

Software
Architecture

Blackboard
Architecture Style

Drools

Eunmi Choi
Kookmin University



Distributed Intelligent
System Lab.

Rule Engine Drools 소개

drools 설치부터
활용예제까지

이시현, 김유정

국민대학교



목차

1. Rule Engine & Drools 소개
2. Drools 설치법
3. Drools 구조
4. Drools 사용예제

Rule Engine & Drools 소개



Rule Engine

- 최소한의 지식으로 결론을 추론 할 수 있는 도구
- 인간의 의사 결정능력을 모방하는 전문가 시스템의 기본 구성 요소
- Rule Engine의 핵심은 추론엔진의 패턴매칭

Drools 소개



오픈소스 룰 엔진

아파치 라이선스에
서
배포된 java 라이브
러리

전후방 추론 기반
규칙관리 시스템

Drools 특징

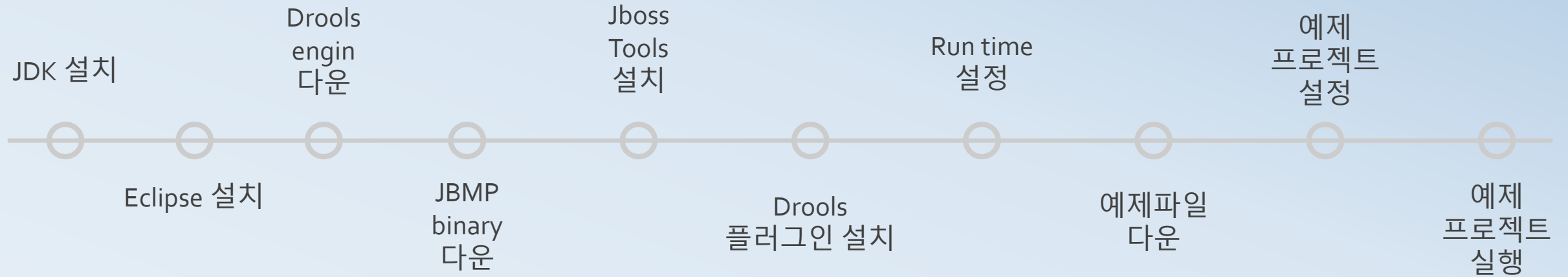
drl

- 자바소스파일과 분리되어 규칙들을 관리하기 때문에 자주 바뀌게 되는 규칙들을 유지 보수하기에 편리하다.
- 모든 규칙이 단일 저장소에 저장되기 때문에 작업흐름을 손쉽게 제어할 수 있다.
- 소스파일과 분리되어 있기 때문에 규칙들이 바뀌어도 다시 빌드하지 않아도 된다.
- 개발자가 아닌 일반 사용자 그룹들이 소스코드에 영향을 주지 않고 규칙을 변경하는 것이 가능하다.
- 복잡한 규칙들이 얹혀있는 Expert System에 활용 할 수 있다.

Drools 설치법



Drools 설치순서



JDK 설치

<https://www.oracle.com/technetwork/java/javase/downloads/index.html>

Java SE 8u201 / Java SE 8u202

Java SE 8u201 / Java SE 8u202 includes important bug fixes. Oracle strongly recommends that all Java SE 8 users upgrade to this release.

[Learn more](#) ➔

- [Installation Instructions](#)
- [Release Notes](#)
- [Oracle License](#)
- [Java SE Licensing Information User Manual](#)
 - Includes Third Party Licenses
- [Certified System Configurations](#)
- [Readme Files](#)
 - [JDK ReadMe](#)
 - [JRE ReadMe](#)

JDK

DOWNLOAD ↕

Server JRE

DOWNLOAD ↕

JRE

DOWNLOAD ↕

원하는 버전의
JDK DOWNLOAD 버튼 클릭



Java Se Development Kit 에서
Accept License Agreement 클
릭



운영체제에 맞는
Java Se Development Kit 설
치

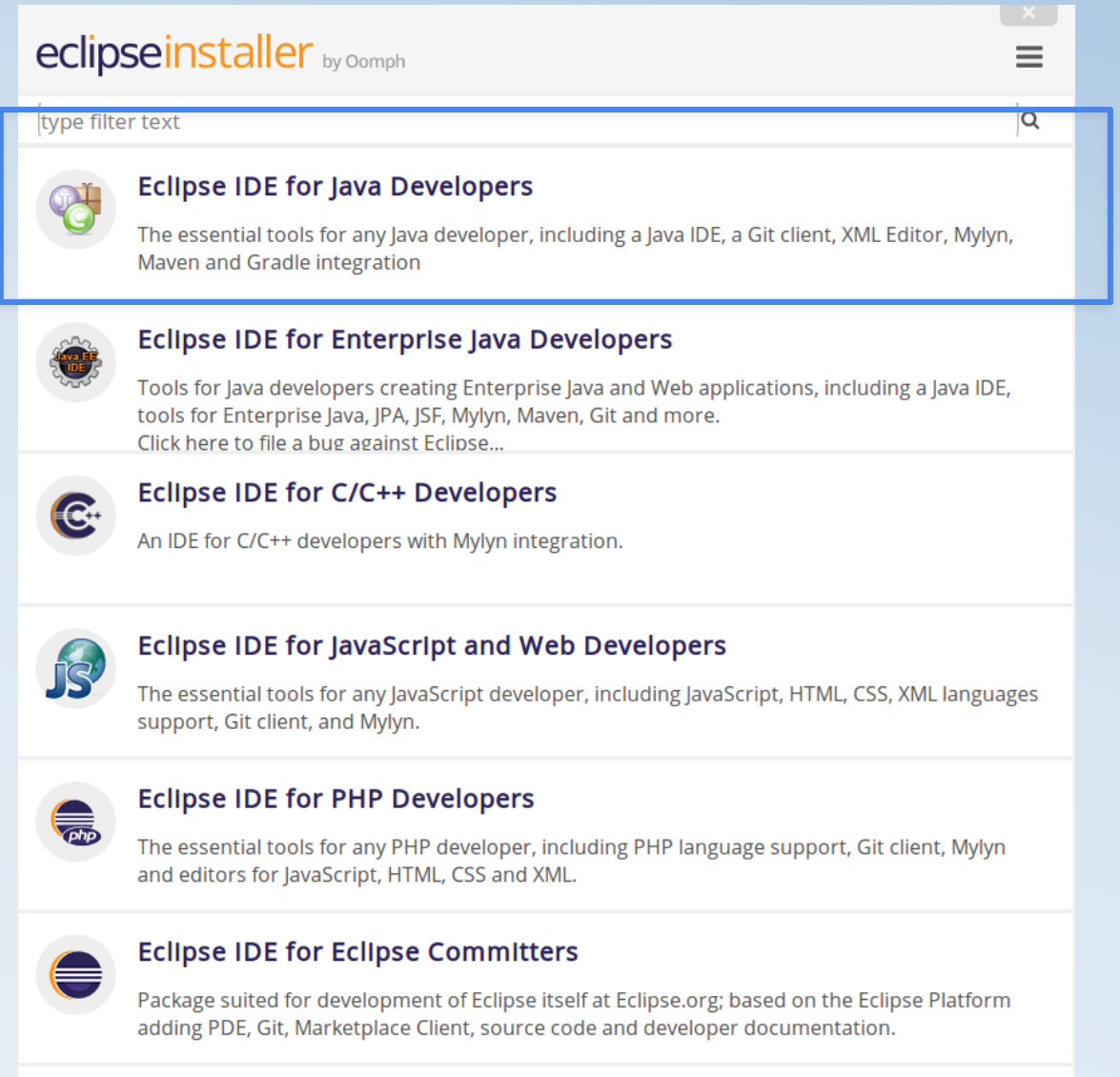
Eclipse 설치

<https://www.eclipse.org/downloads/>

이클립스 공식홈페이지에서
최신버전의 이클립스 다운로드



installer 실행 후
Eclipse IDE for Java Developers 설
치



Drools engine 파일 다운

<https://www.drools.org/download/download.html>

위의 링크 또는
Drools 홈페이지 - Download로
이동



Drools engine Final 버전 다운

Download

Latest final version: 7.16.0.Final

- Release date: 2019-01-14
- [New and Noteworthy in drools 7.16.0.Final](#)
- [Release Notes](#)
- License: [ASL 2.0](#)

