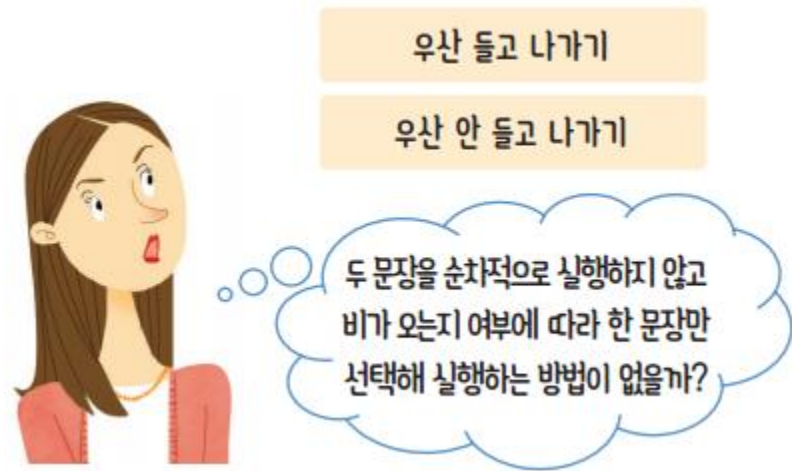


# 제어문과 메서드



# 제어문

## ■ 필요성



# 제어문

- 제어문은 실행문의 수행 순서를 변경



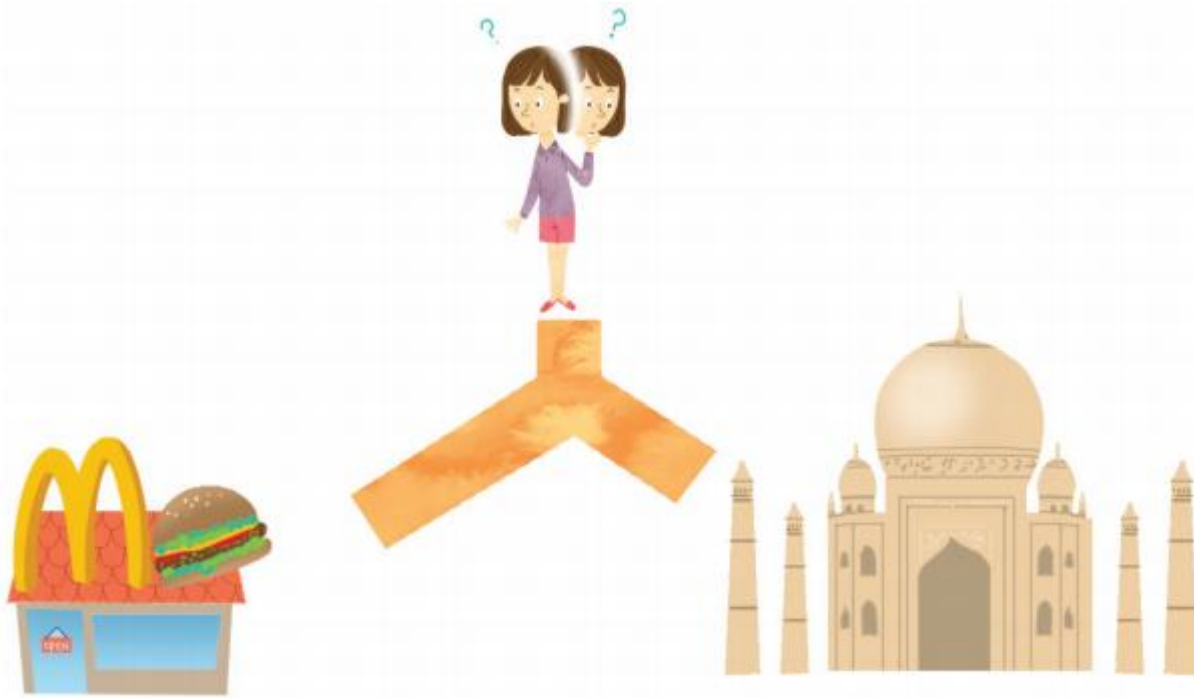
- 종류

- 조건문, 반복문, 분기문



# 조건문

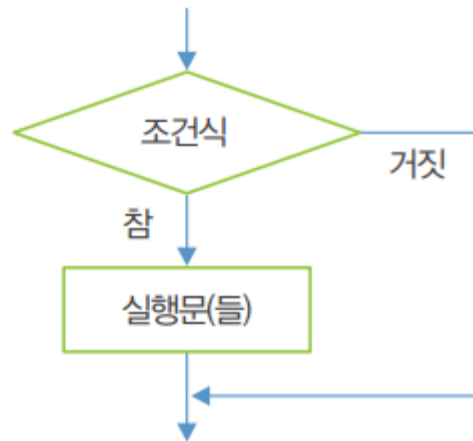
- 조건에 따라 실행문을 선택을 할 때 사용



# 조건문

## ■ 단순 if 문

```
if (조건식) {  
    실행문(들);  
}
```



- 예제 : [sec02/SimpleIfDemo](#)

숫자를 입력하세요 : 2  
짝수!  
종료

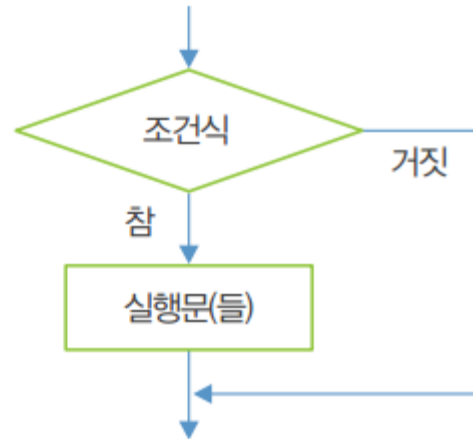
숫자를 입력하세요 : 3  
홀수!  
종료



# 조건문

## ■ 단순 if 문

```
if (조건식) {  
    실행문(들);  
}
```



### ● 예제 : [sec02/SimpleIfDemo](#)

```
숫자를 입력하세요 : 2  
짝수!  
종료
```

```
숫자를 입력하세요 : 3  
홀수!  
종료
```

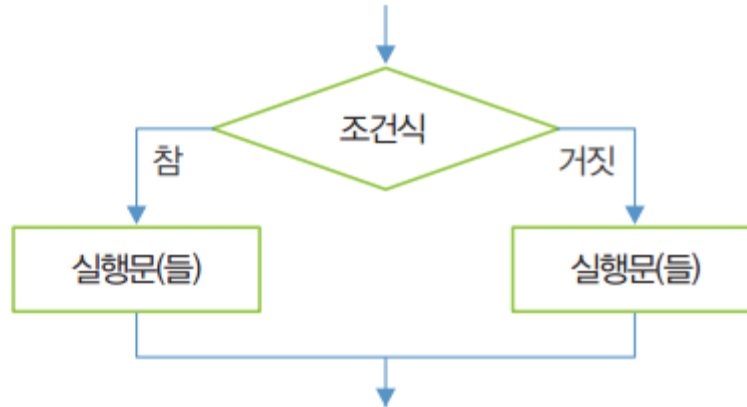
```
SimpleIfDemo - 메모장  
파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)  
import java.util.Scanner;  
  
public class SimpleIfDemo {  
    public static void main(String[] args) {  
        Scanner in = new Scanner(System.in);  
        System.out.print("숫자를 입력하세요 : ");  
        int number = in.nextInt();  
  
        if (number % 2 == 0)  
            System.out.println("짝수!");  
        if (number % 2 == 1)  
            System.out.println("홀수!");  
        System.out.println("종료");  
    }  
}
```



# 조건문

## ■ if~else 문

```
if (조건식) {  
    실행문(들);  
} else {  
    실행문(들);  
}
```



- 예제 : [sec02/IfElseDemo](#)

```
IfElseDemo - 메모장
파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)
import java.util.Scanner;

public class IfElseDemo {
    public static void main(String[] args) {
        Scanner in = new Scanner(System.in);
        System.out.print("숫자를 입력하세요 : ");
        int number = in.nextInt();

        if (number % 2 == 0)
            System.out.println("짝수!");
        else
            System.out.println("홀수!");
        System.out.println("종료");
    }
}
```



