Java 2

2장 객체지향 프로그래밍과 클래스

/* elice */

Contents

01

클래스

클래스

자바 프로그래밍의 중심에는 **클래스**가 있다.

프로그램 작성한다. =

필요한 클래스의 객체를 생성하고

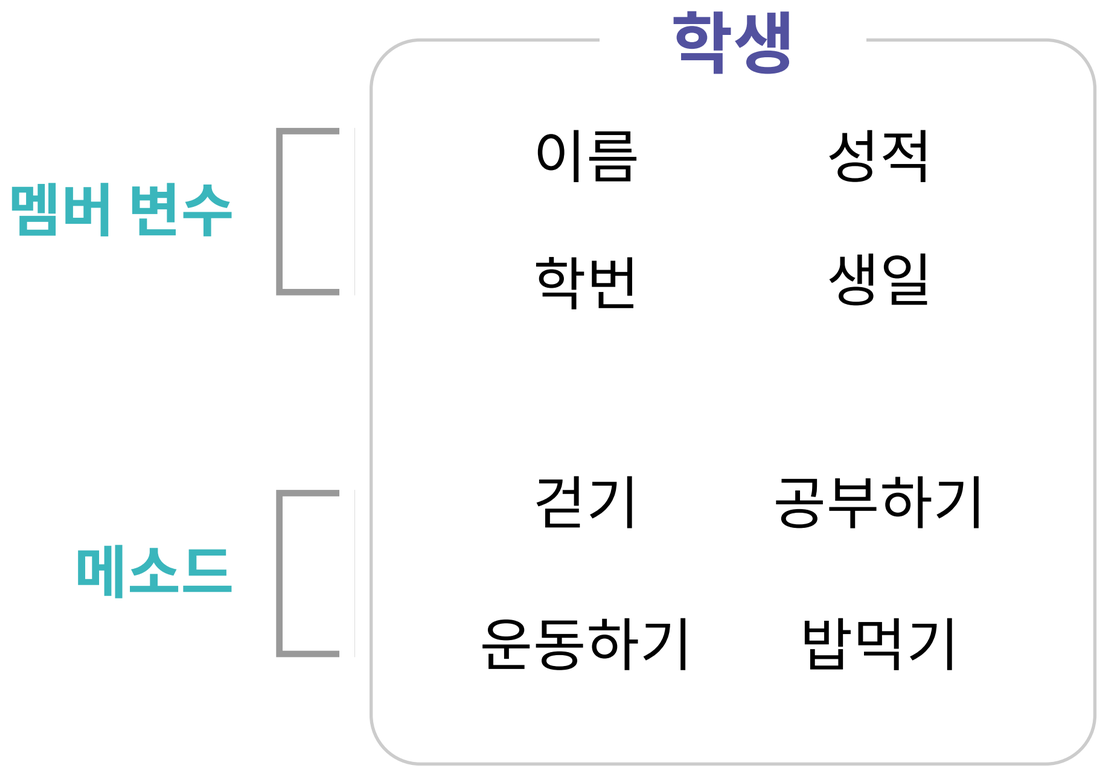
그 객체의 **속성**과 **기능**을 호출한다.

클래스

클래스는 **멤버 변수**, **메소드**로 구성된다.

클래스

속성(멤버 변수)과 **기능**(메소드)을 묶어 놓은 집합체



클래스

클래스는 **멤버 변수**, **메소드**로 구성된다.