Name	Description	Download
Drools Engine	Drools Expert is the rule engine and Drools Fusion does complex event processing (CEP). Distribution zip contains binaries, examples, sources and javadocs.	Distribution ZIP
Drools and jBPM integration	Drools and jBPM integration with third party project like Spring. Distribution zip contains binaries, examples and sources.	Distribution ZIP
Business Central Workbench	Business Central Workbench is the web application and repository to govern Drools and jBPM assets. See documentation for details about installation.	WildFly 14 WAR EAP 7 WAR
Drools and jBPM tools	Eclipse plugins and support for Drools, jBPM and Guvnor functionality. Distribution zip contains binaries and sources.	Distribution ZIP
KIE Execution Server	Standalone execution server that can be used to remotely execute rules using REST, JMS or Java interface. Distribution zip contains WAR files for all supported containers.	Distribution ZIP

JBPM binary파일 다운

<https://www.jbpm.org/download/download.html>

위의 링크 또는
JBPM 공식 홈페이지 - download
로 이동



JBPM X.X.X Final-bin.zip 파일 다운

Latest Final version: 7.16.0.Final

- New and Noteworthy in jBPM 7.16.0.Final
- Release notes
- License: [ASL 2.0](#)
- Release date: 2019-01-14

The jBPM Server distribution is the easiest way to start with jBPM.

-  [Download jBPM 7.16.0.Final server \(single zip\) distribution](#)

Just download, unzip and run

jbpms-server/bin/standalone.sh	(Unix / Linux)
jbpms-server/bin/standalone.bat	(Windows)


It includes:

- WildFly server
- Business Central
- KIE Server
- jBPM Case Management showcase app
- jBPM Service repository (with community work items)

all configured to work smoothly together.

Once download is completed have a look at [Get started guide](#)

The jBPM binaries include documentation, examples and sources.

-  [Download jBPM 7.16.0.Final binaries](#)
 - [jBPM 7.16.0.Final-bin.zip](#)
 - [jBPM 7.16.0.Final-examples.zip](#)
 - [jBPM-installer 7.16.0.Final.zip](#)
 - [jBPM-installer-full 7.16.0.Final.zip](#)
 - [updatesite 7.16.0.Final](#)

binary

!!'

Eclipse JBoss Tools 설치

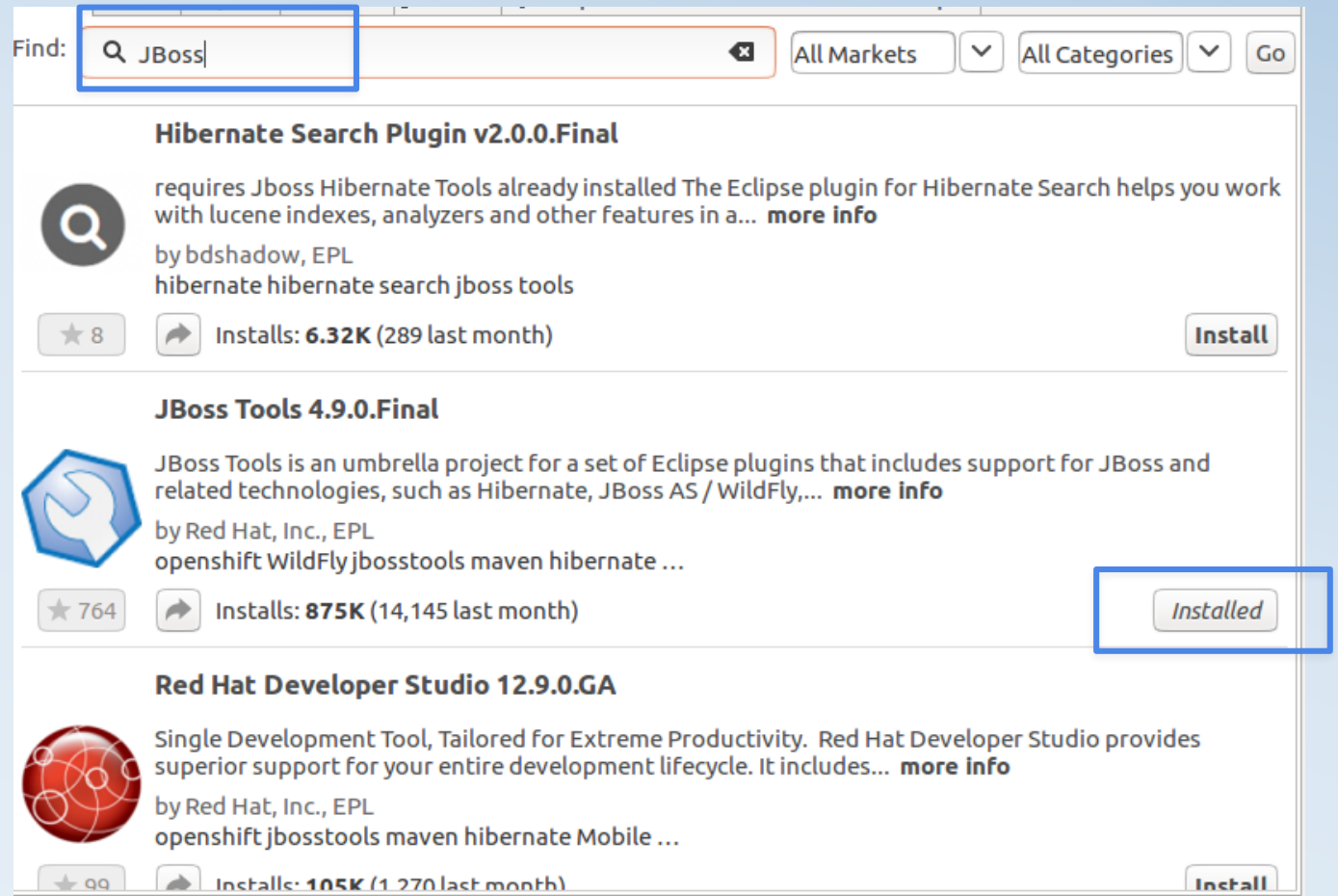
Eclipse 실행 후 상단메뉴의
Help - Eclipse Marketplace 실행