# 조건문

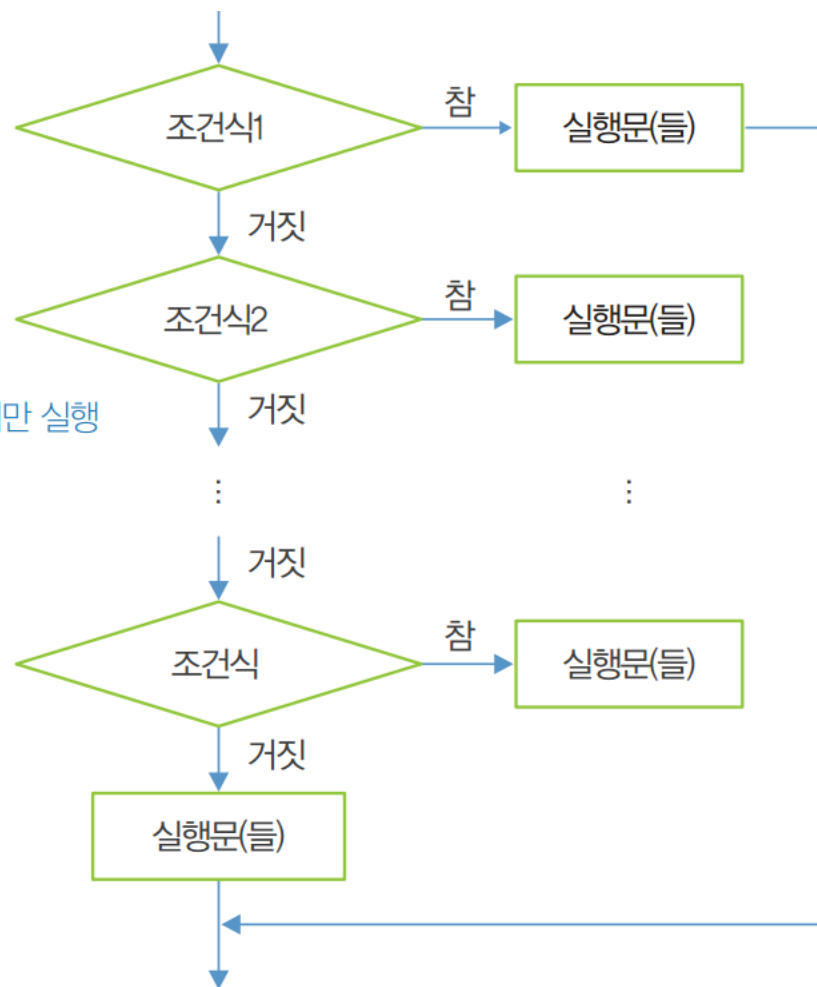
## ■ 다중 if 문

```
if (조건식1) {  
    실행문(들); 조건식1이 참일 때만 실행  
}  
else if (조건식2) {  
    실행문(들); 조건식1이 거짓이며 조건식2가 참일 때만 실행  
}  
else if (조건식3) {  
    ...  
}  
else {  
    실행문(들); 모든 조건을 만족하지 않을 때만 실행  
}
```

- 예제 : [sec02/MultilfDemo](#)

점수를 입력하세요 : 95  
당신의 학점은 A

점수를 입력하세요 : 87  
당신의 학점은 B





# 조건문

## ■ 다중 if 문

- 예제 : [sec02/MultilfDemo](#)



```
MultilfDemo - 메모장
파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)
import java.util.Scanner;

public class MultilfDemo {
    public static void main(String[] args) {
        Scanner in = new Scanner(System.in);
        String grade;
        System.out.print("점수를 입력하세요 : ");
        int score = in.nextInt();

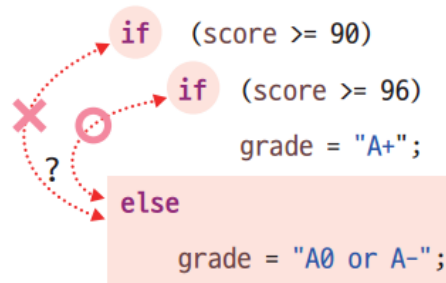
        if (score >= 90)
            grade = "A";
        else if (score >= 80)
            grade = "B";
        else if (score >= 70)
            grade = "C";
        else if (score >= 60)
            grade = "D";
        else
            grade = "F";
        System.out.println("당신의 학점은 " + grade);
    }
}
```



