



5월 26일 복습

이태양



```
import java.util.Scanner;

class DiceDeathGame {
    final int MAX_PLAYER = 2;
    final int DICE_IDX = MAX_PLAYER + 1;
    final int START_MONEY = 1000;

    final int FIRST_IDX = 0;
    final int SECOND_IDX = 1;
    final int TOTAL_IDX = 2;

    final int DICE_RANGE = 6;
    final int DICE_START_OFFSET = 1;

    int comMoney;
    int usrMoney;
    int betMoney;

    int roundNum;

    Boolean isAlive;
    Scanner scan;

    int[] comDice;
    int[] usrDice;
```

필요한 변수들을 클래스 안에 선언해주었다.
DICE_IDX에 첫번째 두번째 합이 들어가야해서 3의 크기의 배열이 필요해서 MAX_PLAYER+1 을 해주었다
시작돈은 천만원이기에 고정해주고
인덱스 번호를 가르키는 변수들

주사위 범위를 지정하기 위해 선언해줌

배팅머니와 컴과 유저의 돈을 받을 변수

라운드 카운트하기위해 선언해줌

반복문을 돌리기위해 선언해주었고
입력을 위해 객체생성

주사위 눈금을 배열로 받기 위해 배열을 선언해주었다





```
public DiceDeathGame () {
    roundNum = 0;

    comDice = new int[DICE_IDX];
    usrDice = new int[DICE_IDX];

    isAlive = true;
    scan = new Scanner(System.in);

    comMoney = START_MONEY;
    usrMoney = START_MONEY;
}

private void bettingMoney () {
    System.out.printf("현재 %d 라운드입니다.\n", ++roundNum);
    System.out.printf("이게임을 시작한 당신 오늘 둘중 하나는 인생 로그아웃합니다.\n" +
        "돈이 부족하면 손모가지나 장기로 대체 가능합니다.\n" +
        "판돈을 입력하시오: ");
    betMoney = scan.nextInt();
}

public void gameStart () {
    do {
        bettingMoney();
        runDice();
        checkMagicDiceSkill();
        whoGetBetMoney();
        canWePlay();
        checkEachMoney();
    } while (isAlive);
}
```

라운드넘을 초기화
배열 크기를 지정해주고

True로 설정
입력을 위한 객체를 생성하고
둘다 1000으로 지정해준다(시작금액)

배팅금액을 입력하는 함수

중재자역할이라고 했던 부분
만들어놓았던 함수를 호출한다
위에 선언했던 isAlive가 false가될 때까지



chapter

```
private void checkEachMoney () { System.out.printf("사용자 수중 금전: %d, 컴퓨터 수중 금전: %d\n", usrMoney, comMoney); }
private void canWePlay () {
    if (usrMoney <= 0 || comMoney <= 0) {
        isAlive = false;
    }
}
private void whoGetBetMoney () {
    int usrTotDice = usrDice[TOTAL_IDX];
    int comTotDice = comDice[TOTAL_IDX];

    if (usrTotDice > comTotDice) {
        System.out.printf("이번판은 사용자가 이겼습니다! %d(사용자) vs %d(컴퓨터)\n",
            usrTotDice, comTotDice);
        usrMoney += betMoney;
        comMoney -= betMoney;
    } else if (usrTotDice < comTotDice) {
        System.out.printf("이번판은 컴퓨터가 이겼습니다! %d(사용자) vs %d(컴퓨터)\n",
            usrTotDice, comTotDice);
        usrMoney -= betMoney;
        comMoney += betMoney;
    } else {
        System.out.printf("승부가 나지 않았습니다! %d(사용자) vs %d(컴퓨터)\n",
            usrTotDice, comTotDice);
    }
}
```

남은 금액을 확인하기 위한 함수
돈이 0원 밑으로 떨어지면 반복문을
종료할 수 있게 false로 바꿔준다.

주사위 눈금을 담을 수 있게 배열 선언

크기 비교해서 usr나 com의 돈 계산



```
private void checkSkill (int[] curDice, int[] targetDice) {
    switch (curDice[SECOND_IDX]) {
        case 1:
            curDice[TOTAL_IDX] = curDice[FIRST_IDX] + 3;
            break;
        case 3:
            targetDice[TOTAL_IDX] = targetDice[FIRST_IDX] - 2;
            break;
        case 4:
            curDice[TOTAL_IDX] = 0;
            break;
        case 6:
            curDice[TOTAL_IDX] = (curDice[FIRST_IDX] + curDice[SECOND_IDX]) * 2;
            targetDice[TOTAL_IDX] = (targetDice[FIRST_IDX] + targetDice[SECOND_IDX]) * 2;
            break;
        default:
            curDice[TOTAL_IDX] = curDice[FIRST_IDX] + curDice[SECOND_IDX];
            break;
    }
}

private void checkMagicDiceSkill () {
    // 사용자 관점에서의 2번째 주사위 스킬 발동
    checkSkill(usrDice, comDice);
    // 컴퓨터 관점에서의 2번째 주사위 스킬 발동
    checkSkill(comDice, usrDice);
}

private void runSecondDice (int[] dice) {
    if (dice[FIRST_IDX] % 2 == 0) {
        dice[SECOND_IDX] = getRandomValue(DICE_RANGE, DICE_START_OFFSET);
    }
}
```

curDice와 targetDice 두개를 입력받는데
Switch문의 조건은 curDice[SECOND_IDX]이다
curDice로 스위치가 한번돌고 targetDice로 한번
또 돌게 되는건가요?

주사위눈금에 따른 조건들 나열

checkSkill을 동작시키는 함수

두번째 다이스의 값을 구하기 위한 함수



```
,
private void runFirstDice (int[] dice) {
    dice[FIRST_IDX] = getRandomValue(DICE_RANGE, DICE_START_OFFSET);
}

private void runDice () {

    runFirstDice(usrDice);
    runFirstDice(comDice);

    runSecondDice(usrDice);
    runSecondDice(comDice);
}

public int getRandomValue (int range, int startOffset) { return (int)(Math.random() * range + startOffset); }

}

public class Prob45Answer {
    public static void main(String[] args) {
        DiceDeathGame ddg = new DiceDeathGame();

        ddg.gameStart();
    }
}
```

첫번째 주사위의 값을 구하는 함수

첫번째 주사위값과 두번째 주사위값을
배열에 넣기 위한 함수

범위를 입력받아서 주사위 눈금의 값을
반환한다