

2023-03-07

Math.random() : 난수를 추출해주는 API ($0.0 \leq ? < 1.0$)

자바 산술 이항 연산

int 타입보다 작은 타입(char, short, byte)은 타입으로 변환되어 연산 수행

두 항의 타입이 다를 때는 값이 손실되지 않는 범위내에서 하나의 타입으로 일치시켜서

```
char + char -> int
short + short -> int
byte + byte -> int
char + long -> long
int + long -> long
long + float -> float
double + float -> double
```

```
i-value = r-value
    >      자동 형변환
    <      강제 형 변환(연산자를 이용해서) I-value = (I-value의 타입)-r-value
(예외 : boolean)
```

대입연산

```
int result = ++num; // 먼저
int result = num ++; //나중에
```

```
System.out.println(num);
System.out.println(++num);
System.out.println(num++);
```

조건연산자(3항연산자)

조건식(연산결과가 boolean타입인식) ? 참일 때 수행할 식 : 거짓일 때 수행할 식

```
public static void main(String[] args) {
    int num = (int)(Math.random() * 10) + 1; // 1~10
    // CASE 1
    char result = num % 2 == 1 ? '홀' : '짝';
    System.out.println(num+"은 "+result+"수입니다.");
}
```

```
// CASE 2
System.out.println(num+"은 "+(num % 2 == 1 ? '홀' : '짝')+ "수입니다.");
}
```

같은 조건식을 반복해서 사용한다면 CASE1처럼 변수를 선언한다.

println은 전달된 값이 char이면 문자형태, int면 10진수로 출력한다.

연산자

논리연산자 : &&, || → 이항연산자로서 피연산자가 반드시 boolean타입이어야함

비트연산자 : &, | → 이항연산자로서 피연산자가 정수형, 논리형 모두 가능

```
8 & 7 --> 00001000 & 00000111 --> 00000000
8 | 7 --> 00001000 | 00000111 --> 00001111
```

Switch 문

식 : int(byte, short, char), String, enum

```
switch (등가비교식) {
    case 비교값1 : 수행문장 1;
        break;
    case 비교값2 : 수행문장 2;
        수행문장 3;
        break;
    case 비교값3 : 수행문장 4;
        break;
    default : 수행문장 5;
        수행문장 6;
}
```

제어문

조건제어문(선택제어문): if, switch

반복제어문: for, while, do~while

분기제어문: break, continue(반드시 반복문 안에서만 사용한다)

for문 : 반복 횟수를 이미 알고 있을 때 / 배열이나 컬렉션과 같은 데이터들의 묶음으로 반복하고자 할 때

while문 : 어떠한 조건이 만족되는 동안 반복, 무한루프

for문 while문 모두 무한루프에서 사용할 수 있으나 대부분의 개발자는 while을 많이 사용한다.

for문

```
for(int su = 1; su <=10; su++) {  
    result = su;  
}
```

가급적이면 반복문내에서 변수선언하는 코드는 짜지말자! → 메모리를 낭비하는 코드