

Eclipse에서 디버깅

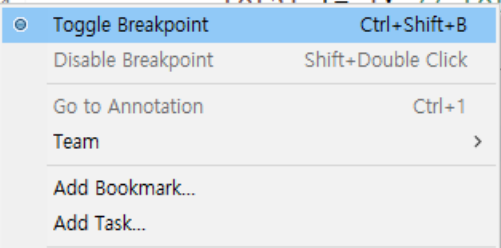
1. Break point 설정

디버깅을 시작할 코드 위치(중단 시킬 위치)에 편집기 왼쪽 파란 부분에 더블 클릭

```
8      for (int i = 1; i <= 100; i++) {
9
10     Line breakpoint:Lab [line: 10] - main(String[]) total = total + i;
11
12     System.out.println("1부터 " + i + "
13     }
14 }
```

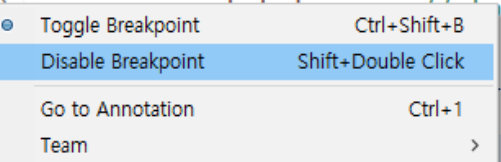
또는 마우스를 우 클릭하여 Toggle Breakpoint를 선택합니다

```
8      for (int i = 1; i <= 100; i++) {
9
10     total = i; // total = total + i;
11
12     System.out.println("1부터 " + i + "
13     }
14 }
```

A context menu is shown over the code editor. The menu has the following items: 'Toggle Breakpoint' with a radio button and keyboard shortcut 'Ctrl+Shift+B', 'Disable Breakpoint' with keyboard shortcut 'Shift+Double Click', 'Go to Annotation' with keyboard shortcut 'Ctrl+1', 'Team' with a right arrow, 'Add Bookmark...', and 'Add Task...'. The 'Toggle Breakpoint' option is currently selected.

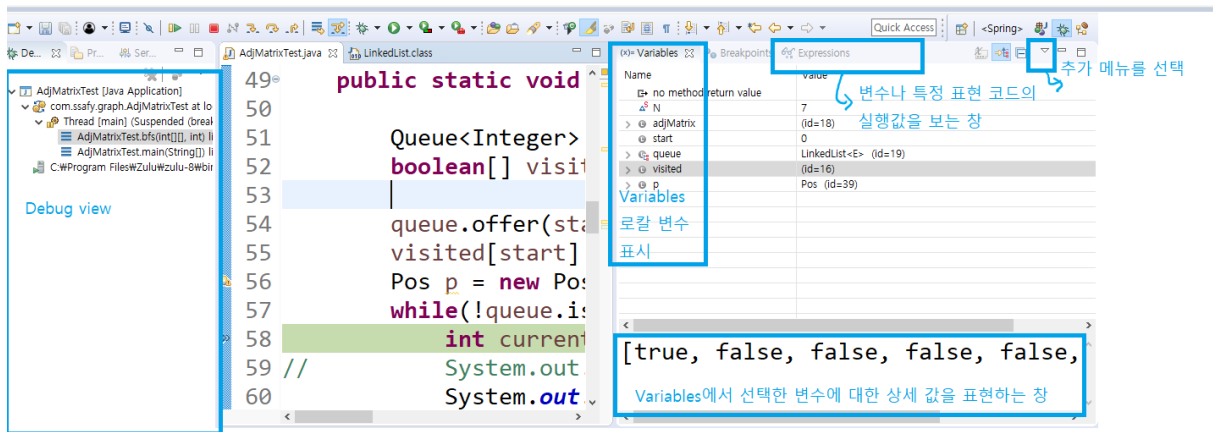
2. Break point 해제

브레이 포인트에서 마우스 우 클릭 >> Disable Breakpoint

A context menu is shown over the code editor, similar to the one in the previous image. In this image, the 'Disable Breakpoint' option, which has the keyboard shortcut 'Shift+Double Click', is selected.

3. Debug 모드 실행하기

단축키 f11 또는  로 실행한다.