# 조건문

## ■ 중첩 if 문

- if 문에 다른 if 문이 포함되는 것을 중첩 if 문이라고 한다
- 주의 사항



```
if (score >= 90) {  
    if (score >= 96)  
        grade = "A+";  
    else  
        grade = "A0 or A-";  
}
```

- 예제 : [sec02/NestedIfDemo](#)

```
import java.util.Scanner;
```

```
public class NestedIfDemo {  
    public static void main(String[] args) {  
        Scanner in = new Scanner(System.in);  
        String grade;  
        System.out.print("점수를 입력하세요 : ");  
        int score = in.nextInt();
```

```
        if (score >= 90)  
            grade = "A";  
        else {  
            if (score >= 80)  
                grade = "B";  
            else {  
                if (score >= 70)  
                    grade = "C";  
                else {  
                    if (score >= 60)  
                        grade = "D";  
                    else  
                        grade = "F";  
                }  
            }  
        }  
        System.out.println("당신의 학점은 " + grade);  
    }  
}
```



# 반복문

## ■ 조건에 따라 같은 처리를 반복



while 문, do~while 문

종이 한 쪽을 다 채울 때까지 반복해 쓰기  
반복할 조건을 안다.



for 문

100번 반복해 쓰기  
반복 횟수를 안다.

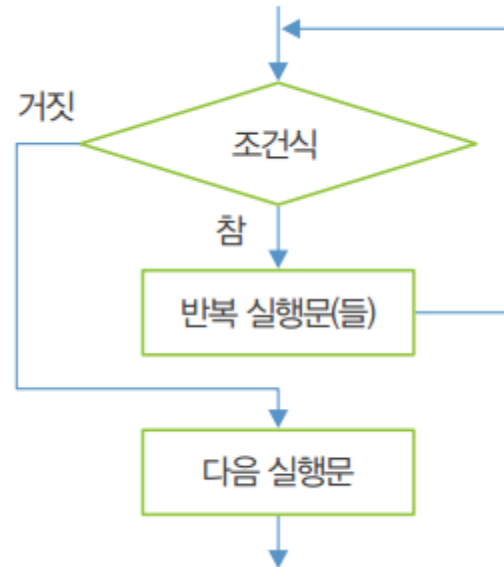


# 반복문

## ■ while 문

조건식이 거짓이면 본체를  
한 번도 실행하지 않는다.

```
while (조건식) {  
    반복 실행문(들);  
}
```



본체를 탈출할  
실행문이 필요하다.

```
while (true) {  
    반복 실행문(들);  
}
```

(a) 오류 미발생

```
while (false) {  
    반복 실행문(들);  
}
```

도달하지 않는 코드라는  
오류를 발생시킨다.

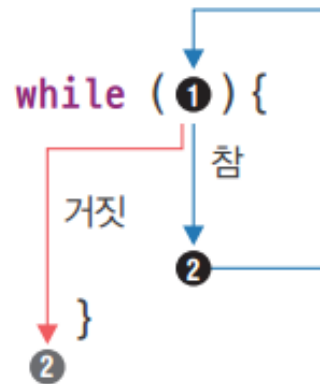
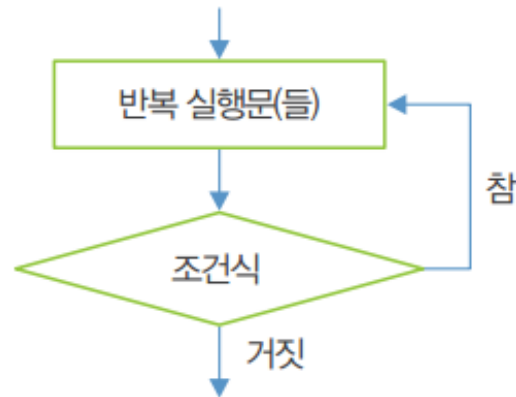
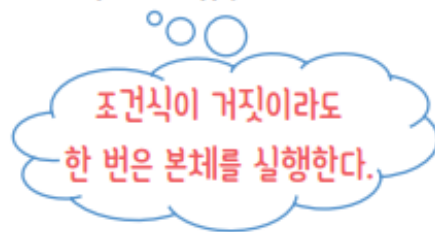
(b) 오류 발생



