

```

public class _4th_Quiz45 {
    public static void main(String[] args) {
        ---Casino gaem---
        Start Money 1000만원.
        베팅금액은 Scanner를 통해 입력.
        둘 중 하나가 파산하면 game set.
        주사위 2개를 사용.
        숫자가 높은 사람이 round 승자.
        첫 번째 주사위가 짝수가 나온 경우에만 두 번째 주사위를 굴릴 수 있다.
        두 번째 주사위는 특수 능력을 가지고 있다.
        두 번째 주사위가 1이 나오면 첫 번째 주사위의 값을 + 3 할 수 있다.
        두 번째 주사위가 3이 나오면 상대방 첫 번째 주사위 값을 -2 할 수 있다.
        두 번째 주사위가 4가 나오면 내 첫 번째 주사위 값이 0이 된다.
        두 번째 주사위가 6이 나오면 상대방과 나의 주사위 값을 2배로 한다.
        (2배 적용은 (첫 번째 주사위 + 두 번째 주사위)를 진행한 후 2배 적용한다)
        DiceGame casino = new DiceGame();
    }
}

```

2021.05.26 Java

〈Quiz.45 재도전〉

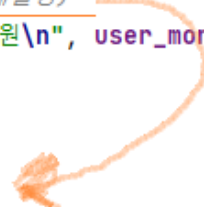
- Q. 계속 run해본 결과
아래 질문한 설계 상 오류로
판단 되는 것 외에
다른 문제는 없어보이는데
혹시 코드 옮겨서 run 하면서
제가 찾지 못 한
문제점이 있는지 봐주실 수 있을까요..?

<https://github.com/KHWeb18/LectureContents/pull/413/commits/113f4d710c2638387adfd5f80135b268bdd3c866#diff-68a87df1e772634781ade1f642476f4f87a8aa51f0c00790e5a3f11a4da64a3eL1>

```
import java.util.Scanner;

class DiceGame {
    private int user_money = 1000;
    private int computer_money = 1000;
    Scanner scan;
    public int user_Dice;
    public int user_Dice2;
    public int computer_Dice;
    public int computer_Dice2;
    public int bet_money;
    public boolean one_more_round = true;
    public int round = 1;
```

```
    public void start_Game(){ // 현재금액 + 배팅금액(배팅금액 오기입시 배팅금액 재설정)
        System.out.printf("현재 user_money:%d만원\n현재 computer_money: %d만원\n", user_money, computer_money);
        System.out.print(">>>라운드를 진행하려면 배팅하시오(1~1000만원): ");
        scan = new Scanner(System.in);
        bet_money = scan.nextInt();
        while(bet_money > 1000 || bet_money <= 0){
            System.out.print(">>>범위 안의 값을 배팅하시오(1~1000만원): ");
            bet_money = scan.nextInt();
        }
    }
```



```
    public DiceGame() { // 생성자
        while (one_more_round) {
            start_Game(); // 현재금액 + 배팅금액(배팅금액 오기입시 배팅금액 재설정)
            user_Dice = getRandDice( range: 6, offset: 1);
            computer_Dice = getRandDice( range: 6, offset: 1);
            let_Me_Know_firstDice(); // 첫번째 주사위 print
            if_User_Dice(); // 주사위 홀수/짝수 판정 후 짝수인 경우 두번째 주사위
            if_Computer_Dice(); // 주사위 홀수/짝수 판정 후 짝수인 경우 두번째 주사위
            let_Me_Know_secondDice(); // 두번째 주사위 print
            calc_Betting_Money(); // Round승자 판단 후 betting money 계산
            gameSet(); // 파산여부 검사
        }
    }
```

```

public void if_User_Dice() { // user주사위 홀수/짝수 판정 후 짝수인 경우 두번째 주사위
    user_Dice2 = 0;
    // 다음 라운드에서 첫번째 주사위가 홀수 일 경우를 대비해서 두번째 주사위 0으로 초기화
    if (user_Dice % 2 == 0) {
        user_Dice2 = getRandDice( range: 6, offset: 1);
    }
    switch (user_Dice2) {
        case 1:
            user_Dice = user_Dice + 3;
            break;
        case 3:
            computer_Dice = computer_Dice - 2;
            break;
        case 4:
            user_Dice = 0;
            break;
        case 6:
            user_Dice = (user_Dice + user_Dice2) * 2;
            computer_Dice = (computer_Dice + computer_Dice2) * 2;
            break;
        default:
            break;
    }
}

