

Revisit Java Programming (Loop)

Fall, 2020

Table of Contents

- Loop
 - for
 - while
 - do while
 - nested loop
 - continue & break

Loop

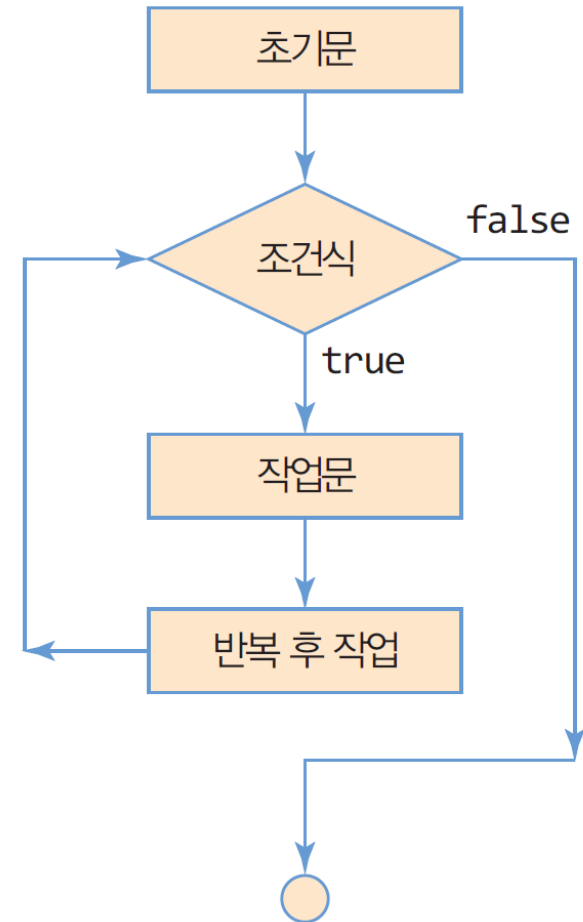
- Loop statements
 - for
 - while
 - do while



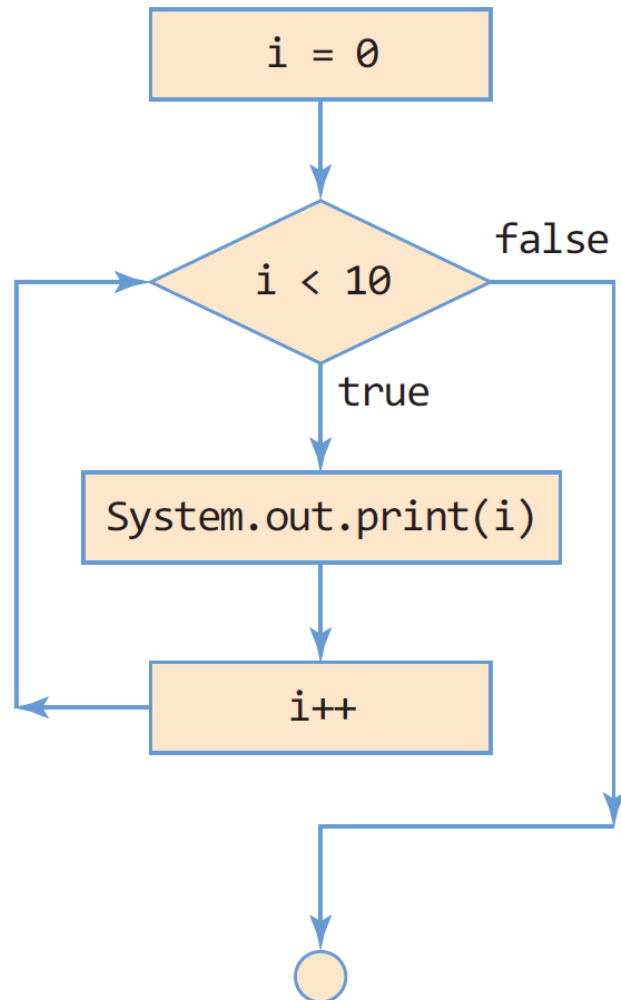
Loop statement: for

for (초기문; 조건식; 반복 후 작업) {
 ..작업문..
}

1 2 4
3



Loop statement: for



```
for(i=0; i<10; i++) {  
    System.out.print(i);  
}
```

