

01

01-3. 기초 규칙 및 코드 컨벤션

Java Overview

자바 프로그램 기초 규칙 - 파일과 클래스

확장자가 java 인 소스 파일로 작성되어야 한다.

- 자바 소스 파일의 확장자는 java 입니다.
- User.java Order.java Hello.java 등

하나의 소스 파일에는 여러 개의 클래스가 선언될 수 있다.

- 컴파일되면 모든 클래스가 독립 class 파일로 만들어진다.

하나의 소스 파일에는 public 으로 선언된 클래스는 하나만 가능하며 public 으로 선언된 클래스는 파일명과 대소문자가 동일한 이름으로 선언되어야 한다.

동일 패키지내에 이름이 중복된 여러 클래스가 선언될 수는 없다.

자바 프로그램 기초 규칙 - 자바 파일 구성

자바 파일 내에는 `package`, `import`, 클래스가 선언될 수 있다.

- `package` 가 선언된다면 파일의 가장 첫줄에 한번만 선언 가능하다.
- `package` 는 자바 파일의 물리적인 디렉토리 위치와 동일해야 한다.
- `import` 는 여러줄 선언 가능하다.
- 동일 패키지의 다른 클래스는 `import` 없이 사용이 가능하다.
- 다른 패키지의 클래스를 이용하려면 꼭 `import` 해야 한다.
- `package`, `import` 는 컴파일러를 위한 정보이며 컴파일 되면 소멸된다.

자바 프로그램 기초 규칙 - 자바 파일 구성

자바 파일 내에는 `package`, `import`, 클래스가 선언될 수 있다.

- Top Level 구성요소는 클래스만 가능하다.
- 자바 코드에는 클래스, 변수, 함수가 선언될 수 있으며 변수, 함수는 클래스 내부에 선언되어야 한다.

자바 프로그램 기초 규칙 - main() 함수

JVM 에 의해 실행되는 함수는 main() 함수이다.

- public 으로 선언된 클래스내에 main() 함수를 선언할 수 있으며 main() 함수를 가지는 클래스를 JVM 으로 구동이 가능하다.
- public static void main(String[] args) { }
- System.out.println() 은 JVM 콘솔 출력이다.
- args 는 JVM 으로 실행시키면서 대입시키는 값이다.

```
public class Exam3 {  
    public static void main(String[] args) {  
        for(String arg : args){  
            System.out.println(arg);  
        }  
    }  
}
```

>java ch1.section3.Exam3 hello world

자바 프로그램 기초 규칙 - 주석

주석은 컴파일시 사라지는 코드를 작성하는 사람을 위한 정보이다.

3가지 타입의 주석을 제공한다.

- 한줄 주석 : //
- 여러줄 주석 : /* */
- Doc 주석 : /** */

자바 프로그램 기초 규칙 - 코드

대소분자를 구분한다.

문장의 끝을 ; 으로 명시적으로 선언해야 한다.

자바 프로그램 코드 컨벤션

- 코드 컨벤션(Code Convention)이란 프로그래밍 코드를 작성하는 일종의 가이드입니다.
- 코드를 작성할 때 읽고 관리하기 쉽게 작성하기 위한 일종의 관습이라고 보시면 됩니다.

하나의 파일에 하나의 클래스만 선언되는 것이 바람직하다.

들여쓰기를 해야 한다.

- 코드 컨벤션에서 가장 기본적인 것은 들여쓰기입니다.
- 어떤 실행영역을 묶기 위해서 중괄호({ })을 사용하는데 이 중괄호 부분을 중괄호 밖의 코드보다 들여쓰기를 해서 한눈에 중괄호로 묶인 부분이 어디까지인지를 쉽게 식별할 수 있게 해주어야 합니다.

자바 프로그램 코드 컨벤션

식별자에 대소문자를 구분해서 사용한다.

- 코드를 작성하면서 변수명, 함수명, 클래스명 등 개발자가 직접 식별을 위해 이름을 지정하는 경우가 있습니다.
- 이때 대소문자를 구분해서 지정해 주어야 합니다.
- 변수명은 소문자로 시작하는 명사형을 이용한다.
- 함수명은 소문자로 시작하는 동사형을 이용한다.
- 클래스명은 대문자로 시작하는 명사형을 이용한다.
- 두 단어를 결합시켜서 이름을 주는 경우에는 뒷 단어의 첫 글자를 대문자로 지정한다.

자바 프로그램 코드 컨벤션

식별자에 대소문자를 구분해서 사용한다.

- 뒤 단어를 대문자로 해서 식별자를 지정하는 것을 camel case 라고 합니다.
- 두 단어를 언더스코어(_) 로 연결해서 지정할 수도 있는데 이를 snake case 라고 합니다.
- 자바 포함 대부분의 소프트웨어 언어에서는 대소문자를 구분하며 이렇게 대소문자를 구분하는 언어에서는 관습상 camel case 를 주로 사용합니다.
- 만약 어떤 소프트웨어 언어가 대소문자를 구분하지 않는 경우에는 snake case 를 관습상 사용합니다.

자바 프로그램 코드 컨벤션

주석을 열심히 추가하자.

- 코드에 주석을 추가하지 않아도 애플리케이션은 작성되고 잘 실행됩니다.
- 하지만 적절한 주석을 추가하지 않은 코드는 나중에 그 코드가 왜 무엇을 위해 작성되었는지를 간파하기 어려워 전체 코드를 해석하기 어려워 집니다.

스페이스를 적절히 사용하여 보기 좋게 작성

- 스페이스를 추가하지 않았다고 에러가 발생하지는 않지만 코드 해석의 용이성을 위해 적절하게 스페이스를 추가하는 것이 좋습니다.

자바 프로그램 코드 컨벤션

의미있는 식별자를 사용하자.

- 식별자는 변수명, 함수명, 클래스명등을 지칭합니다.
- 이름만 보고도 이 변수가 어떤 데이터가 저장되는 변수인지, 이 함수가 어떤 행위를 하는 함수인지를 간파할 수 있게 지정하는 것이 좋습니다.

디버깅

디버깅도 실력이다.

- 우리가 작성한 애플리케이션은 에러가 발생할 수 있습니다.
- 에러는 크게 컴파일 에러와 런타임 에러로 구분할 수 있습니다.
- 컴파일 에러는 소프트웨어 언어의 문법에 맞게 작성하지 않은 에러이며 런타임 에러는 문법에는 맞게 작성했지만 실행하다가 발생한 에러입니다.

디버깅

디버깅도 실력이다.

- 컴파일 에러는 문법에 맞지 않게 작성한 에러임으로 대부분의 IDE 에서 무엇이 잘못 된 것인지를 알려줍니다.
- 런타임 에러는 코드가 문법에 맞게 작성되었지만 실행할 수 없는 상태입니다.
- 아무리 코드를 잘 작성하더라도 런타임 에러는 발생할 수 있습니다.
- 런타임 에러도 애플리케이션을 실행시키는 플랫폼에 에러 메시지가 나옵니다.



감사합니다

단단히 마음먹고 떠난 사람은
산꼭대기에 도착할 수 있다.
산은 올라가는 사람에게만 정복된다.



윌리엄 셰익스피어
William Shakespeare