jboss를 검색하여
JBoss Tools x.x.x.Final 설치




Eclipse 재실행



Eclipse drools & JBPM 플러그인 설치

<http://download.jboss.org/drools/release/>



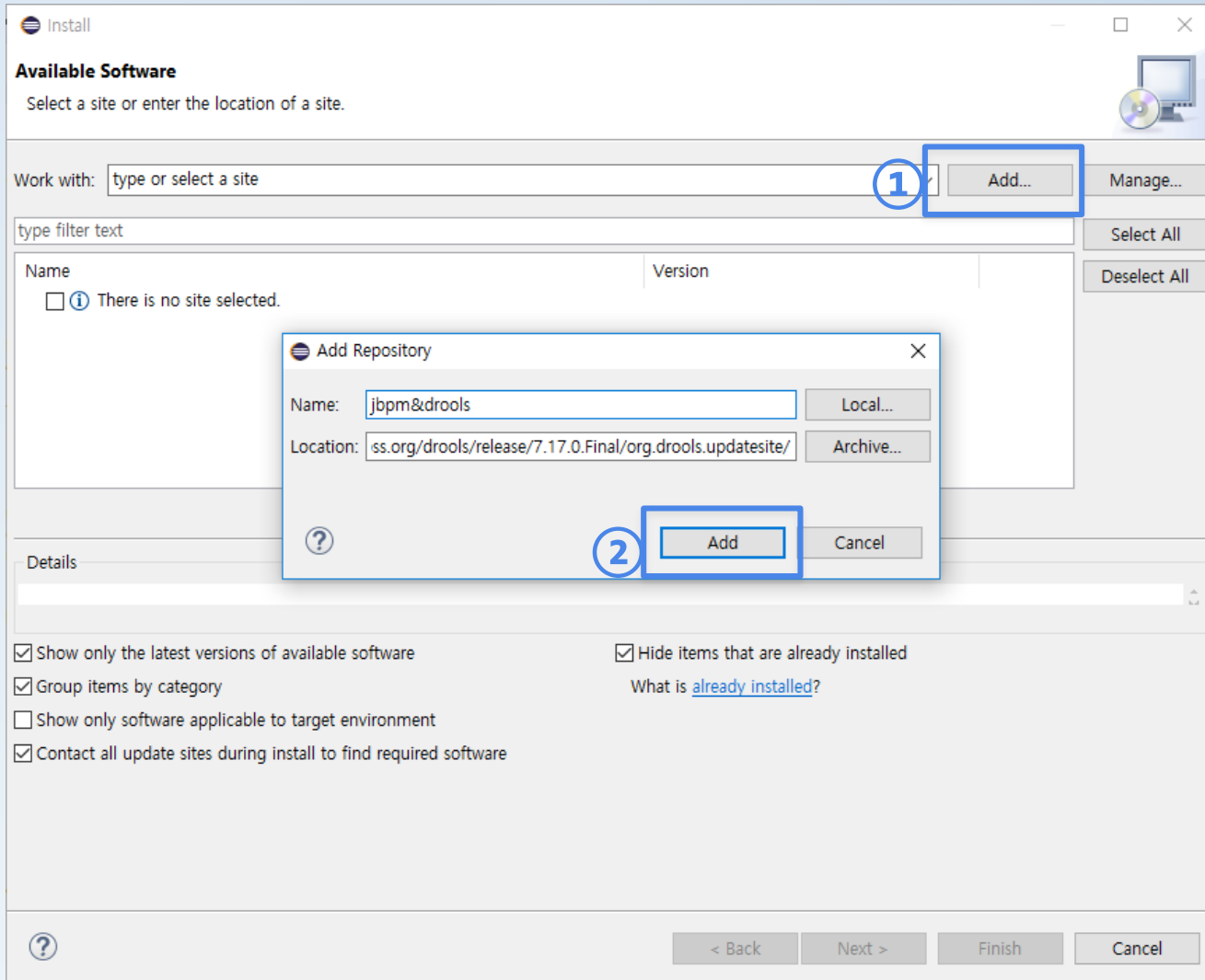
Name	Last modified	Size	Description
Parent Directory	-	-	-
business-central-7.17.0.Final-eap7.war	31-Jan-2019 03:16	175M	
business-central-7.17.0.Final-wildfly14.war	31-Jan-2019 03:23	175M	
drools-distribution-7.17.0.Final.zip	31-Jan-2019 03:05	53M	
droolsjbpm-integration-distribution-7.17.0.Final.zip	31-Jan-2019 03:06	31M	
droolsjbpm-tools-distribution-7.17.0.Final.zip	31-Jan-2019 03:08	54M	
kie-server-distribution-7.17.0.Final.zip	31-Jan-2019 03:31	186M	
org.drools.update.site/	31-Jan-2019 03:01	-	

