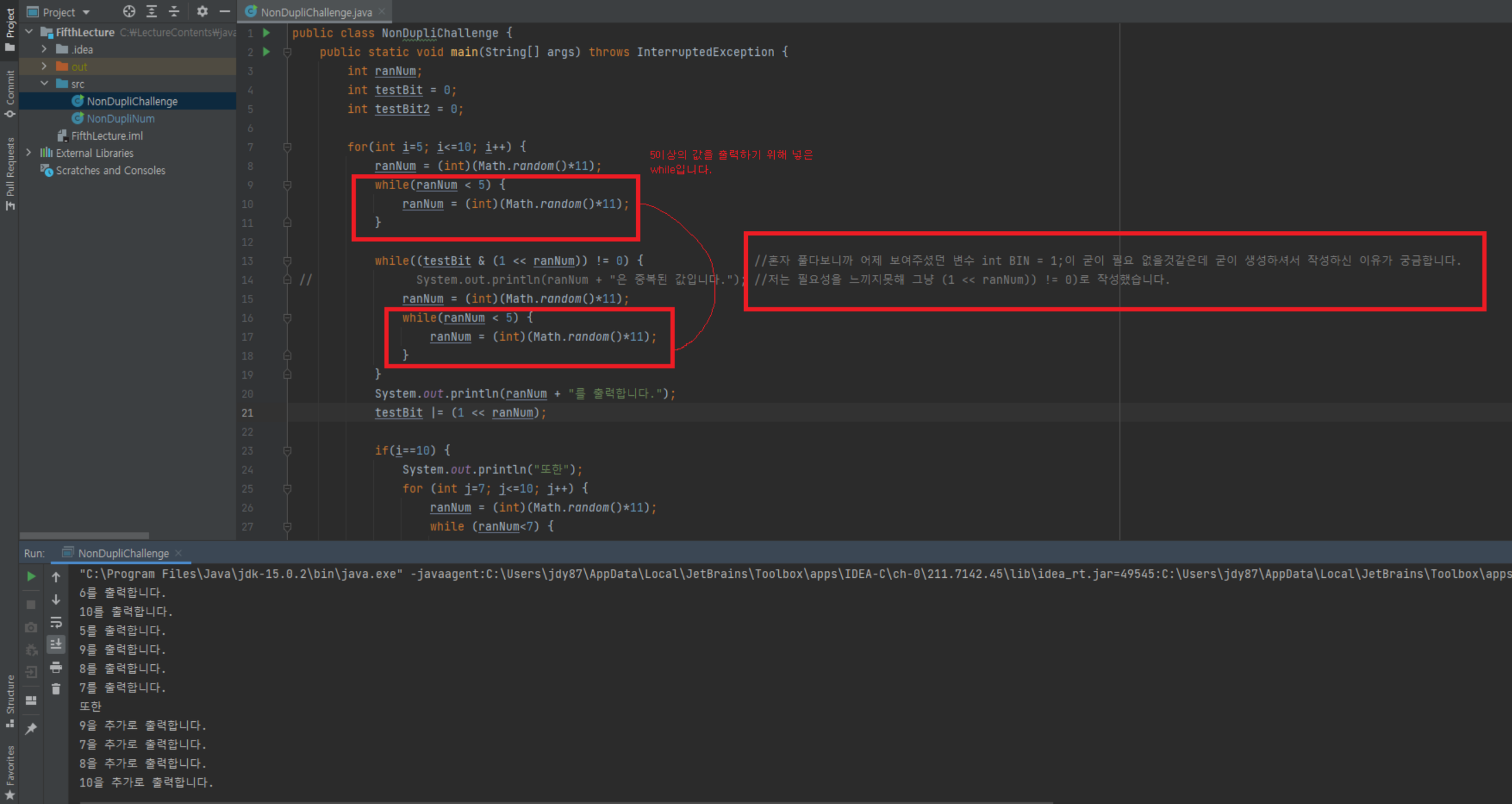
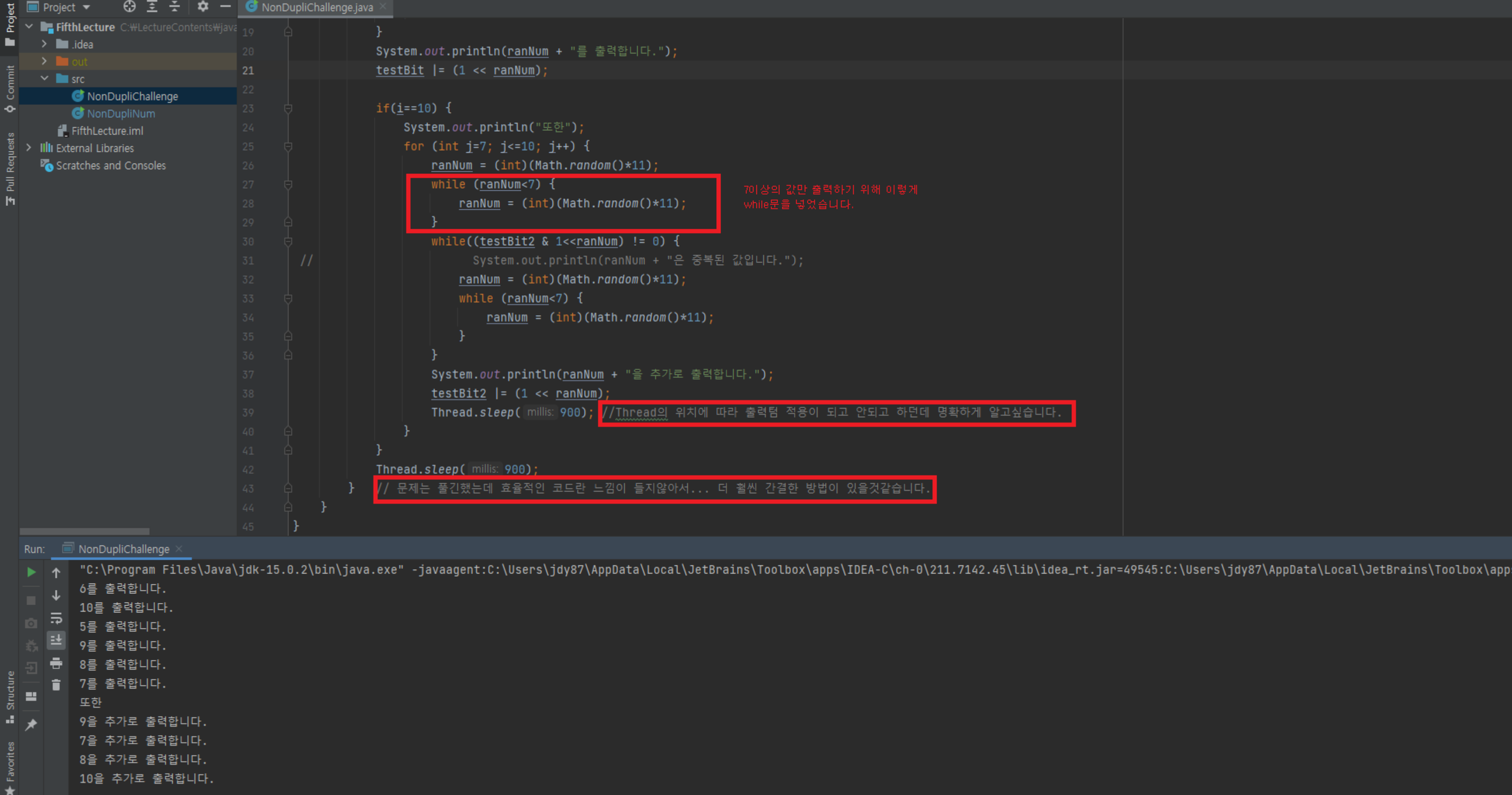
```
public class NonDupliNum {
    public static void main(String[] args) throws InterruptedException {
        int testBit = 0;
        int random;
        int bin = 1;

        for(int i=0; i<10; i++) {
            random = (int)(Math.random()*10);

            while((testBit & (bin << random)) != 0) {
//                System.out.println(random + "은 중복입니다!");
                random = (int)(Math.random()*10);
            }
            System.out.println(random + "을 출력");
            testBit |= (bin << random);

        Thread.sleep( millis: 800);
        }
//        System.out.println(19 & 10); //10011 & 1010 ===> 00010 (2)
//        System.out.println(19 | 10); //10011 | 1010 ===> 11011 (27)
        // 워낙 헷갈리는 개념이다보니 좀 안쓰면 바로 까먹을것같은데 기억하는 팁이나 주로 쓰이는 분야?가 궁금하네요.