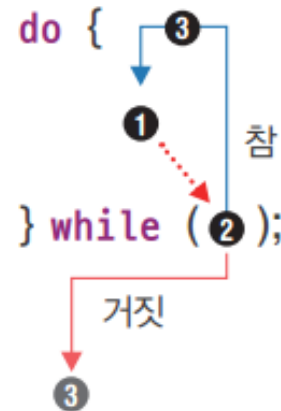
# 반복문

## do~while 문

```
do {  
    반복 실행문(들); 본체  
} while (조건식);
```



(a) while 문



(b) do~while 문



# 반복문

## ■ while 문

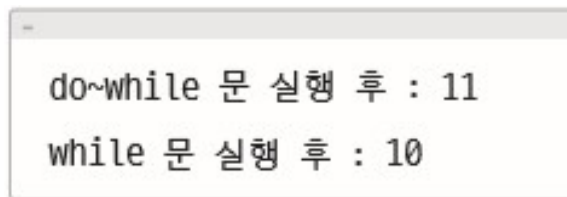
- 예제 : [sec03/While1Demo](#)



1234

```
public class DoWhile1Demo {  
    public static void main(String[] args) {  
        int i = 1;  
        do {  
            System.out.print(i);  
            i++;  
        } while (i < 5);  
    }  
}
```

- 예제 : [sec03/While2Demo](#)



do~while 문 실행 후 : 11  
while 문 실행 후 : 10

```
public class DoWhile2Demo {  
    public static void main(String[] args) {  
        int i = 10;  
        do {  
            i++;  
        } while (i < 5);  
        System.out.println("do~while 문 실행 후 : " + i);  
  
        i = 10;  
        while (i < 5) {  
            i++;  
        }  
        System.out.println("while 문 실행 후 : " + i);  
    }  
}
```



# 반복문

## ■ do~while 문

- 예제 : [sec03/DoWhile3Demo](#)

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

```
public class DoWhile3Demo {  
    public static void main(String[] args) {  
        int row = 2;  
        do {  
            int column = 1;  
            do {  
                System.out.printf("%4d", row *  
column);  
                column++;  
            } while (column < 10);  
            System.out.println();  
            row++;  
        } while (row < 10);  
    }  
}
```

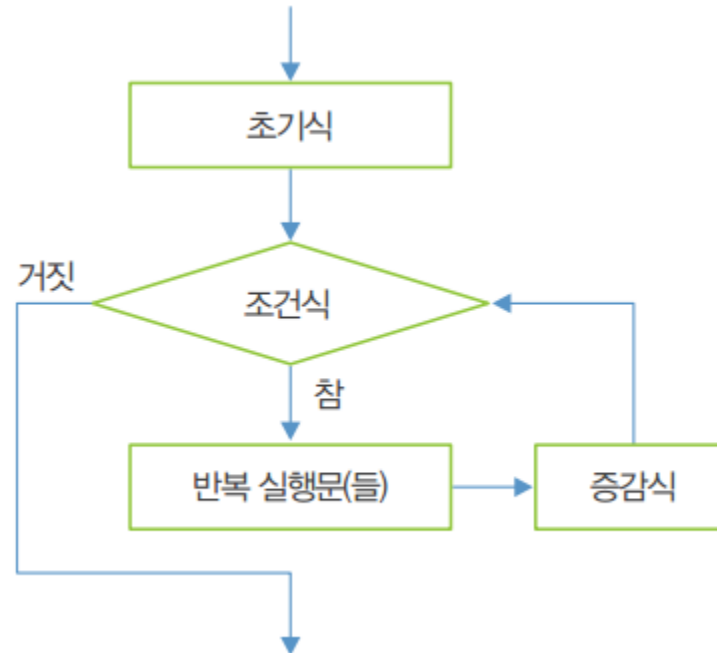


# 반복문

## ■ for 문

조건식이 거짓이면 본체와 증감식은  
한 번도 실행되지 않는다.

```
for (초기식; 조건식; 증감식) {  
    반복 실행문(들); 본체  
}
```