위의 링크에 접속하여 이전에 다운로드 받은 drools-engine과 동일한 버전 클릭



org.drools.update.site/ 로 접속하여 주소 복사

Eclipse drools & JBPM 플러그인 설치



Eclipse의 상단메뉴에서
Help - Install New Software... 실행

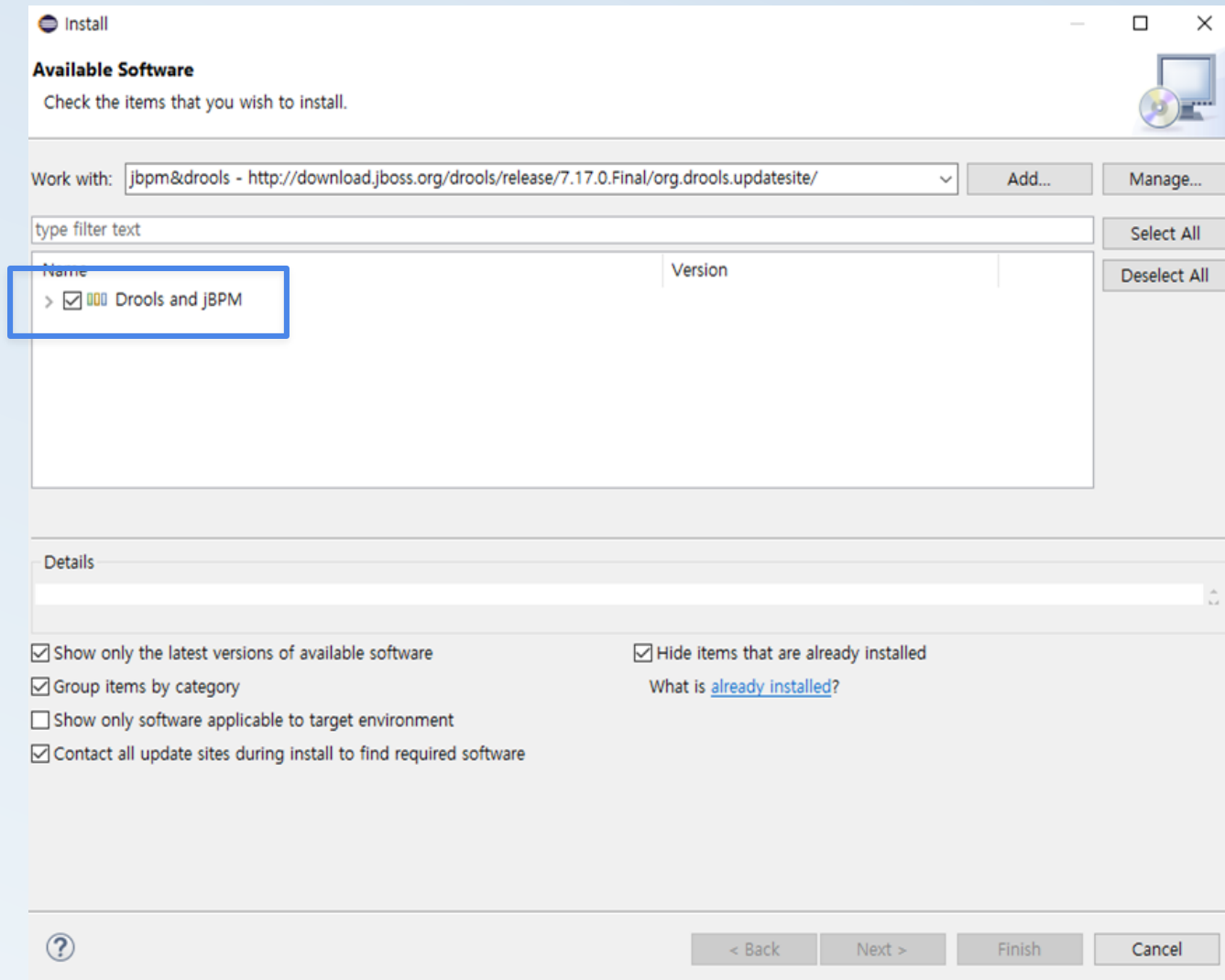


Add버튼 클릭



Name에는 적당한 값을 넣고
Location에 주소를 붙여넣은 후
Add클릭

Eclipse drools & JBPM 플러그인 설치



체크박스 선택 후 설치



설치완료 후 Eclipse 재실행

Drools & JBPM Runtime 설정

Eclipse 상단메뉴의
windows - preferences 실행



