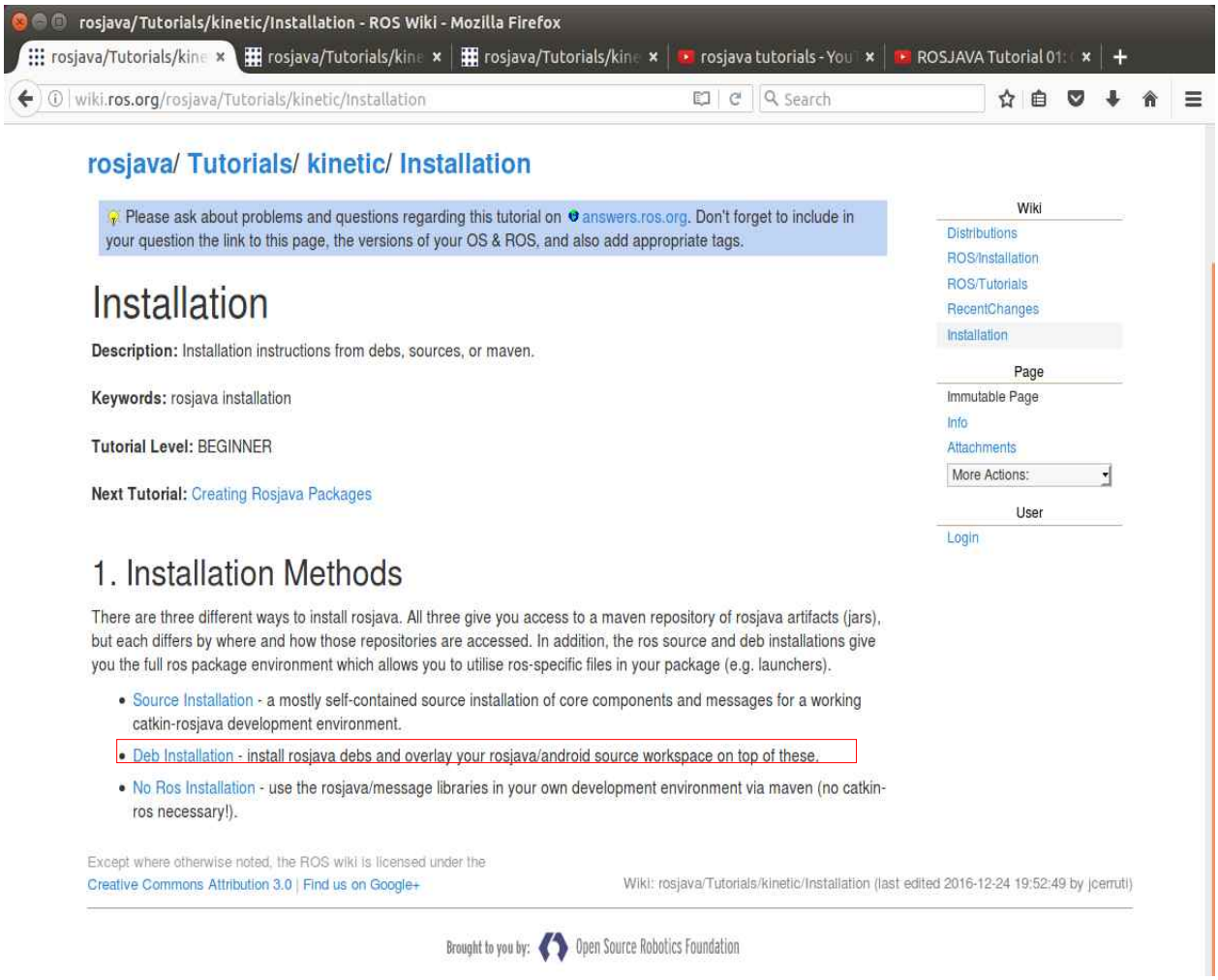
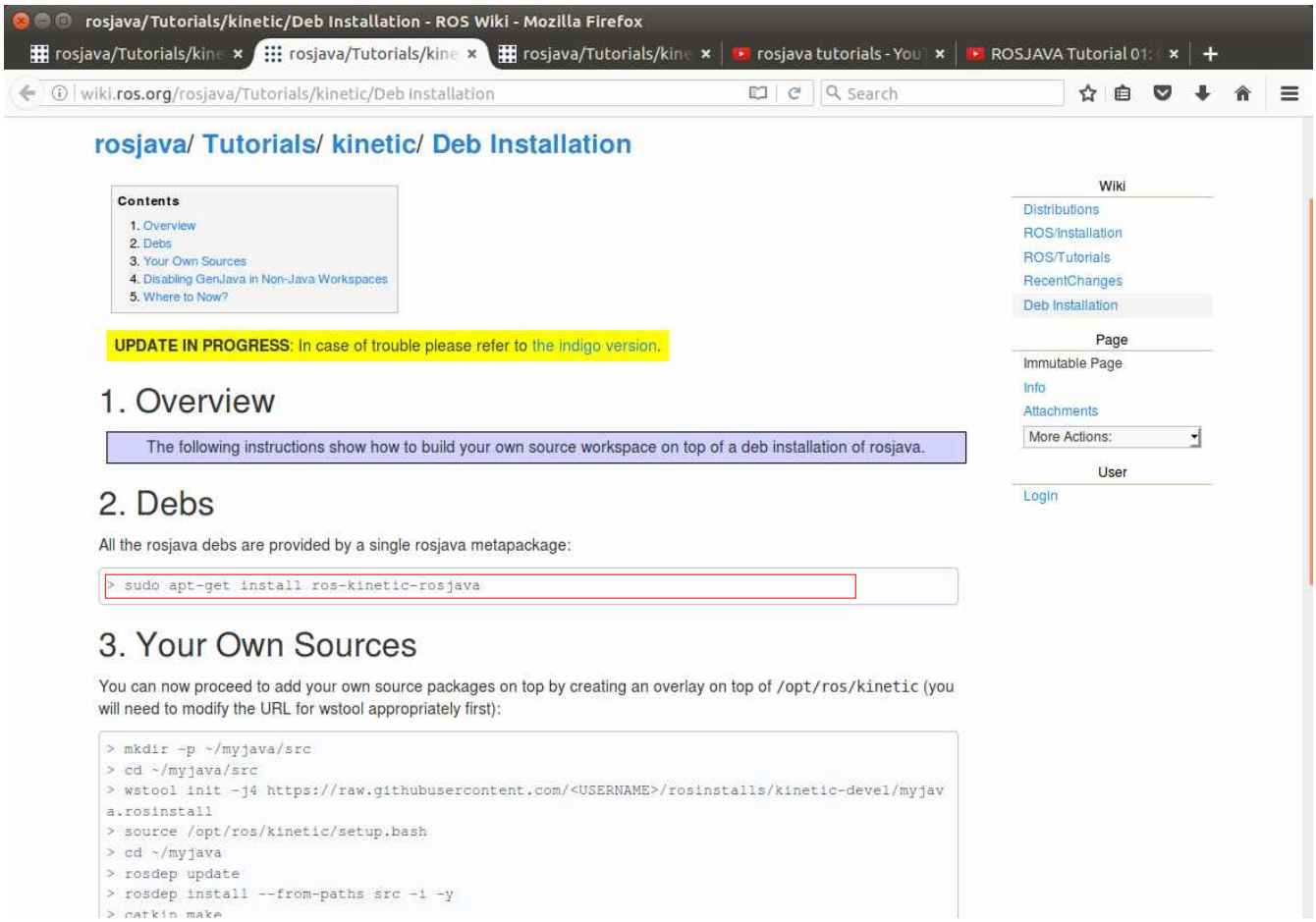
리눅스에서의 rosjava 설치방법과 사용법.

