
> 07. 조건문과 논리연산자

논리 연산자는 boolean값 true, false을 가지고 하는 연산이다.

논리연산자는 and,or,not이 있고 &&, ||, ! 으로 표기 한다.

&& and, || or 연산자는 두개의 boolean값을 가지고 하나의 boolean값을 만드는 일을 한다.

&& and연산자는 두 조건식이 모두 true일때 true를 생성한다.

|| or연산자는 두 조건식중 하나라도 true일때 true를 생성한다.

| 문자는 키보드의 엔터 위에 있다.

! not 조건식의 결과를 반전할때 사용한다. 조건식이 true이면 false, false이면 true가 된다.

if문에 2개의 조건식을 합쳐서 하나로 표현할 때 사용한다.

논리 연산자를 이용해 모든 입력과 출력을 만들어 놓은 표를 진리표라 이야기하고 아래에 진리표가 있으니 확인해 보자.

!은 true일 때 false, false일 때 true를 만들어 내는 not이라는 단항 연산자이고 3개중에 우선순위가 가장 높다. not and or 순으로 우선순위를 가진다.

중복된 여러개의 if문을 하나의 if문으로 쓰고자 할때 논리연산자를 사용한다.

만약 a가 4보다 크고 b가 3보다 크면 합격 아니면 불합격을 출력하는 프로그램을 구현 하려면 다음과 같이 구현하여야 한다.

```
if (a>4) { //2
    if (b>3) {
        System.out.println("합격");
    }
}
```

```

        } else {
            System.out.println("불합격");
        }
    } else {
        if (b>3) {
            System.out.println("불합격");
        } else {
            System.out.println("불합격");
        }
    }
}

```

상위 코드에서 불합격을 찍는 부분의 코드가 중복해서 사용 되었다. 이런 형태의 코드를 개선해서 중복을 없애려면 논리 연산자를 사용하여 $a > 4$ 와 $b > 3$ 이 모두 맞을때 출력 되도록 기술하려면 $a > 4 \&\& b > 3$ 을 하면 된다.

$\&\&$ 연산자는 앞과 뒤에 각각 true 또는 false 값을 가지는 두 개의 조건식을 연결하여, 두 조건식이 모두 true일 경우에만 결과 값으로 true를 반환하는 논리 연산자다. 그 외의 경우에는 false를 반환한다. 즉, $\&\&$ 연산자는 논리적인 AND 연산을 수행하는 연산자이다.

이 연산자를 이용해서 상위 코드를 다음과 같이 변경할 수 있다.

```

if (a>4&&b>3) {//출력값이 상위 코드와 동일하게 동작
    System.out.println("합격");
} else {
    System.out.println("불합격");
}

```

a가 10이고 b가 20이라면 $a > 4 \&\& b > 3$ 은 true $\&\&$ true가 되어 $\&\&$ 앞뒤에 true가 오니 최종 실행 결과가 true가 된다.

a가 10이고 b가 1이라면 $a > 4 \&\& b > 3$ 은 true $\&\&$ false되어 $\&\&$ 앞뒤가 모두 true가 아니므로 $\&\&$ 계산결과는 false가 된다.

$a > 4$ 이 true이고 $b > 3$ 이 true이면 $a > 4 \&\& b > 3$ 이 결과가 true이다. $a > 4$ 와 $a > 3$ 둘다 true일때 true이지 둘중 하나라도 true가 아니면 실행 결과는 false가 된다.

