**로고학습일지**

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

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 학습일시 | 2018. 07. 17 (화) | 장소 | kt ds University B관 201호 | **시 간** | 09:00~18:00 |
| 학습범위 | 자바 | | | | |
| 작 성 자 | 유재헌 | | | **강 사** | 장민창 강사 |

|  |  |
| --- | --- |
| 학습안건 | **01. 자바 프로그램**  **02. 변수**  **03. 데이터 타입**  **04. Naming Convention**  **05. 할당**  **06. Scope of variable**  **07. Arithmetic Operation & Scanner** |

|  |  |
| --- | --- |
| 학습내용 | 내용 |
| **01. 자바 프로그램**  - 자바 설명  1) James Gosling이 개발함  2) 자바는 다른 언어와 다르게 OS의 종속을 받지 않고 어떤 환경에서든 사용이 가능한 플랫폼 독립성을 가진다.  3) 그 외에 자바 특징  -. 객체지향 언어  🡪 프로그램을 개발하는 기법으로 부품에 해당하는 객체들을 먼저 만들고, 이것들을 하나씩 조립 및 연결해서 전체 프로그램을 완성하는 기법이다. 그리고 이것을 객체지향 프로그래밍이라고 한다.  (OOP : Object Oriented Programming)  -. 메모리를 자동으로 관리한다.(JVM)  🡪 C++은 메모리에 생성된 객체를 제거하기 위해 개발자가 직접 코드를 작성해야한다. 만약 이 작업을 성실하게 하지 않으면, 프로그램은 불완전해지고 갑자기 다운되는 현상을 겪게 된다. 즉, C++은 개발자가 직접 메모리를 할당해주고 해제해주어야한다. 그렇지 않으면 메모리 누수가 발생하기 때문에 프로그램이 죽게 된다.  🡪 하지만 자바는 개발자가 직접 메모리에 접근할 수 없도록 설계되었으며, 메모리는 자바가 직접 관리한다. 객체 생성 시 자동적으로 메모리 영역을 찾아서 할당하고, 사용이 완료되면 Garbage Collector를 실행시켜 자동적으로 사용하지 않는 객체를 제거시켜준다.  -. 멀티스레드를 쉽게 구현할 수 있다.  🡪 하나의 프로그램이 동시에 여러 작업을 처리해야 할 경우와 대용량 작업을 빨리 처리하기 위해 서브 작업으로 분리해서 병렬 처리하려면 멀티 스레드 프로그래밍이 필요하다. 프로그램이 실행되는 운영체제에 따라서 멀티 스레드를 구현하는 방법이 다르지만, 자바는 스레드 생성 및 제어와 관련된 라이브러리 API를 제공하고 있게 때문에 실행되는 운영체제에 상관없이 멀티 스레드를 쉽게 구현할 수 있다.  -. 동적 로딩을 지원한다.  🡪 자바 애플리케이션은 여러 개의 객체가 서로 연결되어 실행되는데, 이 객체들은 클래스로부터 생성된다. 애플리케이션이 실행될 때 모든 객체가 생성되지 않고, 객체가 필요한 시점에 클래스를 동적 로딩해서 객체를 생성한다. 또한, 개발 완료 후 유지보수가 발생하더라도 해당 클래스만 수정하면 되므로 전체 애플리케이션을 다시 컴파일할 필요가 없다. 따라서, 유지보수를 쉽고 빠르게 진행할 수 있다.  -. 오픈 소스 라이브러리가 풍부하다.  -. 다른 언어에 비용이 저렴하다.  - 자바 가상 기계(JVM) 원리  🡪 운영체제는 자바 프로그램을 바로 실행할 수 없는데, 그 이유는 자바 프로그램은 완전한 기계어가 아닌, 중간 단계의 바이트 코드이기 때문에 이것을 해석하고 실행할 수 있는 가상의 운영체제가 필요하다. 이것을 자바 가상 기계(JVM)이다. JVM은 실 운영체제를 대신해서 자바 프로그램을 실행하는 가상의 운영체제 역할을 한다.  🡪 운영체제별로 프로그램을 실행하고 관리하는 방법이 다르기 때문에 운영체제별로 자바 프로그램을 별도로 개발하는 것보다는 운영체제와 자바 프로그램을 중계하는 JVM을 두어 자바 프로그램이 여러 운영체제에서 동일한 실행 결과가 나오도록 설계한 것이다.    🡪 자바 프로그램은 확장자가 .java인 파일을 작성하는 것부터 시작된다. 이것을 소스파일이라고 하는데, 이 소스 파일을 컴파일러로(javac.exe)로 컴파일하면 확장자는 .class인 바이트 코드파일이 생성된다.  