

이클립스에서 작업 도중 임의의 변수 값 확인하기

- 목적 : 문법 오류, 실행 오류가 아니라 런타임 오류 잡기

```
package pack;
import java.awt.event.*;
import javax.swing.*;
```

```
public class DebugTest extends JFrame implements ActionListener{
```

```
    int cou = 0;
    int tot = 0;
    JButton btn = new JButton("클릭");
    JTextField txtA = new JTextField();
```

```
    public DebugTest() {
        add("North", txtA);
        add("Center", btn);
        btn.addActionListener(this);

        setBounds(200, 200, 300, 300);
        setVisible(true);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    }
```

```
@Override
```

```
public void actionPerformed(ActionEvent e) {
    for (int i = 0; i < 5; i++) {
        cou++;
        System.out.println(cou); //변수 확인 1 : console로 출력
        tot += cou;
    }
    System.out.println("tot : " + tot);
```

```
//변수 확인 2 : 메시지 다이얼로그로 출력
```

```
//OptionPane.showMessageDialog(this, cou);
```

```
//변수 확인 3 : 임의의 컴포넌트로 출력
```

```
txtA.setText("합은 " + Integer.toString(tot));
```

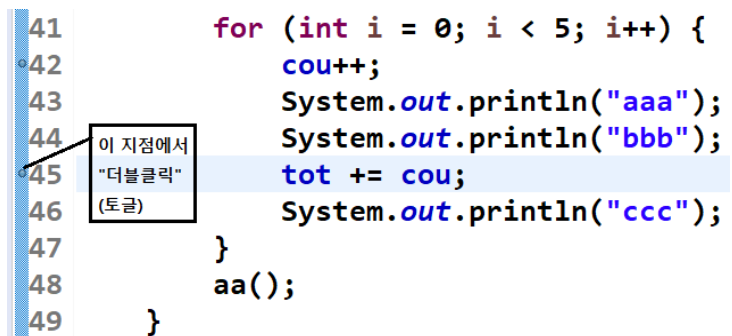
```
cou = 0;
tot = 0;
```

//변수 확인 4 : 디버깅 도구 사용 : 확인하고자 하는 라인에 중단점 표시

```
for (int i = 0; i < 5; i++) {
    cou++;
    System.out.println("aaa");
    System.out.println("bbb");
    tot += cou;
    System.out.println("ccc");
}
aa();
}
```

```
private void aa() {
    for (int i = 0; i < 5; i++) {
        cou++;
        tot += cou;
        bb();
        System.out.println("ddd");
    }
}
```

```
private void bb() {
    int kbs = 9;
    kbs++;
}
```

