반복문
실습 문제

▶ 문제 안내

기능 제공 클래스 : com.br.chap02.practice.example.LoopPractice

기능 실행 클래스 : com.br.chap02.practice.run.Run

한 실습 기능 클래스에 여러 메소드를 넣어 실습 진행

▶ 실습문제1

메소드 명 : public void practice1(){}

사용자로부터 한 개의 값을 입력 받아 1부터 그 숫자까지의 숫자들을 모두 출력하세요.

단, 입력한 수는 1보다 크거나 같아야 합니다.

만일 1 미만의 숫자가 입력됐다면 "잘못 입력하셨습니다."를 출력하세요.

ex.

1이상의 숫자를 입력하세요 : 4

1 2 3 4

1이상의 숫자를 입력하세요 : 0

잘못 입력하셨습니다.

▶ 실습문제2

메소드 명 : public void practice2(){}

practice1() 문제와 동일하나, 1 미만의 숫자가 입력됐다면

"잘못 입력하셨습니다. 다시 입력해주세요."가 출력되면서 다시 사용자가 값을 입력하도록 하세요.

ex.

```
1이상의 숫자를 입력하세요 : 4
1 2 3 4
```

```
1이상의 숫자를 입력하세요 : 0
잘못 입력하셨습니다. 다시 입력해주세요.
1이상의 숫자를 입력하세요 : 8
1 2 3 4 5 6 7 8
```

▶ 실습문제3

메소드 명 : public void practice3(){}

사용자로부터 한 개의 값을 입력 받아 1부터 그 숫자까지의 모든 숫자를 거꾸로 출력하세요.

단, 입력한 수는 1보다 크거나 같아야 합니다.

ex.

1이상의 숫자를 입력하세요 : 4

4 3 2 1

1이상의 숫자를 입력하세요 : 0

잘못 입력하셨습니다.

▶ 실습문제4

메소드 명 : public void practice4(){}

practice3() 문제와 동일하나, 1 미만의 숫자가 입력됐다면

"잘못 입력하셨습니다. 다시 입력해주세요."가 출력되면서 다시 사용자가 값을 입력하도록 하세요.

ex.

```
1이상의 숫자를 입력하세요 : 4
4 3 2 1
```

```
1이상의 숫자를 입력하세요 : 0
잘못 입력하셨습니다. 다시 입력해주세요.
1이상의 숫자를 입력하세요 : 8
8 7 6 5 4 3 2 1
```

▶ 실습문제5

메소드 명 : public void practice5(){}

1부터 사용자에게 입력 받은 수까지의 정수들의 합을 출력하세요.

ex.

정수를 하나 입력하세요 : 8

1 + 2 + 3 + 4 + 5 + 6 + 7 + 8 = 36

▶ 실습문제6

메소드 명 : public void practice6(){}

사용자로부터 두 개의 값을 입력 받아 그 사이의 숫자를 모두 출력하세요.

만일 1 미만의 숫자가 입력됐다면 "1이상의 숫자만을 입력해주세요"를 출력하세요.

ex.

```
첫 번째 숫자 : 8
두 번째 숫자 : 4
4 5 6 7 8
```

```
첫 번째 숫자 : 4
두 번째 숫자 : 8
4 5 6 7 8
```

```
첫 번째 숫자 : 9
두 번째 숫자 : 0
1이상의 숫자만을 입력해주세요.
```

▶ 실습문제7

메소드 명 : public void practice7(){}

practice6() 문제와 동일하나, 1 미만의 숫자가 입력됐다면

"1 이상의 숫자를 입력해주세요"가 출력되면서 다시 사용자가 값을 입력하도록 하세요.

ex.

```
첫 번째 숫자 : 8
두 번째 숫자 : 4
4 5 6 7 8
```

```
첫 번째 숫자 : 4
두 번째 숫자 : 8
4 5 6 7 8
```

```
첫 번째 숫자 : 9
두 번째 숫자 : 0
1 이상의 숫자를 입력해주세요.
첫 번째 숫자 : 6
두 번째 숫자 : 2
2 3 4 5 6
```

