# [**디지털 컨버전스**] 스마트 콘텐츠와 웹 융합 응용SW 개발자 양성과정

강사:이상훈

학생 : 임초롱

## 5일차 수업내용 질문: 배열없이 중복 피하기

링크 https://github.com/limcholong/LectureContents/blob/main/java/CholongLim/Day5/src/NonDuplicateWithoutArrayTest.java

```
public class NonDuplicateWithoutArrayTest2 {
          public static void main(String[] args) {
              final int BIN = 1;
              int testBit = 0;
              int randNum;
              for (int i = 0; i < 10; i++) {
      5//
                   for (초기화 코드 ; 조건식 코드 ; 증감식 코드)
                   int i 값으로 위 세값을 구성하였지만, 아래엔 작성된 int i가 존재하지 않습니다.
10
      //
                   그럼에도 for문이 반복되어 실행하고 있습니다.
11
      //
                   (int)(Math.random() * 10) 가 0 ~ 9 값이어서 작동되는거겠죠?
12
                   하지만 이 값이 int i = randNum 이라고 하지 않았음에도 어떻게 실행되는 건가요?
13
                  randNum = (int)(Math.random() * 10);
14
15
16
                  while ((testBit & (BIN << randNum)) != 0) {</pre>
                     System.out.println("중복이 이렇게나 많이 발생합니다: " + randNum);
17
                     randNum = (int)(Math.random() * 10);
18
19
20
                 System.out.printf("randNum = %d\n", randNum);
21
                  testBit |= (BIN << randNum);</pre>
22
23
              System.out.println("testBit의 최종값은 1023이다. 진짜 ? " + testBit);
24
25
26
```

# 5일차 수업내용 복습: 배열없이 중복 피하기 - 1

링크 https://github.com/limcholong/LectureContents/blob/main/java/CholongLim/Day5/src/NonDuplicateWithoutArrayTest.java

```
public class NonDuplicateWithoutArrayTest2 {
2
           public static void main(String[] args) {
               final int BIN = 1;
               int testBit = 0;
               int randNum;
            1) for (int i = 0; i < 10; i++) {
11
                   randNum = (int)(Math.random() * 10);
12
                   while ((testBit & (BIN << randNum)) != 0) {</pre>
13
                       System.out.println("중복이 이렇게나 많이 발생합니다: " + randNum);
14
                       randNum = (int)(Math.random() * 10);
15
16
17
                   System.out.printf("randNum = %d\n", randNum);
18
19
                   testBit |= (BIN << randNum);
20
21
               System.out.println("testBit의 최종값은 1023이다. 진짜 ? " + testBit);
23
24
```

- ① for문 사이값이 참일때 안의 내용이 반복될 것이다. (i값이 0~9 사이 값일때 반복된다.)
- ② randNum의 값은 (int사용으로 정수) 0 ~ 9 중 랜덤이다.
- ③ while ((testBit & (BIN << randNum))!= 0) (testBit & (BIN << randNum))의 값이 0이 아닐 때 반복될 것이다.

randNum 값이 2일때.

( testBit & ( BIN << randNum)) → ( 0 & ( 1 << 2)) → ( 0 & ( 1 x 2^2)) → ( 0 & 4 ) → (2^0 & 2^2) 이며 2^0 = 0, 2^2 = 100 둘의 and 값은 0이므로 while 조건에 충족하지 못하여 반복문에서 벗어난다.

- ④ System.out.printf("randNum = %d\n", randNum)에서 randNum값이 2이기 때문에 randNum = 2 로 출력된다.
- ⑤ testBit |= (BIN << randNum) 는
  testBit = testBit | (BIN << randNum) 를 뜻한다.
  → 0 |= (1 << 2) → 0 |= (1 x 2^2) → 0 |= 4
  → BitOR연산 → 2^0 |= 2^2
  2^0 = 0, 2^2 = 100 둘의 or 값은 100, 십진수로 4이다.
  testBit 값은 4가 되고, for문이 다시 반복된다.