1. 준비
   - ros 설치
   - eclipse 설치

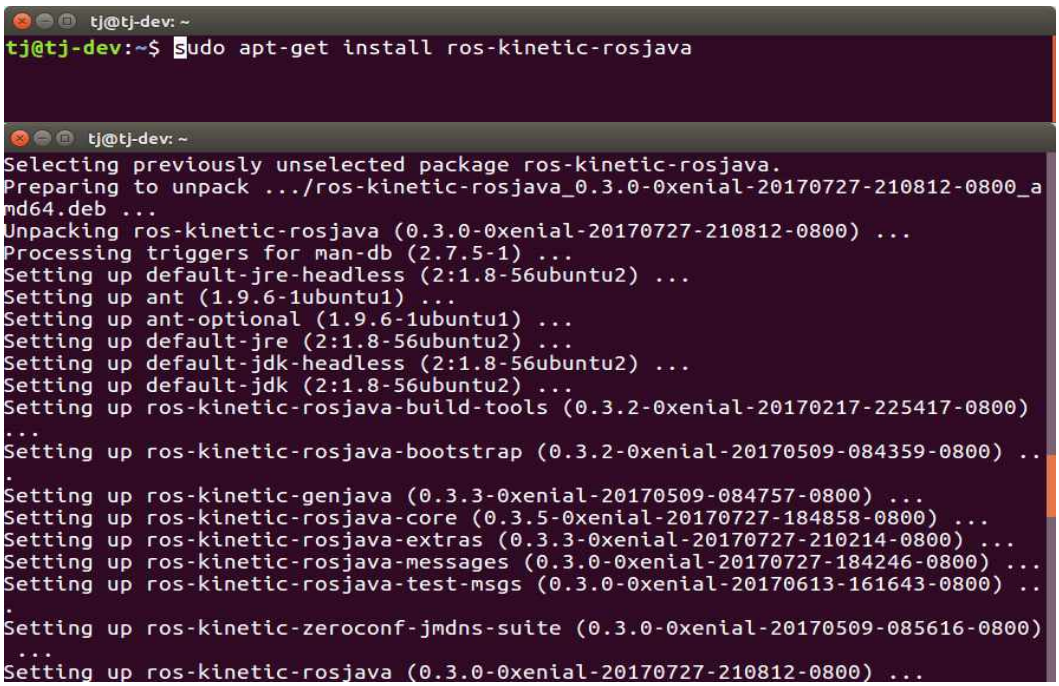
2. rosjava 설치



-rosjava/../installation 페이지에서 deb installation을 선택한다.

