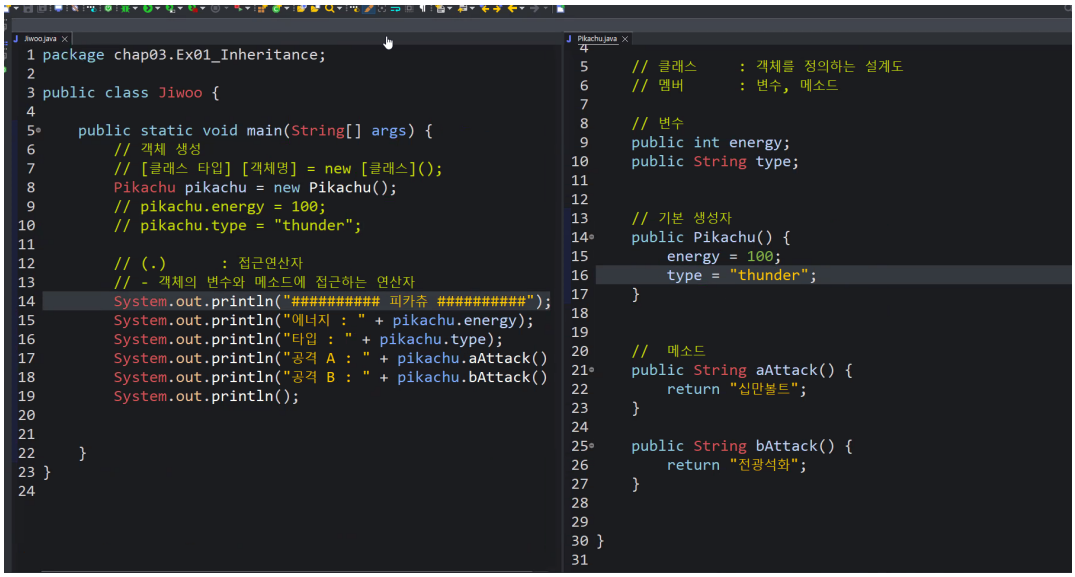
인스턴스: 메모리 공간에 할당된 객체

- 교수 객체, 자동차 객체...
- 인스턴스: 인프라 관리 시, 고유한 서버 하나하나
- 객체: 멤버(변수와 메서드)를 사용할 수 있도록 변수로 선언한 실체 = 인스턴스
- 클래스를 실제 사용할 수 있도록 변수로 선언한 것을 인스턴스라고 하며, 이 인스턴스를 객체라 할 수 있다. 객체는 높은 모듈성과 정보은닉의 장점을 제공한다.
- 생성자(Constructor): 인스턴스 변수를 초기화하기 위해 객체 생성시 호출되는 메서드
    - 필수요건:
        - 클래스명과 생성자명이 동일할 것

```
Class Car {
    Car() {
        ```

    }
}
```

- 객체 생성시(new) 호출되는 메서드 => 이때 메모리 공간 할당
- 클래스명과 생성자명이 동일하다
- 리턴타입을 정의하지 x
- 상속이 되지 않는다. 상속: 변수와 메서드만 상속, 생성자는 x, 자식 클래스는 생성자를 따로 관리
- 기본생성자(매개변수가 없는 생성자)를 정의하지 않으면, 컴파일러가 자동으로 생성해준다.
    - 매개변수가 있는 생성자를 정의하면, 더이상 컴파일러가 기본 생성자를 자동생성하지 않는다.



```
Jiwoo.java
1  package chap03.Ex01_Inheritance;
2
3  public class Jiwoo {
4
5      public static void main(String[] args) {
6          // 객체 생성
7          // [클래스 타입] [객체명] = new [클래스]();
8          Pikachu pikachu = new Pikachu();
9          // pikachu.energy = 100;
10         // pikachu.type = "thunder";
11
12         // (.)       : 접근연산자
13         // - 객체의 변수와 메소드에 접근하는 연산자
14         System.out.println("########## 피카츄 ##########");
15         System.out.println("에너지 : " + pikachu.energy);
16         System.out.println("타입 : " + pikachu.type);
17         System.out.println("공격 A : " + pikachu.aAttack()
18         System.out.println("공격 B : " + pikachu.bAttack()
19         System.out.println();
20
21
22     }
23 }
24
```

```
Pikachu.java
4
5      // 클래스      : 객체를 정의하는 설계도
6      // 멤버        : 변수, 메소드
7
8      // 변수
9      public int energy;
10     public String type;
11
12
13     // 기본 생성자
14     public Pikachu() {
15         energy = 100;
16         type = "thunder";
17     }
18
19
20     //  메소드
21     public String aAttack() {
22         return "십만볼트";
23     }
24
25     public String bAttack() {
26         return "전광석화";
27     }
28
29
30 }
31
```

- 다형성 : 다양한 형태를 갖는 성질, 같은 존재 = 이름이 같은데 형태가 다른 것
- 오버로딩: 매개변수의 순서, 개수, 타입이 반드시 달라야한다.
    - 접근제한자/ 리턴타입은 상관없음
- 상속: 부모 클래스에 정의된 변수와 메서드를 자식 클래스가 물려받는 것 =>
- super-class sub-class 상속 화살표
-
```

성립조건

구분	오버로딩	오버라이딩
메소드 이름	동일	동일
매개변수- 타입/순서/개수	다름	동일
리턴 타입	상관없음	동일

- 병원홈페이지
- 작은 홈페이지 솔루션 => 에이전시 ...
-