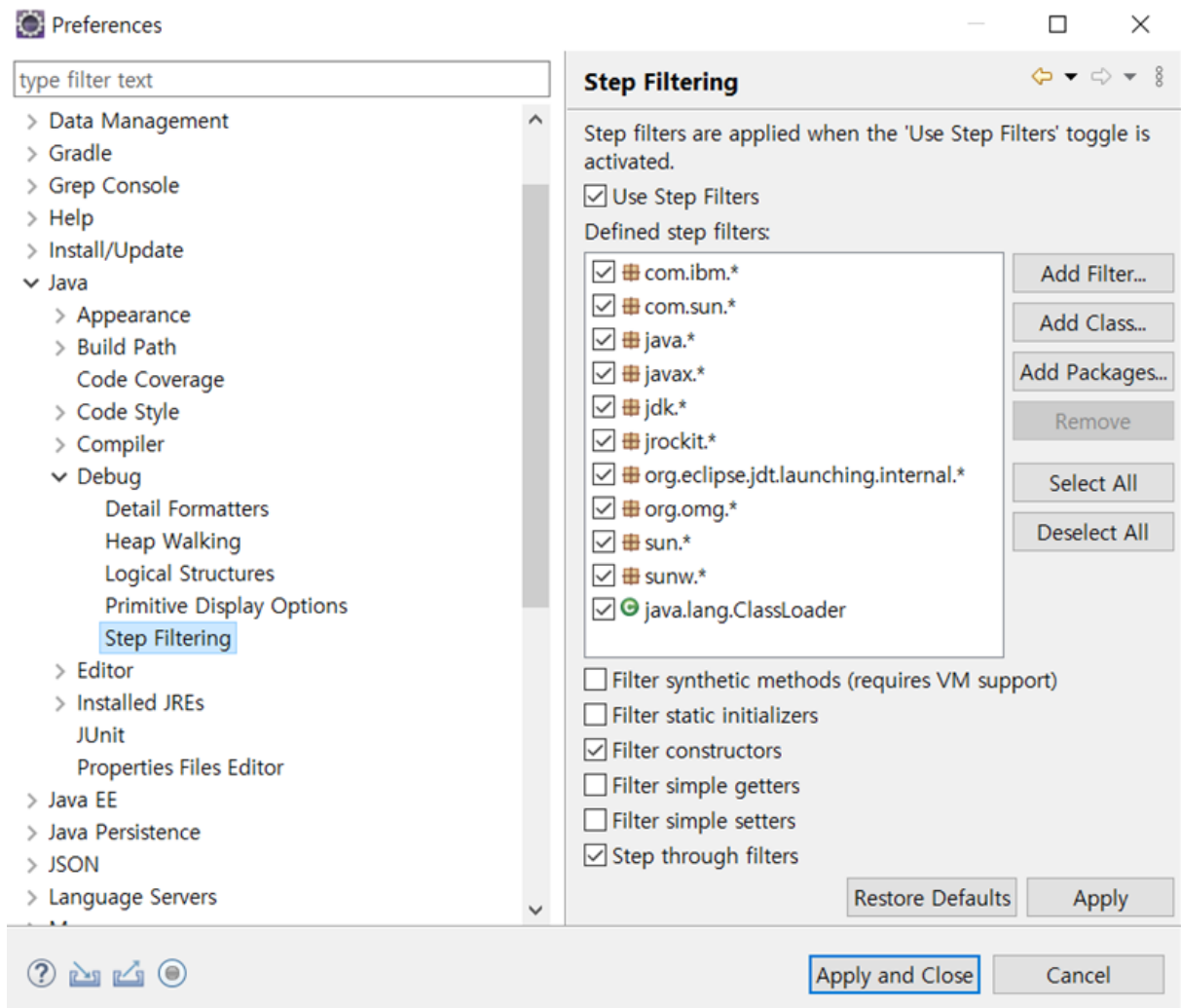
4. 디버그 뷰 버튼 설명

용어	설명
Resume 	<ul style="list-style-type: none"> 다음 브레이크 포인트를 만날 때까지 진행
Suspend 	<ul style="list-style-type: none"> 현재 작동하고 있는 Thread를 멈춤
Terminate 	<ul style="list-style-type: none"> 프로그램(디버그모드 실행)을 종료
Step Into 	<ul style="list-style-type: none"> Method를 포함한 라인을 만나면 Method 안으로 진입
Step Over 	<ul style="list-style-type: none"> 다음 라인으로 이동 Method가 있어도 Method 안으로 진입하지 않고 다음 라인으로 이동
Step Return 	<ul style="list-style-type: none"> Current Method에서 즉시 Return
Drop to Frame 	<ul style="list-style-type: none"> Method를 처음부터 다시 실행