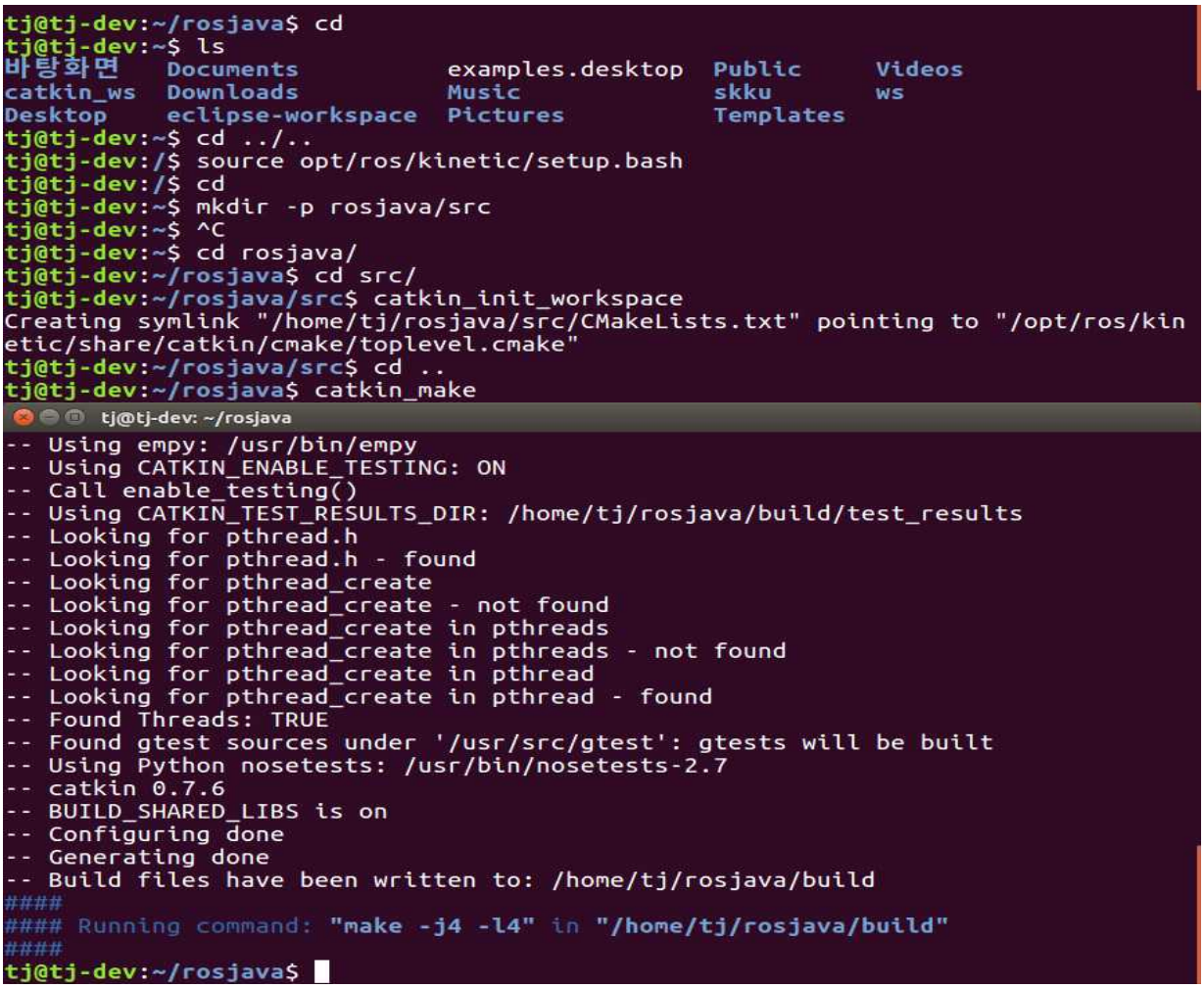


- 터미널에 sudo apt-get install ros-<version name>-rosjava를 입력.



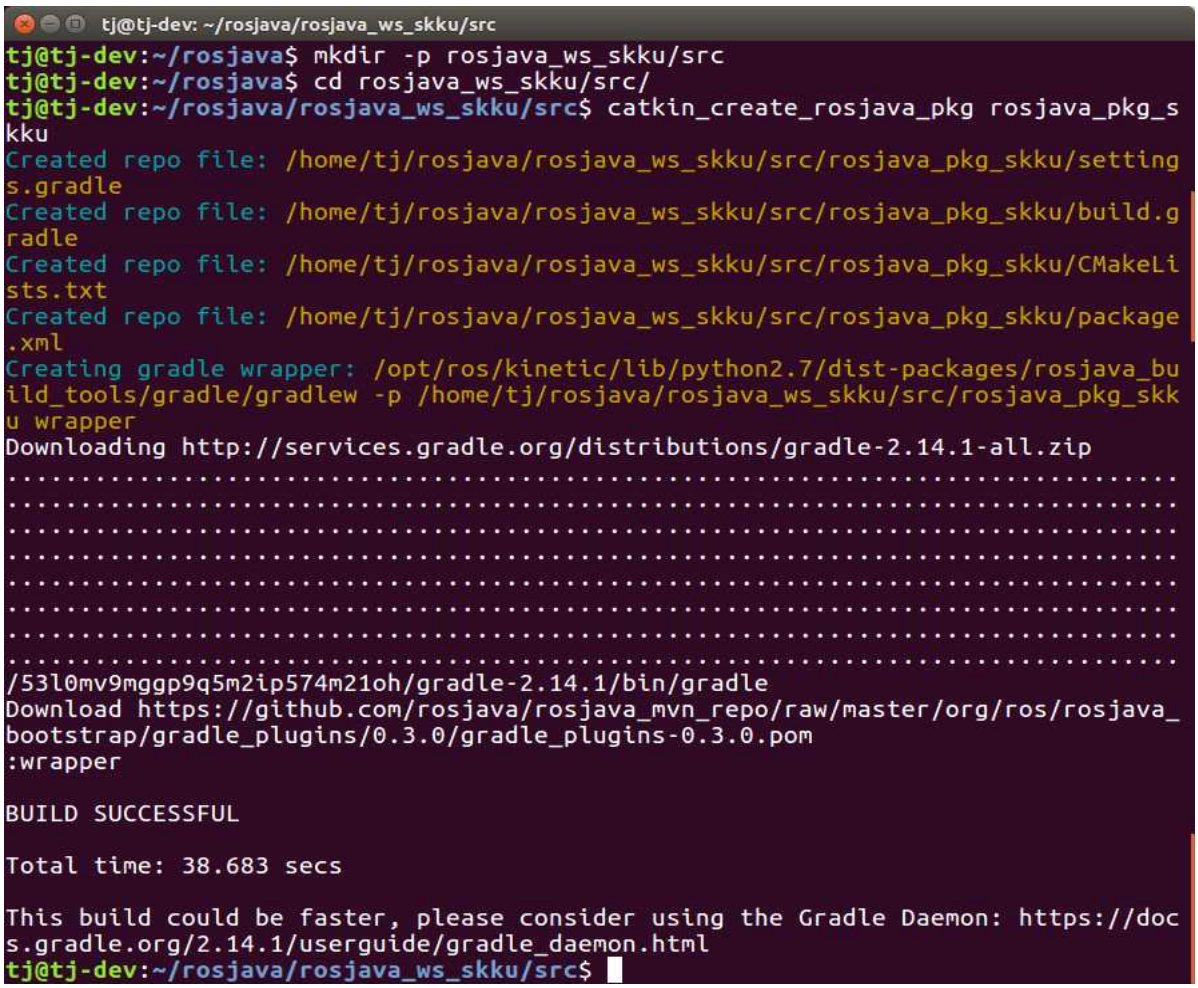
-설치가 끝났다면 위와 같이 출력이 된다.

3. rosjava 작업 환경 생성.



