**로고학습일지**

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

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

|  |  |
| --- | --- |
| 학습안건 | Commant (주석처리) , 사칙연산, Primitve Type, Reference Type, Array, List, Map  실습 내용 : 화씨 <-> 섭씨 변환기, 4가지 과목 평균, 배열을 이용한 실습 |

|  |  |
| --- | --- |
| 학습내용 | 내용 |
| 1. **Commant (주석)** 2. 주석은 3가지 형태로 작성 할 수 있다. 3. 작성방법  // -------- 실무에서는 사용되지 않음 /\* \*/ -------- 실무에서는 사용되지 않음 /\*\* \*/ -------- 실무에서도 사용되며 가이드를 위해서 많이 쓰임. 멤버 변수 또는 클래스 쪽에 가져다 댔을 때 Commant가 나오도록 되어있음. 4. **String (문자열)**   문자열을 저장하는 변수로 String 타입의 데이터를 저장하고자 할 땐 “ “를 사용한다.   1. **Arithmetic Operation (사칙연산)** 2. 더하기 + , 빼기 –, 곱하기 \*, 나누기 /, 나머지 % **나누기는 상황에 따라 다른 값이 나올 수 있으므로 주의 해야함!** # 정수 / 정수 = 정수   실수 / 정수 = 실수 ex) 10/3 = 3.3  정수 / 정수 = 실수로 표현 ex) 10/3 = 3.0  정수 / 정수 = 실수로 표현2 ex) 10/(3\*1.0) = 3.333333   1. 사칙연산의 우선 순위는 수학의 사칙연산 순위와 동일 또한 먼저 계산이 필요한 것은 () 표현한다. 2. 연산자의 규칙   \* 1. 숫자 + 숫자 = 숫자  \* 2. 숫자 + 문자 = 문자  \* 3. 정수 + 정수 = 정수  \* 4. 실수 + 실수 = 실수  \* 5. 실수 + 정수 = 실수  \* 6. 4byte + 8byte = 8byte  \* 7. 1byte + 2byte = 2byte  \* 8. int + long = long  \* 9. float + double = double  \* 10. int + int = int  \* 11. float + float = float |
|  | 1. **Type Casting** 2. 명시적 형 변환을 통해 변하고자 하는 값 앞에 변환 할 타입 (Type) 을 넣어주면 형 변환이 이루어진다.  (단, 높은 값을 낮은 값으로 형 변환을 진행 하고자 할 땐 OverFlow가 발생 되는 것에  주의 하여야 한다.) 3. **Primitive Type** 4. 원시적인 Type으로 앞에 소문자로 시작하도록 되어있다. 5. boolean, int, char, double, byte, short, long, float 6. **Reference Type** 7. 앞글자에 대문자로 시작 하는 것들로 wrapper class 라고 함 (감싸놓은 타입) 8. Boolean, Integer, Character, Double, Byte, Short, Long, Float 9. **Array & List & Map**   하나의 변수에 여러 개의 값을 저장 할 수 있도록 한 것으로 배열이라고 부름   1. Array 배열 선언 방법   \* 데이터 타입[] 배열명 = new 데이터 타입[변수의 개수]; 배열의 값 할당  \* 배열명[index] = 변수 값 사용 방법   \* 결과값 = 배열[필요로 하는 index] 2. List  배열의 크기를 원하는 대로 추가/삭제 할 수 있도록 지원한 것이다. List 선언 방법  \* List<ReferenceType> 배열명 = new ArrayList<ReferenceType>(); List 사용 방법  \* index에 변수 추가 : 배열명.add(변수값);  \* 해당 index에서 값 추출 : 배열명.get(index);  \* List 개수 확인 : 배열명.size();  \* List 삭제 : 배열명.remove(index);  삭제 시에는 가장 앞의 Index를 지울 시 다음 Index 값들이 당겨지는 것을 유의! 3. Map 배열의 index 대신 다른 타입을 사용 할 때 쓰임. Map 선언 방법  \* Map<Key, Value> 배열명 = new HashMap<Key, Value>(); Map 사용 방법  \* 변수 추가 : 배열명.put(key, value);   # List와는 다르게 덮어씌우는 형식을 가지고 있음.  \* 값 추출 : 배열명.get(key);  \* 데이터 삭제 : 배열명.remove(Key); |

|  |  |
| --- | --- |
| 특이사항 |  |