0123456789

Loop statement: for (example)

- Print out from 0 to 9

```
int i;  
for(i = 0; i < 10; i++) {  
    System.out.print(i);  
}
```

```
int i;  
for(i = 0; i < 10; i++)  
    System.out.print(i);
```

- You can define a variable in for statement

```
for(int i = 0; i < 10; i++)    // 변수 i는 for문을 벗어나서 사용할 수 없음  
    System.out.print(i);
```

- Sum from 0 to 100

```
int sum = 0;  
for(int i = 0; i <= 100; i++)  
    sum += i;
```

```
int i, sum;  
for(i = 0, sum=0; i <= 100; i++)  
    sum += i;
```

```
int sum = 0;  
for(int i = 100; i >= 0; i--)  
    sum += i;
```

Loop statement: for (special cases)

```
for(초기작업; true; 반복후작업) { // 반복 조건이 true이면 무한 반복  
.....  
}
```

```
for(초기작업;; 반복후작업) { // 반복조건이 비어 있으면 true로 간주, 무한 반복  
.....  
}
```

```
// 초기 작업과 반복후작업은 ','로 분리하여 여러 문장 나열 가능  
for(i=0; i<10; i++, System.out.println(i)) {  
.....  
}
```

```
// for문 내에 변수 선언  
for(int i=0; i<10; i++) { // 변수 i는 for문 내에서만 사용 가능  
.....  
}
```

Practice 1: 예제 3-1 : for 문을 이용하여 1부터 10까지 합 출력

for문을 이용하여 1부터 10까지 덧셈을 표시하고 합을 구하시오.

$$1+2+3+4+5+6+7+8+9+10=55$$

Practice 1: 예제 3-1 : for 문을 이용하여 1부터 10까지 합 출력

for문을 이용하여 1부터 10까지 덧셈을 표시하고 합을 구하시오.

```
public class ForSample {  
    public static void main(String[] args) {  
        int sum=0;  
  
        for(int i=1; i<=10; i++) { // 1~10까지 반복  
            sum += i;  
            System.out.print(i); // 더하는 수 출력  
  
            if(i<=9) // 1~9까지는 '+' 출력  
                System.out.print("+");  
            else { // i가 10인 경우  
                System.out.print("="); // '=' 출력하고  
                System.out.print(sum); // 덧셈 결과 출력  
            }  
        }  
    }  
}
```