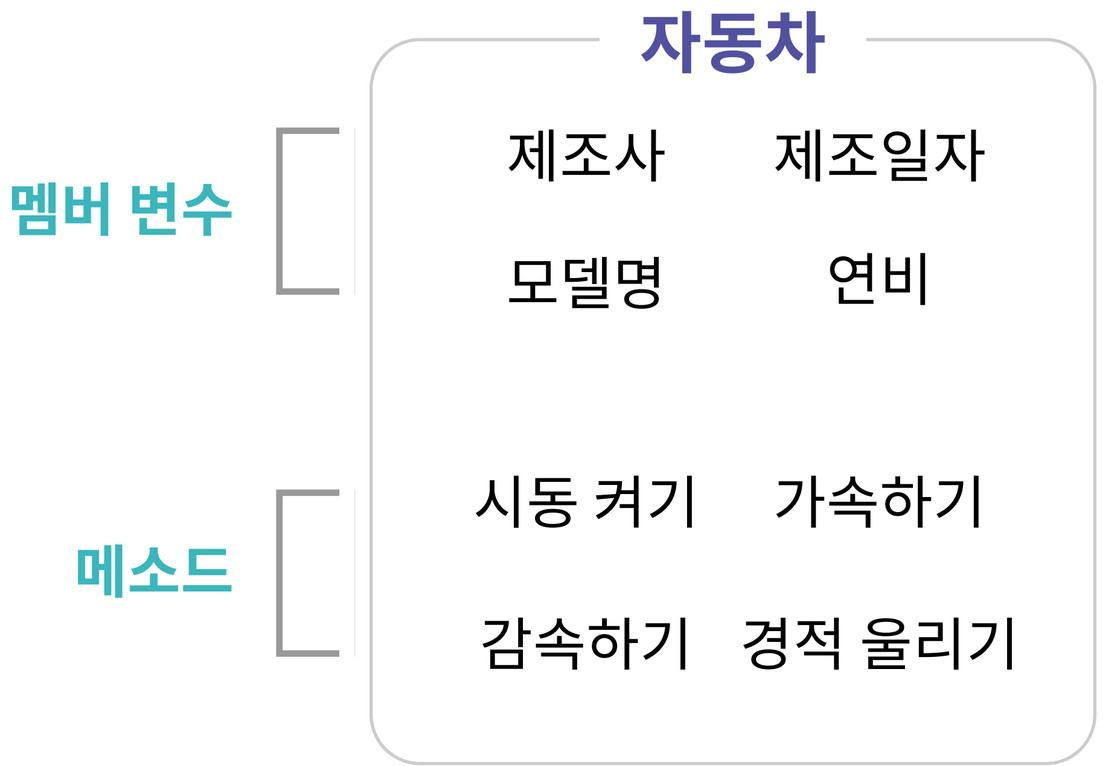
Example

```
class Student {
    int number;     //학번
    int score;      //시험 점수
    String name;    //이름

    void study(){
        System.out.println(“Studying”);  //공부하기
    }
}
```

클래스

속성(멤버 변수)과 **기능**(메소드)을 묶어 놓은 집합체



클래스

클래스는 **멤버 변수**, **메소드**로 구성된다.

Example

```
class Car {
    String model;           //모델명
    String manufactor;      //제조사
    double efficiency;      //연비
    double speed;           //속력

    void accerlate(){
        speed += 5;         //가속하기
    }
}
```

클래스

원하는 **속성**과 **기능**을 넣어 클래스를 **설계**할 수 있다.

클래스는 코드로 만드는 **설계도**

클래스

설계는 **문제 상황(요구 사항, Requirements)**을 잘 반영해야 한다.

클래스

요구 사항은 복잡하지만, 복잡한 것이 모두 필요하지는 않다.

설계자의 역량이 중요하다.

클래스 예시

SNS 게시물을 표현하는 클래스의 속성과 기능

Example

```
class Post {
    String author;          //작성자
    String[] comments;      //댓글
    int likes;              //'좋아요' 수
    String content;         //글 내용
    void like(){
        likes++;    //'좋아요' 누르기 기능
    }

}
```

클래스 예시

항공편을 표현하는 클래스의 속성과 기능

Example

```
class Flight {
    String pilot;                // 기장
    String[] passengers;          // 승객 명단
    int flightTime;              // 비행 시간
    String flightNumber;         // 항공기 번호
    void departure(int startTime){
        // …
    }
    void arrival(int endTime){
        // …
    }
}
```

