**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 | 자바 | | | | |
| 작 성 자 | 유재헌 | | | **강 사** | 장민창 강사 |

|  |  |
| --- | --- |
| 학습안건 | **01. 객체지향**  **02. 클래스** |

|  |  |
| --- | --- |
| 학습내용 | 내용 |
| **01. 객체지향**  - 객체지향 프로그래밍(OOP: Object Oriented Programming) : 소프트웨어를 개발할 때에도 부품에 해당하는 객체들을 먼저 만들고, 이것들을 하나씩 조립해서 완성된 프로그램을 만드는 기법이다.  - 객체란, 물리적으로 존재하거나 추상적으로 생각할 수 있는 것 중에서 자신의 속성을 가지고 있고, 다른 것과 식별 가능한 것을 말한다. 예를 들어, 물리적으로 존재하는 자동차, 자전거, 책, 사람 등이 모두 객체가 될 수 있다. 즉, 객체는 상태와 동작으로 구성되어 있다.  - 자바는 상태과 동작을 각각 멤버 변수(필드)와 메소드라고 부른다.  - 필드(멤버 변수 같은 뜻)에는 데이터 정보들을 담고 있고 메소드에는 클래스를 동작하는 기능에 대한 구현 코드가 들어간다.  - 객체 모델링(Object Modeling)은 현실 세계의 객체를 소프트웨어 객체로 설계하는 것을 말한다.  🡪 현실 세계 객체의 속성과 동작을 추려내어 소프트웨어 객체의 필드와 메소드로 정의하는 과정이라고 볼 수 있다.  - 객체 지향 프로그래밍의 특징  1) 캡슐화  🡪 객체의 멤버 변수, 메소드를 하나로 묶고 실제 구현 내용을 감추는 것을 말한다. 외부 객체는 객체 내부의 구조를 알지 못하며 객체가 노출해서 제공하는 필드와 메소드만 이용할 수 있다.  🡪 멤버 변수와 메소드를 캡슐화하여 보호하는 이유는 외부의 잘못된 사용으로 인해 객체가 손상되지 않도록 하는데 있다.  🡪 자바는 캡슐화된 멤버를 노출시킬 것인지, 숨길 것인지를 결정하기 위해 접근 제한자를 사용한다. 접근 제한자는 객체의 필드와 메소드의 사용 범위를 제한함으로써 외부로부터 보호한다.  2) 상속  🡪 부모가 가지고 있는 재산을 자식에 물려주는 것을 의미한다.  🡪 이것을 소프트웨어식으로 표현하면 상위 객체(슈퍼 클래스)가 가지고 있는 필드과 메소드를 하위 객체(서브 클래스)에게 물려주어 하위 객체가 사용할 수 있도록 한다.  🡪 상속은 상위 객체를 재사용해서 하위 객체를 쉽고 빨리 설계할 수 있도록 도와주고, 이미 잘 개발된 객체를 재사용해서 새로운 객체를 만들기 때문에 반복된 코드의 중복을 줄여준다.  🡪 상속은 상위 객체의 수정으로 모든 하위 객체들의 수정 효과를 가져오므로 유지 보수 시간을 최소화시켜주기도 한다.    3) 다형성  🡪 같은 타입이지만 실행 결과가 다양한 객체를 이용할 수 있는 성질을 말한다. 코드 측면에서 보면 다형성은 하나의 타입에 여러 객체를 대입함으로써 다양한 기능을 이용할 수 있도록 해준다.  🡪 자바는 다형성을 위해 부모 클래스 또는 인터페이스의 타입 변환을 허용한다. 부모 타입에는 모든 자식 객체가 대입될 수 있고, 인터페이스 타입은 모든 구현 객체가 대입될 수 있다. 다형성의 효과로 객체는 부품화가 가능하다.  **02. 클래스**  - 클래스는 어떤 목적을 달성하기 위해 필요한 기능들과 상태들을 하나로 묶어 분리시켜 놓은 모듈이다.  - 한번 정의된 클래스는 필요시 반드시 여러 번 호출해 사용함으로써 코드의 재사용을 극대화시켜준다.  - 클래스에는 객체를 생성하기 위한 멤버 변수와 메소드가 정의되어 있다. 클래스로부터 만들어진 객체를 해당 클래스의 인스턴스(instance)라고 한다.  - 즉, 클래스로부터 객체를 만드는 과정을 객체화 혹은 인스턴스화라고 한다.  - 객체 지향 프로그래밍 개발의 3가지 단계  1) 클래스를 설계한다.  2) 설계된 클래스를 가지고 사용할 객체를 생성한다.  3) 생성된 객체를 이용한다.  - new 클래스();  🡪 new는 클래스로부터 객체를 생성시키는 연산자다.  🡪 new 연산자 뒤에는 생성자가 오는데, 생성자는 클래스() 형태를 가지고 있다.  🡪 new 연산자로 생성된 객체는 메모리 힙(heap) 영역에 생성된다.  🡪 객체 지향 프로그램에서 메모리 내에서 생성된 객체의 위치를 모르면 객체를 사용할 없다. 그래서 new 연산자는 힙 영역에 객체를 생성시킨후, 객체의 주소를 리턴하도록 되어 있다. 이 주소를 참조 타입인 변수에 저장해두면, 변수를 통해 객체를 사용할 수 있다.  🡪 클래스 안에는 크게 멤버 변수, 생성자, 메소드 영역으로 나뉜다.  🡪 여기서 주의할 점은 멤버변수에 값을 초기화 하지 않는다.  🡪 객체 생성 시 메모리 구조 |
|  |

|  |  |
| --- | --- |
| 특이사항 |  |