    }
}
```

```
Run:  NonDupliNum
"C:\Program Files\Java\jdk-15.0.2\bin\java.exe" -javaagent:C:\Users\jdy87\AppData\Local\JetBrains\Toolbox\apps\IDEA-C\ch-0\211.7142.45\lib\idea_rt.jar=
8을 출력
7을 출력
1을 출력
5을 출력
3을 출력
2을 출력
6을 출력
0을 출력
9을 출력
4을 출력
```


Project
FifthLecture C:₩LectureContents₩java
.idea
out
src
NonDupliChallenge
NonDupliNum
FifthLecture.iml
External Libraries
Scratches and Consoles
NonDupliChallenge.java
public class NonDupliChallenge {
public static void main(String[] args) throws InterruptedException {
int ranNum;
int testBit = 0;
int testBit2 = 0;
for(int i=5; i<=10; i++) {
ranNum = (int)(Math.random()*11);
while(ranNum < 5) {
ranNum = (int)(Math.random()*11);
}
while((testBit & (1 << ranNum)) != 0) {
// System.out.println(ranNum + "은 중복된 값입니다.");
ranNum = (int)(Math.random()*11);
while(ranNum < 5) {
ranNum = (int)(Math.random()*11);
}
}
System.out.println(ranNum + "를 출력합니다.");
testBit |= (1 << ranNum);
if(i==10) {
System.out.println("또한");
for (int j=7; j<=10; j++) {
ranNum = (int)(Math.random()*11);
while (ranNum<7) {
50이상의 값을 출력하기 위해 넣은
while입니다.
//혼자 풀다보니까 어제 보여주셨던 변수 int BIN = 1;이 굳이 필요 없을것같은데 굳이 생성하셔서 작성하신 이유가 궁금합니다.
//저는 필요성을 느끼지못해 그냥 (1 << ranNum)) != 0)로 작성했습니다.
Run:
NonDupliChallenge
"C:\Program Files\Java\jdk-15.0.2\bin\java.exe" -javaagent:C:\Users\jdy87\AppData\Local\JetBrains\Toolbox\apps\IDEA-C\ch-0\211.7142.45\lib\idea_rt.jar=49545:C:\Users\jdy87\AppData\Local\JetBrains\Toolbox\apps
6를 출력합니다.
10를 출력합니다.
5를 출력합니다.
9를 출력합니다.
8를 출력합니다.
7를 출력합니다.
또한
9을 추가로 출력합니다.
7을 추가로 출력합니다.
8을 추가로 출력합니다.
10을 추가로 출력합니다.


Project
FifthLecture C:₩LectureContents₩java
.idea
out
src
NonDupliChallenge
NonDupliNum
FifthLecture.iml
External Libraries
Scratches and Consoles
NonDupliChallenge.java
}
System.out.println(ranNum + "를 출력합니다.");
testBit |= (1 << ranNum);
if(i==10) {
System.out.println("또한");
for (int j=7; j<=10; j++) {
ranNum = (int)(Math.random()*11);
while (ranNum<7) {
ranNum = (int)(Math.random()*11);
}
while((testBit2 & 1<<ranNum) != 0) {
// System.out.println(ranNum + "은 중복된 값입니다.");
ranNum = (int)(Math.random()*11);
while (ranNum<7) {
ranNum = (int)(Math.random()*11);
}
}
System.out.println(ranNum + "을 추가로 출력합니다.");
testBit2 |= (1 << ranNum);
Thread.sleep(millis: 900); //Thread의 위치에 따라 출력텀 적용이 되고 안되고 하던데 명확하게 알고싶습니다.
}
}
Thread.sleep(millis: 900);
} // 문제는 풀긴했는데 효율적인 코드란 느낌이 들지않아서... 더 훨씬 간결한 방법이 있을것같습니다.
}
}
7이상의 값만 출력하기 위해 이렇게
while문을 넣었습니다.
Run: NonDupliChallenge
"C:\Program Files\Java\jdk-15.0.2\bin\java.exe" -javaagent:C:\Users\jdy87\AppData\Local\JetBrains\Toolbox\apps\IDEA-C\ch-0\211.7142.45\lib\idea_rt.jar=49545:C:\Users\jdy87\AppData\Local\JetBrains\Toolbox\app
6를 출력합니다.
10를 출력합니다.
5를 출력합니다.
9를 출력합니다.
8를 출력합니다.
7를 출력합니다.
또한
9을 추가로 출력합니다.
7을 추가로 출력합니다.
8을 추가로 출력합니다.
10을 추가로 출력합니다.