🡪 바이트 코드 파일은 JVM 구동 명령어(java.exe)에 의해 JVM에서 해석되고 해당 운영체제에 맞게 기계어로 번역된다. 바이트 코드는 하나지만, JVM에 의해서 번역되는 기계어는 운영체제 따라서 달라진다.  🡪 자바의 가장 큰 장점 중의 하나인 “Write Once, run anywhere(한 번 작성하면 어디서든 실행된다.)”이다.  - 자바 프로그램 개발 순서  🡪 자바 프로그램을 개발하기 위해서는 우선 파일 확장명이 .java인 텍스트 파일을 생성하고 프로그램 소스를 작성한다.  🡪 이렇게 만들어진 파일을 자바 소스 파일이라고 한다. 작성 완료된 자바 소스 파일은 컴파일로(javac.exe)로 컴파일해야한다. 컴파일이 성공되면 확장명은 .class인 바이트 코드 파일이 생성된다.  🡪 바이트 코드 파일은 완전한 기계어가 아니므로 단독으로 실행할 수 없고 JVM이 실행되어야 한다. 여기서 JVM을 구동시키는 명령어는 java.exe이다.  🡪 java.exe 명령어가 실행되면 JVM은 바이트 코드 파일(.class)을 메모리로 로드하고, 최적의 기계어로 번역한다. 그리고 main() 메소드를 찾아 실행시킨다.  - 자바 소스 작성에서부터 실행까지의 과정을 도식화로 표현  - 자바 설치  오라클 사이트 접속 후 jdk-10.0.1\_windows-x64\_bin 다운로드 및 설치  - 환경 설정  내컴퓨터 속성-> 고급 시스템 설정 -> 환경 변수  1) JAVA\_HOME 만든 후 C:\Program Files\Java\jdk-10.0.1 경로 지정  2) path C:\Program Files (x86)\Common Files\Oracle\Java\javapath;C:\Program Files (x86)\Intel\iCLS Client\;C:\Program Files\Intel\iCLS Client\;C:\Windows\system32;C:\Windows;C:\Windows\System32\Wbem;C:\Windows\System32\WindowsPowerShell\v1.0\;C:\Program Files (x86)\Intel\Intel(R) Management Engine Components\DAL;C:\Program Files\Intel\Intel(R) Management Engine Components\DAL;C:\Program Files (x86)\Intel\Intel(R) Management Engine Components\IPT;C:\Program Files\Intel\Intel(R) Management Engine Components\IPT;C:\Program Files (x86)\NVIDIA Corporation\PhysX\Common;%JAVA\_HOME%/bin; 경로 지정  3) 명령프롬프트에서 javac –version 명령을 입력하여 환경 설정이 제대로 되어있는지 확인  - 메모장에 Hello.java 파일을 만든 후 명령프롬프트에서  1) d: -> javac Hello.java 컴파일 실행  2) java Hello 실행하여 출력 예제 실습  **∴ JAVA 동작 원리 : Hello.java 파일을 컴파일하게 되면 Hello.Class 파일이 만들어지고 JVM에서 Hello.Class 파일을 실행하게 된다.**  - STS 설치  1) spring.io 사이트에 접속한 후 Win 64bit STS 파일 다운로드 및 설치  - STS 개발 환경 설정  1) 자바 인코딩 설정  🡪 Window -> Preferences -> Java -> Intalled JREs -> jre 삭제 -> Add -> Standard VM  🡪 Standard VM에서 C:\Program Files\Java 경로 들어가서 jdk-10.0.1 선택하고 finish한 후, 추가한 경로 체크 및 적용  🡪 Window -> Preferences -> General -> Workspace Text file encoding쪽에서 other를 선택하고 utf-8로 지정  2) 웹 페이지 인코딩 설정  🡪 Window -> Preferences -> Web -> CSS Files  -> HTML Files  -> JSP Files 설정에서 UTF-8로 지정  3) 실무에서 사용하는 폰트  🡪 Lucida Console (유료)  🡪 Monaco (무료)  -. Fira code  4) Monaco 폰트 설치 및 STS 개발 환경에 적용  -. 