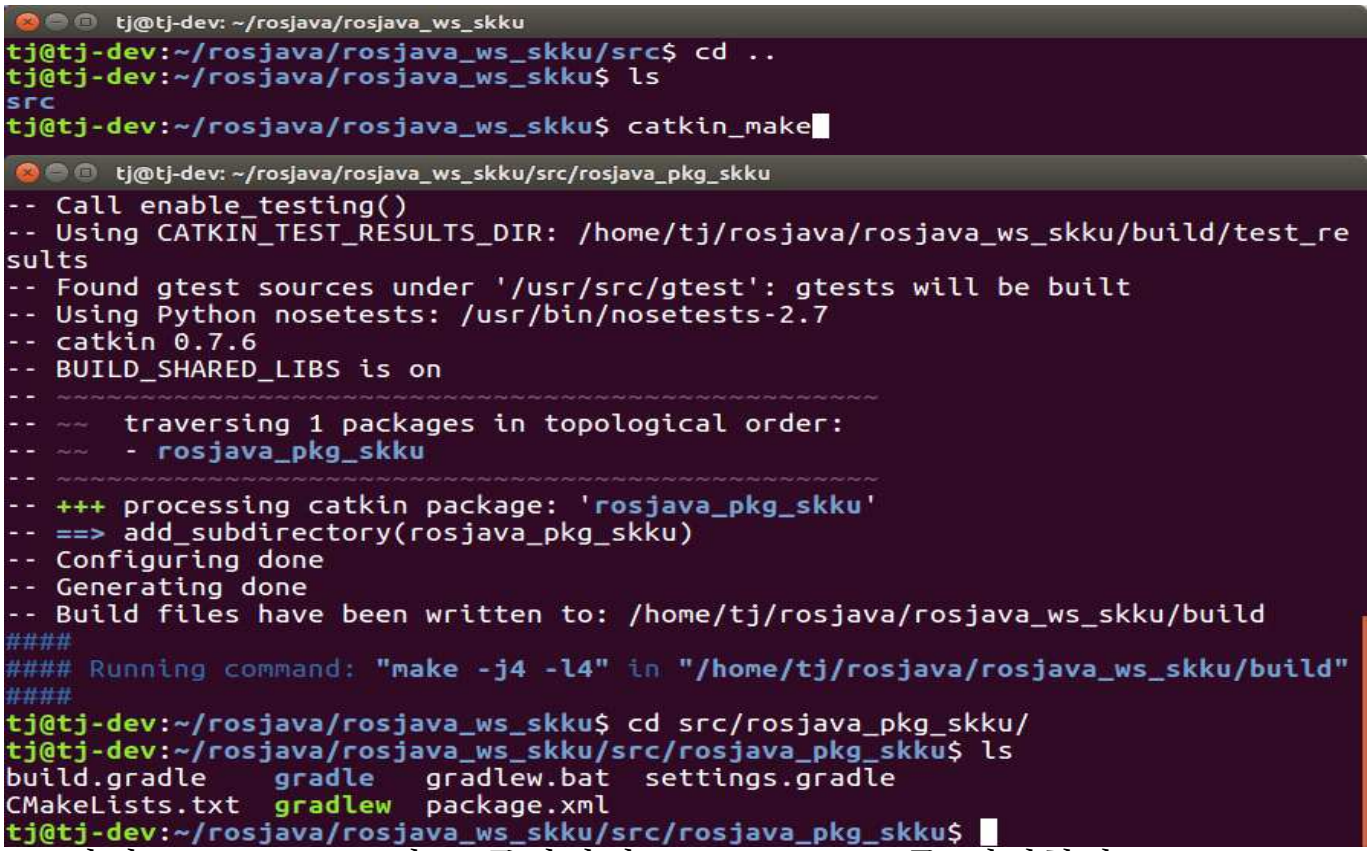
- ros의 명령어를 사용하기 위해서
  source opt/ros/<version name>/setup.bash를 터미널에 입력을 한다.
- mkdir -p rosjava/src 입력 하여 디렉토리를 생성 한다.
- /src에서 catkin_init_workspace를 입력하면 workspace를 초기화하기 위한 파일들이 생성된다.
- /rosjava에서 catkin_make를 입력하면 위의 이미지처럼 출력된다.

4. rosjava workspace 생성.



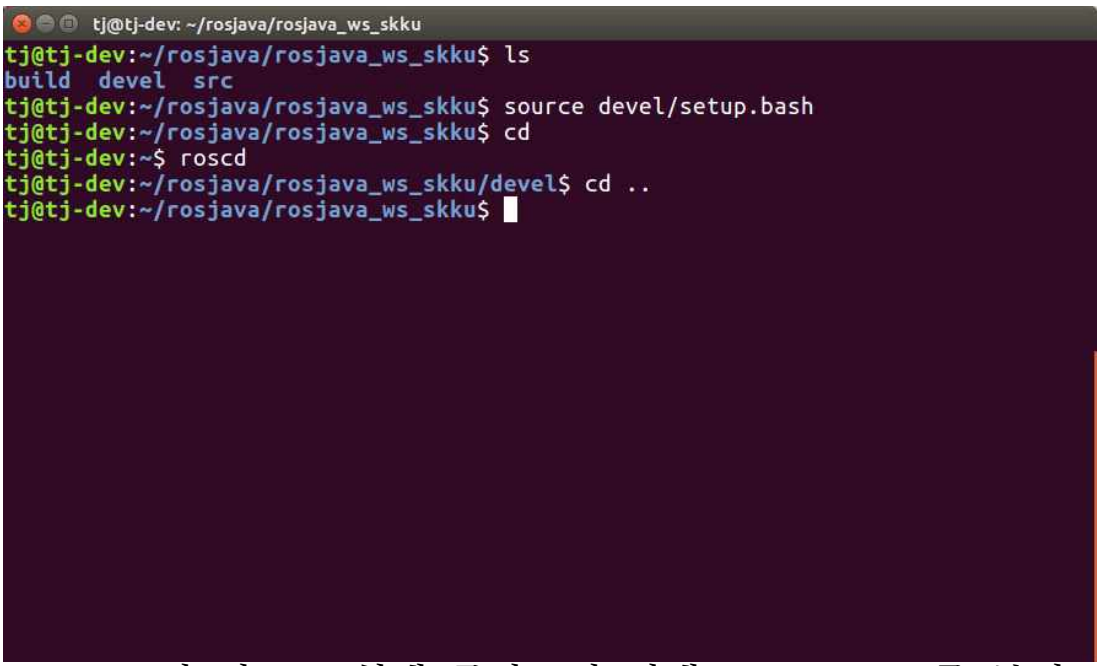
- /rosjava에서 mkdir -p <workspace명>/src를 입력하여 디렉토리를 생성한다.

