)를 하나로 묶고, 실제 구현 내용을 외부로부터 감추는 기법(정보은닉)
- 만일의 상황(타인이 외부에서 조작)을 대비해서 특정 속성이나 메소드를 사용자가 조작할 수 없도록 숨겨 놓은 것.
- 외부에서는 공개된 메소드(기능)의 인터페이스를 통해 접근할 수 있다.


추상화
구체화
핸드폰
전화걸기
전화걸기
wifi
스마트폰
전화걸기
nate
2G폰
갤럭시10
아이폰XS
햅틱
롤리팝
전화걸기
wifi
삼성페이
전화걸기
wifi
무음버튼
전화걸기
nate
터치가능
전화걸기
nate
불빛 조절

상속(Inheritance)

- 이미 작성된 클래스(상위클래스)의 특성을 그대로 이어받아 새로운 클래스 (하위클래스)를 생성하는 기법

- 기존 코드를 그대로 재사용하거나 재정의 -> 재사용 + 확장

Java 다형성

정해진 규격만 맞다면
다양한 곳에서 기능 수행 가능!

다형성(Polymorphism)

- 사전적 의미 '다양한 형태로 나타날 수 있는 능력'
- 같은 기능(메소드)를 호출하더라도 객체에 따라 다르게 동작하는 것
- 상위클래스의 동작을 하위클래스에서 다시 정의하여 사용하는 것
  또한 다형성으로 볼 수 있다.

객체지향프로그래밍의 장점

- 신뢰성 있는 소프트웨어를 쉽게 작성할 수 있다.
- 코드를 재사용하기 쉽다.
- 유지보수가 용이하다.
- 직관적인 코드 분석이 가능하다.
- 소프트웨어 생산성이 향상된다.

Java 클래스의 구조

스마트인재개발원
Smart Human Resources Development

```
public class Person {

    String name;
    int age;

    public void eat() {
        System.out.println("먹다.");
    }

    public String tell() {
        return "말하다";
    }
}
```

클래스명

객체의 속성 정의
(Field)

객체의 기능 정의
(Method)

Java 클래스 설계해보기!

스마트인재개발원
Smart Human Resources Development

회원
속성(필드)
아이디 이름 나이
기능(메소드)
메시지 보내기 로그인 하기 기프티콘 보내기

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

클래스 명
(자료형 개념)

객체 명
(레퍼런스 변수)

반드시 () 필수

객체 명
(레퍼런스변수)

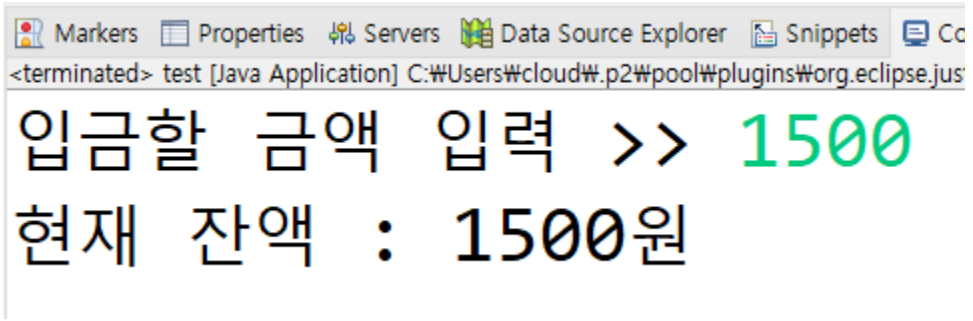
클래스타입의 객체

Java OOP실습 - 통장관리 프로그램

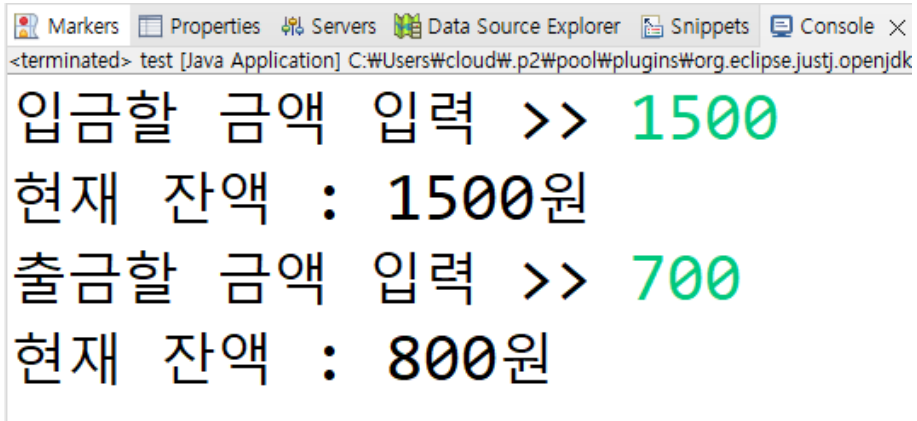
스마트인재개발원
Smart Human Resources Development

통장(Bankbook)
속성(필드)
잔액(money)
기능(메소드)
입금하다(deposit) 출금하다(withdraw) 잔액보기(showMoney)

1. 통장에서 1500원을 입금한 후 출력하세요.



