

지금까지는 인스턴스에 해당되는 '인스턴스 변수'와 '인스턴스 메소드'를 사용했습니다. 특정 인스턴스의 속성을 담당하는 변수와, 특정 인스턴스에 대해서 이루어지는 동작만 사용했다는 뜻이죠.

```
Person p1 = new Person("문종모", 25, 50000);
BankAccount a1 = new BankAccount(100000);

a1.setOwner(p1);      // a1에 대한 인스턴스 메소드
a1.withdraw(20000);    // a1에 대한 인스턴스 메소드
```

이번에는 인스턴스에 해당되지 않고 클래스에 해당되는 '클래스 변수'와 '클래스 메소드'를 사용해봅시다!

클래스 변수

현재는 **Person** 클래스에 **count** 라는 인스턴스 변수가 있습니다. **count** 는 현재 총 몇 개의 **Person** 인스턴스가 있는지 보관합니다.

```
public class Person {
    int count;
}
```

그런데 새로운 **Person** 인스턴스를 생성할 때마다 각 인스턴스의 **count** 변수를 바꿔줘야 하죠? 굉장히 번거롭습니다.

```
public static void main(String[] args) {  
    Person p1 = new Person();  
    p1.count++;  
  
    Person p2 = new Person();  
    p1.count++;  
    p2.count = p1.count;  
  
    Person p3 = new Person();  
    p1.count++;  
    p2.count++;  
    p3.count = p2.count;  
  
    Person p4 = new Person();  
    p1.count++;  
    p2.count++;  
    p3.count++;  
    p4.count = p3.count;  
  
    System.out.println(p1.count);  
    System.out.println(p2.count);  
    System.out.println(p3.count);  
    System.out.println(p4.count);  
}  
  
4  
4  
4  
4
```

이런 문제를 해결할 수 있는 게 바로 클래스 변수입니다. 클래스 변수를 정의하기 위해서는 **static** 이라는 키워드를 붙여주면 됩니다.

```
public class Person {  
    static int count;  
}
```

이제 **count** 는 특정 인스턴스에 해당되는 게 아니라, **Person** 클래스 전체에 해당되는 것입니다. 따라서 **count** 를 부를 때는 대문자로 쓴 클래스 이름을 사용해서 **Person.count** 를 써주면 됩니다.

```

public static void main(String[] args) {
    Person p1 = new Person();
    Person.count++;

    Person p2 = new Person();
    Person.count++;

    Person p3 = new Person();
    Person.count++;

    Person p4 = new Person();
    Person.count++;

    System.out.println(Person.count);
}

```

4

많이 깔끔해졌죠? 그런데 아직도 반복적인 코드가 많습니다. **Person.count++** 를 매번 해줘야 한다는 것 인데요. 반복적인 코드는 안 좋은 것이기 때문에 개선시킬 방법이 필요하겠죠?

우리는 지금 새로운 인스턴스를 생성할 때마다 **Person.count** 를 1 씩 늘려주고 있죠? 그러면 아예 생성자에 이 줄을 넣으면 되겠군요!

```

public class Person {
    static int count;

    public Person() {
        count++;
    }
}

public static void main(String[] args) {
    Person p1 = new Person();
    Person p2 = new Person();
    Person p3 = new Person();
    Person p4 = new Person();

    System.out.println(Person.count);
}

```

4

훨씬 깔끔해졌습니다! 이런 식으로 변수가 클래스 자체에 해당될 때에는 **static** 을 써서 클래스 변수로 만들어주면 되겠죠?

상수

우리가 자주 접하게 될 클래스 변수는 바로 상수입니다. **final** 을 공부할 때 상수를 보긴 했지만, 상수를 더 상수답게 쓰려면 **static** 과 함께 쓰는 것이 좋습니다. 상수는 인스턴스에 해당되는 것이 아니고, 따라서 **여러 복사본 대신 한 값만** 저장해두는 것이 맞기 때문입니다.

상수 이름은 보통 모두 대문자로 쓰고, 단어가 여러 개인 경우 **_** 로 구분 짓습니다.

```
public class CodeitConstants {
    public static final double PI = 3.141592653589793;
    public static final double EULERS_NUMBER = 2.718281828459045;
    public static final String THIS_IS_HOW_TO_NAME_CONSTANT_VARIABLE = "Hello";

    public static void main(String[] args) {
        System.out.println(CodeitConstants.PI + CodeitConstants.EULERS_NUMBER);
    }
}
```

5.859874482048838



수업을 완료하셨으면 체크해주세요.



수강생 Q&A 보기



[\(/questions?](#)
질문하기

[assignment_id=455&sort_by=popular\)](#)
[\(/questions/new?](#)

[assignment_id=455&op1=%EA%B0%9D%EC%B2%B4+%EC%A7](#)

< 이전 강의
final 예시 [\(/assignments/443\)](#)

다음 강의 > [\(/assignments/353\)](#)
클래스 메소드