##### 소수 찾기

Level 2

numberOfPrime 메소드는 정수 n을 매개변수로 입력받습니다.

1부터 입력받은 숫자 n 사이에 있는 소수의 개수를 반환하도록 numberOfPrime 메소드를 만들어 보세요.

소수는 1과 자기 자신으로만 나누어지는 수를 의미합니다.  
(1은 소수가 아닙니다.)

10을 입력받았다면, 1부터 10 사이의 소수는 [2,3,5,7] 4개가 존재하므로 4를 반환  
5를 입력받았다면, 1부터 5 사이의 소수는 [2,3,5] 3개가 존재하므로 3를 반환

class NumOfPrime {

int numberOfPrime(int n) {

int result = 1;

for(int i=3;i<=n;i++){

int cnt = 0;

for(int j=2;j<i;j++){

if(i%j==0){

cnt++;

}

}

if(cnt==0)

result++;

}

// 함수를 완성하세요.

return result;

}

public static void main(String[] args) {

NumOfPrime prime = new NumOfPrime();

System.out.println( prime.numberOfPrime(5) );

\*\*참조\*\*

**class** **NumOfPrime** {

**int** numberOfPrime(**int** n) {

**int** result = **0**;

**for**(**int** i=**2**; i<=n; i++){

**for**(**int** j=**2**; j<=i; j++){

**if**(j == i){

++result;

} **else** **if**(i%j == **0**){

**break**;

}

}

}

**return** result;

}

**public** **static** **void** main(**String[]** args) {

NumOfPrime prime = **new** NumOfPrime();

**System**.out.println( prime.numberOfPrime(**10**) );

}

}

}