## 5일차 수업내용 복습: 배열없이 중복 피하기 - 2

링크 https://github.com/limcholong/LectureContents/blob/main/java/CholongLim/Day5/src/NonDuplicateWithoutArrayTest.java

```
1
       public class NonDuplicateWithoutArrayTest2 {
 2
           public static void main(String[] args) {
               final int BIN = 1;
               int testBit = 0;
               int randNum;
               for (int i = 0; i < 10; i++) {
                  randNum = (int)(Math.random() * 10);
11
12
             7-a while ((testBit & (BIN << randNum)) != 0) {
13
                       System.out.println("중복이 이렇게나 많이 발생합니다: " + randNum);
14
                       randNum = (int)(Math.random() * 10);
15
16
17
             7-b System.out.printf("randNum = %d\n", randNum);
18
19
                   testBit |= (BIN << randNum);
20
21
               System.out.println("testBit의 최종값은 1023이다. 진짜 ? " + testBit);
23
24
```

- ⑥ randNum의 값은 (int사용으로 정수) 0 ~ 9 중 랜덤이다.
   a) 앞부분과 똑같이 2가 나온다면?
   b) 앞부분과 똑같은 2가 아닌 3이 나온다면?
- ⑦ while ((testBit & (BIN << randNum))!= 0) (testBit & (BIN << randNum))의 값이 0이 아닐 때 반복될 것이다.
  - a) 앞의 결과로 testBit의 값은 4이며, randNum = 2일때, (testBit & (BIN << randNum)) → (4 & (1 << 2))
    - $\rightarrow$  (4 & (1 x 2^2))  $\rightarrow$  (4 & 4)  $\rightarrow$  (2^2+2^1 & 2^2)
    - → 100 & 100 이며 둘의 BitAnd 값은 100 이다.
    - → 즉, 4!= 0으로 4는 0이 아니다 라는 조건에 충족하여 중복이 이렇게나 많이 발생합니다 : 2 가 출력되며 randNum 값을 다시 뽑는다.
  - b) 앞의 결과로 testBit의 값은 4이며, randNum = 3일때, (testBit & (BIN << randNum)) → (4 & (1 << 3))
    - $\rightarrow$  (4 & (1 x 2^3))  $\rightarrow$  (4 & 8)  $\rightarrow$  (2^2 & 2^3)
    - → 100 & 1000 이며 둘의 BitAnd 값은 0 이다.
    - → 즉, 4!= 0으로 4는 0이 아니다 라는 조건에 충족하지 못하며, while 반복문에서 벗어나 randNum = 3을 출력하고, testBit |= (BIN << randNum) 은 testBit = testBit | (BIN << randNum) → testBit = 4 | (1 << 3 ) → 4 | (1 x 2^3) → 4
      - → testBit =  $4 \mid (1 << 3)$  →  $4 \mid (1 \times 2^3)$  →  $4 \mid (1 \times 8)$
      - → 4 | 8 → 100 | 1000 → 1100 → 12으로 값이 변한다.

# 23번 : 중복 방지 문제 (Challenge) - 1유형

링크 https://github.com/limcholong/LectureContents/blob/main/java/CholongLim/Day5/src/Quiz23.java

```
public class Quiz23 {
          public static void main(String[] args) {
2
    1
              final int BIN = 1;
              int testBit1 = 0;
8
              int randNum1;
              int randNum2;
9
              for(int i = 1; i < 11; i++) {
                  randNum1 = ((int)(Math.random() * 6 + 5));
                  randNum2 = ((int)(Math.random() * 4 + 7));
                  while((testBit1 & (BIN << randNum1)) != 0) {</pre>
                      randNum1 = ((int)(Math.random() * 6 + 5));
                      randNum2 = ((int)(Math.random() * 4 + 7));
                  if(randNum1 != randNum2) {
                      System.out.printf("randNum1 = %d, randNum2 = %d\n", randNum1, randNum2);
24
                      testBit1 |= (BIN << randNum1);
                              "C:\Program Files\Java\jdk-15.0
                             randNum1 = 10, randNum2 = 8
28
                             randNum1 = 8, randNum2 = 9
                             randNum1 = 6, randNum2 = 8
                             randNum1 = 9, randNum2 = 10
```