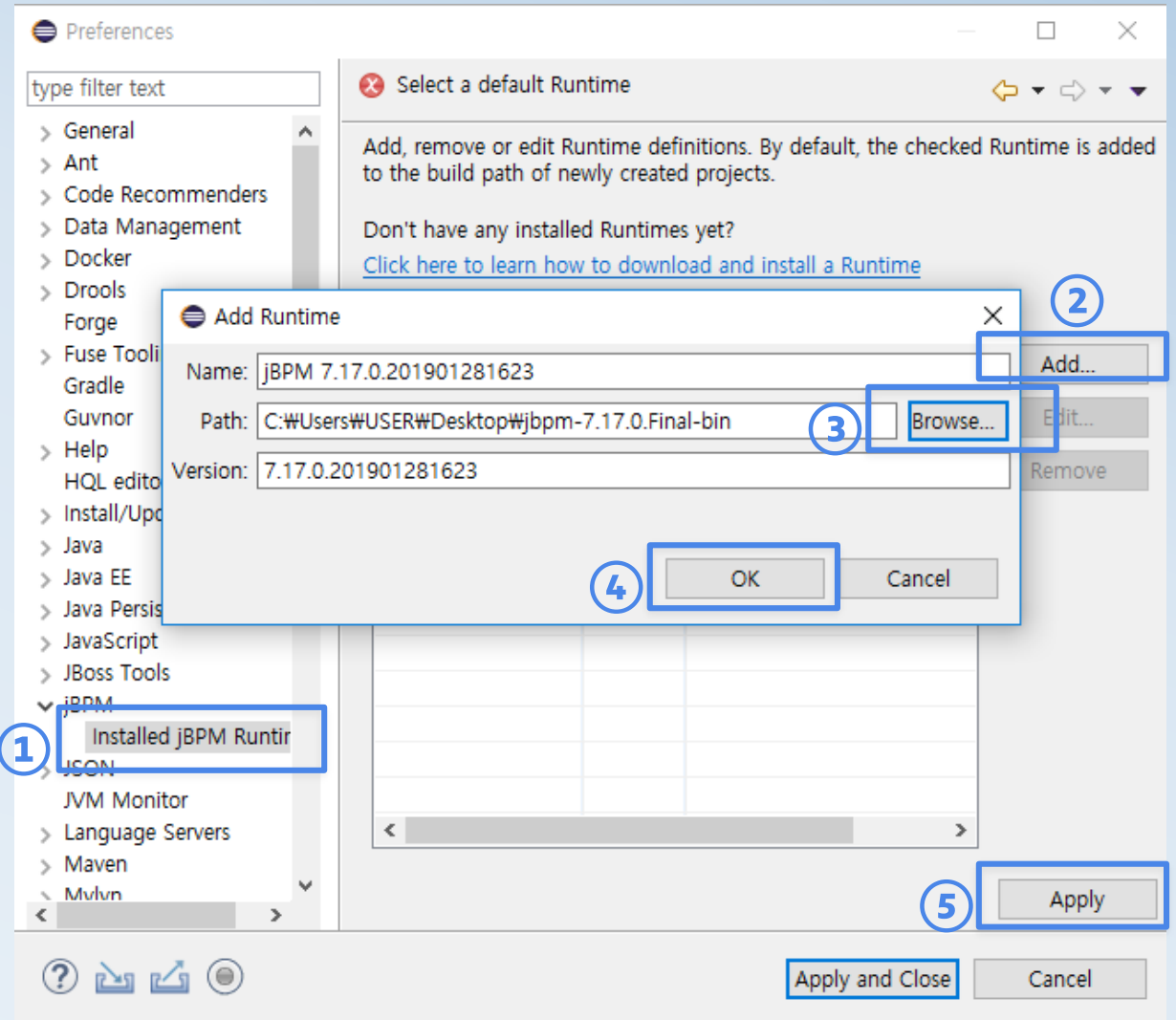
좌측메뉴의
jBPM - Installed jBPM Runtime 선택



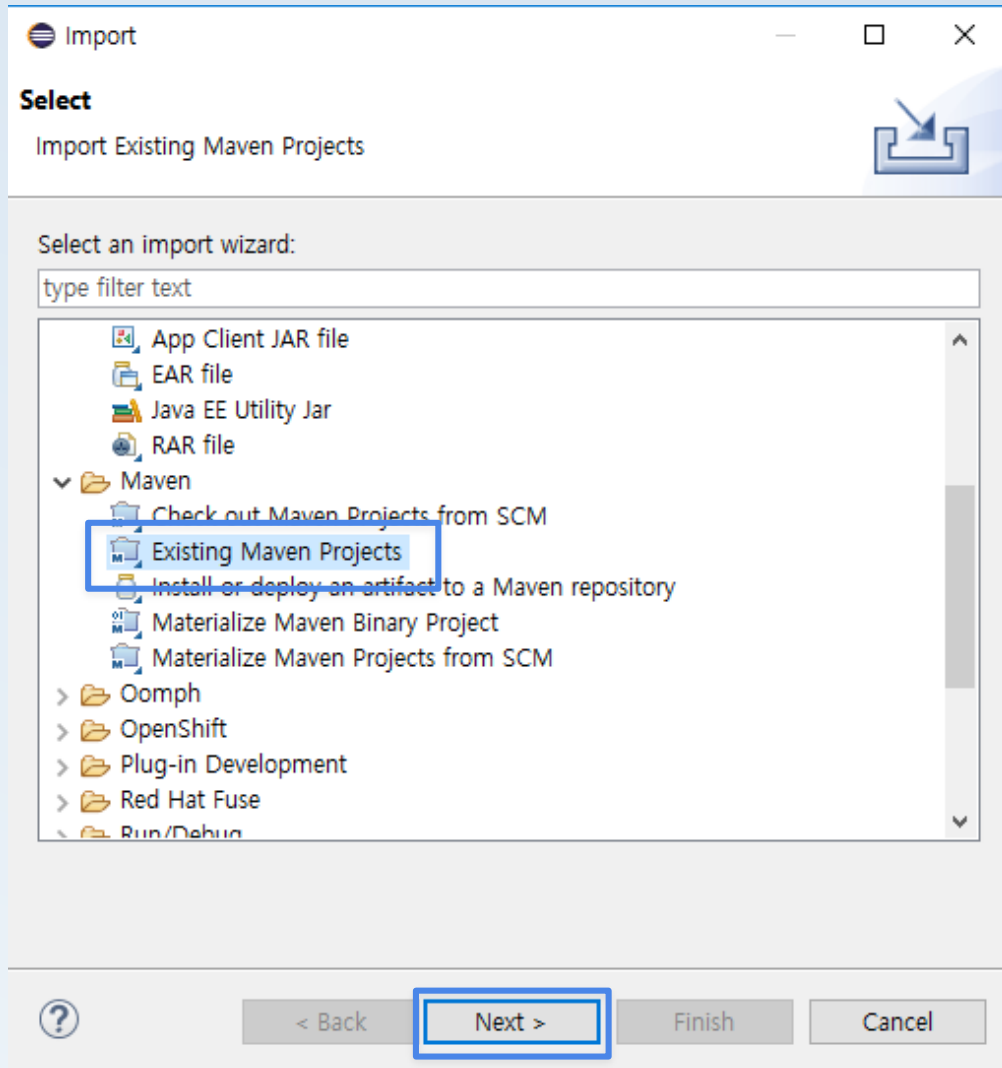
Browse 클릭 후 이전에 다운받은
jBPM 디렉토리 선택 후 ok



동일한 과정으로
Drools의 binaries 디렉토리 선택
후 Eclipse 재실행



예제 파일 다운 및 추가



master.zip

repository in GitHub

위의 파일 또는 drools document 사이트의 example 탭에서 파일을 찾아 다운로드



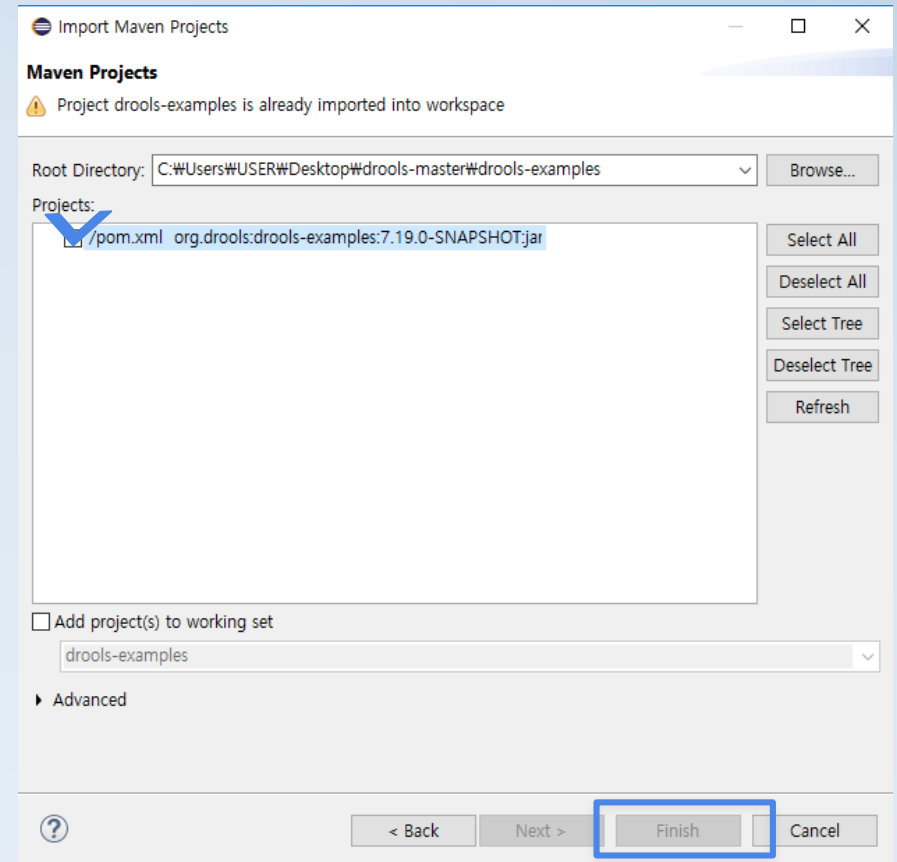
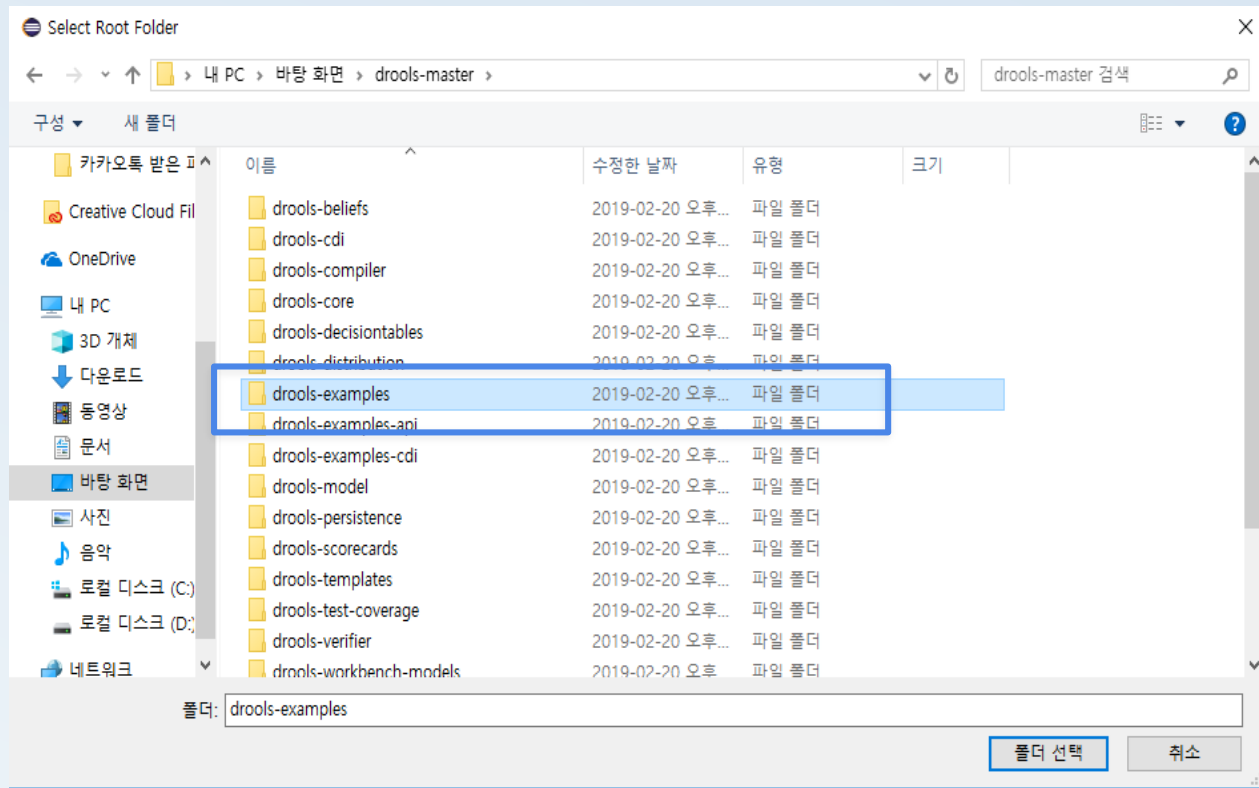
파일 압축해제