_ ../src에서 catkin_create_rosjava_pkg 명령어를 사용하여 rosjava package를 생성한다.



- 다시 workspace 경로 돌아와서 catkin_make를 입력한다.

- 이 후에 build작업은 대부분 이 경로에서 catkin_make를 입력하여 진행을 한다. 위의 이미지처럼 디렉토리 내에 여러 파일들이 생성된다.



-workspace의 경로로 쉽게 돌아오기 위해 setup.bash를 불러오면 roscd를 입력했을 때 쉽게 경로로 들어올수 있다.

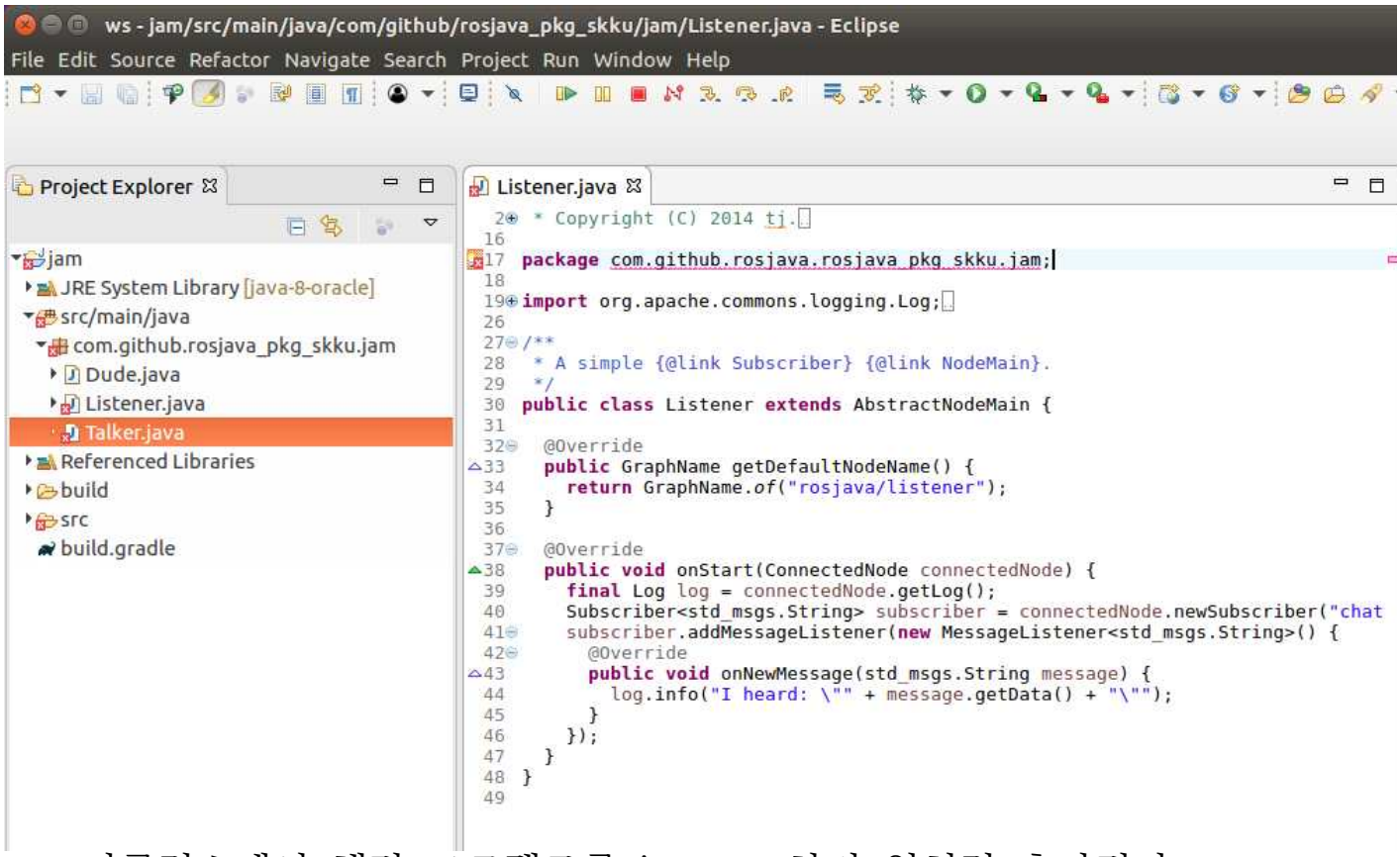
5. 프로젝트 생성.

```
tj@tj-dev: ~/rosjava/rosjava_ws_skku/src/rosjava_pkg_skku
tj@tj-dev:~/rosjava/rosjava_ws_skku/src/rosjava_pkg_skku$ catkin_create_rosjava_
library_project jam

Creating rosjava project
  Name       : jam
  File       : build.gradle
  File       : settings.gradle
  File       : Dude.class
  File       : CMakeLists.txt (gradle task update)
tj@tj-dev:~/rosjava/rosjava_ws_skku/src/rosjava_pkg_skku$ catkin_create_rosjava_
project jam

Creating rosjava project
  Name       : jam
  File       : build.gradle
  File       : settings.gradle
  File       : Talker.java
  File       : Listener.java
  File       : CMakeLists.txt (gradle task update)
tj@tj-dev:~/rosjava/rosjava_ws_skku/src/rosjava_pkg_skku$
```

