(디지털컨버전스) 스마트 콘텐츠와 웹 융합 응용 SW개발자 양성과정

-5일차 학습 및 질문 노트-

강사 - Innova Lee(이상훈)
gcccompil3r@gmail.com
학생 – Kyeonghwan Lee(이경환)
airtrade7@naver.com

학습노트: 퀴즈 17번

```java
public class Prob17Answer {
    public static void main(String[] args) {
        System.out.println("컴퓨터가 주사위를 굴립니다.");

        int com_dic1 = (int)(Math.random()* 6 + 1);
        int com_dic2 = (int)(Math.random()* 6 + 1);

        System.out.println("사용자가 주사위를 굴립니다");

        int user_dic1 = (int)(Math.random() * 6 + 1 );
        int user_dic2 = (int)(Math.random() * 6 + 1 );

        int com_total = com_dic1 + com_dic2;
        int user_total = user_dic1 + user_dic2;

        if( com_total > user_total){
            System.out.printf("컴퓨터 승! 점수는 %d(컴퓨터) vs %d(사용자)\n,", com_total, user_total);
        } else if (user_total > com_total) {
            System.out.printf("사용자 승! 점수는 %d(컴퓨터) vs %d(사용자)\n,", com_total, user_total);
        } else {
            System.out.printf("무승부! 점수는 %d(컴퓨터) vs %d(사용자)\n,", com_total, user_total);
        }

    }
}
```

*랜덤 주사위 변수 지정 및 if 를 활용한 경우의 수 값 도출

*랜덤 주사위 변수 지정 및 if 를 활용한 경우의 수 값 도출

학습노트: AND OR 비트 연산자

```java
public class BitAndTest {
    public static void main(String[] args) {
        int num1 = 10, num2 = 8;

        // & 이 비트연산자 AND
        // 관계 연산자에서는 && 형태로 존재하였음
        // 10 ===> 1010
        //  8 ===> 1000 AND
        // -----------------
        //  8 ===> 1000
        System.out.printf("%d AND %d = %d\n", num1, num2, num1 & num2);

        num2 = 138;

        // 138 ===> 10001010
        //  10 ===>     1010 AND
        // ----------------------
        //  10 ===> 00001010
        System.out.printf("%d AND %d = %d\n", num1, num2, num1 & num2);
    }
}
// AND 연산자의 경우 둘다 중첩되는 부분만 출력 시킨다.
```

```java
public class BitOrTest {
    public static void main(String[] args) {
        int num1 = 10, num2 = 5;

        // | 이 비트연산자 OR
        // 관계 연산자에서는 || 형태로 존재하였음
        // 10 ===> 1010
        //  5 ===> 0101 OR
        // -----------------
        // 15 ===> 1111
        System.out.printf("%d OR %d = %d\n", num1, num2, num1 | num2);

        num2 = 136;

        //  10 ===> 00001010
        // 136 ===> 10001000 OR
        // ---------------------
        // 138 ===> 10001010
        System.out.printf("%d OR %d = %d\n", num1, num2, num1 | num2);

        // OR 연산은 합집합 개념, AND 연산은 교집합 개념
        // *주의 OR 연산의 경우 단순 숫자합의 개념이 아니다.
```

OR 연산은 합집합, AND 연산은 교집합 개념

학습노트: Interrupt(이벤트)

```java
public class InterruptComment {
    public static void main(String[] args) throws InterruptedException {
        for (int i = 0;; i++) {
            if (i % 2 == 0) {
                System.out.println("안녕 난 짝수야!");
            } else {
                System.out.println("하이 난 홀수야!");
            }

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

// Interrupt: 인터럽트란 무엇인가 ?
// 사실 인터럽트라는 용어는 하드웨어 개발자들이 주로 사용하는 단어다.
// 보통 자바나 GUI 개발자들 혹은 애플리케이션 개발자들은 이벤트라고 표현한다.
// 결국 이벤트와 인터럽트는 동의어란 뜻이다.
// 그렇다면 인터럽트라고 부르지말고 이벤트라고 불러보자!
// 이벤트는 뭘까 ?

```

인터럽트란? 몰컴시 갑자기 열리는 방문과 같이
최우선적으로 처리해야 하는 작업으로 이해

학습노트: Interrupt(이벤트)

```java
public class InterruptComment {
    public static void main(String[] args) throws InterruptedException {
        for (int i = 0;; i++) {
            if (i % 2 == 0) {
                System.out.println("안녕 난 짝수야!");
            } else {
                System.out.println("하이 난 홀수야!");
            }

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

// Interrupt: 인터럽트란 무엇인가 ?
// 사실 인터럽트라는 용어는 하드웨어 개발자들이 주로 사용하는 단어다.
// 보통 자바나 GUI 개발자들 혹은 애플리케이션 개발자들은 이벤트라고 표현한다.
// 결국 이벤트와 인터럽트는 동의어란 뜻이다.
// 그렇다면 인터럽트라고 부르지말고 이벤트라고 불러보자!
// 이벤트는 뭘까 ?

```

인터럽트란? 몰컴시 갑자기 열리는 방문과 같이
최우선적으로 처리해야 하는 작업으로 이해

학습노트: 10진수 -> 2진수 , 2진수

```
// 21 ---> 16(2^4) + 4(2^2) + 1(2^0)
//          10101
// 1, 3, 5 번째 비트지만
// 실제 표현할때는 0번 비트, 2번 비트, 4번 비트로 표현해주도록 한다.

// 73 ---> 64(2^6) + 8(2^3) + 1(2^0)
//            1001001
// 0번 비트, 3번 비트, 6번 비트로 표현됨
```

10진수 ->2진수

1. 2진수 변경 시에는 해당 숫자와 가장 근접한 2의n 승을 찾고 값이 0이 나올 때까지 반복
2. 구한 숫자들이 각각 이진수의 자리 수 에 해당하며, 없는 숫자의 경우 0으로 대입한다.

```
// 2진수 10101000 을 10진수로 바꿔보자!
// 2^7 + 2^5 + 2^3 = 8 + 32 + 128 = 168

// 2진수 11111100 을 10진수로 바꿔보자!
// 11111111 = 2^0 + 2^1 + ... + 2^7 => 2^8 - 1 = 256 - 1 = 255
// 255 - 3 = 252
```

2진수 ->10진수

1) 1이 적은 경우
   -> 해당 자리 수에 n승을 계산하여 합한다.
2) 0이 적은 경우
   -> 0의 자리에 1을 넣은 후 총 값에서 0에 해당되는 자리값을 빼고 계산한다.

*주의 :2진수의 경우 1의 자리가 아닌 0의 자리부터 시작된다

학습노트: NonDuplicateWithoutArrayTest

```java
public class sss {
    public static void main(String[] args) {
        final int BIN = 1;

        int testBit = 0;
        int randNum;

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

            while ((testBit & (BIN << randNum)) != 0) {
                System.out.println("중복이 이렇게나 많이 발생합니다: " + randNum);
                randNum = (int)(Math.random() * 10);
            }

            System.out.printf("randNum = %d\n", randNum);

            testBit |= (BIN << randNum);
        }

        System.out.println("testBit의 최종값은 1023이다. 진짜 ? " + testBit);
    }
}
```

Final int : 값을 한번만 할당, 변경 불가
설명은 듣고 이해는 갔으나, 혼자서 짜기는 힘들어 보인다.