구글에서 monaco.ttf 검색 -> <https://github.com/todylu/monaco.ttf> -> monaco.ttf 다운로드 및 설치  -. Window -> Preferences -> General -> Apperance -> Colors and Font -> Basic -> Text Font에서 Monaco 글꼴 선택  5) 반디집 설치  **02. 변수**  1) 변수란 : 결과값을 담는 저장공간  🡪 사용자로부터 입력받은 값  🡪 계산을 위해 잠시 저장하는 값  🡪 계산의 결과값  2) 변수 사용 방법  🡪 변수 타입 변수명 = 변수값 ex) int age = 29;  🡪 변수타입 변수명 ex) int age  **03. 데이터 타입**  1) 컴퓨터는 0,1만 인식한다.  2) 사용자가 입력할 수 있는 값(자바가 처리할 수 있는 값)  🡪 숫자  🡪 문자  🡪 True/False  3) 숫자  🡪 정수(byte, short, int, long) -> byte와 short는 사용하지 않고 디폴트는 int형이다.  🡪 Ex) long data = 10000000000; 에서 Error가 발생하는 이유는 정수의 디폴트인 int의 범위가 넘어가렸기 때문에  long data = 10000000000L;로 표기한다.  🡪 실수 (float, double) -> 실수 데이터 타입의 디폴트는 double형이다.  🡪 Ex) float pi = 3.14; 에서 Error가 발생하는 이유는 실수 데이터 타입이 double이기 때문에  float pi = 3.14f; 로 뒤에 f나 F로 명시해야한다.  -.  C:\Users\Admin\Desktop\1.PNG    4) 문자  🡪 문자 타입은 char과 String이 있다.  🡪 Ex) char ch = ‘A’;  String name = “Yoo Jae Heon”;  5) true/false  🡪 true/false 표현하기 위해서 boolean이라는 타입을 사용한다.  🡪 Ex) boolean zombie = true;  🡪 플래그 값을 지정할 때 사용하는 용도로도 쓰인다.  **04. Naming Convention**  1) 코드를 짤 때 다른 개발자와 협업을 하기 때문에 변수명 정하는 것은 매우 중요하다.  2) JAVA에서 변수를 짓는 방법  🡪 변수의 의미와 목적이 분명히 나타나야한다.  🡪 비슷한 이름은 사용하지 않는다.  🡪 단축어 사용을 지양한다. Ex) int scope (o), int scpe(x)  🡪 숫자로 시작할 수 없다.  🡪 띄어쓰기를 할 수 없다.  🡪 특수 기호를 사용하지 않는다.  🡪 예약어를 사용하지 않는다.  🡪 Camel Case를 지킨다. Ex) int resultOfCalculateTwoNumbers; 첫 글자는 소문자로 시작하고 다음 단어 첫 단어를 대문자로 작성한다.  🡪 대소문자를 구분한다.  **05. 할당**  1) ex) int price = 1500; -> int형 price 변수에 1500이라는 값을 할당한다. ‘=’은 할당 연산자이다.  2) 상수  🡪 final 키워드를 사용하고 한번 초기한 후에는 값을 변경할 수 없다. 중요한 데이터를 고정값으로 할 때 사용한다.  🡪 Ex) final int price = 1500;  Price = 2000; -> 에러 발생  **06. Scope of variable**  1) 멤버변수 : 클래스 아래에 선언되어 클래스 안에서 자유롭게 사용할 수 있는 변수  2) 지역변수 : 클래스가 아닌 곳 중괄호 안에 선언되어 특정 영역에서만 사용할 수 있는 변수  3) Ex)  C:\Users\Admin\Desktop\2.PNG  🡪 현재 코드에서 zipCode 변수는 클래스 중괄호 범위까지이기 때문에 어느 위치에서든 선언이 가능하지만 age변수는 클래스 중괄호 아닌 곳에 선언되어 있어 지역변수에 속하고 지역변수 안에서 끝난다.  🡪 즉, 범위의 위치에 따라 멤버 변수인지 지역변수인지 구별된다.  🡪 참고로 매개변수의 args는 지역변수이다.  **07. Arithmetic Operation & Scanner**  1) 입력한 값을 덧셈 연산하여 출력값 구하기 예제  C:\Users\Admin\Desktop\3.PNG |
|  |

|  |  |
| --- | --- |
| 특이사항 |  |