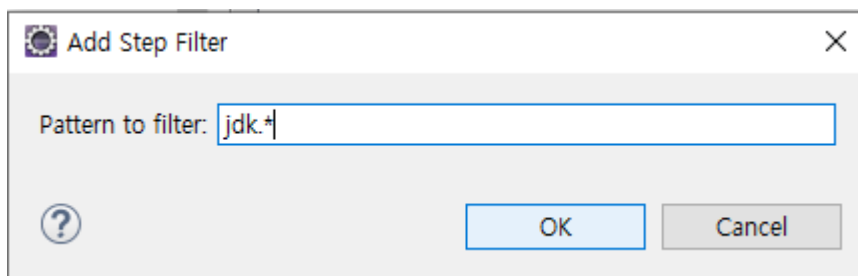
5. Step Filters

Step into(F5)로 debugging 할때 이미 확정된 api는 디버깅 대상이 아니니까 건너뛰기 설정

Window >> preferences >> java >> debug >> step filtering

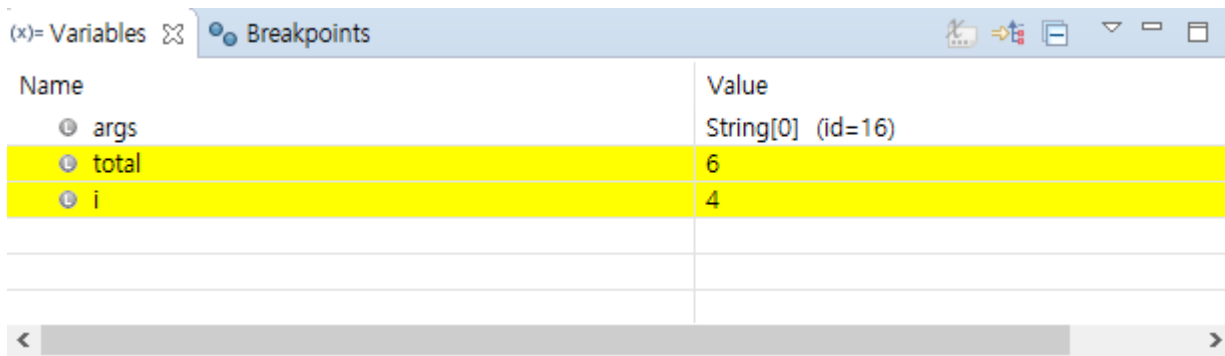


건너뛰기할 API를 선택하거나 Add Filter를 선택해서 추가 한다.




6. Variables

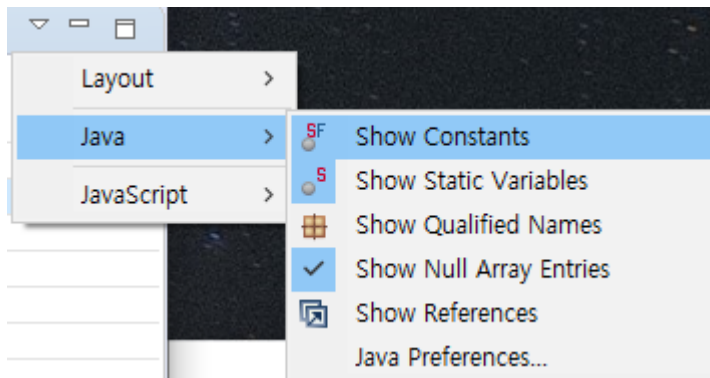
중단점을 지정한 곳의 함수에 선언된 로컬 변수를 표시한다.



Name	Value
args	String[0] (id=16)
total	6
i	4

Static 변수나, 상수도 Variables에 표현 하고 싶다면

추가 메뉴  에 마우스를 올리면 아래와 같이 추가 메뉴가 나오고
자바 > show constants 와 show static variables를 선택한다.