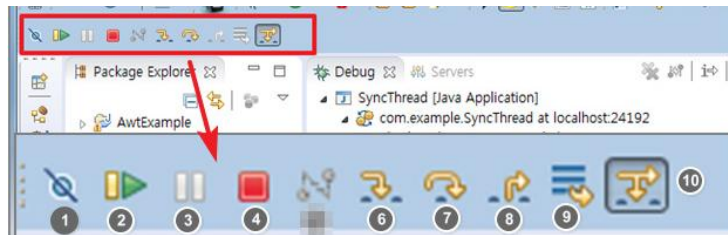
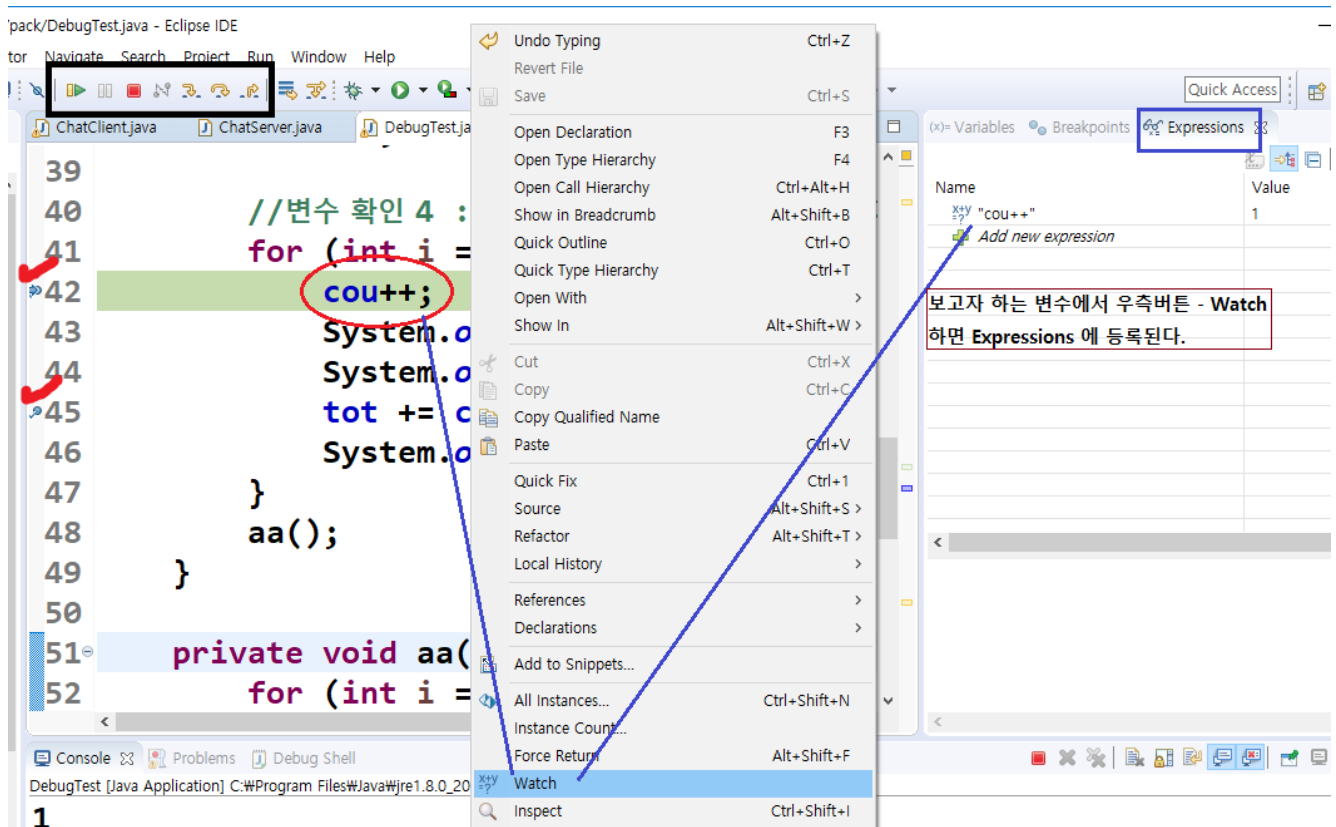


이 지점에서
"더블클릭"
(토글)

```
41 for (int i = 0; i < 5; i++) {
42     cou++;
43     System.out.println("aaa");
44     System.out.println("bbb");
45     tot += cou;
46     System.out.println("ccc");
47 }
48 aa();
49 }
```

```
public static void main(String[] args) {
    new DebugTest();
}
```

```
}
```



- 1) Skip All Breakpoints : 모든 브레이크 포인트 건너뛰
- 2) Resume(F8키) : 다음 브레이크포인트까지 진행함
- 3) Suspend : 스레드를 일시 정지하며 현재 수행문에 지정한 것과 같음
- 4) Terminate : 스레드 종료
- 6) Step Into(F5키) : 한단계 진행하는데 다음 라인이 함수 안이면 함수 안으로 들어감.
- 7) Step Over(F6키) : 함수 호출을 지나치고 현재 위치에서 한 단계 진행
- 8) Step Return(F7키) : 현재 함수 끝까지 바로 가서 리턴한 후 함수 호출부로 되돌아 감
- 9) Drop to Frame : 선택한 스택 프레임의 첫 행으로 이동. 처음부터 다시 하고자 할 때
- 10) Use Step Filters(Shift+F5) : 스텝 필터링

F5는 해당 메소드를 모두 수행하고 호출된 다음 메소드를 진행한다.

F6은 하나의 메소드를 수행하고 끝난다. 호출되는 메소드로 진행되지 않는다.

F7은 중단점이 있는 명령문만 수행한다.