만약 $a > 4$ 와 $b > 3$ 중 하나만 만족하면 합격 아니면 불합격을 출력하는 프로그램을 구현하려면 다음과 같이 구현하여야 한다.

```

if (a>4) {//2
    if (b>3) {
        System.out.println("합격");
    } else {
        System.out.println("합격");
    }
} else {
    if (b>3) {

```

```

        System.out.println("합격");
    } else {
        System.out.println("불합격");
    }
}

```

상위 코드는 합격을 찍는 부분의 코드가 중복해서 사용 되었습니다. 이런 코드 중복을 제거하기 위해, $a > 4$ 또는 $b > 3$ 중 하나라도 맞을 때 합격 메시지를 출력 하도록 코드를 수정하려면 $a > 4 \ || \ b > 3$ 을 사용하면 됩니다. $\ ||$ 연산자는 앞과 뒤에 각각 true 또는 false 값을 가지는 두 개의 조건식을 연결하여, 두 조건식 중 하나라도 true일 경우 결과값으로 true를 반환하는 논리 연산자입니다. 둘 다 false일 경우에는 false를 반환합니다. 즉, $\ ||$ 연산자는 논리적인 OR 연산을 수행하는 연산자입니다.

이 연산자를 이용해서 상위 코드를 다음과 같이 변경할 수 있다.

```

if (a>4 || b>3) {
    System.out.println("합격");
} else {
    System.out.println("불합격");
}

```

논리 연산자를 좀더 자세히 살펴보자.

논리 연산자는 불리언 (Boolean)데이터들을 가지고 새로운 불리언 결과를 만들어 내는 연산자이다. 조건문과 나중에 배울 반복문에서 조건식을 2개 이상 사용하여 하나의 결과를 얻고자 할 때 논리 연산자를 사용한다. 다음 같은 경우가 조건식이 2개 있는 경우이다.

집에들어 가는 방법이 비밀번호와 key 두 가지가 있다.

비밀번호가 111이고 key는 108호 키이다.

이때 집에 들어갈 수 있는 방법은 비밀번호와 key가 둘다 맞아야 가능한 경우가 있고, 둘중에 하나만 맞아도 들어갈 수 있는 방법이 있을 것이다. 결과가 true일때 집에 들어갈 수 있다고 본다면 다음과 같이 생각할 수 있다.

둘 다 맞아야 열어 줄 경우 $\&\&$ 연산자, 둘 중에 하나만 맞아도 열어 줄 경우 $\ ||$ 연산자를 사용한다. 전자의 경우 수식은 비밀번호==1111 $\&\&$ key==108 이고 후자의 경우 수식은 비밀번호==1111 $\ ||$ key==108 이다. $\&\&$ (앤드) 연산자는 비교 대상이 둘 다 true 일 때만 결과가 true가 되고 $\ ||$ (오어) 연산자는 비교 대상 둘 중에 하나 라도 true가 존재하면 true를 결과로 남겨 준다.

둘다 맞아야 들어가는 경우 비밀번호==1111 $\&\&$ key==108 은 비밀번호는 1111이고 key는 108이어야 둘다 true가 되어 true $\&\&$ true == true 다음과 같이 연산결과가 true가 된다. 비밀번호가 123 이면 key가 108이든 1111이든 관계없이 비밀번호, key 둘다 true가 아니므로 연산 결과는 false가 된다.

둘중에 하나만 맞아도 들어가는 경우 비밀번호==1111 $\ ||$ key==108 의 경우 비밀번호가 1111 이고 key가 100이면 true $\ ||$ false 가 되어서 둘중에 하나라도 true가 있으면

true이므로 결과가 true가 되어 집에 들어갈 수 있다. 비밀번호가 112 이고 key가 123 이면 둘다 false여서 false || false == false 가되어 최종 결과가 false가 되어 집에 못들어 간다. 둘다 맞을 경우는 당연히 들어간다. 다음 코드를 확인해 보자.

```
int num=1111;
String key="108호";
//비밀번호가 111이고 key는 "108호"이다.
//1. 둘다 맞아야 들어갈 수 있음 &&
if(num==111&&key.equals("108호")) {//객체의 비교는 .equals를 사용한다.
    System.out.println("들어감");
}else {
    System.out.println("못들어감");
}
//2. 둘중에 하나만 맞으면 들어 갈 수 있음 ||
if(num==111||key.equals("108호")) {
    System.out.println("들어감");
}else {
    System.out.println("못들어감");
}
//3. 비밀번호만 맞으면 들어감
if(num==111) {
    System.out.println("들어감");
}else {
    System.out.println("못들어감");
}
//4. 키만 맞으면 들어감
if(key.equals("108호")) {
    System.out.println("들어감");
}else {
    System.out.println("못들어감");
}
```