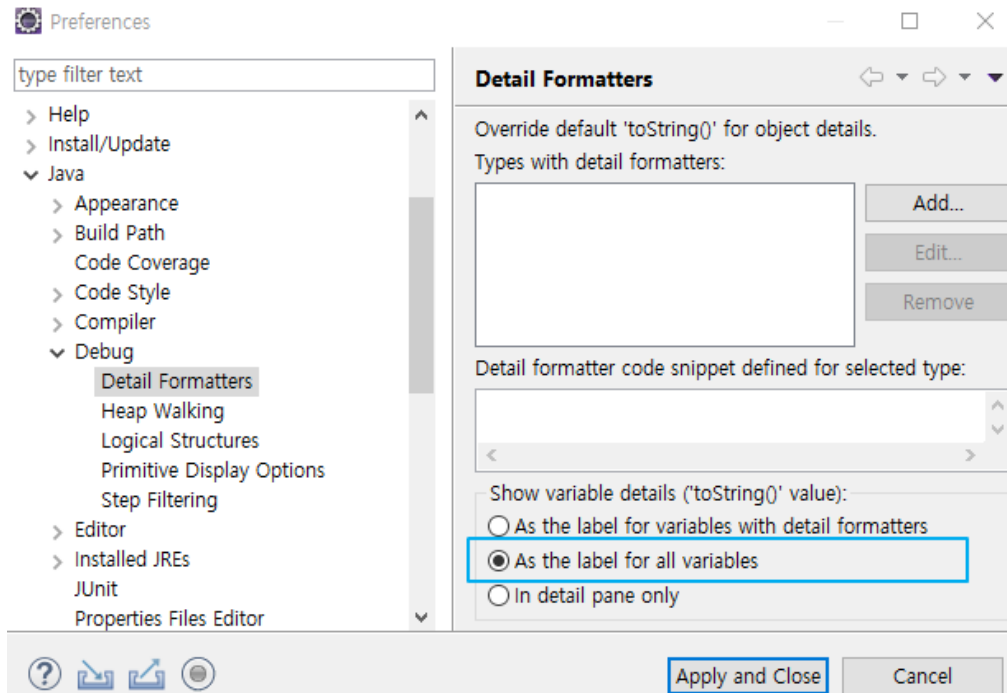
7. 객체의 내용을 value 에 표현하기

클래스에 선언한 toString()의 결과를 value에 표시 하기 위해서는

주 메뉴 ▾ >> java >> java Preferences >> detail Formatters 또는

Windows >> preferences >> java >> debug >> detail Formatters 를 선택후

Show variable details의 As the label for all variables를 선택한다.



선택 전

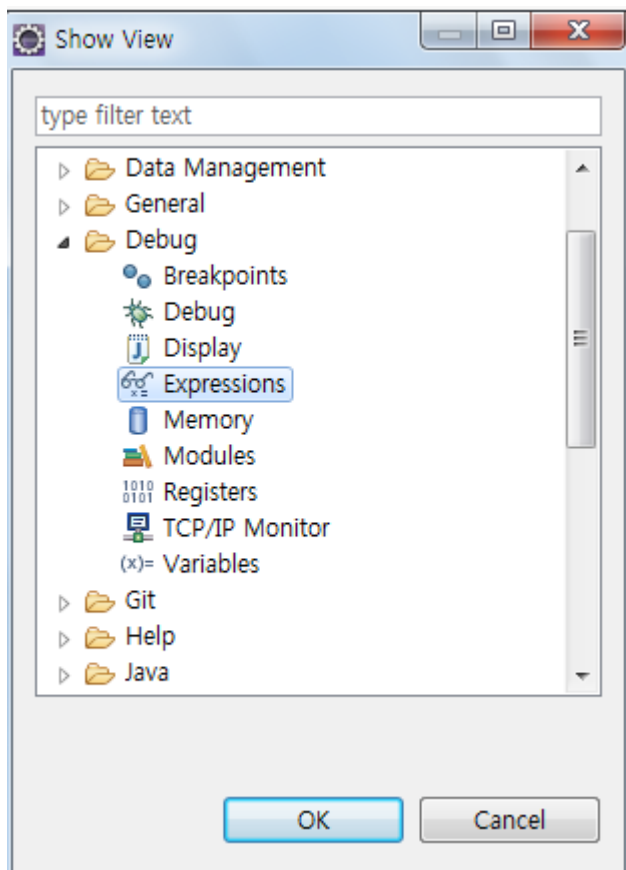
(x)= Variables Breakpoints Expressions	
Name	Value
no method return value	
N	7
adjMatrix	(id=18)
start	0
queue	LinkedList<E> (id=19)
visited	(id=16)
p	Pos (id=39)

