

jlink로 응용프로그램 맞춤형 실행 환경 만들기

A. jlink로 응용프로그램 맞춤형 실행 환경 만들기

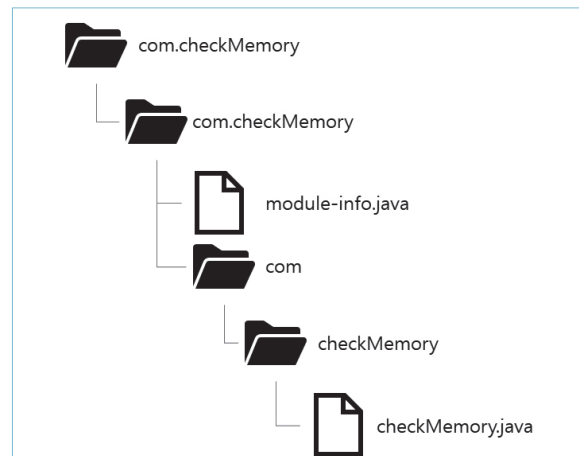


예제 개요

Java 9부터는 JDK에 jlink라는 도구가 포함되어 있다. 이 도구를 이용하면 개발자가 개발한 응용프로그램의 실행에 최적화된 커스텀 런타임 환경(custom JRE)을 생성할 수 있다.

jlink 사용법에 대해 알아보자. 설명을 위해 다음 [그림 1]과 같은 디렉터리 구조를 갖는 예제를 만들어본다. 예제는 매우 간단하지만, 처음 보면 무슨 말인지 적응이 안 될 것이다. 일단 계속 읽어보자.

이 예제는 Java 9의 JDK 9가 설치된 환경에서 만들 사례이다. 만일 사용자가 JDK 10이나 그 이상의 버전을 설치하고 실행하는 경우 경로명을 바꾸어야 한다. 이것은 다시 아래 본문에서 해당하는 부분이 나올 때 언급하겠다.



[그림 1] 예제 디렉터리 구조

module-info.java의 내용은 다음과 같다.

```

module com.checkMemory {
    exports com.checkMemory;
}
  
```

예제 프로그램인 checkMemory.java의 내용은 다음과 같다.

```
package com.checkMemory;

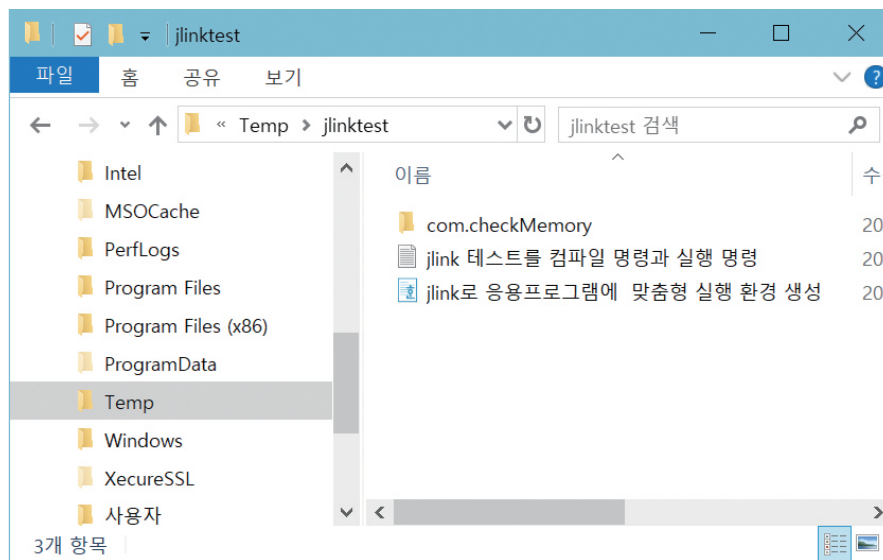
public class checkMemory {
    public static void main(String[] args) {
        System.out.println("KB: " + (double) (Runtime.getRuntime().
            totalMemory() - Runtime.getRuntime().freeMemory()) / 1024);
    }
}
```

예제는 매우 간단한 구조이다. 소스라고 해 봤자, 이 두 파일뿐이다.

이 예제의 목적은, 이 예제를 실행시켜 custom JRE를 만들고, custom JRE의 구조가 어떠한지, 자바 JRE에 배포되는 기본 modules 파일(109MB)에 비해 얼마나 modules 파일이 작아지는지 관찰하는데 있다.

