

제 10 강

제어문3: 반복문

교재:p64~68

목차

1. 제어문: 반복문2
 - while 문
 - do-while 문

제어문 복습

1) 제어문의 정의

프로그램의 흐름을 제어하는 문법

2) 제어문의 종류

- (1) 조건문: 정해진 조건에 따라 흐름을 제어하는 문법
- (2) 반복문: 정해진 조건에 따라 흐름을 반복하는 문법

제어문: 반복문이란?

1) 반복문의 정의

“ 반복문이란 특정 작업을 반복적으로
수행하고 싶을 때 사용하는 문법 ”

2) 반복문의 종류

- (1) for문
- (2) while문
- (3) do-while문

1. while문(p64)

“조건이 참일 동안에 ~ 명령을 반복해라!”

구조

조건이 참일 동안에 { } 안에 명령을 반복!

```
while [조건식]{  
    반복할 명령;  
}
```

조건식

: true 혹은 false의 결과값을 갖는 식

1. while문(p64)

<실습 WhileEx.java>

```
1 package day10;
2
3 public class WhileEx {
4     public static void main(String[] args) {
5         int x=1;
6         while(x<=10) {
7             System.out.println("x:"+x);
8             x++;
9         }
10    }
11 }
```

Problems @ Javadoc Declaration Console

<terminated> WhileEx [Java Application] C:\Program Files\J...

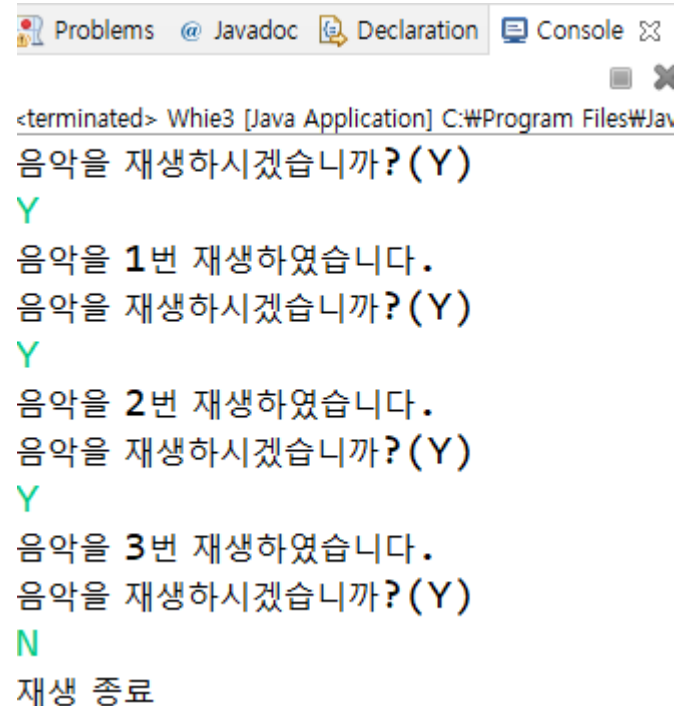
x:1
x:2
x:3
x:4
x:5
x:6
x:7
x:8
x:9
x:10

1. while문(p64)

<실습 Exam-22.java>

사용자에게 입력 받은 값이 Y라면 음악재생을 반복하고 아니라면 반복을 멈추는 코드를 구현해보자!

```
1 package day10;
2
3 import java.util.Scanner;
4
5 public class Whie3 {
6     public static void main(String[] args) {
7         Scanner sc= new Scanner(System.in);
8         String answer="Y"; //answer값을 Y로 초기화해야 while 실행 가능
9         int count=0;
10        while(answer.equals("Y")){
11            System.out.println("음악을 재생하시겠습니까?(Y)");
12            answer=sc.nextLine();
13            if(answer.equals("Y")) {
14                System.out.printf("음악을 %d번 재생하였습니다.\n",++count);
15            }
16        }
17        System.out.println("재생 종료");
18    }
19 }
```



Problems @ Javadoc Declaration Console

<terminated> Whie3 [Java Application] C:\Program Files\Jav
음악을 재생하시겠습니까?(Y)
Y
음악을 1번 재생하였습니다.
음악을 재생하시겠습니까?(Y)
Y
음악을 2번 재생하였습니다.
음악을 재생하시겠습니까?(Y)
Y
음악을 3번 재생하였습니다.
음악을 재생하시겠습니까?(Y)
N
재생 종료

1. while문(p64)

<실습 UpdownGame.java>

랜덤으로 1~100까지의 숫자 하나가 지정되며, 사용자에게 숫자를 입력 받아

- 랜덤숫자 < 입력 값 : Down 출력 후 반복
- 랜덤숫자 > 입력 값 : Up 출력 후 반복
- 랜덤숫자 == 입력 값 : 정답 출력 후 반복종료

위 조건으로 실행되는 프로그램을 구현하기

```
1 package day10;
2 import java.util.*;
3 public class UpdownGame {
4     public static void main(String[] args) {
5         Scanner sc = new Scanner(System.in);
6         Random random = new Random();
7         int user = -1;
8         int rnum = random.nextInt(100) + 1;
9         while(rnum != user) {
10             System.out.print("숫자 입력:");
11             user = sc.nextInt();
12             if(rnum < user) {
13                 System.out.println("Down!");
14             }
15             else if(rnum > user) {
16                 System.out.println("Up!");
17             }
18             else {
19                 System.out.println("정답!");
20             }
21         }
22     }
23 }
```

Problems @ Javadoc Declaration Console

<terminated> UpdownGame [Java Application] C:\Program F

숫자 입력: 50

Down!

숫자 입력: 25

Down!

숫자 입력: 10

Up!

숫자 입력: 15

Up!

숫자 입력: 20

Up!

숫자 입력: 23

정답

2. do-while문(p66)

“ 일단 한번은 실행하고
조건이 참이라면, 명령을 반복해라!”

구조

일단 실행 한 후 조건을 비교해서, 참이라면, { } 안에 명령을 반복!

```
do{  
    반복할 명령;  
} while(조건식);
```

조건식

: true 혹은 false의 결과값을 갖는 식

2. do-while문(p66)

<실습 Exam-23.java>- while문과의 비교

```
1 package day10;
2
3 public class Do_while {
4     public static void main(String[] args) {
5         int i=100;
6         do{
7             System.out.println("적어도 한번은 출력되는 문장");
8         }while(i<10);
9         System.out.println("종료");
10    }
11 }
```

Problems @ Javadoc Declaration Console

:terminated> Do_while [Java Application] C:\Program Files\W
적어도 한번은 출력되는 문장
종료

```
1 package day10;
2
3 public class Do_while {
4     public static void main(String[] args) {
5         int i=100;
6         while(i<10){
7             System.out.println("적어도 한번은 출력되는 문장");
8         }
9         System.out.println("종료");
10    }
11 }
```

Problems @ Javadoc Declaration Console

<terminated> Do_while [Java Application] C:\Program Files\W
종료

3. while문 vs do-while문

“

while 문은 조건식의 결과값이 거짓이라면
단 한번도 실행되지 않고 종료하나

do - while 문은 조건식의 결과값이 거짓이라면
한번은 실행되고 종료한다.

”