선택 후

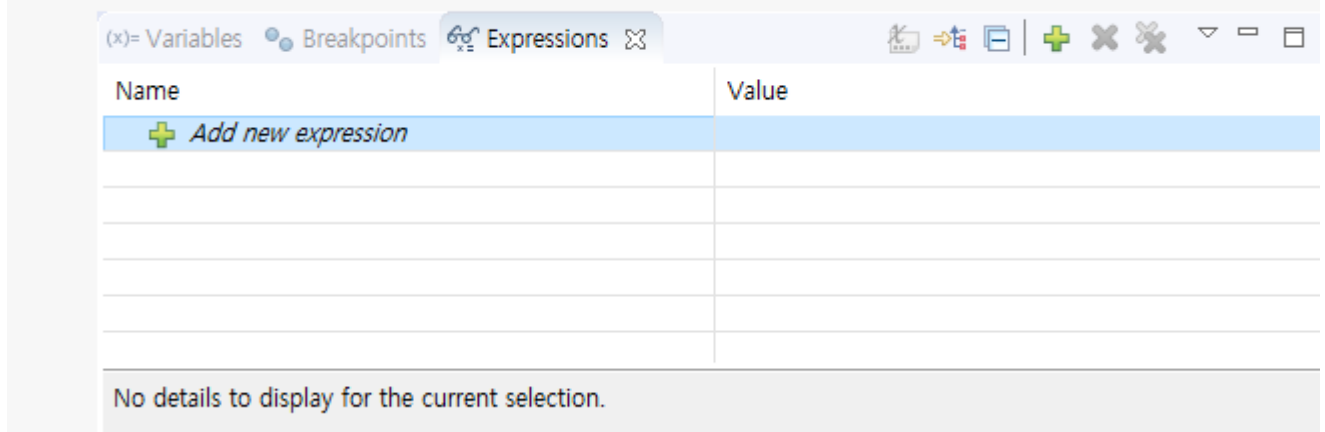
(x)= Variables Breakpoints Expressions	
Name	Value
no method return value	
N	7
adjMatrix	[[0, 1, 1, 0, 0, 0, 0], [1, 0, 0, 1, 1, 0, 0], [1, 0, 0, 0, 1, 0, 0], [0, 1, 0, 0, 0, 1, 0], [...
start	0
queue	[0]
visited	[true, false, false, false, false, false, false]
p	x:3 y:3

8. Expressions

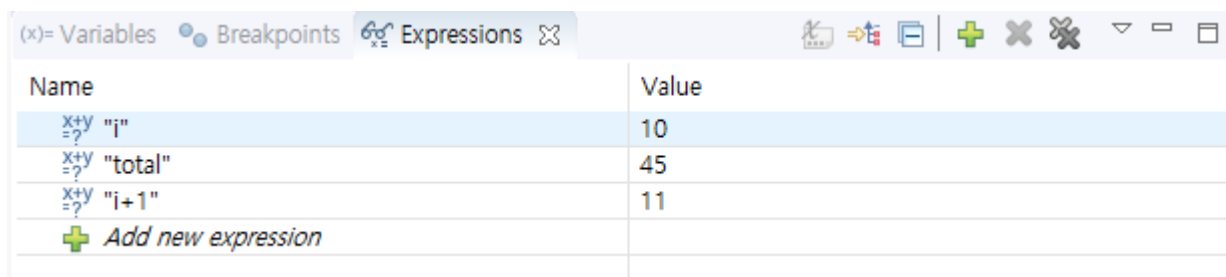
메뉴 windows >> show view >> other >> debug