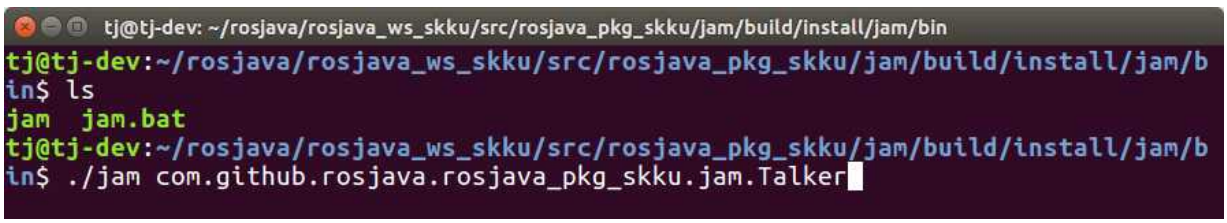
- 패키지 경로 내에서 catkin_create_rosjava_library_project 명령어와 catkin_create_rosjava_project 명령어 뒤에 프로젝트 이름과 함께 입력을 한다.
- library_project를 생성해야 다른 라이브러리를 import 할 수 있다.
- 빌드 후에는 Talker와 Listener, Dude를 지우고 새로운 Node클래스를 추가하여 작업을 할 수 있다.(이 후에 소개)

```
tj@tj-dev: ~/rosjava/rosjava_ws_skku
-- Call enable_testing()
-- Using CATKIN_TEST_RESULTS_DIR: /home/tj/rosjava/rosjava_ws_skku/build/test_re
sults
-- Found gtest sources under '/usr/src/gtest': gtests will be built
-- Using Python nosetests: /usr/bin/nosetests-2.7
-- catkin 0.7.6
-- BUILD_SHARED_LIBS is on
-- ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
-- ~~  traversing 1 packages in topological order:
-- ~~  - rosjava_pkg_skku
-- ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
-- +++ processing catkin package: 'rosjava_pkg_skku'
-- ==> add_subdirectory(rosjava_pkg_skku)
-- Configuring done
-- Generating done
-- Build files have been written to: /home/tj/rosjava/rosjava_ws_skku/build
####
#### Running command: "make -j4 -l4" in "/home/tj/rosjava/rosjava_ws_skku/build"
####
[100%] Gradling tasks for rosjava_pkg_skku
Could not find metadata com.github.rosjava.rosjava_pkg_skku:jam/maven-metadata.x
ml in remote (file:/home/tj/rosjava/rosjava_ws_skku/devel/share/maven/)
[100%] Built target gradle-rosjava_pkg_skku
tj@tj-dev:~/rosjava/rosjava_ws_skku$
```

- workspace로 돌아와 catkin_make를 입력하면 위처럼 빌드가 된다.



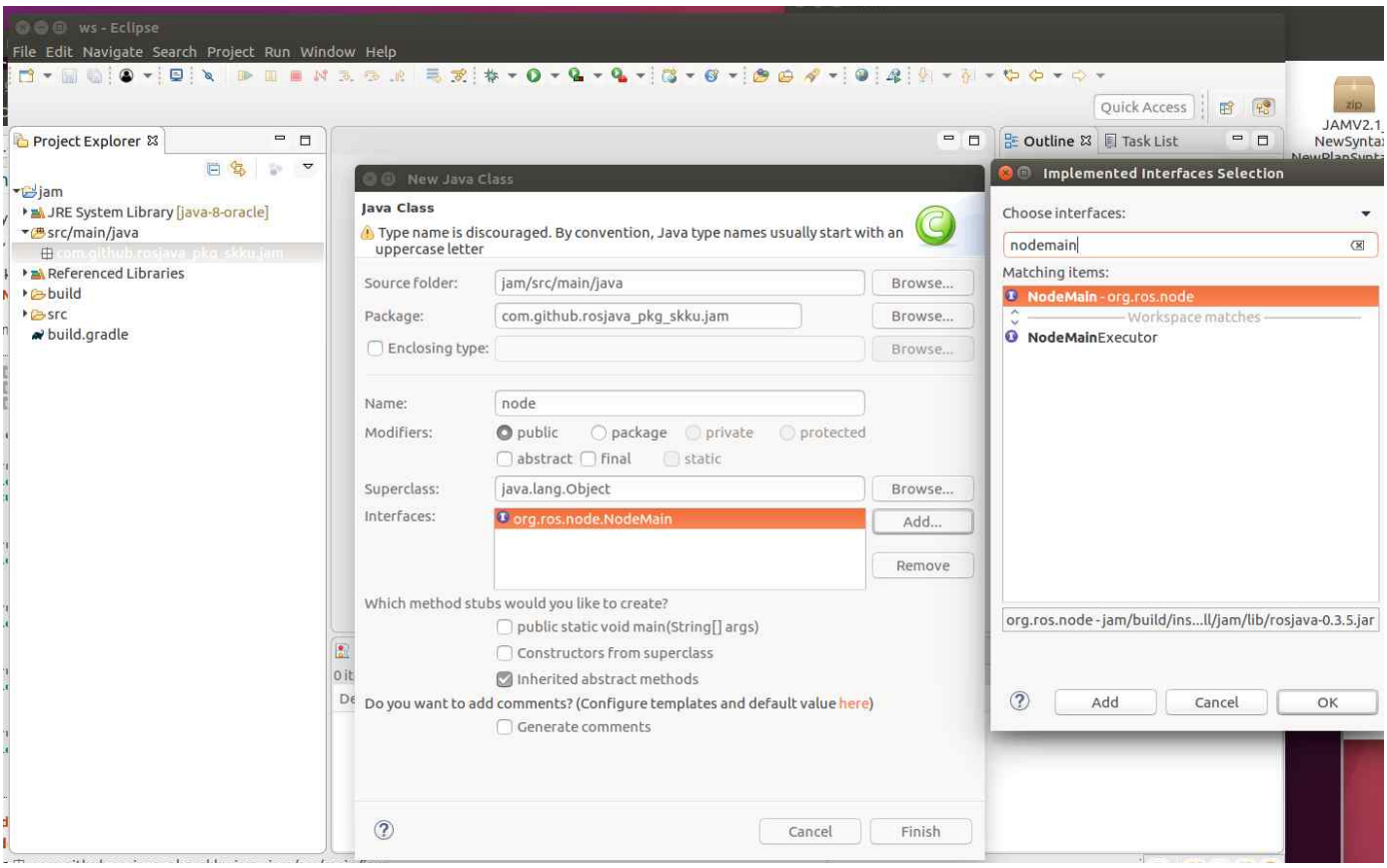
-이클립스에서 해당 프로젝트를 import 하면 위처럼 추가된다.