다음은 진리표라고 하는 것이다. 불 연산은 연산 대상이 많지 않아 모든 연산 방법을 표로 만들어서 외우고 다니는 경우가 많다. 아래 표는 연산에 필요한 boolean자료형 a,b,c , 둘다 true일때만 true인 and연산자, 둘 중에 하나가 true면 true인 or연산자, true이면 false false이면 true 반전되는 not연산자를 가지고 연산한 결과표이다. 논리 연산자는 연산자 우선순위가 모두 다르다는 것을 조심하자.

!이 가장 높고 다음은 <,>,<=,>= 다음은 == 다음은 &&, 다음은 ||이다.

최상단 부분만 남겨놓고 나머지를 가린 다음 답을 맞출수 있을때 까지 공부해 보자.

| | | | | | | | | |
|---|---|------|------|----|---|---------|---------|---------|
| a | b | a&&b | a b | !a | c | a b c | a&&b&&c | a b&&c |
|---|---|------|------|----|---|---------|---------|---------|

| | | | | | | | | |
|-------|-------|-------|-------|-------|---|-------|-------|-------|
| true | true | true | true | false | f | true | false | true |
| true | false | false | true | false | f | true | false | true |
| false | true | false | true | true | f | true | false | false |
| false | false | false | false | true | f | false | false | false |

논리 연산자를 사용하여 아래표의 왼쪽을 보고 오른쪽을 구현 할 수 있도록 확인해 보자.

| 조건식 | 판별식 |
|------------------------|--------------------------------|
| 6보다 큰수 | a>6 |
| 두수 모두 100 이 넘는 경우 | a>100 && b>100 |
| 세수중 적어도 하나가 100이 넘는 경우 | a>100 b>100 c>100 |
| 부모 동의가 있거나 20살이 넘는 경우 | isState==true a>20 |
| 세수중 하나라도 음수가 있는 경우 | a<0 b<0 c<0 |
| 30과 40 사이 숫자 | a>=30 && a<=40 30<a<40 은 안됨 |
| 30과 40 사이를 제외한 모든 숫자 | !(a>=30 && a<=40) |
| 60보다 작거나 100보다 큰 숫자 | a<60 a>100 |

다음 문제는 2개 씩 뽑아서 각의 프로그램의 ?표부분에 적절한 내용을 적어서 동일하게 동작 하도록 만들어 보자.

1. 같은 문자열이 2번 출력 되도록 적절한 조건식을 ?부분에 넣어보자.

```
// 입력값에 따라 동일한 결과가 나오도록 두 조건식을 완성 하시오.
boolean isFlag1 = true or false, isFlag2 = true or false;// 입력값
```

```

if (isFlag1) {
    if (isFlag2) {
        System.out.println("맞음");
    } else {
        System.out.println("틀림");
    }
} else {
    if (isFlag2) {
        System.out.println("틀림");
    } else {
        System.out.println("틀림");
    }
}
if (?) {//적절한 조건식을 ?부분에 넣어보자.
    System.out.println("맞음");//출력값이 상위 코드와 동일하게 동작하도록
} else {
    System.out.println("틀림");
}

```

2. 같은 문자열이 2번 출력 되도록 적절한 조건식을 ?부분에 넣어보자.

```

boolean isFlag1 = true or false, isFlag2 = true or false;// 입력값
if (isFlag1) {
    if (isFlag2) {
        System.out.println("맞음");
    } else {
        System.out.println("맞음");
    }
} else {
    if (isFlag2) {
        System.out.println("맞음");
    } else {
        System.out.println("틀림");
    }
}
if (?) {// 적절한 조건식을 입력하시오
    System.out.println("맞음");//출력값이 상위 코드와 동일하게 동작하도록
} else {
    System.out.println("틀림");
}

