

모듈

514770
2019년 가을학기
11/5/2019
박경신

모듈

□ Java 9 Module System

- Project Jigsaw
 - Modular JDK
 - Modular Java Source Code
 - Modular Run-time Images
 - Encapsulate Java Internal APIs
 - Java Platform Module System
- 편하고 효율적인 Java 개발 환경을 만들기 위해서 시작
- Jar 기반 모노리틱 방식을 개선하여 모듈 지정 및 모듈별 버전 관리 기능 가능
- 필요한 모듈만 구동하여 크기와 성능 최적화가 가능
- 임베디드 환경에서 필요한 모듈만 탑재되어 적은 메모리로 로딩 가능

모듈

□ Java 9 모듈

- 패키지들을 묶어서 추상화시켜 관리할 수 있는 방법
 - 패키지는 관련 있는 클래스들을 묶어서 관리할 수 있도록 했고, 그 안에서 private, protected, public 등의 가시성을 부여
 - 패키지 내부 또는 외부에서 사용할 수 있는 클래스를 구별
 - 추상화와 캡슐화 때문
- 패키지의 특성이 오히려 추상화나 캡슐화를 해치는 경우가 발생함
 - 예: sun.misc.BASE64Encoder나 sun.misc.BASE64Decoder 클래스
- 모듈은 내부적으로 사용하려고 만든 패키지들의 public 클래스들이 모두에게 공개되는 것을 막을 수 있음

모듈

□ 기존 패키지 대비 모듈의 장점

- public 클래스를 숨길 수 있으므로 강한 캡슐화 지원
- 모듈 단위로 추상화와 캡슐화를 지원함으로써 패키지 간의 관계를 모듈 단위로 정리할 수 있어 복잡한 의존성을 줄일 수 있음
- 모듈은 컴파일된 자바 코드 외에 이미지 파일이나 XML 파일 같은 다른 리소스 파일들을 포함시킬 수 있음
- 응용 프로그램을 모듈에 넣는 것 보다는 자주 사용될 수 있는 라이브러리를 구축할 때 더 유용함

module-info.java

- 모듈은 `module-info.java` 파일 안에 아래 3가지를 선언하는 소프트웨어적인 단위



```
module com.mycompany { // 내 모듈 이름
  exports com.mycompany; // com.mycompany 사용할 수 있음
  requires com.yourcompany; // com.yourcompany 사용할 것임
}
```

module-info.java



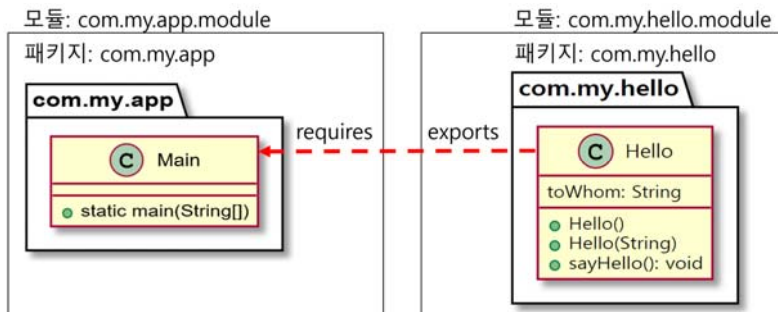
module-info.java

- Module name (모듈 이름)
 - 모듈이름은 충돌을 피하기 위해 패키지 명명 규칙과 유사
- Export (어떤 것을 제공하는가?)
 - 해당 모듈이 다른 외부 모듈에서 사용할 수 있도록 공개 API로 간주되는 모든 패키지 목록을 제공
 - 만약 어떤 클래스가 `public`이라 할지라도 `export`된 패키지에 없으면 모듈 외부의 어떤 것도 이 클래스에 접근 불가능
- Require (어떤 것들이 필요한가?)
 - 해당 모듈과 의존 관계가 있는 다른 모듈 목록을 명시

모듈 구현 방법

모듈 구현 방법

- `module-info.java`를 작성해야 함
- `com.my.app.module`에서는 어떤 모듈을 사용할 지 지정
- `com.my.hello.module`에서는 어떤 패키지를 제공할 것인지 지정



모듈 구현 방법

다른 모듈을 사용하는 쪽

- `com.my.hello.module`을 사용함

```
module com.my.app.module {
  requires com.my.hello.module;
}
```

모듈에서 기능을 제공하는 쪽

- `com.my.hello` 패키지의 `public` 클래스를 사용할 수 있도록 함

```
module com.my.hello.module {
  exports com.my.hello;
}
```

모듈 구현 방법

```
package com.my.hello;

public class HelloModule {
    String toWhom = "World";
    public HelloModule() {
    }
    public HelloModule(String toWhom) {
        this.toWhom = toWhom;
    }
    public void sayHello() {
        System.out.println("Hello " + toWhom);
    }
}
```

모듈 구현 방법

```
package com.my.app;

import com.my.hello.HelloModule;

public class Main {
    public static void main(String[] args) {
        HelloModule hm = new HelloModule("java19-2");
        hm.sayHello();
    }
}
```

모듈 구현 방법

ModuleAppTest->Build Path->Configure Build Path
Project Tab->Modulepath->Add->Select ModuleHelloTest