Eclipse의 상단메뉴 File에서
Import - Maven - Existing Maven Project
선택

예제 파일 다운 및 추가

Browse 클릭하여 drools-master 디렉토리 안
의
drools-examples 디렉토리 선택 후 Finish 클릭



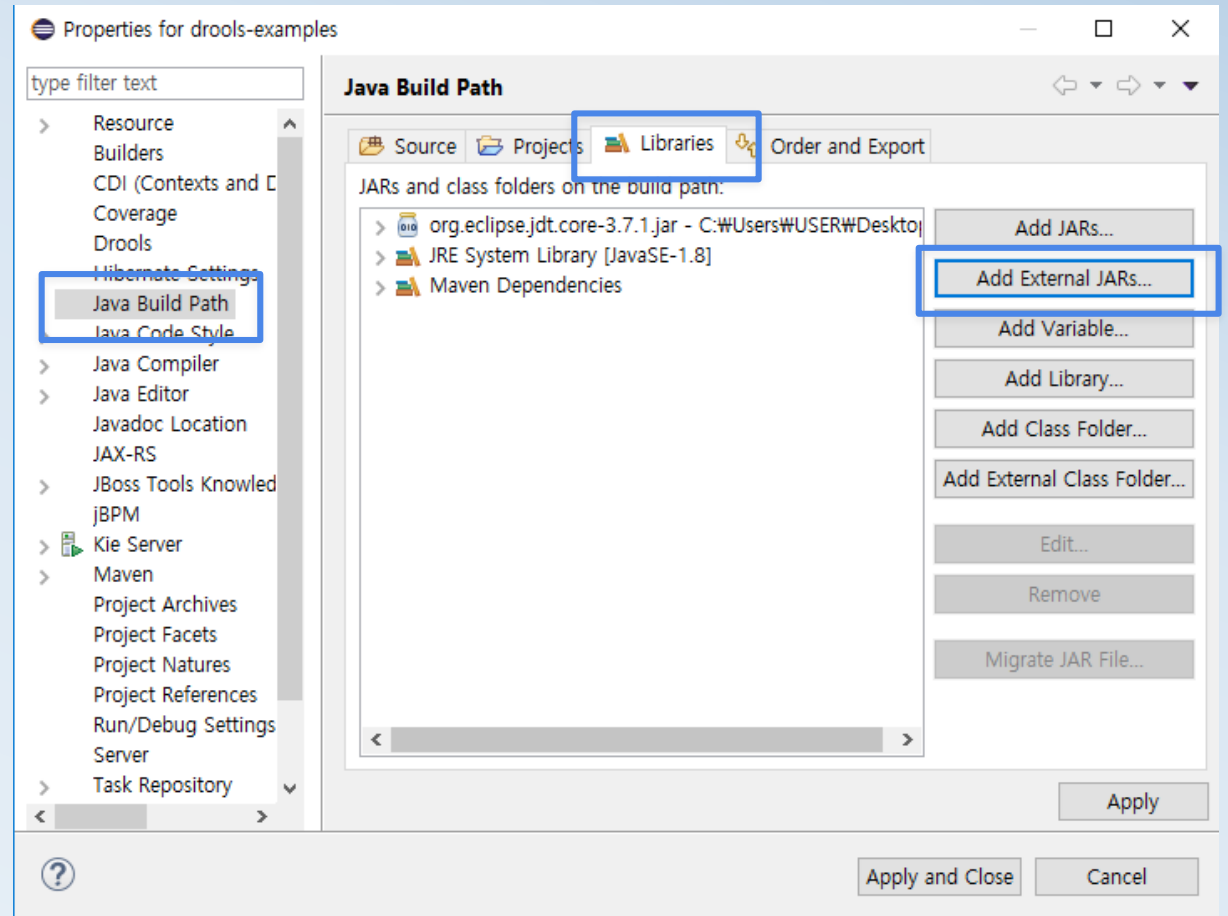
예제 프로젝트 설정

<http://www.java2s.com/Code/Jar/o/Downloadorg eclipsejdtcore371jar.htm>

위의 주소로 접속하여 jar파일 다운로드

프로젝트 우클릭 후 properties 클릭 후
java build path에서 Libraries 탭으로 이동

Add External JARs... 클릭하여
다운받은 jars파일 선택



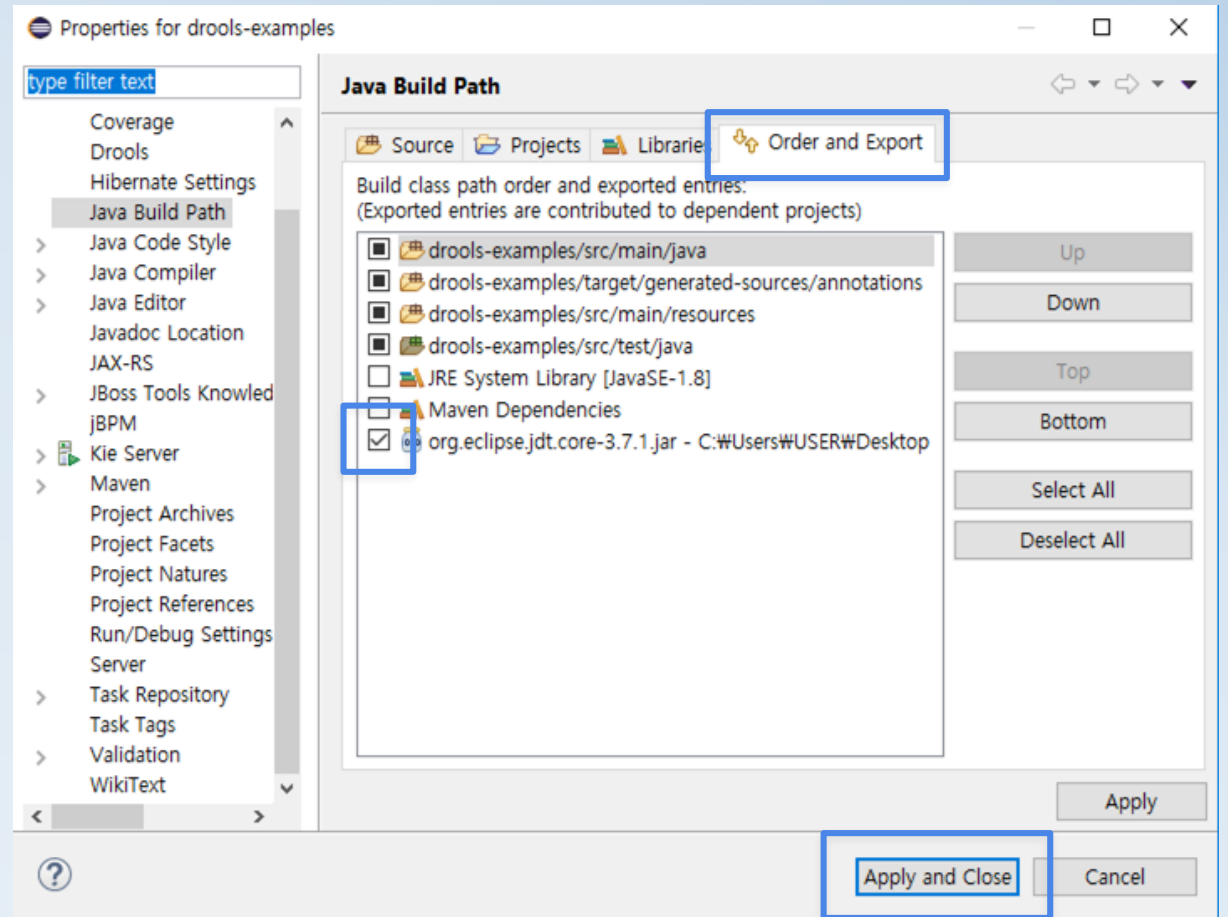
예제 프로젝트 설정

<http://www.java2s.com/Code/Jar/o/Downloadorg eclipsejdtcore371jar.htm>

Order and Export 탭으로 이동



추가한 jars 파일 선택 후 Apply



예제 프로젝트 설정

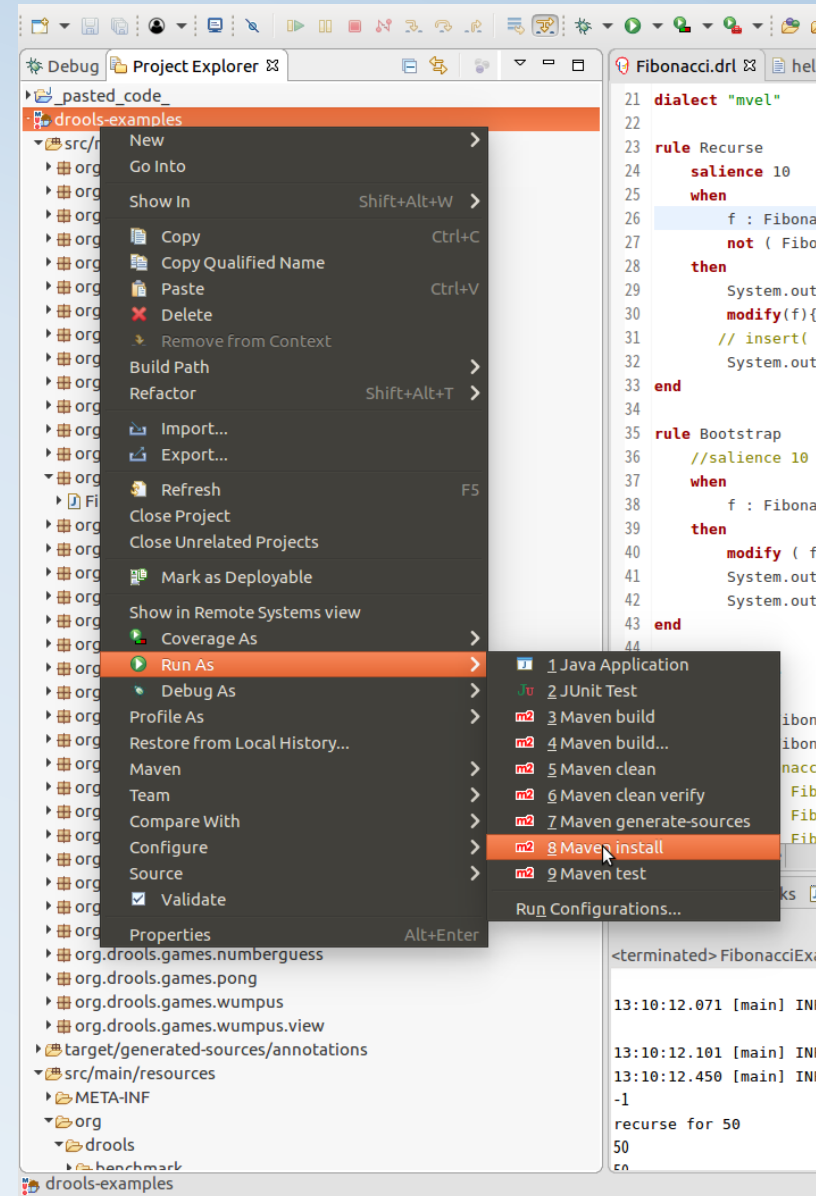
프로젝트 우클릭 후
Run As - Maven Install 클릭



FibonacciExample.java 파일을 찾아
Run as java application으로 실행

java파일이 프로젝트에 없을 경우
상단메뉴의 File - Open File을 이용해
java파일을 열어서 동일한 방법으로 실행

Maven Install을 실패하거나 에러가 발생하면
Maven clean - Maven install을 시도해보고
안된다면 이클립스설치부터 재시도



예제 프로젝트 설정

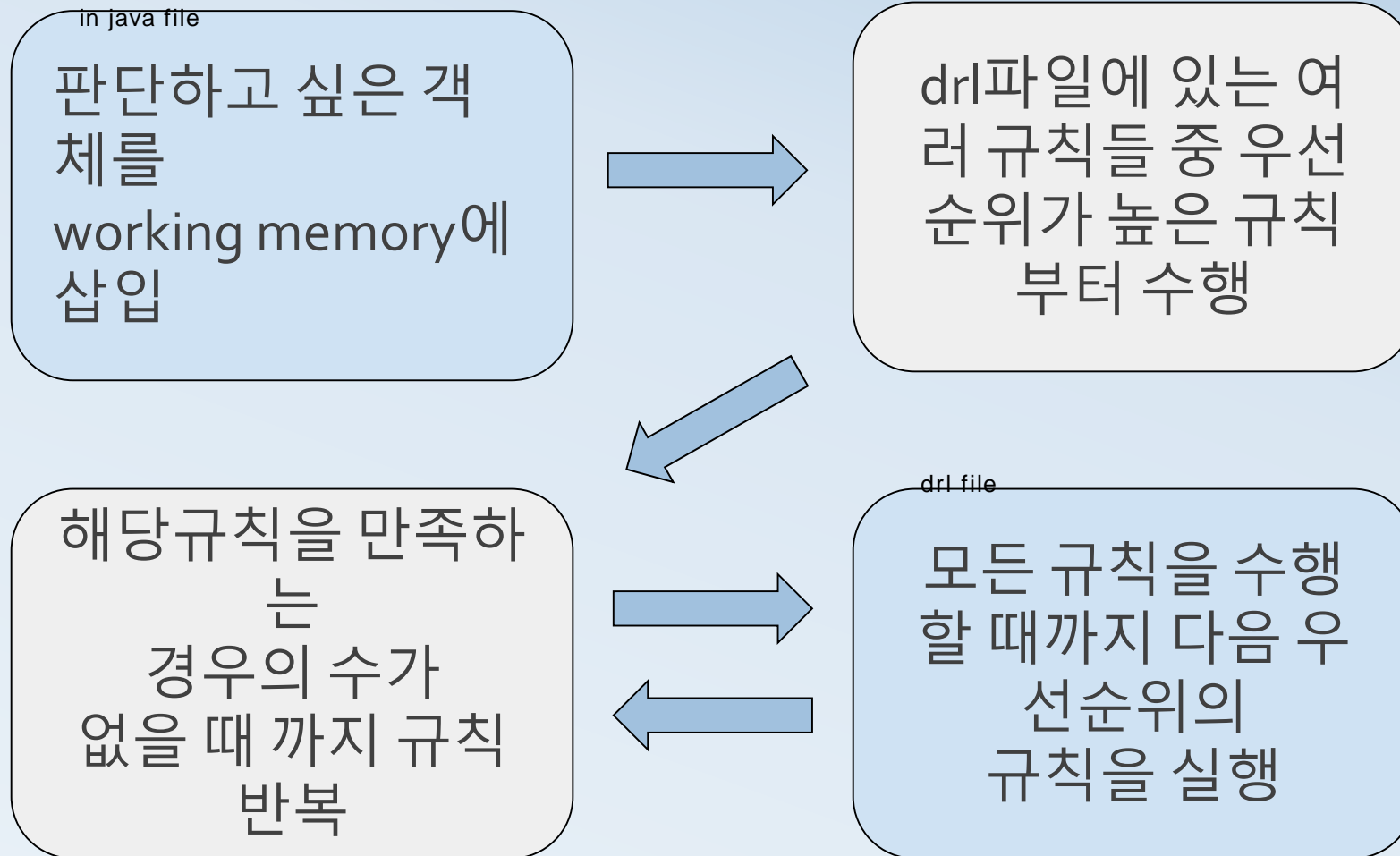
```
[INFO] -----  
[INFO] BUILD FAILURE  
[INFO] -----  
[INFO] Total time: 4.850 s  
[INFO] Finished at: 2019-02-21T17:10:35+09:00  
[INFO] -----  