[+ Add new expression] 클릭

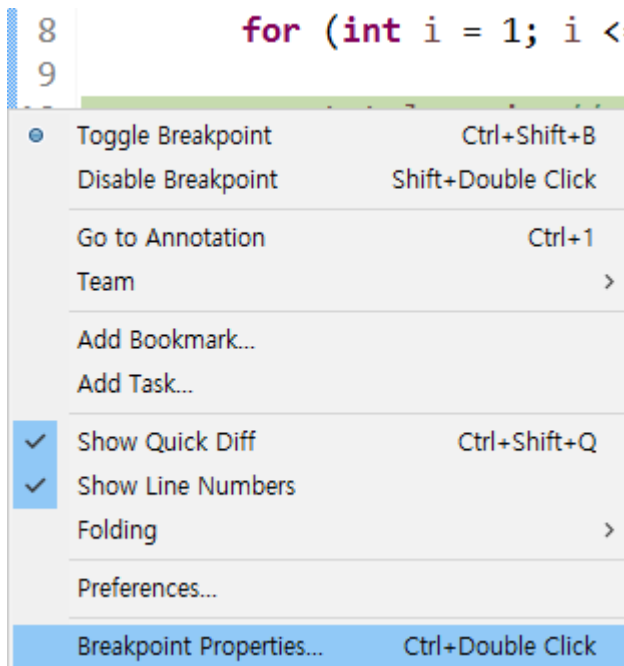


변수 i 가 있다면 추가, $i+1$ 이란 표식식도 추가 할 수 있다.