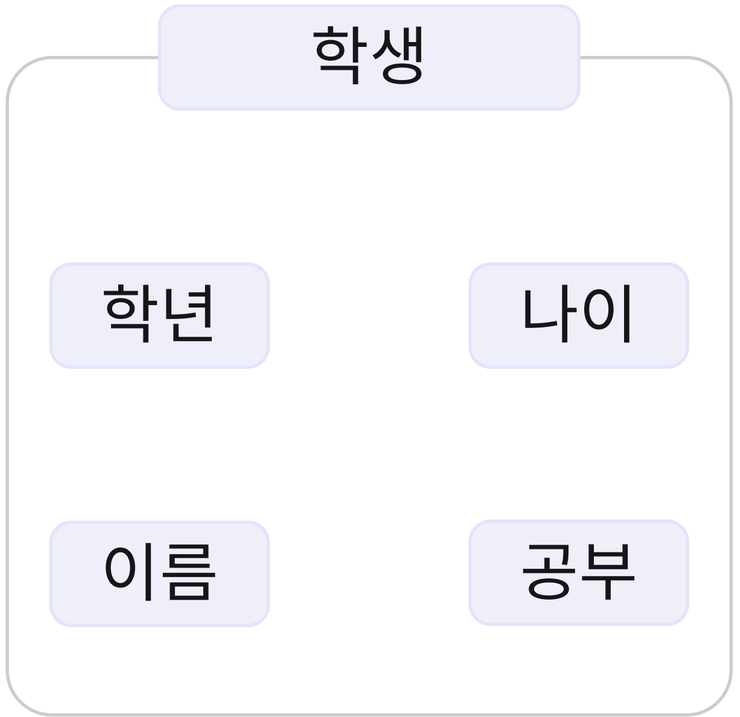
클래스 명명 규칙

변수, 메소드 명명 규칙은 **camelCase**

클래스 명명 규칙은 **PascalCase**

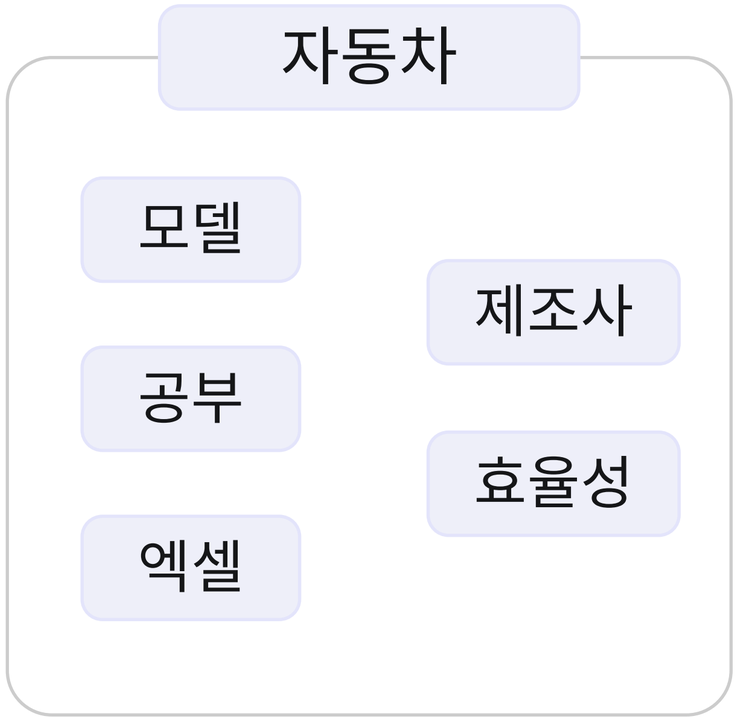
[실습1] 클래스 만들어보기(1)

Student 클래스를 만들어 봅시다!



[실습2] 클래스 만들어보기(2)

Car 클래스를 만들어 봅시다!



02

클래스와 인스턴스

인스턴스(객체)란?

클래스가 **설계도**라면, 인스턴스는 **제품!**

인스턴스란?

학생 클래스	학생 인스턴스
학생은 고유한 **학번**을 가진다.	김OO 학생
이름, 과목별 **성적**	이OO 학생
공부를 할 수 있다.	Elice 학생
…	…

인스턴스란?

게시물 클래스	게시물 인스턴스
게시물은 최대 20개의 **사진**을 가짐	Charlie의 4월 4일 게시물
댓글을 달 수 있음	David의 7월 17일 게시물
‘좋아요’를 누를 수 있음	Elice의 10월 10일 게시물
…	…

인스턴스란?

클래스	인스턴스
어떤 **데이터**가 있는지	그 클래스로 만든
어떤 **기능**이 있는지	**실제 예시(사례)**
명시한 **설계도**	

02 클래스와 인스턴스

인스턴스란?

new 키워드를 이용해 인스턴스 생성

Example

```
class Student {
  ...
}


...

Student s1 = new Student();
```

/* elice */

인스턴스의 사용

인스턴스의 **멤버변수**와 **메소드**를 코드에서 호출하자!

인스턴스의 사용

`.` 을 이용해 멤버변수, 메소드 호출

Example

```
Student s1 = new Student();

s1.name = “Elice”;
s1.score = 100;
s1.study();
```

/* elice */

참조 자료형

기본 자료형 vs 참조 자료형

Example

```
int num = 3;
char chr = ‘a’;

String str = new String();
Student s1 = new Student();
```

참조 자료형

기본 자료형은 단 8가지지만

참조 자료형은 무수히 많다.

참조 자료형의 예시

배열, 클래스, 인터페이스, 열거형

참조 자료형

기본 자료형	참조 자료형
크기가 정해져 있다.	클래스로 선언
자바에서 정해진 **키워드**	크기가 정해져있지 않다.
기본적으로 제공된다.	**String , 배열도 참조 자료형**

이전 과목에서의 궁금증 해결

String 자료형은 왜 대문자로 시작했나요?

