**로고학습일지**

**kt ds University 자바 기반의 데이터 사이언티스트 양성과정**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 학습일시 | 2018. 07. 23 (월) | 장소 | kt ds University B관 201호 | **시 간** | 09:00~18:00 |
| 학습범위 | UI/UX | | | | |
| 작 성 자 | 류복기 | | | **강 사** | 장민창 강사 |

|  |  |
| --- | --- |
| 학습안건 | -클래스 |

|  |  |
| --- | --- |
| 학습내용 | 내용 |
| 오늘은 클래스에 대해 알아보겠습니다.  클래스는 크게 상태와 행동으로 구성됩니다.  우리는 상태를 변수로 행동은 기능, 즉 method로 표현할 것입니다.  여기서 행동은 상태의 변화를 가져오게 됩니다.  예를 들어, 사람의 상태가 갈증이라고 하면, 물을 마시는 행동을 통해 갈증은 해소될 수 있습니다.  우리가 배우고 있는 java라는 언어는 객체지향 언어라는 특징을 가지고 있습니다.  말 그대로, 왠만하면 객체를 사용하라는 뜻입니다.  실제 세계는 대부분 객체들로 이루어져있습니다.  객체 지향이라는 말을 설명하기 위해서 간단히 예를 들어보겠습니다.  병원이라는 세계를 생각해봅시다.  이 세계에는 의사, 간호사 등 상태와 행동을 가진 객체들로 가득 차 있습니다.  하지만 이러한 객체들은 혼자서 작동하지 않고 의사의 처방전을 받아 간호사가 주사를 놓는 등  객체 간의 무수한 상호작용으로 이루어져있습니다.  이와 같은 객체간의 통신을 객체지향이라고 표현할 수 있습니다.  지금 이러한 설명을 듣고 객체와 클래스가 많이 혼동될 것입니다.  객체와 클래스는 명백히 다릅니다.  예를 들어, 클래스를 설계도라고 표현한다면,  객체는 설계도를 바탕으로 만들어진 실제 제품이라고 할 수 있습니다.  이러한 개념을 바탕으로 실습을 진행해 보겠습니다.    위의 예제는, 합을 출력하는 기능과 평균을 출력하는 기능을 담고 있는 class입니다.  크게 상태와 기능으로 나뉠 수 있습니다.  먼저, 상태에 해당하는 부분은 변수입니다. 여기에서는 numberOne, numberTwo라는 변수가 해당합니다.  여기서, 빨간색 테두리는 멤버변수의 사용 범위를 나타냅니다.  멤버변수는 생성시 값을 주지 않습니다.  따라서, 위를 보면 int numberOne; int numberTwo;라고 선언해주었습니다.  다음으로, 기능에 해당하는 부분은 파란색 테두리에 해당합니다.  우리는 이것을 method라고 부릅니다.  메소드의 특징은 이름 끝에 소괄호(), 중괄호{}가 나온다는 것입니다.  파란색 테두리를 보면  numberOne = 50; numberTwo = 40; 처럼  멤버변수의 값이 할당되고 result값에 위 두 값을 더한 결과를 할당하는 것을 볼 수 있습니다.  이는 기능이 실행됨에 따라 상태(변수)가 변한다는 것을 의미합니다.  이제 CalculatorMain이라는 class파일을 또 생성해보겠습니다.    파란색 테두리를 보면 main뒤에 소괄호와 중괄호가 나오는 것을 볼 수 있습니다.  따라서 이것도 method에 해당합니다.  Main method는 application을 실행시키는 역할을 하는 method입니다.  내려와서 빨간색 테두리를 봅시다.  Calculator calc와 = new Calculator();로 구성되어 있습니다.  앞의 부분은 클래스의 선언이라고 부르고 뒤의 부분은 클래스의 객체화라고 부릅니다.  클래스 안의 메소드는 혼자서 사용될 수 없습니다.  클래스 안의 메소드를 불러오기 위해서는 클래스 이름.메소드 이름()을 사용합니다.  앞서 메소드를 정의 할 때는 소괄호와 중괄호를 모두 사용했지만  불러올 때는 소괄호만 사용하면 됩니다.  이를 메소드 콜이라고 합니다.  그렇다면 calc이라는 객체안에는 어떤 값이 담겨있을까요?  System.out.println()을 통해 객체 안의 값을 확인해 봅시다.  출력결과 Calculator@3caeaf62라는 값이 담겨있는 것을 확인할 수 있습니다.  이와 비교하기 위해,  Number 라는 int형 변수에 100이라는 값을 할당하고 출력해 봅시다.  Calc와 다르게 100이라는 값이 나타나는 것을 확인할 수 있습니다.  이러한 이유는 변수는 primitive 타입이고 객체는 reference 타입이기 때문입니다.  Primitive 타입의 경우 변수의 값이 하나에 불과하지만,  객체의 경우 여러 개의 멤버변수와 다양한 메소드를 가지고 있습니다.  그래서 하나의 값이라는 것을 담지 못하고 메모리 주소라는 것으로 표현하는 것입니다.  하지만, reference 타입에도 값을 담는 것이 존재합니다.  바로, wrapper class와 String입니다. 이 두 가지를 객체와 구분해서 알아두도록 합시다.  여기서 new는 새로운 객체를 만들어 그것을 메모리에 쓰는 역할을 합니다.  그리고 메모리의 가장 처음 부분을 calc에 할당하는 것이죠.  그 결과 calc를 console창에 출력하게 되면 calc의 메모리 값이 나오게 됩니다.   * Calculator@3caeaf62에서 Calculator는 reference type에 해당하고,   뒷부분은 메모리의 주소에 해당합니다.  실습 : 유저가 몬스터를 사냥하는 과정을 클래스로 만들어 보기 |