```

```

public void if_Computer_Dice() { // computer주사위 홀수/짝수 판정 후 짝수인 경우 두번째 주사위
    computer_Dice2 = 0;
    // 다음 라운드에서 첫번째 주사위가 홀수 일 경우를 대비해서 두번째 주사위 0으로 초기화
    if (computer_Dice % 2 == 0) {
        computer_Dice2 = getRandDice( range: 6, offset: 1);
    }
    switch (computer_Dice2) {
        case 1:
            computer_Dice = computer_Dice + 3;
            break;
        case 3:
            user_Dice = user_Dice - 2;
            break;
        case 4:
            computer_Dice = 0;
            break;
        case 6:
            computer_Dice = (computer_Dice + computer_Dice2) * 2;
            user_Dice = (user_Dice + user_Dice2) * 2;
            break;
        default:
            break;
    }
}

```

```

public void let_Me_Know_firstDice() {
    System.out.printf("-----\nuser의 첫번째 dice는 %d\ncomputer의 첫번째 dice는 %d\n-----\n"
        , user_Dice, computer_Dice);
}

public void let_Me_Know_secondDice() {
    System.out.printf("user의 두번째 dice는 %d \ncomputer의 두번째 dice는 %d\n-----\n"
        , user_Dice2, computer_Dice2);
    System.out.printf("***round%d의 결과***\n", round++);
}

```

위의 두 번째 주사위를 돌리는 method를 합치고 싶었으나 method 내부에 Dice_2 때문에 실패. 매개변수 3개를 사용하는 method로 다시 도전했으나 매개변수들이 인지가 안 되는 현상 발생.

Q. 위 현상 관련해서 아래쪽에 질문 적어두었습니다.

```

public void calc_Betting_Money() { // Round승자 판단 후 betting money 계산
    if (user_Dice > computer_Dice) {
        System.out.printf("user승!\nuser: %d\ncomputer: %d\n", user_Dice, computer_Dice);
        user_money += bet_money;
        computer_money -= bet_money;
    } else if (user_Dice < computer_Dice) {
        System.out.printf("computer승!\nuser: %d\ncomputer: %d\n", user_Dice, computer_Dice);
        user_money -= bet_money;
        computer_money += bet_money;
    } else {
        System.out.printf("무승부!\nuser: %d\ncomputer: %d\n", user_Dice, computer_Dice);
    }
}

public void gameSet() { // 파산여부 검사
    if (user_money <= 0 || computer_money <= 0) {
        System.out.printf("최종 user_money:%d만원\n최종 computer_money: %d만원\n", user_money, computer_money);
        one_more_round = false;
    } else {
        System.out.println("-----\n-----");
    }
}

public int getRandDice(int range, int offset) { return (int) (Math.random() * range + offset); }

```

첫 번째 주사위가 홀수가 나왔음에도 두 번째 주사위를 돌리는 오류 발생

- 두 번째 주사위를 돌리는 method에서 시작 전 두 번째 주사위를 0으로 초기화 하는 코드 생성 후 해결.

```
_4th_Quiz45 x
-----
현재 user_money:700
현재 computer_money: 1300
>>>배팅금액(1~1000): 100
-----
user의 첫번째 dice는 5
computer의 첫번째 dice는 1
-----
user의 두번째 dice는 5
computer의 두번째 dice는 6
-----
```

```
현재 user_money:600
현재 computer_money: 1400
>>>배팅금액(1~1000): 100
-----
user의 첫번째 dice는 3
computer의 첫번째 dice는 1
-----
user의 두번째 dice는 2
computer의 두번째 dice는 2
-----
***round8의 결과***
user승!
user: 3
computer: 1
-----
현재 user_money:700
현재 computer_money: 1300
>>>배팅금액(1~1000):
```

```
user의 첫번째 dice는 6
computer의 첫번째 dice는 3
-----
user의 두번째 dice는 6
computer의 두번째 dice는 1
-----
***round13의 결과***
user승!
user: 24
computer: 9
```