```

3. 같은 문자열이 2번 출력 되도록 적절한 조건식을 ?부분에 넣어보자. !연산자가 필요하다.

```

boolean isFlag1 = true or false, isFlag2 = true or false;// 입력값
if (isFlag1) {
    if (isFlag2) {
        System.out.println("틀림");
    } else {

```

```

        System.out.println("틀림");
    }
} else {
    if (isFlag2) {
        System.out.println("틀림 ");
    } else {
        System.out.println("맞음");
    }
}
}
if (?) { // 적절한 조건식을 입력하시오
    System.out.println("맞음"); // 출력값이 상위 코드와 동일하게 동작하도록
} else {
    System.out.println("틀림");
}

```

4. 같은 문자열이 2번 출력 되도록 적절한 조건식을 ?부분에 넣어보자.

```

boolean isFlag1 = true or false, isFlag2 = true or false; // 입력값
if (isFlag1) {
    if (isFlag2) {
        System.out.println("틀림");
    } else {
        System.out.println("맞음");
    }
} else {
    if (isFlag2) {
        System.out.println("맞음 ");
    } else {
        System.out.println("맞음");
    }
}
}
if (?) { // 적절한 조건식을 입력하시오.
    System.out.println("맞음"); // 출력값이 상위 코드와 동일하게 동작하도록
} else {
    System.out.println("틀림");
}
}

```

- | | |
|------------------------|------------------------|
| 1. isFlag1&&isFlag2 | 2. isFlag1 isFlag2 |
| 3. !(isFlag1 isFlag2) | 4. !(isFlag1&&isFlag2) |

다음 문제를 풀어보자. 문자열의 비교는 .equals를 사용해야 한다. 복잡한 내용이 있는데 일단은 참조 데이터는 .equals로 비교한다고 생각하면된다. scanner를 이용해서 사용자 입력으로 얻어온 데이터는 모두 문자열이므로 .equal 를 사용해야 한다.

```
String str1 = "Hello"; String str2 = "Hello";
```

```

if (str1.equals(str2) == true) {

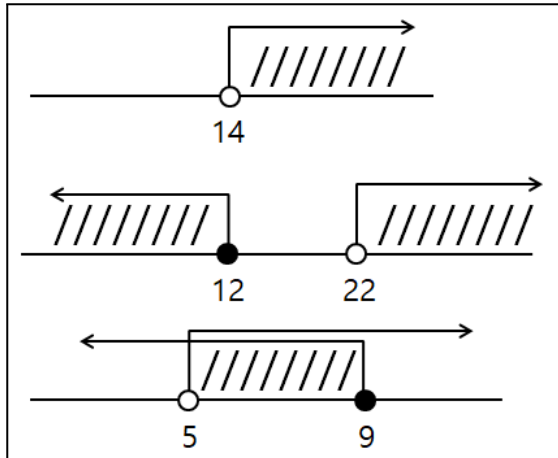
    System.out.println("str1과 str2는 같습니다.");

} else {

    System.out.println("str1과 str2는 다릅니다.");

}

```



1. 왼쪽 이미지에서 색칠한 부분의 값들이 들어 왔을때 true가출력 되는 조건식을 가진 if문을 각각 3개 만들어 색칠한 부분인지 아닌지 출력하는 프로그램 형태로 구현해 보자.

2. 왼쪽 이미지에서 색칠 안한 부분의 값들이 들어 왔을때 true가되는 조건식을 만들어 보자. not연산자는 사용방법과 not를 사용하지 않는 방법 각각 2개씩 총 6개 if문 수식에 넣어서 만들어 보자.

3. 입력받은 수가 3의 배수 이거나 7의 배수이면 ‘3또는 7의 배수’가 출력되고, 2의 배수도 되고 5의 배수도 되면 ‘2와 5의 배수’를 출력하고 두가지 경우 다 만족하면 2가지 다 출력하는 프로그램을 만들어 보자.

| | |
|------|------|
| 1번위치 | 2번위치 |
| 4번위치 | 3번위치 |

4. 왼쪽 이미지는 하나의 점이 x,y좌표를 가질때 위치할 수 있는 장소를 4개로 구분한 것이다. x,y가 모두 양수이면 2번 위치에 점이 존재하게 될것이고 모두 음수이면 4번위치에 점이 위치하게 될것이다. 해당 점의 좌표를 입력받아 해당 점이 어느 위치에 있는지 출력해 보자. x,y축 위에 있을 때는 x,y축

위에 있다고 출력하면 된다.

