

# 문제은행 [3] 복습풀이+질문

12/20 개강 SW개발자 양성과정 [손현지]

2021. 12. 31

# <문제은행[3] - 1번, Answer1 보며 다시 풀이하기>

## 기존의 코드

```
import java.util.Scanner;

public class Practice1 {
    public static void main(String[] args) {
        //문제1. 아래와 같은 등비 수열이 있다.
        // 1, 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024, ...
        // 사용자 입력을 통해 원하는 위치의 값을 뽑아내도록 프로그래밍 해보자!
        //(1~32번째 혹은 31번째 항까지만 올바른 결과가 나올 것임)

        int[] arr;

        //1. import Scanner를 사용해 내가 구할 위치를 입력할 수 있는 문장이 출력되도록 한다.
        Scanner scan = new Scanner(System.in);
        System.out.print("몇 번째 항의 결과를 찾고 싶나요? : ");
        int end = scan.nextInt(); // 정수를 입력할 수 있음.

        //2. 매 항마다 숫자가 x2씩 커지는 배열을 만들어준다.
        //2-1. 이 때, 1항의 숫자는 1이므로 arr[0] = 1;은 직접 입력해준다.

        arr = new int[end];
        arr[0] = 1;

        //2-2. for문을 이용하여 반복문이 내가 입력하는 수 만큼만 arr를 배열하도록 만든다.

        int num = 1;

        for(int i = 1; i < end; i++){
            arr[i] = num * 2;
            System.out.printf("arr[%d] = %d\n", i, arr[i]);

            num = arr[i]; //for 문 맨 위로 돌아가 반복하기 전, num의 값을 바꿔준다.
        }

        //꼭 31번째 항(0포함 32번째)부터 -가 붙어서 나옵니다.
    }
}
```

## 복습 결과

```
//복습 포인트: 1. final int 사용해보기!
//                2. 1항도 검색하면 나올 수 있도록 해보기!

//31번째 항 부리는 숫자가 제대로 나오지 않으므로
final int MAX = 31;
final int START = 0;

int[] arr;

//1. import Scanner를 사용해 내가 구할 위치를 입력할 수 있는 문장이 출력되도록 한다.
Scanner scan = new Scanner(System.in);
System.out.print("몇 번째 항의 결과를 찾고 싶나요? : ");
int end = scan.nextInt(); // 정수를 입력할 수 있음.

//2. 매 항마다 숫자가 x2씩 커지는 배열을 만들어준다.
//2-1. 이 때, 1항의 숫자는 1이므로 arr[0] = 1;은 직접 입력해준다.

arr = new int[end];

//2-2. for문을 이용하여 반복문이 내가 입력하는 수 만큼만 arr를 배열하도록 만든다.

if(end > MAX){ //찾고자 하는 항이 MAX보다 큰 경우
    System.out.println("출력값이 너무 커 계산할 수 없습니다!");
} else{
    for(int i = 0; i < end; i++){
        arr[i] = (int) Math.pow(2, i); //1에 2를 제곱해준다.
        System.out.printf("arr[%d] = %d\n", i, arr[i]);
    }

    //배열이니까 따로 값을 백업해줄 변수가 없어도 arr[i]의 값이 알아서 백업이 되는거구나?
}

//꼭 31번째 항(0포함 32번째)부터 -가 붙어서 나옵니다.
}
```

0항에서 1의 값은 0인데 제공하면 1이 나오는 이유...  
실제로 java 내에서 0은 딱 떨어지는 숫자가 아니라 0.nnnnn인 건가요?