예제 코드 풀고 컴파일하기

우선 독자들은 zip으로 제공되는 예제를 적당한 디렉터리에 풀도록 하라. 저자는 [그림 2]와 같이 C:\Temp\jlinktest 밑에 풀었다.



[그림 2] C:\Temp\jlinktest 밑에 예제 풀기

그리고 윈도우에서 cmd 명령창을 실행하고 그 폴더로 이동하였다.

```

C:\Windows\system32\cmd.exe
C:\Temp\jlinktest>
C:\Temp\jlinktest>
  
```

[그림 3] 윈도우에서 cmd 명령으로 창을 출력하고, C:\Temp\jlinktest으로 이동

이제, 현재 디렉터리에서 다음 명령으로 예제를 컴파일한다.

```
javac --module-source-path . -d output -m com.checkMemory
```

컴파일된 클래스는 [그림 4]와 같이 output 디렉터리 밑에 생기는데 [그림 1]의 구조로 클래스 파일이 생성된다.

```

C:\Windows\system32\cmd.exe
C:\Temp\jlinktest>javac --module-source-path . -d output -m com.checkMemory
C:\Temp\jlinktest>dir
C 드라이브의 볼륨: BOOTCAMP
볼륨 일련 번호: 0246-9FF7

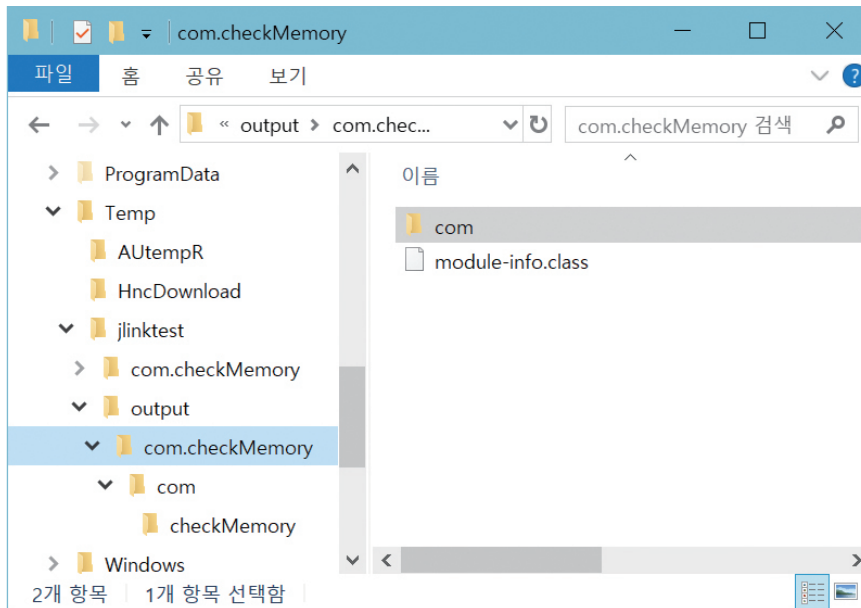
C:\Temp\jlinktest 디렉터리

2018-04-26 오후 05:01 <DIR>      .
2018-04-26 오후 05:01 <DIR>      ..
2018-04-26 오후 04:32 <DIR>      com.checkMemory
                305  jlink 테스트를 컴파일 명령과 실행 명령.txt
2018-04-14 오후 01:53                38,912 jlink로 응용프로그램에 맞춤형 실행 환경 생성.hwp
2018-04-26 오후 04:21                output
2018-04-26 오후 05:01 <DIR>      output
                2개 파일                39,217 바이트
                4개 디렉터리 288,094,822,400 바이트 남음

C:\Temp\jlinktest>
  
```

[그림 4] 윈도우에서 cmd 명령으로 창을 출력하고, C:\Temp\jlinktest으로 이동

[그림 5]는 output 디렉터리에 만들어진 컴파일된 결과를 보여준다.



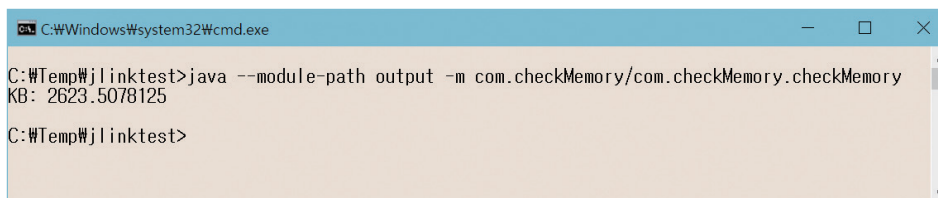
[그림 5] output 디렉터리에 컴파일된 결과들

모듈 실행

작성된 모듈을 실행해보자. cmd로 생성한 명령창으로 가서 다음 명령을 입력하여 실행해보자.

```
java --module-path output -m com.checkMemory/com.checkMemory.checkMemory
```