4. 버스요금계산하는 프로그램을 구현해 보자. 15세 이하는 1300원 초과는 1600원이고 버스카드로 결제하면 100원 할인해 준다.

문자열을 boolean형으로 변경하는 방법은 다음과 같다.

```
Boolean a= Boolean.parseBoolean("true");
```

입력: 16엔터 true엔터

출력: 버스요금은 1500원 입니다.

5. 다음을 구현하려면 어떤문을 사용해야 하는지 고민해 보고 사용자 입력을 받아 결과가 출력되도록 구현해 보자. (if elseif ifelse 논리연산자)

a. 밥을 먹었으면 ‘밥을 먹었음’ 이 출력 되고 밥을 먹지 않았으면 아무것도 출력되지 않는 형태의 프로그램구현.

b. 빵을 먹었으면 ‘빵을 먹었음’이 출력되고 빵을 먹지 않았으면 ‘밥을 먹었음’이 출력되는 형태의 프로그램 구현.

c. 밥을 먹었으면 ‘밥을 먹었음’, 빵을 먹었으면 ‘빵을 먹었음’ 아무것도 먹지 않았으면 ‘아무것도 먹지 않음’이 이 출력 되도록 프로그램 구현.

d. 식사로 ‘밥’과 ‘빵’이 있고 후식으로 국,우유, 아이스크림, 커피가 있는데 밥을 먹으면 국과 아이스크림중 하나를 빵을 먹으면 우유 커피 중 하나를 후식으로 먹을수 있도록 구현해 보자.

6. 다음과 같이 기술하면 오늘의 요일이 i에 숫자로 들어간다. i가 7일 경우 일요일이고 월화수목금토일은 1234567과 같다. i값을 가지고 오늘의 요일을 출력하는 프로그램을 만들어 보자.

//빨간줄에 마우스를 올리면 импорт 할 패키지가 뜬다.

//확인해서 해당 클래스를 클릭해서 자동으로 импорт하자.

```
LocalDateTime now = LocalDateTime.now();
```

```
System.out.println(now); //현재시간 출력
```

```
DayOfWeek dayOfWeek = now.getDayOfWeek();
```

//1 (월요일)부터 7 (일요일)까지의 값을 가집니다.

```
int i = dayOfWeek.getValue();
```

```
System.out.println(i);
```

7. 한국사이즈를 입력받아 미국사이즈로 출력하시오.

| 미국 SIZE | S | M | L | XL | XXL |
|---------|-------|--------|---------|---------|-------|
| 한국 SIZE | 90~95 | 95~100 | 100~105 | 105~120 | 110이상 |

8. java,html,db과목의 점수를 입력받아. 평균이 60점 이상이면 합격, 평균이 60점 미만이면 불합격, 40점 이하인 과목이 하나라도 있으면 과락이 출력되도록 프로그램을 구현해 보자. 최종 결과 화면은 다음과 같다. java 35 과락, html 10과락,db 20과락 또는 평균 40로 60점미만 불합격 또는 과락 없이 평균 60이상 합격이 출력된다.

| 초·중·고교생 | | | 성인 | | |
|-----------|-------|--|-----------|-------|----------------------------|
| | 분류 | 비고 | | 분류 | 비고 |
| 95 미만 | 체중 미달 | 120 미만을 모두 정상 체중으로 나타내기도 함 | 100 미만 | 체중 미달 | 체중 미달을 95 미만으로 보기도 함 |
| 95 ~ 120 | 정상 | | 100 ~ 110 | 정상 | |
| 120 ~ 130 | 경도 비만 | | 110 ~ 120 | 과체중 | |
| 130 ~ 150 | 중도 비만 | | 120 ~ 150 | 비만 | |
| 150 이상 | 고도 비만 | | 150 이상 | 고도 비만 | |