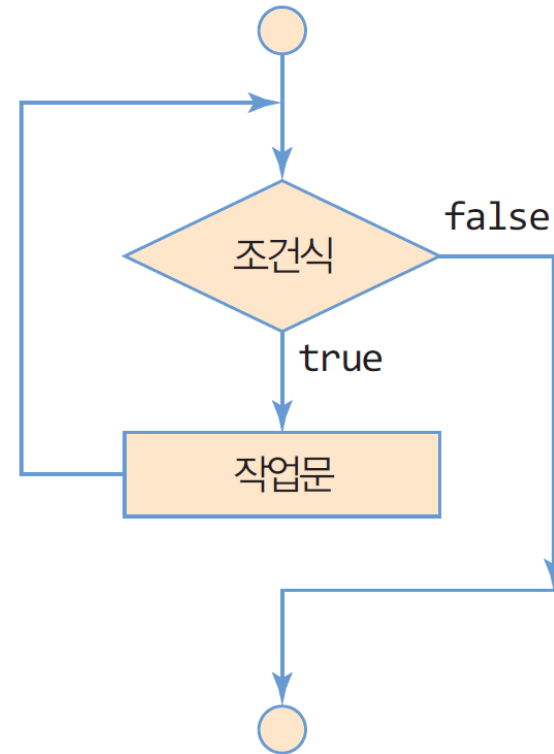
1+2+3+4+5+6+7+8+9+10=55

Loop statement: while

while (조건식) {
 ..작업문..
}

1

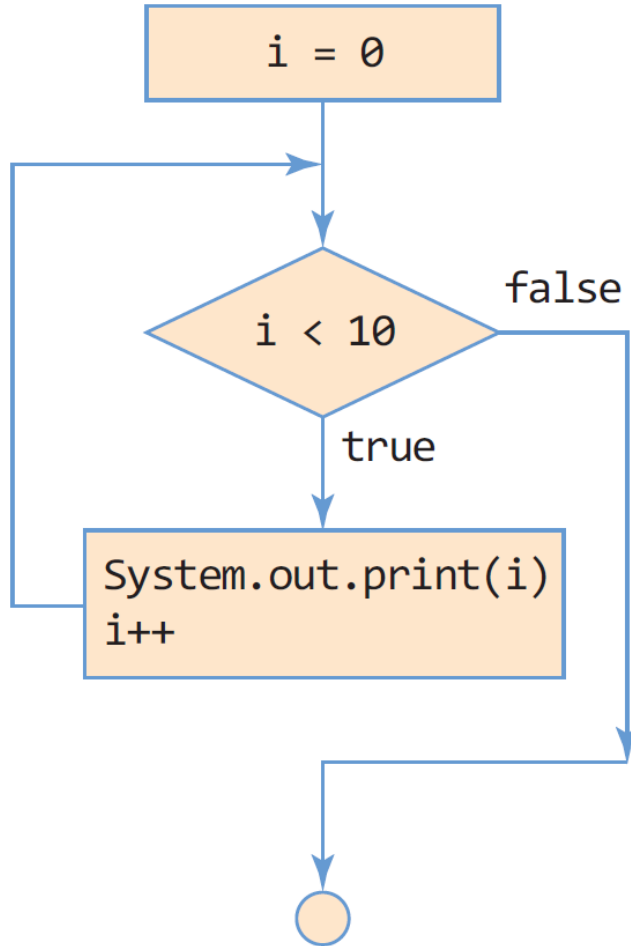
2



- The conditional statement will be evaluated in every repetition
 - true → do a task
 - false → escape from while statement

Loop statement: while

Each loop statement has the identical expressive power



```
i = 0;  
while(i<10) {  
    System.out.print(i);  
    i++;  
}
```

0123456789

Practice 2: 예제 3-2 : -1이 입력될 때까지 입력된 수의 평균 구하기

while문을 이용하여 정수를 여러 개 입력 받고 평균을 출력하라. -1이 입력되면 입력을 종료한다.

정수를 입력하고 마지막에 -1을 입력하세요.

10 30 -20 40 -1

-1은 마지막 입력을 뜻함

정수의 개수는 4개이며 평균은 15.0입니다.

Practice 2: 예제 3-2 : -1이 입력될 때까지 입력된 수의 평균 구하기

while문을 이용하여 정수를 여러 개 입력 받고 평균을 출력하라. -1이 입력되면 입력을 종료한다.

```
import java.util.Scanner;
public class WhileSample {
    public static void main(String[] args) {
        int count=0; // count는 입력된 정수의 개수
        int sum=0; // sum은 합
        Scanner scanner = new Scanner(System.in);
        System.out.println("정수를 입력하고 마지막에 -1을 입력하세요.");

        int n = scanner.nextInt(); // 정수 입력
        while(n != -1) { // -1이 입력되면 while 문 벗어남
            sum += n;
            count++;
            n = scanner.nextInt(); // 정수 입력
        }
        if(count == 0) System.out.println("입력된 수가 없습니다.");
        else {
            System.out.print("정수의 개수는 " + count + "개이며 ");
            System.out.println("평균은 " + (double)sum/count + "입니다.");
        }
        scanner.close();
    }
}
```

정수를 입력하고 마지막에 -1을 입력하세요.

10 30 -20 40 -1

정수의 개수는 4개이며 평균은 15.0입니다.

