**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 | 장영선 | | | **강 사** | 장민창 강사 |

|  |  |
| --- | --- |
| 학습안건 | Class, 객체 지향 언어  Method |

|  |  |
| --- | --- |
| 학습내용 | 내용 |
| 1. **Class => 객체**   상태(변수), 행동(기능)을 가지고 있다.   1. 상태(변수) : 행동없이 상태만 가지고 있어도 객체가 될 수 있다. 2. 행동(기능) : 상태가 변화한다. (SideEffect) , 스스로 할 수 없으며, 누군가가 시켜야 한다.   ※ 한번 정의된 Class는 코드의 재사용을 극대화 한다!  3) 객체는 실제로 존재 한다면, Class는 이런 것들을 설계한 설계도가 된다고 생각하면 된다. 예를 들어 KTDS 건물이 객체라면, Class는 이 건물 안의 교실이 몇 개의 책상이 들어가는가? 이런것들이 Class가 된다.   1. **객체 지향** 2. 실제 세계라고 표현하면 된다. -> 회사, 기관, 단체 3. ex) 병원 -> 상태 : 의사, 간호사, 환자, 진찰에 필요한 기계 등 , 행동 : 의사가 진찰을 한다. 간호사가 주사를 넣는다. 등 4. 변수 = 항상 Primitive Type 객체 = 항상 Reference Type  Reference Type 은 항상 메모리 주소를 출력한다.   단, String 과 wrapper Class 는 메모리 주소를 가지지 않고, 각자의 값을 가진다.(출력)   1. **Class 객체화 = new Calculator();** 2. Class의 기능들은 () 소괄호가 붙고 { 중괄호가 붙는다. 3. **Method** Method는 데이터의 처리 과정을 하나로 묶어 놓은 것을 말한다. 1) Method를 생성 할 때 앞의 void는 빈 값을 리턴 즉, 원하고자 하는 Type을 리턴하기 위해  맨앞에 원하는 data type을 명시해야한다. Ex) int 메소드명() {}  현재까지 자주 사용하던 public static void main 에서 void는 반환 값이 없는 메소드를 뜻하고, main은 메소드 명을 뜻한다. 2) Method의 구조는 \* 접근제한자 반환자 메소드명 (파라미터 타입 파라미터명) {  return 반환 변수 혹은 수식  }   3) Arguments : 생성 된 메소드에 파라미터 타입과 파라미터 명이 적혀 있을 때 이것을 Arguments라고  한다.  4) Parameter : 파라미터는 메소드를 쓸 때 넣어줘야하는 값을 말한다. |
|  |  |

|  |  |
| --- | --- |
| 특이사항 |  |