# 〈문제은행[3] - 3번, Answer3 보며 다시 풀이하기(1)〉

## 기존의 코드1

```
//배열로 모든 문제를 만들어보기!
// 100명 중 5명이 당첨되도록 만들어보자!
// 배열길에 당첨되는 자리를 배치해놓고 사용자가 들어서 당첨되는지 안되는지를 판정하도록 한다.

//2. 매 할마다 숫자가 +1씩 커지는 배열을 만들어준다.
final int START = 1;
int[] arr;

//1. import Scanner를 사용해 사용자가 1~100중 원하는 숫자를 입력할 수 있도록 한다.
Scanner scan = new Scanner(System.in);

System.out.print("1~100 중 숫자 1개를 골라보세요! : ");
int end = scan.nextInt();
arr = new int[end];

boolean defense = false;
arr[0] = 1;

//3. for문을 이용해 arr[99](100번)번째 배열이 나올때 까지 반복되는 반복문을 만들어준다.
for(int i = START; i < end; i++){
    arr[i] = i + 1;

    //3-1. 조건문 if를 사용해 11, 33, 55, 77, 99 (5개)의 숫자를 사용자가 입력한 경우,
    // = 당첨되었습니다. 라는 문구가 출력되도록 한다.
    if(i == 10 || i == 32 || i == 54 || i == 76 || i == 98){
        System.out.printf("arr[%d] = 당첨입니다!\n", arr[i]);
        defense = false;
    } else {
        //3-2. 그 외의 숫자는 else를 사용해 = 참! 이라는 문구가 출력되도록 한다.
        System.out.printf("arr[%d] = 참!\n", arr[i]);
        defense = false;
    }
}

}
```

//문제점. 키보드로 입력한 숫자에 도달할 때 까지 결과가 모두 출력됨.

## 기존의 코드2

```
//1. 매 할마다 숫자가 +1씩 커지는 배열을 만들어준다.
final int START = 1;
final int MAX = 100;
int[] arr = new int[MAX];

int answer = 0;

//2. for문을 이용해 arr[99](100번)번째 배열이 나올때 까지 반복되는 반복문을 만들어준다.
for(int i = START; i < MAX; i++) {
    arr[i] = i + 1;

    answer = arr[i]; //2-1. 루프마다 arr[i] 값의 int answer를 넘겨줌
}

//3. import Scanner를 사용해 사용자가 1~100중 원하는 숫자를 입력할 수 있도록 한다.
Scanner scan = new Scanner(System.in);

System.out.print("1~100 중 숫자 1개를 골라보세요! : ");
int choice = scan.nextInt();

arr[0] = 1;

//3. int answer의 값이 11, 33, 55, 77, 88 이라면 당첨 / 아니라면 참 이 출력되도록 한다.
if(choice == 11 || choice == 33 || choice == 55 || choice == 77 || choice == 99){
    System.out.printf("arr[%d] = 당첨입니다!\n", choice);
} else {
    //3-2. 그 외의 숫자는 else를 사용해 = 참! 이라는 문구가 출력되도록 한다.
    System.out.printf("arr[%d] = 참!\n", choice);
}
```

//배열을 만들었으나 활용하지 못하는 형태가 됨

배열을 만들 때 boolean을 사용하는 방법을 몰라서 해매었으므로  
복습할 때 이 부분에 유의하여 다시 살펴보았습니다.  
또한, Answer3의 구조를 이해하여 활용해보았습니다.