그러나 computer_dice 첫 번째 주사위가 3이 나왔음에도

user_dice의 두 번째 주사위가 6이 나왔기 때문에(6은 모두 2를 곱해줘서 결국 다 짝수가 됨)

computer_dice의 첫 번째 주사위가 짝수로 인식 되어서 두 번째 주사위도 돌아감.

이건 초반 설계상의 판단 미스인 것 같습니다.

해결 방법은

'짝수/홀수 판정 후 두 번째 주사위를 돌리는' method를 합쳐서

switch-case로 or/and 비교를 이용해서 엄청 만들어야 하지 않을까 싶은데 엄두가 안 납니다..

```

System.out.print(">>>배팅 (1~1000): ");
bet_money = scan.nextInt();
user_Dice = getRandDice( range: 6, offset: 1);
computer_Dice = getRandDice( range: 6, offset: 1);
let_Me_Know_firstDice();
if_Dice(user_Dice, computer_Dice,user_Dice2);
if_Dice(computer_Dice, user_Dice,computer_Dice2);
let_Me_Know_secondDice();
System.out.printf("***round%d의 결과***\n", round++ );
calcMoney();
gameSet();
}
}

```

```

public void if_Dice(int my_Dice, int your_Dice, int my_second_dice ){
    if(my_Dice % 2 == 0){
        my_second_dice = getRandDice( range: 6, offset: 1);
    }
    switch(my_second_dice){
        case 1 : my_Dice = my_Dice + 3;
            break;
        case 3 : your_Dice = your_Dice - 2;
            break;
        case 4 : my_Dice = 0;
            break;
        case 6 : my_Dice = (my_Dice + my_second_dice) * 2;
            break;
        default: break;
    }
}

public void let_Me_Know_firstDice(){

```

```

현재 user_money:600
현재 computer_money: 1400
>>>배팅금액(1~1000): 100
-----
user의 첫번째 dice는 6
computer의 첫번째 dice는 6
-----
user의 두번째 dice는 0
computer의 두번째 dice는 0
-----
***round5의 결과***
무승부!
user: 6
computer: 6
-----
현재 user_money:600
현재 computer_money: 1400
>>>배팅금액(1~1000): 100
-----
user의 첫번째 dice는 2
computer의 첫번째 dice는 4
-----
user의 두번째 dice는 0
computer의 두번째 dice는 0
-----
***round6의 결과***
computer승!
user: 2
computer: 4
-----

```

Q. '짝수/홀수 판정 후 두 번째 주사위를 돌리는' method를 com/user 두 개로 하지 않고 하나로 만들고 싶었는데 왜 switch-case에서 매개변수들을 인식 못 하는지 이유를 모르겠습니다..

The value (my_Dice + my_second_dice) * 2 assigned to 'my_Dice' is never used
 Remove redundant assignment Alt+Shift+Enter More actions... Alt+Enter
 int my_Dice

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

Q. Quiz.45 강사님 풀이에서.

dice[FIRST_IDX]의 값이 홀수가 나오면

if문이 실행되지 않기 때문에 dice[SECOND_IDX]의 값이

setting이 안 되고 비어있기 때문에

```
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;
    }
}
```

case6의 경우에 error가 나야되지 않나
생각되는데

정상적으로 작동하는 원리가 궁금합니다..

〈class를 잘 만들어야 하는 이유〉

다른 작업에서도 이용할 수 있기 때문에
처음부터 잘 만들어놔야된다.