이 명령을 입력하여 실행하는 모양은 [그림 6]과 같다. 이 예제는 자신이 실행되는데 필요한 메모리의 양을 출력한다. 현재는 Java 9의 실행 환경(Java SE 9, JRE)에서 이 예제를 실행하고 있으며, 출력된 결과를 보면 예제가 실행되는데 약 2623KB(약 2.6MB)의 메모리가 소모됨을 보여준다.



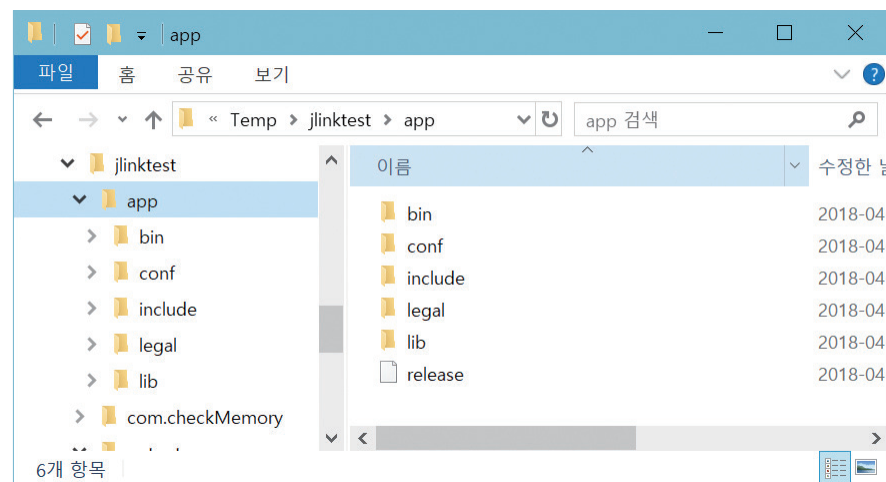
[그림 6] 모듈로 만들어진 예제를 Java SE의 자바 실행 환경(JRE)에서 실행

jlink로 커스텀 실행 환경(custom JRE) 만들기

그러면, 이제 jlink 도구를 이용하여 이 예제 응용프로그램에 최적화된 커스텀 실행 환경을 만들어 보자. 다음 명령을 실행하면 [그림 7]과 같이 app 디렉터리 밑에 최적화된 커스텀 실행 환경이 생성된다.

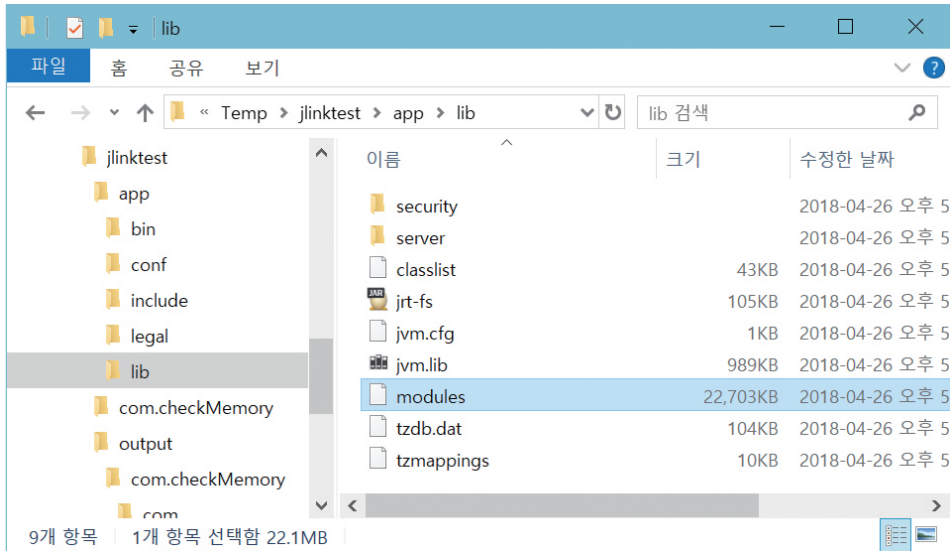
```
jlink --module-path "c:\Program Files\Java\jdk-9\jmods";output --add-modules com.checkMemory --output app
```

참고로 이 명령은 JDK 9를 설치한 상황에서 실행되는 명령이므로 사용자가 JDK 10이나 그 이상 버전을 설치한 경우라면 앞의 명령 부분에서 'jdk-9' 대신 JDK가 설치된 디렉터리를 지정하도록 하라.



