

JAVA 2주차 수업

복습

반복문

특정한 행동을 반복시키기 위해 사용

```
for (int i = 0 ; i < 10 ; i++) {  
    System.out.println("이 라인은 " + i + "번 째 라인!");  
}  
  
int i = 0;  
while (i < 10) {  
    System.out.println("이 라인은 " + i + "번 째 라인!");  
}
```

반복문을 중간에 끊고 싶다면 break 사용

조건문

조건이 참이라면 코드 블록 실행

```
int a = 10;

if(a == 10) {
    System.out.println("a는 10이다!");
} else if (a == 11) {
    System.out.println("a는 11이다!");
} else {
    System.out.println("a는 10도 11도 아니다!");
}
```


복습

계산기를 만들어보자!

ex)

더할 첫번째 숫자를 입력하세요 : 1

더할 두번째 숫자를 입력하세요 : 2

두 수의 합은? : 3

실습

조건문을 사용하여 더 나은 계산기를 만들어보자!

ex)

어떤 연산을 하시겠습니까?

1. 더하기 2. 빼기 3. 곱하기 4. 나누기

입력: 4

나눌 숫자를 입력하세요 : 10

몇으로 나눌지 입력하세요 : 4

결과 : 몫 - 2, 나머지 - 2

실습

조건문과 반복문을 이용하여 숫자 업다운 게임을 만들어보자

게임 규칙

- 프로그램 실행시 게임을 시작하시겠습니까?(1 : 예, 2: 아니오) 문구 출력
- 아니오를 입력하면 게임 종료
- 예를 입력하면 1~100까지중 랜덤 숫자를 만든다
- 사용자가 답을 맞출때까지 Up Down을 출력
- 게임이 끝나면 처음으로 돌아감 (게임을 시작하시겠습니까?)

Ex) 예상결과

게임을 시작하시겠습니까 ? (1: 예, 2: 아니오) : 1
랜덤숫자를 생성하였습니다.
어떤 값일까요? : 10
UP
어떤 값일까요? : 30
UP
어떤 값일까요? : 99
DOWN
어떤 값일까요? : 35
정답입니다! : 35
게임을 시작하시겠습니까 ? (1: 예, 2: 아니오) : 2
종료

HINT (여기까지 보고 시작!)

- Google에 "JAVA Random 1 - 100" 이라 검색

HINT

1~100까지중 랜덤숫자를 가져오는 방법

```
Random rn = new Random();  
int answer = rn.nextInt(100) + 1;
```

정답코드

```
boolean isGame = true;

while(isGame) {
    System.out.print("게임을 시작하시겠습니까 ? (1: 예, 2: 아니오)");
    Scanner scanner = new Scanner(System.in);
    int gameStartNumber = scanner.nextInt();

    if(gameStartNumber == 2) {
        System.out.println("게임 종료");
        isGame = false;
        break;
    }

    Random random = new Random();
    int randomNumber = random.nextInt(100) + 1;
    System.out.println("랜덤숫자를 생성하였습니다.");
}
```

```
boolean isAnswer = false;

while(!isAnswer) {
    System.out.print("어떤 값일까요? :");
    Scanner scanner2 = new Scanner(System.in);
    int answerNumber = scanner2.nextInt();

    if(answerNumber == randomNumber) {
        System.out.println("정답입니다! : " + randomNumber);
        isAnswer = true;
        break;
    }
    else if (answerNumber < randomNumber) {
        System.out.println("UP");
    } else {
        System.out.println("DOWN");
    }
}
}
```

배열

변수의 집합!

: 데이터를 저장하는 공간들의 집합

배열

변수로 만든면?

```
int a1 = 0, a2 = 1, a3 = 2;
```

매우매우 귀찮다..

HOW?

```
int array[] = new int[5];  
int[] array = new int[5];  
  
array[0] = 0;  
array[1] = 1;  
array[2] = 2;
```

이것도 귀찮아 보이는데?

TIP

- new 키워드는 배열이나 클래스를 할당할 때 사용한다.

HOW?

```
int array[] = {0, 1, 2, 3};  
int[] array = new int[]{0, 1, 2, 3};
```

한번에 하는 방법도 있다!

실습

배열에 들어있는 값을 모두 더한 결과를 보여주는 프로그램을 만들자.

실습 2

배열에 들어있는 값을 순서대로 정렬해서 출력하는 프로그램을 만들어 보자.

?

지금껏 짜온 코드가 너무 길지 않은가요?

메서드 (method)

프로그래밍의 효율성을 위해서 프로그램을 쪼개는 것

다른 언어에서는 함수(function)라고 한다.

쓰기 전에!

- 한가지 메서드는 한가지 동작만 하는것이 좋다.
- 메서드의 이름은 신중하게 짓자

Method의 기본 구조

```
[결과 값 타입] [함수이름]([전달된 값의 타입] [사용될 변수 이름]...) {  
    //함수 내용, 로직  
    return [결과 값]  
}
```

```
int add(int a, int b) {  
    return a + b;  
}
```

TIP

- return type이 없으면 void로 표기한다.
- 입력값은 없어도 된다.
- 함수명의 표기는 카멜표기법을 사용한다.
- 함수명의 시작은 소문자로 시작한다.

실습

지금껏 만든 프로그램들을 함수로 나눠서 만들어보자!