# <문제은행[3] - 3번, Answer3 보며 다시 풀이하기(1)>

## 복습 결과

```
final int TOTAL = 10;
final int CHOOSE = 5;

boolean[] lottoBox = new boolean[TOTAL]; //로또박스 [배열숫자] = true/false로 값이 정해질 수 있는 배열
int[] chooseNum = new int[CHOOSE]; // 선택된번호 [배열숫자] = 중복 검사에 사용할 배열

System.out.println("당첨 번호를 발표합니다.");

boolean dupCheck = true; // 중복 검사용

int lottoNum = 0; // 랜덤 로또 번호
int dupNum = 0; // 중복 검사할 때 번호를 백업해줄 변수

for (int i = 0; i < CHOOSE; i++) { //1가 0~4가 될 때 까지, 총 5번 돌아가는 반복문
    while (dupCheck) {
        lottoNum = (int) (Math.random() * TOTAL);

        dupCheck = false;

        for (int j = 0; j < dupNum; j++) {
            if (chooseNum[j] == lottoNum) {
                System.out.println("중복 발생!");
                dupCheck = true;
                break;
            }
        }

        lottoBox[lottoNum] = true;

        //여기에서부터 중복 발생 여부를 체크하는 루틴을 만들어준다!
        chooseNum[dupNum++] = lottoNum; //로또번호의 값을 dupNum(중복숫자)+1에 백업해준다.
        // +1 해주는 이유: 반복문 3 for(int j = 0; j < dupNum; j++)에 걸리지 않기 위해서

        System.out.println("lottoBox[" + lottoNum + "] = " + lottoBox[lottoNum]);

        dupCheck = true; // boolean값을 다시 true로 바꿔주어 반복문 1 처음으로 돌아가게 함
    }
}
```

방어루틴이 어떤 식으로 작용되는지,  
중복 발생 여부를 체크하는 루틴의 구조,  
↑ 두 가지를 이해하는 것에 중점을 두고 문제를 풀었습니다.

풀이를 하면서 방어루틴이 어떻게 사용되는지는 이제 이해가 되는데  
아직 혼자 코드를 짜면서 방어루틴을 잘 활용하는 것은 조금 어려울  
것 같아서  
연습이 더 필요할 것 같습니다.

lottoNum의 값을 chooseNum[dupNum]에 백업해준다.  
지금 백업한 값이 백업되므로써, 다음에 반복문 3까지 갈  
때에는  
lottoNum이 아니라 chooseNum으로 들어가게 되어 중복이  
발생하는지 확인할 수 있게 되는 것

이때 +1을 더하여 백업하도록 한다.  
반복문 3으로 돌아갔을 때  
for(int j = 0; j < dupNum; j++)  
// j는 dupNum보다 작아야 반복문이 작동한다는 조건을  
만족할 수 있도록!

ex) 3이라는 숫자가 dupNum에 저장되었다.  
+1을 해주지 않으면 j < dupNum 이라는 조건을 통과할 수  
있는 수는 (0,1,2)뿐이기 때문에  
중복검사가 제대로 작동되지 않는다.

# <문제은행[3] - 8번, Answer8 보며 다시 풀이하기(1)>

## 선생님 답 풀이 중 의문점(1)

```
// 각 플레이어들의 주사위 값을 비교해서 특수 주사위를 굴릴지 여부 판정
for (int i = 0; i < PLAYER_NUM; i++) {

    if (diceSum[i] % 2 == 0) { // diceSum[i] 값이 짝수인 경우만 if문 조건을 만족한다.
        dice = (int) (Math.random() * range + MIN);

        if (dice == SKILL_NUM1) {
            System.out.println("상대방 주사위 눈금을 2 뺄군다.");

            for (int j = 0; j < PLAYER_NUM; j++) {
                if (j == i) {
                    continue; // skip의 의미임
                }
                diceSum[j] -= 2; //
            }
        }
        else if (dice == SKILL_NUM2) {
            System.out.println("모두 함께 자폭 ^^ ->");

            for (int j = 0; j < PLAYER_NUM; j++) {
                diceSum[j] -= 0;
            }
        }
    }
}
```

j가 i와 같으면 스킵하라는 말을...  
SKILL\_NUM1을 발동시킨 플레이어의 주사위 합에도 -2되면 안되니까  
continue로 처리하는 건가요?

diceSum[j]안의 j가 어떤 식으로 작동하는 것인지 아직 이해가 잘 가지 않습니다.  
처음으로 옥면체주사위 2개를 굴려서 주사위값 두 개를 더한 변수는  
dice[i], dice[i+1]위치에 저장된 것 아닌가요?  
어떻게 주사위를 처음 굴렸을 때 쓴 적이 없는 dice[j]에 -=2가 적용될 수 있는 건지 모르겠어요.  
dice[i]였던 것이 왜 조건식을 특수주사위를 굴리지 못한 나머지 dice[j]로 처리되는 걸까요?