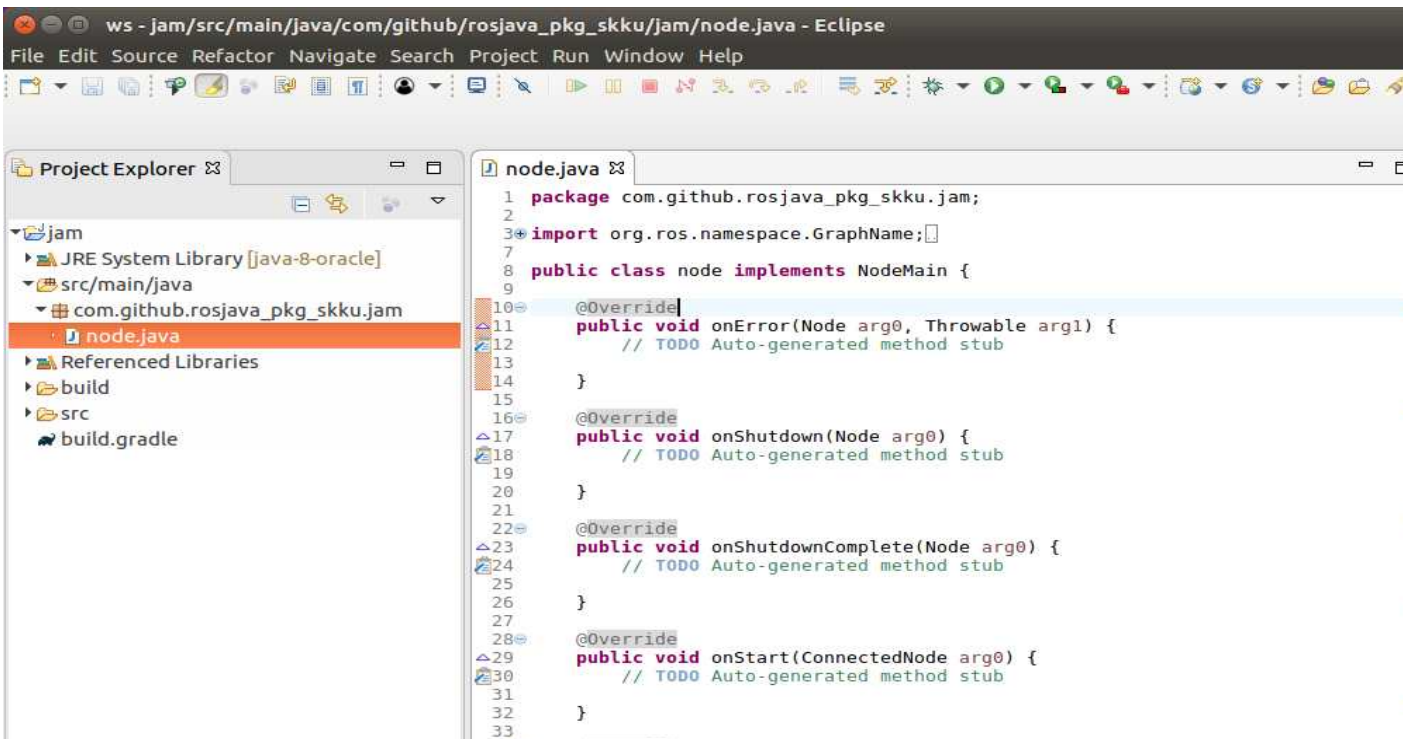
-빌드된 프로그램의 실행은 위처럼 입력을 하면 된다.

./<프로그램명> <패키지.클래스명>

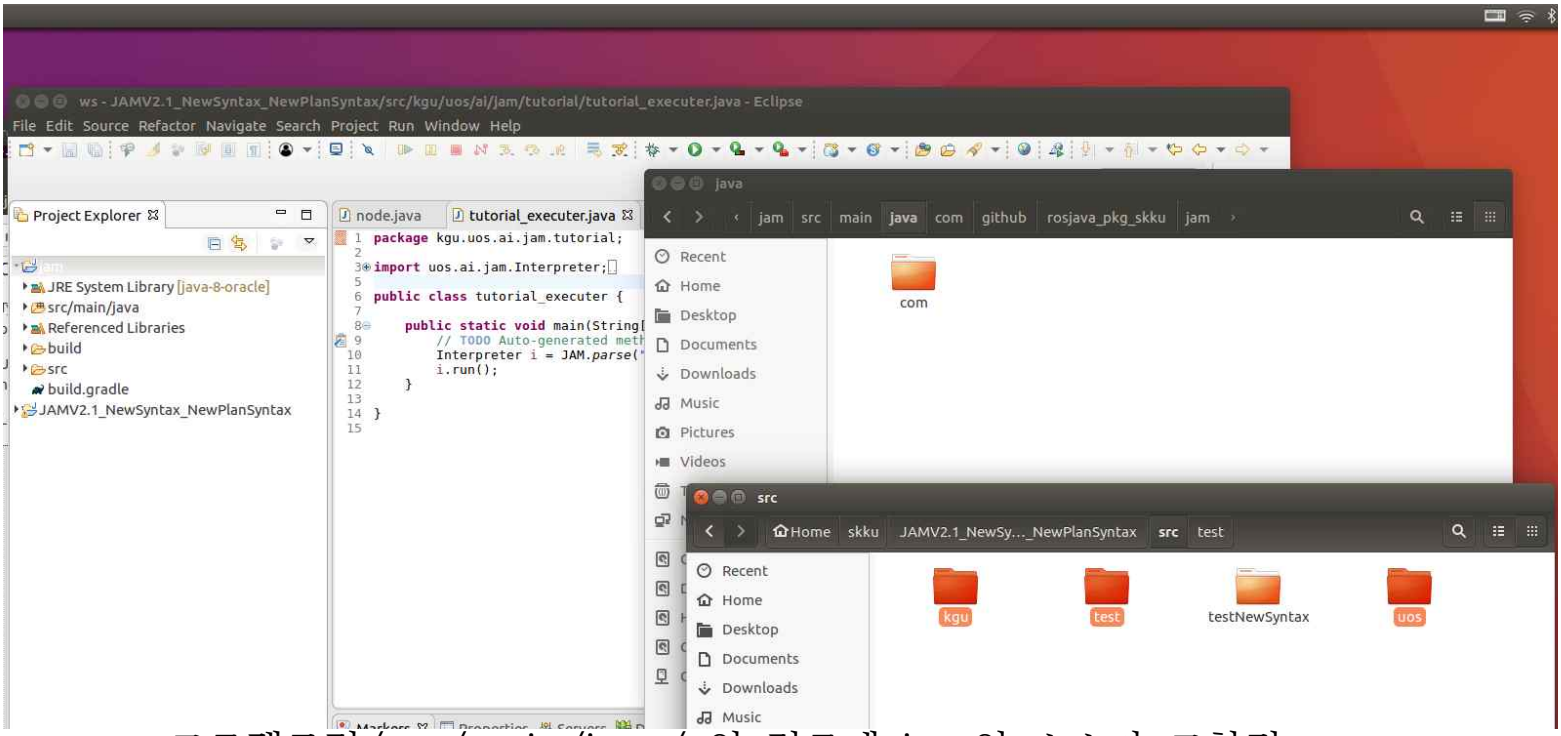
6. eclipse에서 작업.