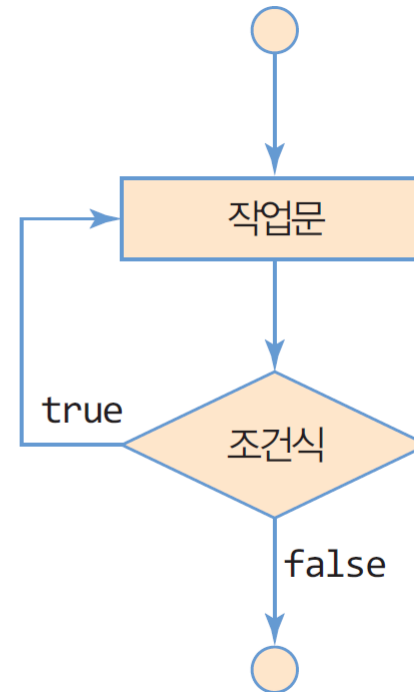
-1은 마지막 입력을 뜻함

Loop statement: do-while

```
do {  
    ..작업문..  
} while(조건식);
```

1

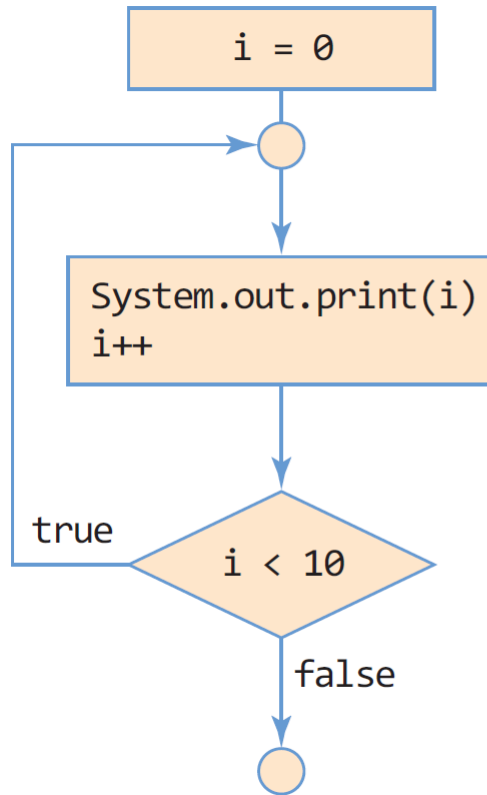
2



- 무조건 최소 한번 작업문 실행

- 반복 조건이 true이면 반복, false이면 반복 종료
- 반복 조건이 없으며 컴파일 오류

Loop statement: do-while



```
i = 0;  
do {  
    System.out.print(i);  
    i++;  
} while(i < 10);
```

0123456789

Practice 3: 예제 3-3 : a-z까지 출력

do-while 문을 이용하여 'a'부터 'z'까지 출력하는 프로그램을 작성하시오.

abcdefghijklmnopqrstuvwxyz

Practice 3: 예제 3-3 : a-z까지 출력

do-while 문을 이용하여 'a'부터 'z'까지 출력하는 프로그램을 작성하시오.

```
public class DoWhileSample {  
    public static void main (String[] args) {  
        char c = 'a';  
  
        do {  
            System.out.print(c);  
            c = (char) (c + 1);  
        } while (c <= 'z');  
    }  
}
```

abcdefghijklmnopqrstuvwxyz

Nested Loop

- Nested Loop
 - A loop statement inside another loop statement
 - Example
 - 9 x 9 multiplication table

1*1=1	1*2=2	1*3=3	1*4=4	1*5=5	1*6=6	1*7=7	1*8=8	1*9=9
2*1=2	2*2=4	2*3=6	2*4=8	2*5=10	2*6=12	2*7=14	2*8=16	2*9=18
3*1=3	3*2=6	3*3=9	3*4=12	3*5=15	3*6=18	3*7=21	3*8=24	3*9=27
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
5*1=5	5*2=10	5*3=15	5*4=20	5*5=25	5*6=30	5*7=35	5*8=40	5*9=45
6*1=6	6*2=12	6*3=18	6*4=24	6*5=30	6*6=36	6*7=42	6*8=48	6*9=54
7*1=7	7*2=14	7*3=21	7*4=28	7*5=35	7*6=42	7*7=49	7*8=56	7*9=63
8*1=8	8*2=16	8*3=24	8*4=32	8*5=40	8*6=48	8*7=56	8*8=64	8*9=72
9*1=9	9*2=18	9*3=27	9*4=36	9*5=45	9*6=54	9*7=63	9*8=72	9*9=81

Practice 4: 예제 3-4 : 2중 중첩을 이용한 구구단

2중 중첩 for문을 사용하여 구구단을 출력하는 프로그램을 작성하시오. 한 줄에 한 단씩 출력한다.

```
public class NestedLoop {  
    public static void main(String[] args) {  
        for(int i=1; i<10; i++) { // 1단에서 9단  
            for(int j=1; j<10; j++) { // 각 단의 구구셈 출력  
                System.out.print(i + "*" + j + "=" + i*j); // 구구셈 출력  
                System.out.print("\t"); // 하나씩 탭으로 띄기  
            }  
            System.out.println(); // 한 단이 끝나면 다음 줄로 커서 이동  
        }  
    }  
}
```