→ 클래스(참조 자료형)이기 때문이다.

String의 예외적 허용

String은 예외적으로 기본 자료형처럼 사용 가능하다.

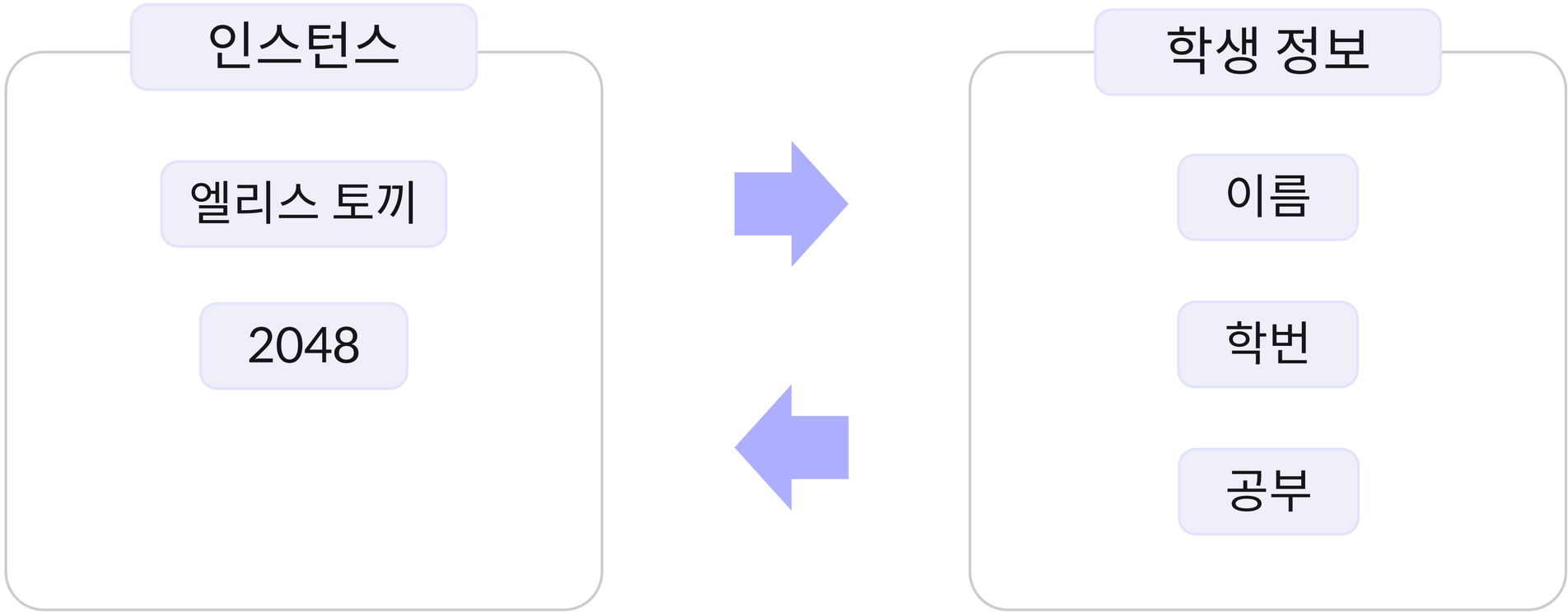
Example

```
String str = new String();
String name = “Elice”
```

/* elice */

[실습3] 인스턴스 사용하기

인스턴스를 생성하여 호출해봅시다!



03

객체지향 프로그래밍

객체지향 프로그래밍

속성과 **기능**으로 이루어진 **클래스**를 설계하고

클래스의 **객체**(인스턴스)로 원하는 로직을 구현

객체지향 프로그래밍

객체지향 프로그래밍의 4가지 특징

추상화

캡슐화

상속성

다형성

추상화

세상의 **모든** 학생을 다 나타낼 수는 없지만…

세상의 **모든** 자동차를 다 나타낼 수는 없지만…

추상화

공통된 부분을 묶어 **클래스**로 표현한다.

추상화

공통된 부분을 묶어 **클래스**로 표현한다.

추상화

객체지향 프로그래밍

객체지향 프로그래밍의 4가지 특징

추상화

캡슐화

상속성

다형성

Credit

/* elice */

코스 매니저
강윤수

콘텐츠 제작자
강윤수

강사
유동환 선생님

디자인
박주연

Contact

TEL
070-4633-2015

WEB
https://elice.io

E-MAIL
contact@elice.io