9. 왼쪽은 비만도 산출식이다. 사용자 입력을 받아 결과가 아래 표와 같은 분류가 나오도록 프로그램을 구현하여라.

$$\frac{y}{(x-100) \times 0.9} \times 100$$

10. 다음 표를 확인하여 사용자가 알고 있는 신체사이즈중 하나를 입력받아 해당 미국사이즈와 한국사이즈를 출력하시오. 겹치는 부분은 본인이 원하는 쪽으로 해석 해서 출력하자. 명확하지 않은 경계는 본인이 적절히 결정하자.

| 미국 SIZE | | S | M | L | XL | XXL |
|---------|--------|-------|--------|---------|---------|---------|
| 한국 SIZE | | 90~95 | 95~100 | 100~105 | 105~120 | 110이상 |
| 목둘레 | (cm) | 33~35 | 36~37 | 38~39 | 41~42 | 43~45 |
| 가슴둘레 | (cm) | 86~91 | 96~102 | 107~112 | 117~122 | 127~132 |
| 팔길이 | (cm) | 78~81 | 81~84 | 84~86 | 86~89 | 89~91 |
| 허리둘레 | (inch) | 28~29 | 30~31 | 32~34 | 34~37 | 38~40 |

11. 다음과 같이 실행되는 계산기 프로그램을 만들어보자.

첫번째수 입력>>5, 연산자선택 1. + 2. - 3. * 4. / >>3 두번째수 입력>>12

5*12=60 입니다.

12. 가위,바위,보 문자열중 하나를 랜덤하게 화면에 출력하는 프로그램을 구현해보자.하나를 내는 것 처럼 구현 하려면 0,1,2 3개의 수를 랜덤하게 리턴하게 만든 다음 가위,바위,보 3개에 1:1 매핑하면 된다.

13.가위 바위 보 게임을 만들어서 이겼는지 졌는지 출력하는 프로그램을 만들어 보자.

```
Random random = new Random();
```

```
int number = random.nextInt(10); // 0부터 9까지의 정수를 반환
```

- 매개변수를 통해 0 이상 10 미만의 정수를 반환할 수 있습니다.
- 예를 들어, `random.nextInt(10)`은 0부터 9까지의 랜덤한 정수를 반환합니다. (즉, 10상한값이 아니라 범위의 크기를 지정합니다.)
- 이 형태는 가장 많이 사용되며, 특정 범위의 양의 정수를 얻을 때 유용합니다.
- `int randomNum = random.nextInt(3) + 5; //5,6,7를 얻을 수 있다.`

14. 6~10사이의 랜덤한 수를 만들어 보자. 총 수에 범위는 6,7,8,9,10 5개 이므로 랜덤 함수에 5를 곱하고 시작을 6부터 하니 더하기 6를 하면 될 것이다. 제대로 동작하는지 확인해 보자.

15. 1~45까지의 수를 랜덤하게 만들어 3개의 수를 뽑은 다음 본인이 입력한 3개의 수와 몇개 같은지 확인하는 프로그램을 배열로 만들어 보자.

| 소득금액 | 세율 | 누진공제액 |
|----------------------|-----|------------|
| 1,200만원 이하 | 6% | 없음 |
| 1,200만원 ~ 4,600만원 이하 | 15% | 1,080,000 |
| 4,600만원 ~ 8,800만원 이하 | 24% | 5,220,000 |
| 8,800만원 ~ 1억 5천만원 이하 | 35% | 14,900,000 |
| 1억 5천만원 ~ 3억원 이하 | 38% | 19,400,000 |
| 3억원 ~ 5억원 이하 | 40% | 25,400,000 |
| 5억원 초과 | 42% | 35,400,000 |

16. 왼쪽표를 기본으로 본인이 낼 세금이 얼마인지 출력하는 프로그램을 구현해 보자. 만약에 본인이 번돈이 1600만원 이라면 1200만원은 세율이 6%이고 나머지 400만원은 세율이 15%이다.