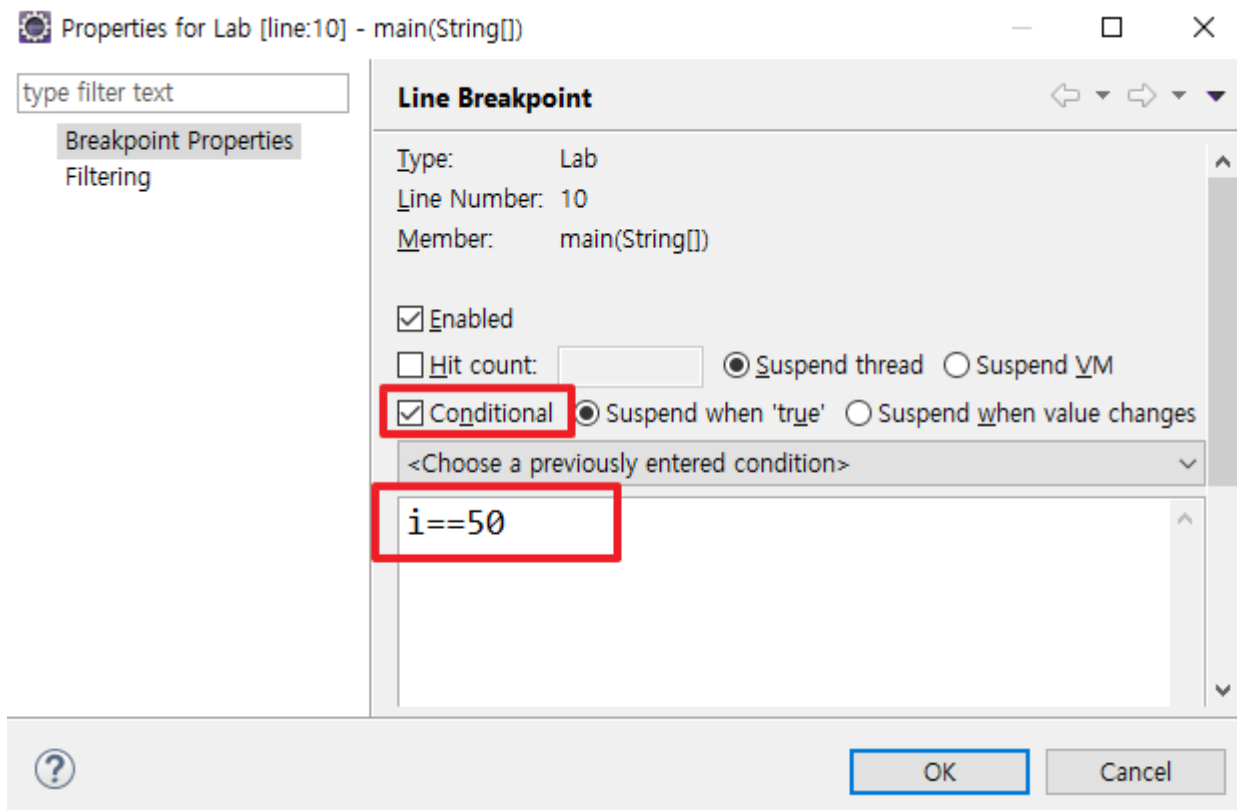


9. Conditional Break Point

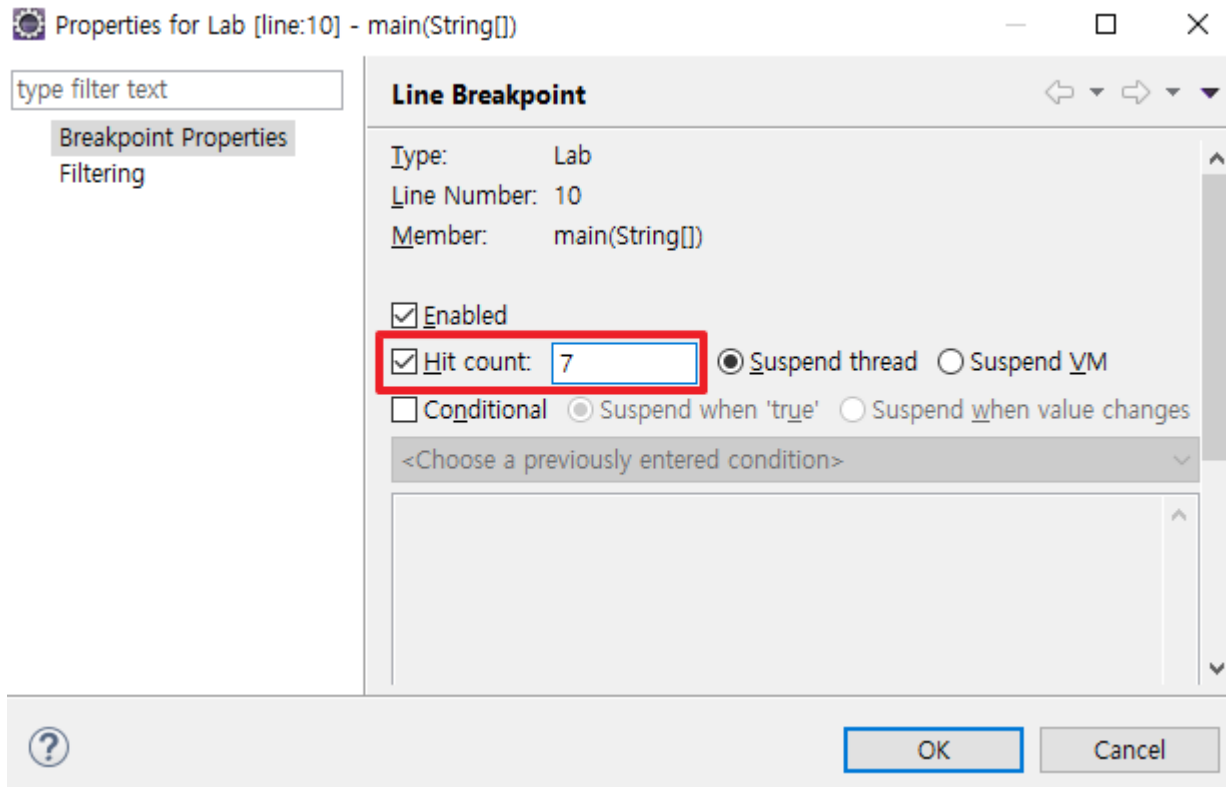
반복문 내부의 코드에 브레이크 포인트를 지정할 때, 중단점의 상태를 특정 조건에 맞게 지정할 수 있다.



반복 변수 `i`가 50이 될 때 중단하도록 설정



Break point로 설정한 코드가 7번 실행됐을 때 중단하도록 설정



10. 재귀함수에서 함수 별 Variables확인 및 return value 확인

재귀함수를 호출하는 조건에 break point를 설정 후 step into(F5)로 확인하기

