



클래스



- 클래스와 객체의 개념을 알고, 구분할 수 있다.
- 클래스 선언에서 중요한 속성과 키워드를 알 수 있다.
- 클래스 선언에서 구성요소들을 알고, 해당 내용에 대하여 명확히 구분할 수 있다.
- 클래스에서 필드, 생성자, 메소드를 활용할 수 있다.
- **this, static, final** 의 개념을 알고 구분할 수 있다.
- 패키지 개념을 알고 사용할 수 있다.
- 접근 제어자를 통해서 필드, 메소드에 대한 접근범위를 알고 활용할 수 있다.



생각해봅시다 :

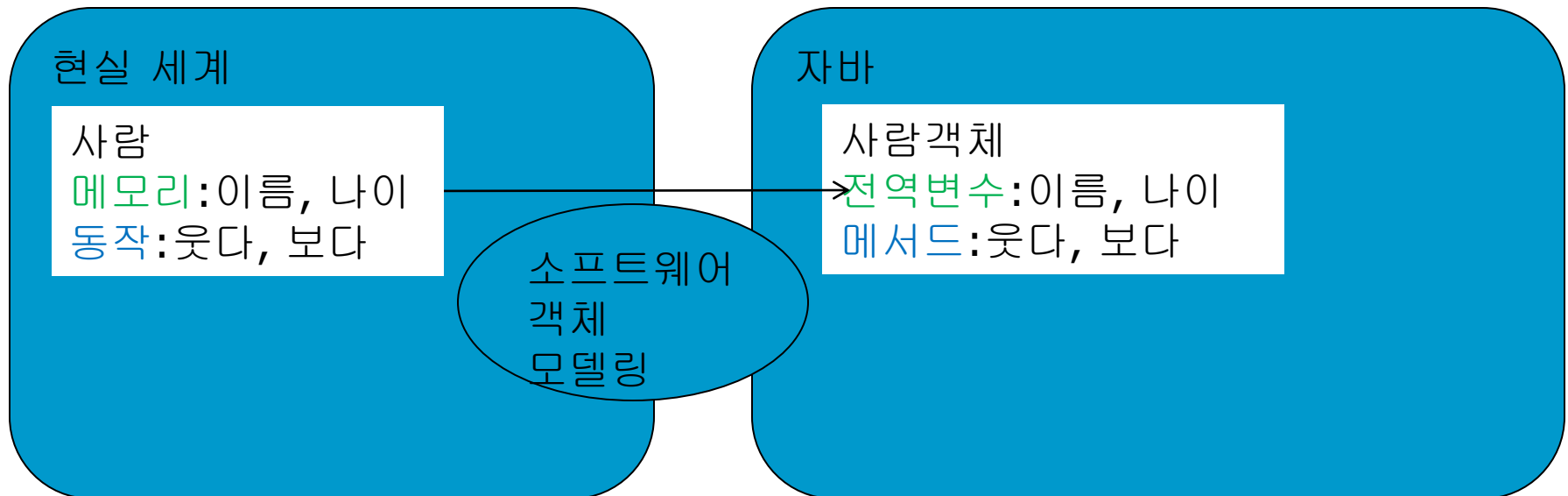
- 객체지향프로그램이 어떤 면에서 효율적인가?
 - 객체에는 속성과 메서드는 상호간의 어떤 작용을 할까?
 - 객체 **VS** 객체는 어떤 프로그램에서 활용될 것인가?
 - 객체 간의 접근제어는 왜 발생하는가?
 - **static**이라는 개념이 있는데, 공통 메모리에 쓰이는 개념이다. 어떤 경우에 이를 활용할까?
-



객체지향 프로그램!! :

■ 객체란?