1*1=1	1*2=2	1*3=3	1*4=4	1*5=5	1*6=6	1*7=7	1*8=8	1*9=9
2*1=2	2*2=4	2*3=6	2*4=8	2*5=10	2*6=12	2*7=14	2*8=16	2*9=18
3*1=3	3*2=6	3*3=9	3*4=12	3*5=15	3*6=18	3*7=21	3*8=24	3*9=27
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
5*1=5	5*2=10	5*3=15	5*4=20	5*5=25	5*6=30	5*7=35	5*8=40	5*9=45
6*1=6	6*2=12	6*3=18	6*4=24	6*5=30	6*6=36	6*7=42	6*8=48	6*9=54
7*1=7	7*2=14	7*3=21	7*4=28	7*5=35	7*6=42	7*7=49	7*8=56	7*9=63
8*1=8	8*2=16	8*3=24	8*4=32	8*5=40	8*6=48	8*7=56	8*8=64	8*9=72
9*1=9	9*2=18	9*3=27	9*4=36	9*5=45	9*6=54	9*7=63	9*8=72	9*9=81

Loop statement: continue

- continue
 - Repeat the task again while not executing the following statements

```
for(초기문; 조건식; 반복 후 작업) {  
    .....  
    continue;  
    .....  
}
```

분기

```
while(조건식) {  
    .....  
    continue;  
    .....  
}
```

조건식으로
분기

```
do {  
    .....  
    continue;  
    .....  
} while(조건식);
```

조건식으로
분기

Practice 5: 예제 3-5 : continue 문을 이용하여 양수 합 구하기

5개의 정수를 입력 받고 그 중 양수들만 합하여 출력하는 프로그램을 작성하라.

정수를 5개 입력하세요.

5

-2

6

10

-4

양수의 합은 21

Practice 5: 예제 3-5 : continue 문을 이용하여 양수 합 구하기

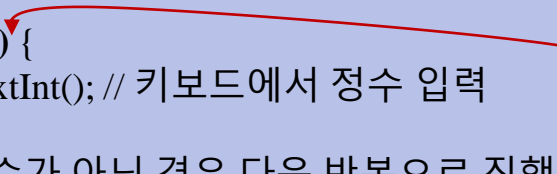
5개의 정수를 입력 받고 그 중 양수들만 합하여 출력하는 프로그램을 작성하라.

```
import java.util.Scanner;

public class ContinueExample {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.println("정수를 5개 입력하세요.");
        int sum=0;
        for(int i=0; i<5; i++) {
            int n = scanner.nextInt(); // 키보드에서 정수 입력
            if(n<=0)
                continue; // 양수가 아닌 경우 다음 반복으로 진행
            else
                sum += n; // 양수인 경우 덧셈
        }
        System.out.println("양수의 합은 " + sum);

        scanner.close();
    }
}
```



정수를 5개 입력하세요.

5

-2

6

10

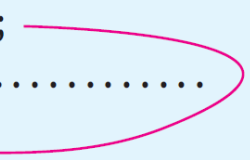
-4

양수의 합은 21

Loop statement: break

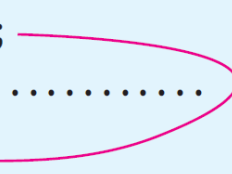
- break
 - Escape from a loop statement

```
for(초기문; 조건식; 반복 후 작업) {  
    .....  
    break;  
    .....  
}  
.....
```



(a) 현재 반복문 벗어나기

```
for(초기문; 조건식; 반복 후 작업) {  
    while(조건식) {  
        .....  
        break;  
        .....  
    }  
    .....  
}  
.....
```



(b) 중첩 반복에서 안쪽 반복문만 벗어나는 경우

Practice 6: 예제 3-6 : break 문을 이용하여 while 문 벗어나기

"exit"이 입력되면 while 문을 벗어나도록 break 문을 활용하는 프로그램을 작성하라.

```
exit을 입력하면 종료합니다.  
>>edit  
>>exit  
종료합니다...
```