▶ 실습문제8

메소드 명 : public void practice8(){}

사용자로부터 입력 받은 숫자의 단을 출력하세요.

ex.

```
숫자 : 4
===== 4단 =====
4 * 1 = 4
4 * 2 = 8
4 * 3 = 12
4 * 4 = 16
4 * 5 = 20
4 * 6 = 24
4 * 7 = 28
4 * 8 = 32
4 * 9 = 36
```

▶ 실습문제9

메소드 명 : public void practice9(){}

사용자로부터 입력 받은 숫자의 단부터 9단까지 출력하세요.

단, 2~9 사이의 숫자가 아닌 경우 "2~9 사이의 숫자만 입력해주세요"를 출력하세요.

숫자 : 4

===== 4단 =====

===== 5단 =====

===== 6단 =====

===== 7단 =====

===== 8단 =====

===== 9단 =====

(해당 단의 내용들은 길이상 생략)

숫자 : 10

2~9 사이의 숫자만 입력해주세요.

▶ 실습문제10

메소드 명 : public void practice10(){}

practice9() 문제와 동일하나, 2~9 사이의 숫자가 아닌 값이 입력됐다면

"2~9 사이의 숫자를 입력해주세요"가 출력되면서 다시 사용자가 값을 입력하도록 하세요.

```
숫자 : 4
===== 4단 =====
===== 5단 =====
===== 6단 =====
===== 7단 =====
===== 8단 =====
===== 9단 =====
(해당 단의 내용들은 길이상 생략)
```

```
숫자 : 10
2~9 사이의 숫자만 입력해주세요.
숫자 : 8
===== 8단 =====
===== 9단 =====
```

▶ 실습문제11

메소드 명 : public void practice11(){}

사용자로부터 시작 숫자와 공차를 입력 받아

일정한 값으로 숫자가 커지거나 작아지는 프로그램을 구현하세요.

단, 출력되는 숫자는 총 10개입니다.

* 부연설명 : '공차'는 숫자들 사이에서 일정한 숫자의 차가 존재하는 것을 말한다.

ex) 2, 7, 12, 17, 22 ...

5 5 5 5 => 여기서 공차는 5

ex.

시작 숫자 : 4

공차 : 3

```
4 7 10 13 16 19 22 25 28 31
```

▶ 실습문제12

메소드 명 : public void practice12(){}

정수 두 개와 연산자(문자열로 입력 받고 입력된 연산자에 따라 알맞은 결과를 출력하세요.

단, 해당 프로그램은 연산자 입력에 "exit"라는 값이 들어올 때까지 무한 반복하며

exit가 들어오면 "프로그램을 종료합니다."를 출력하고 종료합니다.

또한 연산자가 나누기이면서 두 번째 정수가 0으로 들어오면

"0으로 나눌 수 없습니다. 다시 입력해주세요."를 출력하며,

없는 연산자가 들어올 시 "없는 연산자입니다. 다시 입력해주세요."라고 출력하고

두 경우 모두 처음으로 돌아가 사용자가 다시 연산자부터 입력하도록 하세요.

[다음 장 출력 예시 참고]

▶ 실습문제12

```
연산자(+, -, *, /, %) : +
정수1 : 10
정수2 : 4
10 + 4 = 14

연산자(+, -, *, /, %) : /
정수1 : 10
정수2 : 4
10 / 4 = 2

연산자(+, -, *, /, %) : ^
정수1 : 10
정수2 : 4
없는 연산자입니다. 다시 입력해주세요.

연산자(+, -, *, /, %) : /
정수1 : 10
정수2 : 0
0으로 나눌 수 없습니다. 다시 입력해주세요.

연산자(+, -, *, /, %) : exit
프로그램을 종료합니다.
```