그리고 여기에선 왜 변수 두 개가 다 적용되는 거지... dice[i] = dice[j]라고 볼 수 있는 걸까요?  
dice 배열인 게 중요한 거고 인덱스 인지는 항목 번호를 찾아주는 것 뿐이라 상관이 없는건가요?  
변수 이름은 고유한 거니까 이유를 잘 모르겠어요...  
똑같은 내용물을 잠시 다른 박스에 담았을 뿐이라고 생각하면 되는 걸까요?

코드 전문을 보셔야 확인이 더 편하실텐데 8번 문제 코드가 길어서  
한 페이지에 들어가질 않아서 잘랐습니다 πππ

continue란? 반복문 끝으로 이동하여 다음 반복으로 넘어가게 해주는 역할

## 선생님 답 풀이 중 의문점(2) 특수 주사위 돌아간 이후

```
// 음수 처리
for (int i = 0; i < PLAYER_NUM; i++) {
    if (diceSum[i] < 0) {
        diceSum[i] = 0; // int의 음수를 표현하지 못하는 것은 아니지만 음이 0으로 변경해주는 이유가 궁금해요!
    }
}
```

→ int도 음수값을 표현할 수 있을텐데  
왜 음수인 숫자들을 0으로 변경해주는 건가요??

〈문제은행[3] - 8번, Answer8 보며 다시 풀이하기(2)〉

## 복습 도중 발생한 의문점

```
if(firstDiceSum[i] % 2 == 0){
```

```
for(int i = 0; i < PLAYER; i++){
```

}

## 처음 구상했던 구조

처음으로 굴린 육면체주사위 2개 값을 더한 결과가

**조건문1** 짝수여야만 들어갈 수 있는 조건식을 만든다

육면체 주사위 1개를 다시 굴린다.

if / else if / else문으로  
특수 주사위 스킬을 작동시킨다.

그런데 for문 안에 있는 if문 조건에는 배열 인덱스를 넣을 수 있는데 반복문 밖에서는 안되네요.

위에서 이미 firstDiceSum[i](플레이어1)과 [i+1]플레이어2의 주사위 값 결과가 배열에 저장되어서 사용할 수 있을 거라고 생각했는데 왜 반복문 밖에 안되는 걸까요?

```

C:\Users\Administrator\jdk\azul-11.0.13\bin>java.exe -javaagent:C:\
1회차. 플레이어[1]의 주사위 값은 = 10
2회차. 플레이어[2]의 주사위 값은 = 5
짝수가 나온 플레이어[1]는 육면체 주사위를 한 번 더 굴릴 수 있습니다!
2회차. 플레이어[1]의 주사위 값은 = 10
특수스킬4. 플레이어[1]의 가 모든 상태의 눈금을 3씩 빼앗습니다.
짝수가 나온 플레이어[2]는 육면체 주사위를 한 번 더 굴릴 수 있습니다!
2회차. 플레이어[2]의 주사위 값은 = 2
특수스킬1. 플레이어[2]의 가 상태의 주사위 눈금을 2 포인트립니다.
최종 합계. 플레이어[1]의 주사위 값은 = 11
최종 합계. 플레이어[2]의 주사위 값은 = 2

```

그래서 for문 안에 짝수 주사위값을 판별하는 조건식을 넣고

그 안에 특수 주사위 스킬을 작동시킬 수 있도록 했는데 (선생님의 풀이와 같은 방식)

### 결과를 출력해보면

추가 주사위를 굴린다 > 특수 스킬이 발동한다 > 만약 이로 인해 점수가 깎여 기존에 추가 주사위를 굴릴 수 없었던 플레이어의 주사위 눈이 짝수가 될 경우 이 녀석도 갑자기 추가로 주사위를 굴린다.

라는 문제가 발생되어요... 이걸 어떻게 고치면 좋은걸까요...