randNum1 = 5, randNum2 = 10
randNum1 = 7, randNum2 = 10

#### 23번 문제 내용 :

2개의 범위를 가지는 난수 2개를 중복없이 제어해보자! 하나는 5 ~ 10의 범위를 가지고 다른 하나는 7 ~ 10의 범위를 가진다. ex) 5, 6, 7, 8, 9, 10이 모두 출력되어야 하고 또한 7, 8, 9, 10이 출력되어야 한다. 여기서 중복이 발생하면 안됨.

문제에서 언급하는 중복이 ① ② 중 어떤 것인지 모르겠어서 두 가지 경우를 모두 해보았습니다.

randNum1 = 5,6,7,8,9,10 (\_6가지 수) randNum2 = 7,8,9,10 (\_4가지 수)

- ① <u>randNum1의 5,6,7,8,9,10은 모두 출력이 되고,</u> <u>randNum1 값과 randNum2 값이 중복되면 안된다.</u> (결과: randNum1은 6가지수여서 모두 출력하다 보면, randNum2가 중복된다.
- ② randNum1, randNum2값이 각각 중복되면 안되고, randNum1 값과 randNum2 값이 중복되면 안된다.

## 23번 : 중복 방지 문제 (Challenge) - 2유형

링크https://github.com/limcholong/LectureContents/blob/main/java/CholongLim/Day5/src/Quiz23\_2.java

```
public class Quiz23 {
           public static void main(String[] args) {
2
               final int BIN = 1;
               int testBit1 = 0 ;
               int testBit2 = 0;
               int randNum1;
               int randNum2;
               for(int i = 1; i < 11; i++) {
                   randNum1 = ((int)(Math.random() * 6 + 5));
15
                   randNum2 = ((int)(Math.random() * 4 + 7));
                   while((testBit1 & (BIN << randNum1)) != 0) {</pre>
                       randNum1 = ((int)(Math.random() * 6 + 5));
18
                   while((testBit2 & (BIN << randNum2)) != 0) {</pre>
                       randNum2 = ((int)(Math.random() * 4 + 7));
                   if(randNum1 != randNum2) {
                       System.out.printf("randNum1 = %d, randNum2 = %d\n", randNum1, randNum2);
                       testBit1 |= (BIN << randNum1);
                       testBit2 |= (BIN << randNum2);
 Quiz23
    "C:\Program Files\Java\jdk-15.0.2\bin\java.exe" -javaagent:C:\Users\user\AppData\Local\JetBra
    randNum1 = 6, randNum2 = 9
    randNum1 = 10, randNum2 = 8
   randNum1 = 9, randNum2 = 10
    randNum1 = 8, randNum2 = 7
```

#### 23번 문제 내용 :

2개의 범위를 가지는 난수 2개를 중복없이 제어해보자! 하나는 5 ~ 10의 범위를 가지고 다른 하나는 7 ~ 10의 범위를 가진다. ex) 5, 6, 7, 8, 9, 10이 모두 출력되어야 하고 또한 7, 8, 9, 10이 출력되어야 한다. 여기서 중복이 발생하면 안됨.

문제에서 언급하는 중복이 ① ② 중 어떤 것인지 모르겠어서 두 가지 경우를 모두 해보았습니다.

randNum1 = 5,6,7,8,9,10 (\_6가지 수) randNum2 = 7,8,9,10 (\_4가지 수)

- ① randNum1의 5,6,7,8,9,10은 모두 출력이 되고, randNum1 값과 randNum2 값이 중복되면 안된다.
- ② randNum1, randNum2값이 각각 중복되면 안되고, randNum1 값과 randNum2 값이 중복되면 안된다.
   (결과: randNum1의 값도 randNum2 값도 중복되지 않지만, 중복을 피하기 위해 randNum2 값의 4가지까지 출력이 되어 randNum1 값은 몇몇 숫자는 나오지 않을 수 있다.)