

디지털 컨버전스 기반 UXUI Pront 전문 개발자 양성과정

강사 이상훈

gccccompil3r@gmail.com

학생 김도혜

kimdohye0728@gmail.com

질문 없음.

다만 22번은 이해는 했으나 내가 직접 코드를 짜기 어려울 것으로
보인다.

23은 아직 손댈 수 없음.

18.십진수 21을 이진수로 표현해보자!

--->10101

19.십진수 73을 이진수로 표현해보자!

--->1010101

20. $2^1 + 2^3 + 2^5 + 2^6$ 을 2진수로 표현해보자!

--->1101010

21. $2^0 + 2^3 + 2^6 + 2^7 + 2^8 + 2^9$ 을 2진수로 표현해보자!

--->1111001001

22.배열 없이 중복 회피하기 예제 복습

```
public class Review {
    public static void main(String[] args) throws InterruptedException {
        final int BIN = 1;
        int testBit = 0;
        //처음 값만 0이다. 밑에서 새로 치환됨.
        int randNum;
        //애는 지금 값이 주어지지 않았음, 밑에 for문에서 주어짐.

        for (int i = 0; i < 10; i++) {
            randNum = (int)(Math.random() * 10);
            //Math.random()는 0.0이상~10.0미만을 랜덤출력하는데 10을 곱했기 때문에
            //0~9의 숫자가 randNum에 대입될 것이다.
            while ((testBit & (BIN << randNum)) != 0) {
                //testBit와 BIN(1) * 2^random의 값을 비교해서(비교연산자AND) !=0이라면 중복을 출력하고
                //만약 아니라면 밑에 randNum = (int)(Math.random()*10);으로 가서 숫자를 새로 뽑음.
                //새로 뽑은 숫자랑 또 비교해서 !=0이라면 중복 출력하고 만약 0값이 나온다면 while문 벗어남.
                System.out.println("중복이 일어거나 많이 발생합니다: " + randNum);
                randNum = (int)(Math.random() * 10);
            }

            System.out.printf("randNum = %d\n", randNum);
            testBit |= (BIN << randNum);
            //|는 비트연산자의 OR를 말하는 것으로 하나만 1이어도 1로 출력하는 것이고, =있으니까 결과값을 비교하고
            // testBit에까지도 대입까지 하는 것.
        }

        System.out.println("testBit의 최종값은 1023이다. 진짜 ? " + testBit);
        //testBit의 최종 마지막 값은 9가 될 것이므로 2(n+1)-1값이 1023이다.

        Thread.sleep( millis: 500 );
    }
}
```