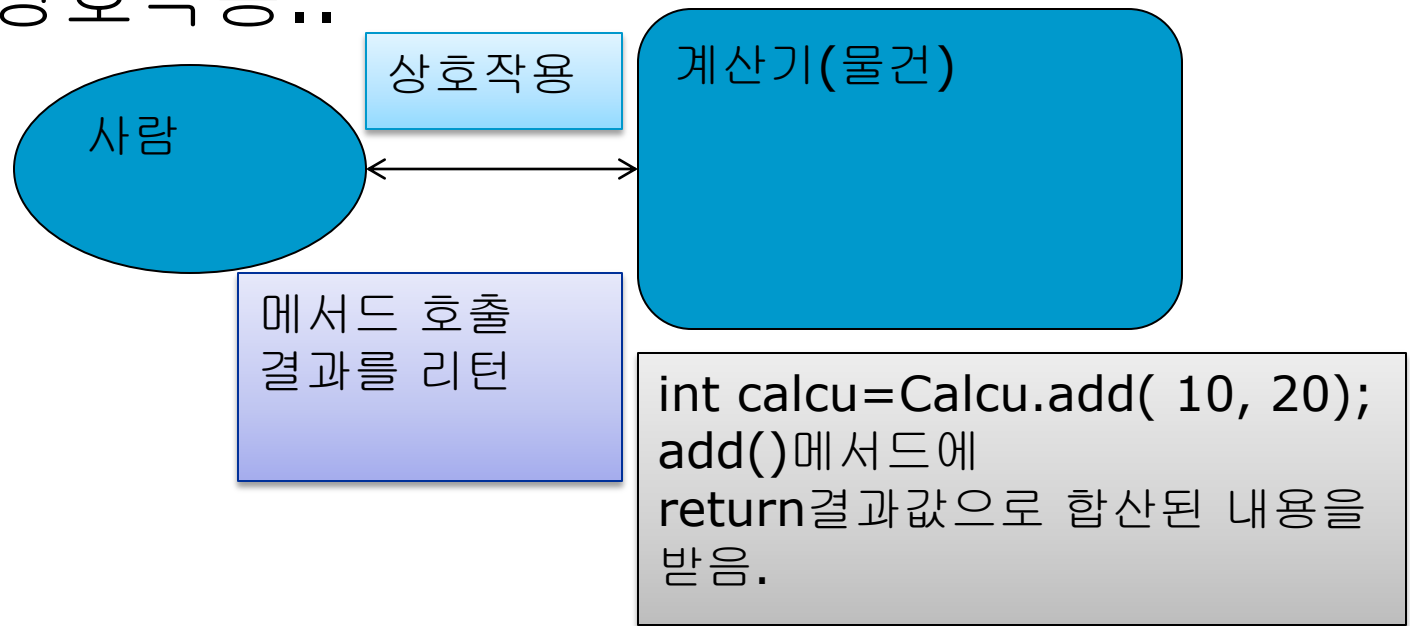
- 물리적으로 존재하거나 추상적으로 생각할 수 있는 것 중에서 자신의 속성이 있고, 다른 것과 식별 것을 말한다.





객체지향 프로그램 :

- 객체 모델링
 - 현실 세계나 추상적인 내용의 속성과 동작을 추려내어 소프트웨어 객체의 필드와 메서드로 정의해 나가는 과정.
- 객체의 상호작용..





객체 지향 프로그램의 특징 :

- 캡슐화(Encapsulation)
 - 객체가 포함한 속성과 메서드는 객체간의 관계에 있어서 감추거나 권한에 따라 접근이 가능하게 처리하는 것을 말한다. 여기서 사용되는 keyword로 접근제어자(access modifier)가 있다.
- 상속(Inheritance)
 - 상속이란 일반적으로 재산을 부모가 물려주는 개념이 있듯이, 자바에서는 부모가 가지고 있는 클래스의 속성과 메서드를 활용할 수 있는 개념으로부터 시작한다.
- 다형성(Polymorphism)
 - 같은 type이지만, 기능적으로 여러 다양한 객체를 이용할 수 있는 성질을 말한다. 프로그래밍 입장에서 상속받는 여러 객체를 대입함으로 다양한 기능을 이용할 수 있다.



객체와 클래스 :

- 건물을 만들 때..
 - 설계
 - 현실성 있는 구현으로 건물을 만들어 낼 수 있다.
- 클래스는 건물을 만들 때, 설계로 보면 된다. 실제 만들어진 건물은 객체라고 비유할 수 있다.
- 자바프로그래밍 단계(코드, 컴파일, 메모리, cpu)
 - `public class Person{}`(코드) `Person.java`
 - `main()` 메서드가 있는 클래스에서 실행하였을 때
 - `Person p1 = new Person();` ➔ 객체로 실제 처리할 수 있는 구체화 된다.
 - `Person p2 = new Person();` ➔ 여러 객체들이 만들어짐.