[ERROR] Failed to execute goal org.apache.maven.plugins:maven-compiler-plugin:3.8.0-jboss-1:compile (default-compile) on project drools-examples: Compilation failure  
[ERROR] No compiler is provided in this environment. Perhaps you are running on a JRE rather than a JDK?  
[ERROR] -> [Help 1]  
[ERROR]  
[ERROR] To see the full stack trace of the errors, re-run Maven with the -e switch.  
[ERROR] Re-run Maven using the -X switch to enable full debug logging.  
[ERROR]  
[ERROR] For more information about the errors and possible solutions, please read the following articles:  
[ERROR] [Help 1] http://cwiki.apache.org/confluence/display/MAVEN/MojoFailureException
```

위와 동일한 에러가 발생할 경우 jars파일 재설치
프로젝트 우클릭 - properties에서 jars파일을 설정해 준 후 maven install

Drools 구조



Drools 실행구조



Drools 기본구조

rule rule의 이름을 정할 때 쓰는 키워드로
옴표는 선택적으로 사용

속성 rule의 옵션을 설정하는 영역
no-loop, salience, data-effective 등

when - 조건 when 키워드의 시작부터 then 키워드 전까지 작
성된 규칙조건

then - 실행 규칙조건이 만족할 때 실행되는 규칙동작

end 해당 규칙이 끝났다는 것을 알려주는 키워드

큰따

rule "룰 이름"

속성

when

조건

then

실행

end

drl

then
java

drl

Drools 기본구조

rule "룰 이름"