[그림 7] app 밑에 만들어진 커스텀 실행 환경(custom JRE)

app\lib 디렉터리 밑의 modules라는 파일의 크기(22MB)에 주목하기 바란다. 이 파일은 예제를 실행하기 위해 최적화된 모듈 파일이다. 기존 JRE의 lib 디렉터리 밑의 modules 파일의 크기(109MB)에 비하여 매우 작은 것을 알 수 있다.



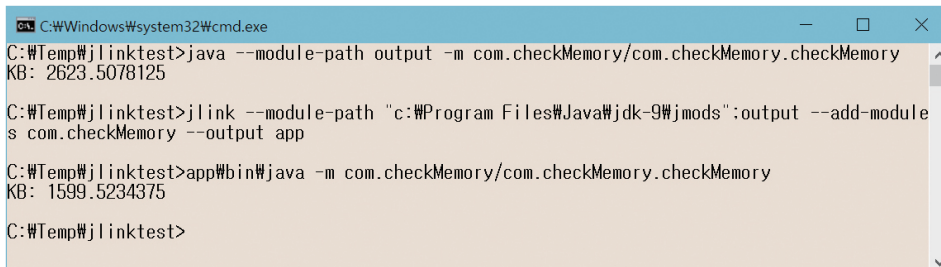
[그림 8] app\lib 밑의 modules 파일의 크기는 약 22MB

커스텀 실행 환경에서 실행

app 디렉터리 밑에 커스텀 실행 환경이 생성되었으므로 이를 이용하여 프로그램을 실행한다. 다음 명령을 입력하여 실행해보자.

```
app\bin\java -m com.checkMemory/com.checkMemory.checkMemory
```

예제를 실행하면 [그림 9]와 같이 출력된다. 커스텀 실행 환경에서 자바는 이 예제를 실행하기 위해 약 1599KB(1.6MB)의 메모리를 사용했다. 이것은 [그림 6]에서 보여준 바와 같이 자바의 기본 JRE에서 소모된 2.6MB의 메모리에 비하면 약 $(1.6/2.6) \times 100\% = 62\%$ 정도밖에 메모리가 소모되지 않는 것으로, 모듈 기반의 커스텀 실행 환경을 사용하면 약 40%가 작은 메모리 환경에서도 실행될 수 있음을 보여준다.



[그림 9] 커스텀 실행 환경에서 예제 실행

런처를 생성하여 실행

jlink 도구를 이용하면 응용프로그램을 실행시키는 런처를 생성할 수도 있다. 다음의 명령은 이름이 launch인 런처를 생성한다.

```
jlink --module-path "c:\Program Files\Java\jdk-9\jmods";output --add-modules com.checkMemory --output app --launcher launch=com.checkMemory/com.checkMemory.checkMemory
```

참고로 이 명령은 JDK 9를 설치한 상황에서 실행되는 명령이므로 사용자가 JDK 10이나 그 이상 버전을 설치한 경우라면 앞의 명령 부분에서 'jdk-9' 대신 JDK가 설치된 디렉터리를 지정하도록 하라.

런처가 생성되었으므로 런처를 이용하여 다음과 같이 응용 프로그램을 실행시킨다.

```
app\bin\launch
```

[그림 10]은 jlink를 이용하여 launch를 만들고 launch를 실행시키는 과정이다.

```
C:\Windows\system32\cmd.exe
C:\Temp\jlinktest>jlink --module-path "c:\Program Files\Java\jdk-9\jmods";output --add-modules com.checkMemory --output app
C:\Temp\jlinktest>app\bin\java -m com.checkMemory/com.checkMemory.checkMemory
KB: 1599.5234375
C:\Temp\jlinktest>jlink --module-path "c:\Program Files\Java\jdk-9\jmods";output --add-modules com.checkMemory --output app --launcher launch=com.checkMemory/com.checkMemory.checkMemory
Error: directory already exists: app
C:\Temp\jlinktest>jlink --module-path "c:\Program Files\Java\jdk-9\jmods";output --add-modules com.checkMemory --output app --launcher launch=com.checkMemory/com.checkMemory.checkMemory
C:\Temp\jlinktest>app\bin\launch
KB: 1313.484375
C:\Temp\jlinktest>
```

[그림 10] jlink를 이용하여 launch를 만들고, 실행