**로고학습일지**

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

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 학습일시 | 2018. 07. 24 (화) | 장소 | kt ds University B관 201호 | **시 간** | 09:00~18:00 |
| 학습범위 | UI/UX | | | | |
| 작 성 자 | 장영선 | | | **강 사** | 장민창 강사 |

|  |  |
| --- | --- |
| 학습안건 | Arguments, Parameter, 접근제한자, 캡슐화, 생성자 |

|  |  |
| --- | --- |
| 학습내용 | 내용 |
| 1. **Arguments, Parameter** 2. Arguments : Arguments는 메소드를 만들 때 그 안에 들어가는 값을 말한다.   Ex) 메소드 (Arguments, Arguments) {}   1. Parameter : 메소드를 불러 올 때 Arguments 가 있는 메소드라면 Parameter 를 입력해 주어야 한다. ex) 불러올 메소드명(파라미터, 파라미터);      1. **접근 제한자 : 접근 범위를 제한 하기 위하여 자바에서 지원** 2. 접근 제한자의 종류와 범위 public : 클래스 내부, 동일 패키지, 하위 클래스, 그 외의 영역 protected : 클래스 내부, 동일패키지, 하위 클래스   default :클래스 내부, 동일패키지  private : 클래스 내부 : 주로 멤버변수 범위는 클래스, 메소드, 멤버변수에서 사용 가능하다.   1. 주로 사용 하는 제한자 멤버변수를 선언 할 때 : private 메소드를 선언 할 때 : public 2. **캡슐화** 3. 복사는 깊은 복사와 얕은 복사가 있다. 깊은 복사 : 메모리를 카피하여 다른 영역에 카피하는 방법 얕은 복사 : 값만 가져가는 방법 ( 동일한 영역을 쓰기에 위험함 )   복사를 하기위해서는 setter와 getter를 통해 메모리를 따로 할당하는 것이 중요하다. this.healthPotionList = healthPotionList; 이러한 구문을  this.healthPotionList = new ArrayList<String>();  this.healthPotionList.addAll(healthPotionList); 이렇게 바꿔주는데.  addAll의 역할은  for ( int i = 0; i < healthPotionList.size(); i++) {  this.healthPotionList.add(healthPotionList.get(i));  }  이것과 같은 역할을 한다.   1. 캡슐화는 Refernce Type 을 캡슐화 하는데 메모리를 가지지 않는 String 과 wrapper 클래스는 캡슐화가 필요없다. 2. **생성자**   new 뒤에 나오면 무조건 생성자!  new 가 마련해 놓은 메모리에 객체를 채워 넣는 것!  생성자의 중요한 역할 중 하나는 초기화!  Refernce Type 은 반드시 초기화가 필요하다.   1. 생성자 메소드 만드는 방법은 Return Type을 생략하고 Class 명과 동일하게 만들어야한다.   일반 생성자와 aguments 생성자가 같이 있는경우 동일한 부분은 한곳에 넣고  this(); 를 aguments 생성자에 넣어 위 구문 실행후 할당받은 값만 넣어주도록 한다.   1. This 는 자신을 지칭하는 키워드 이다. 멤버변수와 메소드를 호출하는데 쓰인다. |
|  |  |

|  |  |
| --- | --- |
| 특이사항 |  |