



자바학개론

자바학개론

● 개발환경 구축

- JDK 설치
 - 자바 개발을 위한 개발 도구
 - oracle JDK, open JDK, zulu JDK 등
 - 설치 : www.oracle.com에서 Java SE의 1.8 버전 다운로드 및 설치
- 환경 변수 설정
 - javac.exe, java.exe는 편의상 어느 경로에서든 실행 가능해야 하며, 이를 위해 환경 변수 설정 진행
 - 시스템 속성 -> 환경 변수 -> 사용자변수에서 새로 만들기
 - 변수 이름 : Path, 변수 값 : "JDK 설치 경로/JDK1.8.x/bin"
 - 명령 프롬프트에서 "java", "javac"를 각각 입력하여 설치 확인
- IDE (IntelliJ)
 - <https://www.jetbrains.com/> 에서 IntelliJ Community 2018.3.6 버전 다운로드 및 설치
 - 인코딩 설정 : Editor -> File Encodings에서 UTF-8로 변경, Transparent ~ 체크
 - 폰트사이즈 : Editor -> Font에서 size는 14, Line spacing은 1.2 추천

자바학개론

- 첫 번째 자바 프로그램의 작성과 실행

- 수업 폴더 생성 : "C:\JavaStudy\오늘날짜"
- 메모장에서 아래 코드를 입력 후 "C:\JavaStudy\오늘날짜" 경로에 "HelloWorld.java"로 파일 저장

```
class HelloWorld {  
    public static void main(String[] args) {  
        System.out.println("Hello Java World!");  
        System.out.println("This is my first java program!");  
    }  
}
```

- 명령 프롬프트에서 "cd C:\JavaStudy\오늘날짜"를 입력하여 폴더 이동
- "javac HelloWorld.java" 입력
- "dir" 입력하여 class 파일 생성 확인
- "java HelloWorld" 입력하여 실행
- 실행 결과 확인

자바학개론

- 자바 프로그램의 실행 구조와 JVM

- 일반적인 프로그램 : 운영체제 위에서 실행되므로 프로그램은 운영체제에 의존
- java 프로그램 : 운영체제 위에 운영체제에 맞는 java virtual machine (JVM)이 존재하고 그 위에서 실행
- 하나의 코드로 OS에 관계 없이 프로그램 실행 가능

- compiler : 자연언어로 사용자가 작성한 java 소스 코드를 컴퓨터가 이해할 수 있는 언어로 번역

- 첫 번째 java 프로그램 관찰 1

```
class HelloWorld {  
    public static void main(String[] args) {  
        System.out.println("Hello Java World!");  
        System.out.println("This is my first java program!");  
    }  
}
```

HelloWorld : 클래스 이름

main : 메소드 이름

- 파일명과 클래스명은 동일
- java의 소스 코드는 클래스 내 작성
- java는 중괄호({ })를 이용하여 영역 구분
- 문장의 끝에는 세미콜론(;)으로 문장의 끝 표현

자바학개론

● 첫 번째 자바 프로그램 관찰 2

- 프로그램 실행 시 main 메소드 안의 코드를 한 줄씩 순차적으로 실행
- System.out.println의 소괄호 안에 출력하고 싶은 내용을 큰 따옴표로 감싸서 넣으면 내용 출력
- System.out.println은 출력 이후 행 바꿈

● System.out.println 활용

```
class SystemOutPrintln {  
    public static void main(String[] args) {  
        System.out.println(7);  
        System.out.println(3.15);  
        System.out.println("3 + 5 = " + 8);  
        System.out.println(3.14 + "는 실수입니다.");  
        System.out.println("3 + 5" + "의 연산 결과는 8입니다.");  
        System.out.println(3 + 5);  
    }  
}
```

자바학개론

1. 다음 두 문장을 출력 결과를 확인하는 프로그램을 작성해 보시오.

- `System.out.println("2 + 5 = " + 2 + 5)`
- `System.out.println("2 + 5 = " + (2 + 5))`

2. 숫자 12를 총 5회 출력하는 프로그램을 작성해 보시오. 단 출력 방법을 모두 다르게 작성하여 출력해야 합니다.

자바학개론

● 주석

- 컴파일의 대상에서 제외되는 문장
- 코드에 대한 기록을 메모의 형태로 남겨둘 수 있음
- /* ~ */ : 범위 주석
- // : 한 줄 주석

```
class Comment {  
    public static void main(String[] args) {  
        // 한 줄 주석 : 다음은 단순한 정수의 출력  
        System.out.println(7);  
  
        /*  
        범위 주석  
        다음은 단순한 실수의 출력  
        */  
        System.out.println(3.15);  
    }  
}
```