속성

when

조건

then

실행

end

```
package org.drools.examples.helloworld

import org.drools.examples.helloworld.HelloWorldExample.Message;

global java.util.List list

rule "Hello World"
    dialect "mvel" mvel
    when
        m : Message( status == Message.HELLO, message : message )
    then
        System.out.println( message );
        // modify ( m ) { setMessage( "Goodbye cruel world" ),
        //               setStatus( Message.GOODBYE ) };
        modify ( m ) { message = "Goodbye cruel world",
                        status = Message.GOODBYE };
    end

rule "Good Bye"
    dialect "java"
    when
        Message( status == Message.GOODBYE, message : message )
    then
        System.out.println( message );
    end
end
```

Message
status, message

가

, java 가

Drools 사용예제



Hello World 예제를 이용한 Drools 문법 설명

```
① rule "Hello World"
  ② dialect "mvel"
    when
      ③ m : Message( status == Message.HELLO, message : message )
    then
      System.out.println( message );
      //      modify ( m ) { setMessage( "Goodbyte cruel world" ),
      //                      setStatus( Message.GOODBYE ) };
      ⑤ modify ( m ) { message = "Goodbye cruel world",
                      status = Message.GOODBYE };
    end
end
```

drools

① 규칙이름은 "Hello World"로 한다.

② dialect 키워드를 이용하여 선언된 표현언어를 사용한다.

③ 콜론을 이용하여 콜론 앞쪽의 변수에 콜론 뒤쪽의 속성을 바인딩한다.

④ // 를 이용하여 한 줄 전체를 주석처리한다.

⑤ modify 키워드를 이용하여 m에 바인딩 된 객체의 속성값을 변경한다.
변수의 값이 변경되었다면 동일한 규칙을 다시 실행한다.

Hello World 예제

```
package org.drools.examples.helloworld

import org.drools.examples.helloworld.HelloWorldExample.Message;

global java.util.List list

rule "Hello World"
    dialect "mvel"
    when
        m : Message( status == Message.HELLO, message : message )
    then
        System.out.println( message );
        //      modify ( m ) { setMessage( "Goodbye cruel world" ),
        //                      setStatus( Message.GOODBYE ) };
        modify ( m ) { message = "Goodbye cruel world",
                        status = Message.GOODBYE };
    end

rule "Good Bye"
    dialect "java"
    when
        Message( status == Message.GOODBYE, message : message )
    then
        System.out.println( message );
    end
```

“Hello World” 규칙

dialect 키워드에 의해 mvel 표현식으로 작성

status == Message.HELLO를 만족하는 객체 message를 호출하여 해당 객체의 message 값을 필드 message값으로 초기화

when 조건을 만족하면 변수 message값을 출력한 후 객체의 message을 "Goodbye cruel world"로, 객체의 status를 상수 GOODBYE로 변경

“Good Bye” 규칙

status == Message.GOODBYE를 만족하는 객체 message를 호출하여 변수message을 필드 message값으로 초기화

when 조건을 만족하면 변수 message 값을 출력

피보나치 수열 예제

```

> rule Recurse
  salience 10
  when
    f : Fibonacci ( value == -1 )
    not ( Fibonacci ( sequence == 1 ) )
  then
    insert( new Fibonacci( f.sequence - 1 ) );
    System.out.println( "recurse for " + f.sequence );
  end

> rule Bootstrap
  when
    f : Fibonacci( sequence == 1 || == 2, value == -1 )
  then
    modify ( f ){ value = 1 };
    System.out.println( f.sequence + " == " + f.value );
  end

> rule Calculate
  when
    f1 : Fibonacci( s1 : sequence, value != -1 )
    f2 : Fibonacci( sequence == (s1 + 1 ), value != -1 )
    f3 : Fibonacci( s3 : sequence == (f2.sequence + 1 ), value == -1 )
  then
    modify ( f3 ) { value = f1.value + f2.value };
    System.out.println( s3 + " == " + f3.value );
  end

```

실행 결과▶

```

recurse for 50
recurse for 49
recurse for 48
recurse for 47
...
recurse for 5
recurse for 4
recurse for 3
recurse for 2
1 == 1
2 == 1
3 == 2
4 == 3
5 == 5
6 == 8
...
47 == 2971215073
48 == 4807526976
49 == 7778742049
50 == 12586269025

```

Drools 예제 피보나치 수열

```
public static class Fibonacci {  
    private int  sequence;  
    private long value;  
  
    public Fibonacci( final int sequence ) { //  
        this.sequence = sequence;  
        this.value = -1;  
    }  
  
    ... setters and getters go here...  
}
```

```
ksession.insert( new Fibonacci( 50 ) ); //  
ksession.fireAllRules();
```

drl

>

피보나치 수열 예제의 자바 클래스 파일에서
sequence는 50, value는 -1인 객체를 insert하
여 working memory에 삽입한 후 규칙을 실행한다.

Recurse Rule

```

> rule Recurse
  salience 10
  when
    f : Fibonacci ( value == -1 )
    not ( Fibonacci ( sequence == 1 ) )
  then
    insert( new Fibonacci( f.sequence - 1 ) );
    System.out.println( "recurse for " + f.sequence );
  end

```

가 1

50

가 2

1 가

1. salience는 규칙에 우선순위를 부여하는 키워드로 규칙이 충돌할 때 salience값이 큰 규칙을 먼저 실행한다. 키워드를 사용하지 않았을 때의 기본값은 0이다. 모든 규칙 중 salience값이 가장 높은 Recurse규칙을 우선 실행한다.

2. 현재 working memory에 있는 객체는 (50,-1)인 객체이므로 when절을 만족하여 then절에서 (49,-1)인 객체를 working memory에 추가한다.

3. Drools는 한 번 규칙을 만족한 대상이 있으면 그 대상으로는 규칙을 실행하지 않는다. (50,-1)객체는 이미 Recurse 규칙을 만족한 적이 있으므로 선택대상에서 무시되고 (49,-1)인 객체가 선택된다.

4. 2~3번을 sequence 값이 2가 될 때까지 반복한다.

5. when절을 수행하지 않은 객체가 더 이상 없으므로 Recurse규칙을 종료한다.

Bootstrap Rule

```
rule Bootstrap
  when
    f : Fibonacci( sequence == 1 || == 2, value == -1 )
  then
    modify ( f ){ value = 1 };
    System.out.println( f.sequence + " == " + f.value );
  end
```

1. 현재 working memory에서 sequence값이 1이거나 2이면서 value값이 -1인 객체를 가져와 value값을 1로 수정
2. 더 이상 when절을 만족하는 객체의 경우가 없으면 Bootstrap 규칙 종료

가 -1

Calculate Rule

```
rule Calculate
  when
    f1 : Fibonacci( s1 : sequence, value != -1 )
    f2 : Fibonacci( sequence == (s1 + 1 ), value != -1 )
    f3 : Fibonacci( s3 : sequence == (f2.sequence + 1 ), value == -1 )
  then
    modify ( f3 ) { value = f1.value + f2.value };
    System.out.println( s3 + " == " + f3.value );
  end
```

1. working memory에서 value값이 -1이 아닌 객체를 가져와 f1에 바인딩
2. sequence가 f1보다 1크고 value가 -1이 아닌 객체를 f2에 바인딩
3. sequence가 f2보다 1크고 value가 -1인 객체를 f3에 바인딩
4. f3의 value를 f1.value + f2.value로 수정한 뒤 f3의 값 출력
5. 1~4를 반복하다 더이상 when절을 만족하는 객체의 경우의 수가 없으면 Calculate Rule 종료

참고자료

Drools 공식 홈페이지 document https://docs.jboss.org/drools/release/7.16.0.Final/drools-docs/html_single/index.html#decision-examples-fibonacci-ref drools-examples

Drools 설치 <https://tramamte.github.io/2018/06/04/eclipse-install/>

Rule Engine <https://elaia.tistory.com/43>