Practice 6: 예제 3-6 : break 문을 이용하여 while 문 벗어나기

"exit"이 입력되면 while 문을 벗어나도록 break 문을 활용하는 프로그램을 작성하라.

```
import java.util.Scanner;

public class BreakExample {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.println("exit을 입력하면 종료합니다.");
        while(true) {
            System.out.print(">>");
            String text = scanner.nextLine();
            if(text.equals("exit")) // "exit"이 입력되면 반복 종료
                break; // while 문을 벗어남
        }
        System.out.println("종료합니다...");

        scanner.close();
    }
}
```

문자열 비교 시 equals()
사용

exit을 입력하면 종료합니다.
>>edit
>>exit
종료합니다...

Practice 7

- Practice for loop and debug!

```
3  public class P1ForLoop {
4
5      public static void main(String[] args) {
6          // For loop enables you to repeat some logics
7
8          // See how the variable i changes
9          for(int i = 0 ; i < 10 ; i++) {
10             System.out.println(i);
11         }
12     }
13 }
```

Practice 8

- Practice for loop and debug!

```
3  public class P2ForLoop2 {
4
5      public static void main(String[] args) {
6          // For loop enables you to repeat some logics
7
8          // Sum 0 to 99
9          int sum = 0;
10         for (int i = 0; i < 100; i++) {
11             sum += i;
12         }
13         System.out.println(sum);
14
15         // How about 30 to 77 ?
16     }
17 }
```

Practice 9

- Practice for loop and debug!

```
3  public class P3ForLoop3 {
4
5      public static void main(String[] args) {
6          // For loop enables you to repeat some logics
7
8          // Sum 0 to 99
9          // What if 2nd element of for statement is fixed to true
10         int sum = 0;
11         for (int i = 0; true ; i++) {
12             // use if and break
13             if(i == 100)
14                 break;
15             sum += i;
16         }
17         System.out.println(sum);
18     }
19 }
```

Practice 10

- Practice for loop and debug!

```
21
20
19      }
18      System.out.println(sum);
17      }
16
15      sum += i;
14      break;
13      if (i == 100)
12          // use if and break
11      for (int i = 0; true;) {
10      int sum = 0;
9      // What if 3rd element of for statement is not used
8      // Sum 0 to 99
7      // For loop enables you to repeat some logics
6      public static void main(String[] args) {
5
4      public class P4ForLoop4 {
3
2
```

Practice 11

- Practice for loop and debug!

```
3  public class P5ForLoop5 {
4
5      public static void main(String[] args) {
6          // For loop enables you to repeat some logics
7
8          // Sum 0 to 99
9          // What if 1st element of for statement is not used
10         int sum = 0;
11         int i = 0;
12         for (; true ;) {
13             // use if and break
14             if (i == 100)
15                 break;
16             sum += i;
17
18             i++;
19         }
20         System.out.println(sum);
21     }
22 }
```

Practice 12

- Practice for loop and debug!

```
3  public class P7WhileLoop2 {
4
5      public static void main(String[] args) {
6          // For loop enables you to repeat some logics
7
8          // Sum 0 to 99
9          // Of course, you can use the terminal condition into while(?)
10         int sum = 0;
11         int i = 0;
12         while (i != 100) {
13             // use if and break
14             sum += i;
15             i++;
16         }
17         System.out.println(sum);
18     }
19 }
```

Practice 13

- Practice for loop and debug!

```
~
3  public class P9LoopPractice2 {
4
5      public static void main(String[] args) {
6          // Print out multiplication table
7          // using nested loops
8
9          // 1*1=1      1*2=2   1*3=3   1*4=4   1*5=5   1*6=6   1*7=7   1*8=8   1*9=9
10         // 2*1=1      2*2=4   2*3=6   2*4=8   2*5=10  2*6=12  2*7=14  2*8=16  2*9=18
11         // ...
12         // ...
13         // ...
14         // ...
15         // ...
16         // ...
17         // ...
18     }
19 }
```


Practice 14

- Practice for loop and debug!

```
3  public class P10LoopPractice3 {  
4  
5      public static void main(String[] args) {  
6          // until the input becomes -1  
7          // get the 'EVEN' integers and print out their averages  
8          // using continue  
9      }  
10 }
```

Summary

- Loop
 - for
 - while
 - do while
 - nested loop
 - continue & break