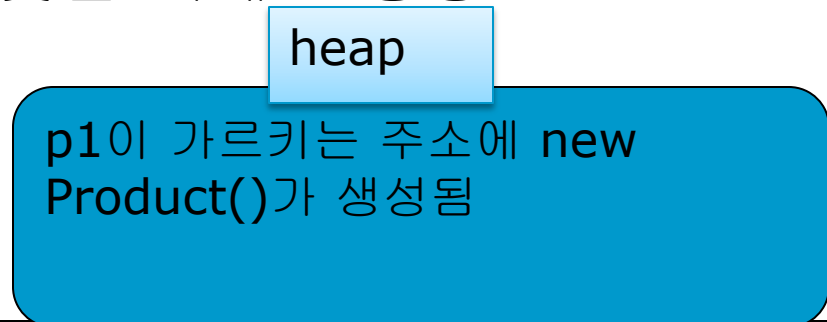
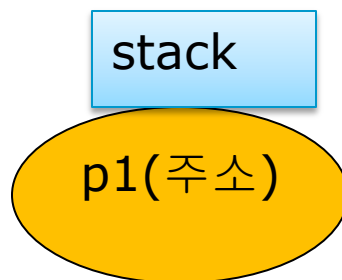
클래스 선언 :

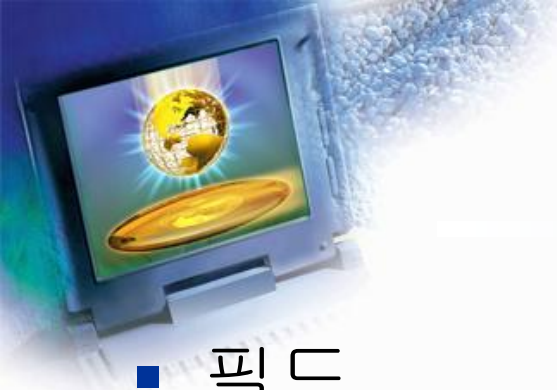
- 문자열로 만들어지고, 변수선언 규칙과 유사
 - 첫번째 글자는 숫자가 올 수 없다.
 - \$, _ 외의 특수 문자를 사용할 수 없다.
 - 자바 내장 키워드를 사용할 수 없다.
- 파일명 .java → public class 클래스명
 - 파일 안에는 여러 클래스명을 선언할 수 있지만, public이 붙은 클래스명은 하나만 사용할 수 있다. public이 붙은 클래스명이 파일명으로 사용된다.
 - Hello.java public class Hello{}
 - class Person{}
- 클래스명은 일반적으로 첫 자를 대문자로 시작한다.
 - public class Person{}
- 모든 객체는 main()메서드가 포함된 클래스에서 객체로 호출해야 실행이 가능하다.



객체 생성 :

- 클래스를 선언한 다음, `main()`에서드가 있는 클래스에서 객체를 아래의 형식으로 생성할 수 있다.
 - `public class Product{}`
 - `main(){`
 - `// stack(주소) = heap(실제객체)`
 - `Product p1 = new Product();`
 - `// class가 선언된 것을 객체로 생성함.`





■ 필드

- 객체의 고유 데이터, 정보를
- 선언형태는 변수와 비슷하지만 **field**로 사용되고 있다.

- 메서드나 생성자에서 선언되는 지역변수와 구분

■ 생성자

- **new** 연산자로 호출되는 클래스명과 동일한 이름을 가진 메서드. 객체 생성할 때, 1번만 호출 됨
- 메서드와 달리 **return**값이 없음.

■ 메서드

- 객체의 동작(기능처리)를 하는 것을 말한다.

```
public class Person{  
    // 필드(전역변수)  
    int fieldName;  
    // 생성자(클래스명과 동일한 메서드)  
    Person(){}  
    // 메서드  
    void show(){}  
}
```

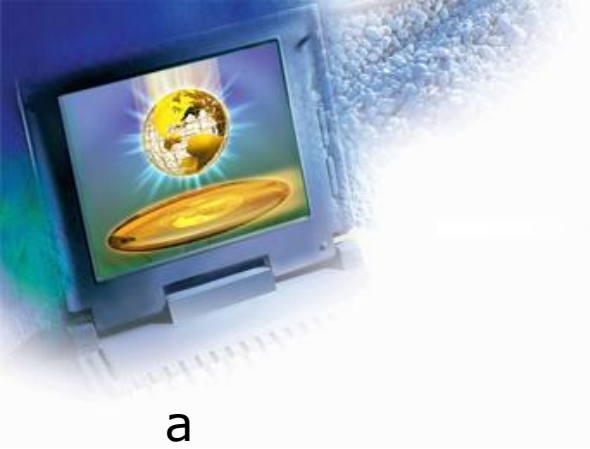
기본예제 :



/*



확인예제 :



a

확인 및 정리:





확인 정리:





감사합니다 !
