**로고학습일지**

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

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 학습일시 | 2018. 07. 30 (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 | 메소드 오버로딩, JVM, Stack, Static, Heap, Class, Intance, 패키지 / Import, Override |

|  |  |
| --- | --- |
| 학습내용 | 내용 |
| 1. **Method OverLoading**   같은 이름이지만 arguments만 다른 경우를 메소드 오버로딩이라고 한다.  메소드 오버로딩은 3가지 조건을 충족해야한다.   1. 같은 Return Type 2. 같은 Method name 3. 다른 파라미터 타입 / 개수   자주 사용하지는 않지만 꼭 알아야 한다! 그나마 Utility에서 많이 사용한다.   1. **JVM** 2. Java App 실행 3. Memory 관리 4. .java => Compile => .class  RunTime 실행 중에 변수를 할당한다. 5. **Static, Stack, Heap** 1) Static 🡪 Heap 과 Stack과 완전히 분리된 독립된 형태로 공유 메모리이다. 2) Stack, Heap 🡪 Runtime : 변수, 메소드 결과, arguments Stack, Heap 에서는 Static에 접근 할 수 있지만, Static은 독립되어 Stack, heap에 접근 할 수 없다. Static은 독립적이기 때문에 하나당 메모리를 각각 가지고 있기 때문에 많이 할당 할수록 메모리를 많이 사용한다. 때문에 최소로 사용하는 것이 좋다. 6. **Class 와 Instance** 1) Class => 추상적 정의 ( 설계도 ) 2) Instance => 구체적 실체 ( 객체 ) 객체  멤버 변수 => Instance Variables 멤버 메소드 => Instance Method 만약, 멤버변수 또는 멤버 메소드 앞에 Static 이 붙으면 더 이상 객체가 아니다. 7. **면접으로 많이 나오는 질문!!!** 1) Static 과 Non Static의 차이   Static은 독립되어 메모리를 따로 가지고 있으며, 메모리를 공유 할 수 있다.  Non Static은 독립되지 않으며 메모리를 따로 가지지 않는다. 공유가 불가능하다. 8. **패키지 Package / Import** 대문자와 숫자를 쓰지 않는다. Java에서 다른 Package의 Class를 사용하기 위해서는 반드시 Import가 필요하다. 예외, 자주 사용되는 Class들에 대해서는 Import를 생략 할 수 있다. 활용용도가 높은 클래스들은 java.lang이라는 곳에 있다.  동일 패키지 일 때에는 Import가 필요하지 않다. 각각 다른 패키지에서 같은 Class 이름을 사용하는 것을 피하는 것이 좋다. //Ctrl + Shift + 0 🡺 Auto Import 9. **Interface 란 ?** 기능을 정의 한 것으로 Interface를 하기 위해서는 약속이 필요하다. 첫번째 약속, 접점을 만들어준다. 두번째 약속, 다형성 : 동작되는 형태를 보여주는 것, 동작이 되는 방식만 다를 뿐. ex) List와 Map은 Interface이다. List <> name = new ArrayList<>(); Map<> name = new HashMap<>(); 리모콘 리모콘 = new 삼성리모콘(); 10. **Override** Override는 재 정의 한다. 즉, 기능을 재 정의 할 때 쓰인다. 11. **상속 == 확장** 1) Interface : 실제로는 public abstract void 형식으로 구현되어있지만 abstract를 생략하고 있다. 2) Class   2-1) Abstract Class : 추상 메소드, 많이 사용됨 ☆ interface(implements) + extends.  2-2) Class  Class 상속   sub Class is a Super Class  까치는 새다. 참새는 새다. 장영선은 사람이다. ktdsFriends is a Friend  SuperClass name = new subclass(); |
|  |  |

|  |  |
| --- | --- |
| 특이사항 |  |