# 반복문

## ■ for 문

첫 번째 반복

```
for (int i = 0; i < 2; i++) {  
    System.out.println(i);  
}
```

두 번째 반복

```
for (int i = 0; i < 2; i++) {  
    System.out.println(i);  
}
```

세 번째 반복

```
for (int i = 0; i < 2; i++) {  
    System.out.println(i);  
}
```

현재는 i가 2므로 조건이  
거짓이 되어 반복 종료



# 반복문

## ■ for 문

```
for ( ; ; )    // 무한 반복문  
;
```

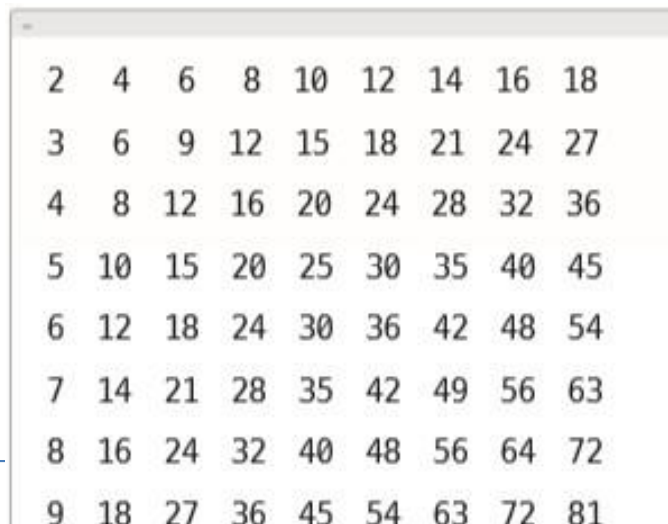
```
      초기식      조건식      증감식  
for (int i = 0, j = 10; i < j; i++, j--) {  
    ... 본체  
}
```

- 예제 : [sec03/For1Demo](#)



1234

- 예제 : [sec03/For2Demo](#)



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



# 반복문

## ■ for 문

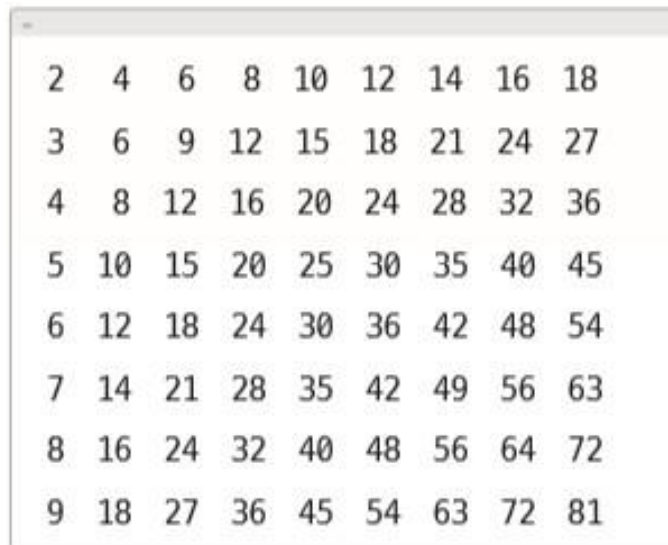
- 예제 : [sec03/For1Demo](#)



1234

```
public class For1Demo {  
    public static void main(String[] args) {  
        for (int i = 1; i < 5; i++)  
            System.out.print(i);  
    }  
}
```

- 예제 : [sec03/For2Demo](#)



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

```
public class For2Demo {  
    public static void main(String[] args) {  
        for (int row = 2; row < 10; row++) {  
            for (int column = 1; column < 10; column++) {  
                System.out.printf("%4d", row * column);  
            }  
            System.out.println();  
        }  
    }  
}
```