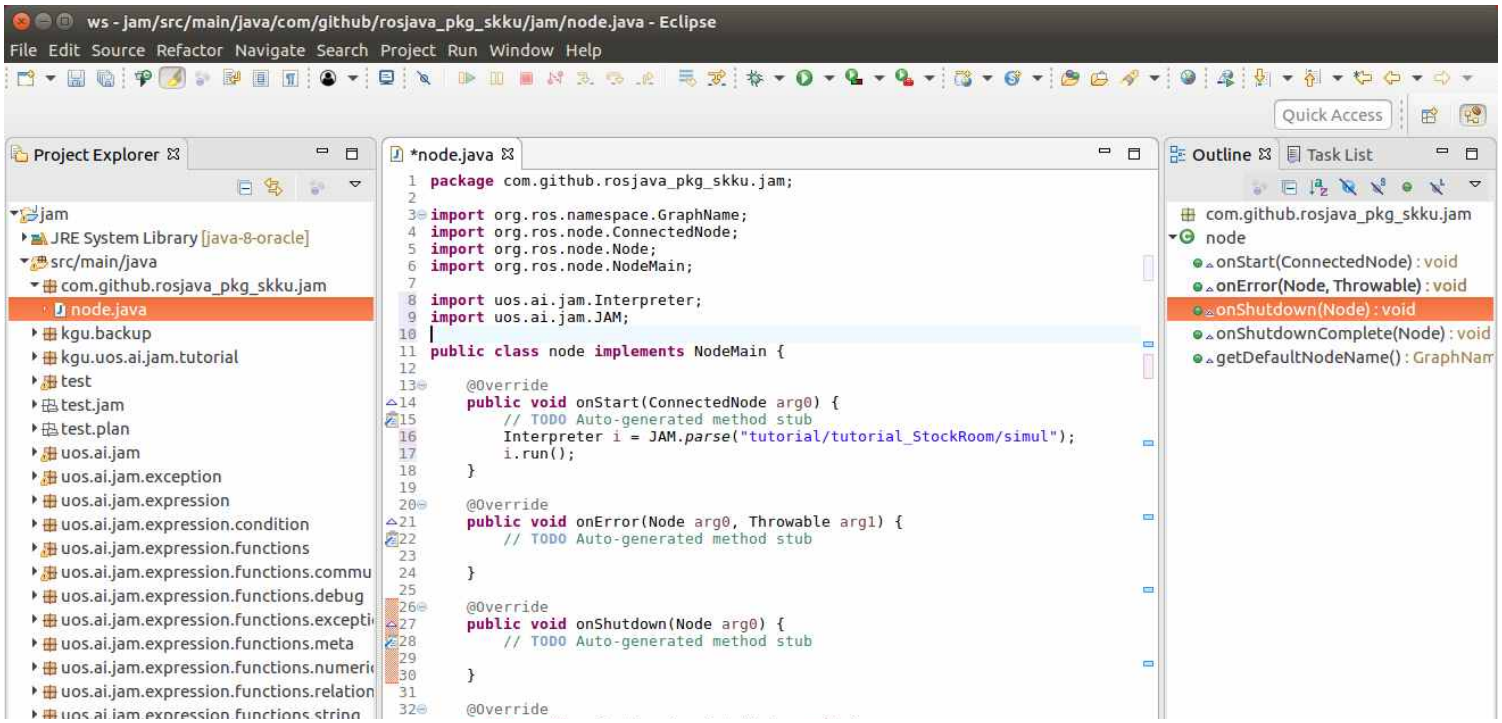
-예제 클래스인 Dude, Talker, Listener를 삭제하고 클래스 추가를 실행하여 위의 이미지처럼 NodeMain의 interface를 상속받는다.



7. 프로젝트에 JAM 추가



- 프로젝트명/src/main/java/ 의 경로에 jam의 소스가 포함된 kgu,test,uos 폴더들을 복사한다.
- eclispe의 프로젝트를 새로고침하면 jam이 추가된 것을 볼 수 있고 아래의 이미지처럼 node클래스의 onStart 메소드에 jam에서 실행했던 interpreter를 추가하면 jam을 실행 시킬 수 있다.
(빌드 작업 후, 빌드 된 실행파일 경로에 tutorial 파일을 옮겨놔야 튜토리얼 프로그램을 실행 할 수 있다.)



- 추가로 eclipse가 사용자 권한보다 높은 권한에서 설치되었을 때, eclipse에서 작업을 하고 catkin_make로 빌드를 실행하면 빌드 에러가 발생 하는 경우가 있다. 이때 sudo catkin_make를 입력하면 해결이 된다.