문제 풀이

8.

```
package com.the.ex;

public class JavaAvg {
    public static final int cutLine=60;
    public static void main(String[] args) {
        // TODO Auto-generated method stub
        //java,html,db과목의 점수를 입력받아.
        //평균이 60점 이상이면 합격, 평균이 60점 미만이면 불합격, 40점 이하인
        과목이
        //하나라도 있으면 과락이 출력되도록 프로그램을 구현해 보자.
        //최종 결과 화면은 다음과 같다.
```

//java 35 과락, html 10과락,db 20과락 또는
//평균 40로 60점미만 불합격 또는 과락 없이 평균 60이상 합격이 출력된다.

```
int java=95;
```

```
int html=80;
```

```
int db=61;
```

```
int avg=(java+html+db)/3;
```

```
if(JavaAvg.cutLine<=java&&JavaAvg.cutLine<=html
```

```
&&JavaAvg.cutLine<=db&& JavaAvg.cutLine<=avg) {//
```

```
System.out.println("과락 없이 평균 "+JavaAvg.cutLine+"이상 합격
```

```
:"+avg);
```

```
}else {
```

```
// if(java<JavaAvg.cutLine) {//java 35 과락, html 10과락,db  
20과락 또는
```

```
// System.out.println("java :"+java+" 과락");
```

```
// }else if(html <JavaAvg.cutLine) {
```

```
// System.out.println("html :"+html+" 과락");
```

```
// }else if(db <JavaAvg.cutLine) {
```

```
// System.out.println("db :"+db+" 과락");
```

```
// }else {
```

```
// System.out.println(avg+"로 "+JavaAvg.cutLine+"점미만");
```

```
// }
```

```
if(java<JavaAvg.cutLine) {//java 35 과락, html 10과락,db
```

```
20과락 또는
```

```
System.out.println("java :"+java+" 과락");
```

```
}
```

```
if(html <JavaAvg.cutLine) {
```

```
System.out.println("html :"+html+" 과락");
```

```
}
```

```
if(db <JavaAvg.cutLine) {
```

```
System.out.println("db :"+db+" 과락");
```

```
}
```

```
if(avg<JavaAvg.cutLine){
```

```
System.out.println(avg+"로 "+JavaAvg.cutLine+"점미만");
```

```
}
```

```

        System.out.println("불합격");
    }
}

```

15.

```

package com.the.ex;
import java.util.Arrays;
import java.util.Scanner;
public class Java01 {
    public static void main(String[] args) {
        //15. 1~45까지의 수를 랜덤하게 만들어 3개의 수를 뽑은 다음
        //본인이 입력한 3개의 수와 몇개 같은지 확인하는 프로그램을 배열로 만들어

```

보자.

```

        int num[] =new int[3];
        int count=0;
        int inputNum=0;
        num[0]=(int)(Math.random()*45+1);
        num[1]=(int)(Math.random()*45+1);
        num[2]=(int)(Math.random()*45+1);

        Scanner scanner =new Scanner(System.in);
        System.out.println("숫자 입력>>");
        inputNum= scanner.nextInt();
        if(num[0]==inputNum) {
            count++;
        }
        if(num[1]==inputNum) {
            count++;
        }
        if(num[2]==inputNum) {
            count++;
        }

        System.out.println("숫자 입력>>");
        inputNum= scanner.nextInt();
        if(num[0]==inputNum) {
            count++;

```

```

    }
    if(num[1]==inputNum) {
        count++;
    }
    if(num[2]==inputNum) {
        count++;
    }

    System.out.println("숫자 입력>>");
    inputNum= scanner.nextInt();
    if(num[0]==inputNum) {
        count++;
    }
    if(num[1]==inputNum) {
        count++;
    }
    if(num[2]==inputNum) {
        count++;
    }

    System.out.println("배열내용:" + Arrays.toString(num));
    System.out.println("총 "+count+"개수 만큼 같은 값을 가진다.");
}
}

```