2. 통장에서 500원을 인출한 후 출력하세요.



```
public static void main(String[] args) {

    Bankbook bank = new Bankbook();
    bank.money = 1500;
    bank.money -= 700;
    System.out.println(bank.money);
}
```

접근을 의미하는 기호

Tip!
.을 "객체가 가지고 있는" 의 의미로 외우기!

캡슐화(정보은닉)가 지켜지지 않음!

학생의 정보를 담을 수 있는 Student클래스를 작성하세요.
Student클래스는 다음과 같은 필드를 갖습니다.

자료형태	변수 이름	설명
String	name	이름
String	number	학번
int	age	나이
int	scoreJava	Java 점수
int	scoreWeb	Web 점수
int	scoreAndroid	Android 점수

Main클래스에서 Student 설계도를 가지고 stu1, stu2 객체를 생성하고 다음과 같이 초기화하세요.

stu1		stu2	
필드	학생 정보	필드	학생 정보
name	이도연	name	임경남
number	20241111	number	20242222
age	20	age	20
scoreJava	50	scoreJava	90
scoreWeb	99	scoreWeb	25
scoreAndroid	77	scoreAndroid	50

```
public class Student {

    private String name;

    public String getName() {
        return name;
    }
    public void setName(String name) {
        this.name = name;
    }
```

직접 필드 접근 불가능

필드 값을 가져갈 수 있는 Getter메소드

필드에 새로운 값을 넣을 수 있는 Setter메소드

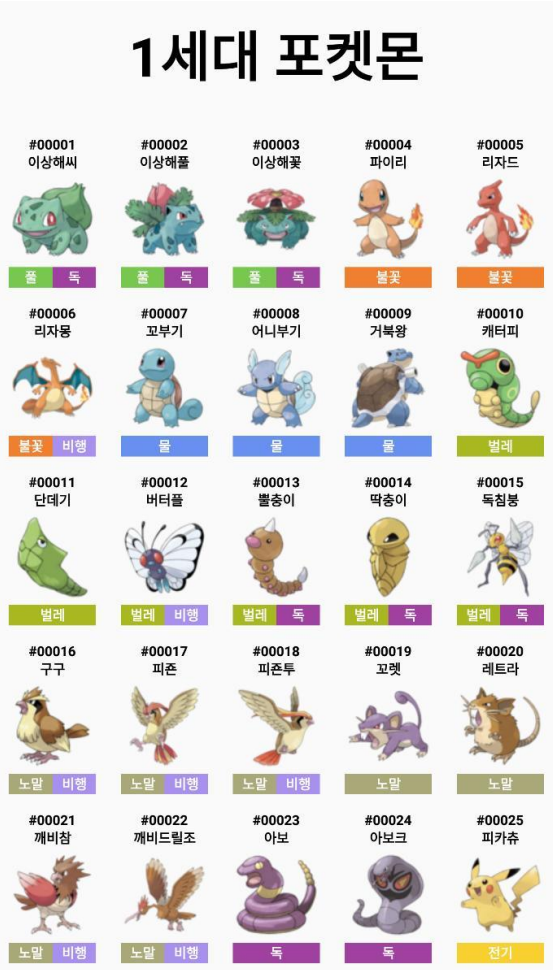
생성자의 특징

- 생성자도 메소드(오버로딩 가능)
- 생성자 이름은 클래스 이름과 동일
- 생성자는 리턴 타입을 지정할 수 없음
- 생성자는 new를 통해 객체를 생성할 때 호출됨
- 생성자는 하나 이상 선언되어야 함
- 기본생성자는 자동으로 생성, 생략 가능

단, 개발자가 새로운 생성자 선언 시
기본 생성자는 사라진다!

Java OOP최종 실습 – 포켓몬 게임

스마트인재개발원
Smart Human Resources Development



게임 rule.

1. 두 마리의 포켓몬을 생성한다.
2. 사용자로부터 포켓몬을 선택하게 한다.
3. 공격 혹은 스킬 공격 둘 중 하나를 선택하게 한다.
4. 선택한 포켓몬이 다른 포켓몬을 공격한다.

   (공격 시 포켓몬 hp – 공격력)
4. 스킬 공격 시 공격력은 1.5배 증가한다.
5. 한 마리의 포켓몬이 죽을 때까지 게임을 반복한다.

다음시간에 배울 내용

ArrayList