

*bantaehak*

# Chapter 05

## 조건문



# 목차

1. 조건문의 개요
2. if문
3. switch~case문

[실전 예제] 사춘기 거북이 지도하기

## Preview



# 학습목표

- 세 가지 기본 제어 구조에 대해 알아봅니다.
- 기본 if문 사용법을 익힙니다.
- if~else문과 else if문 사용법을 익힙니다.
- switch~case문 사용법을 익힙니다.

# Section 01

## 조건문의 개요

# 1. 기본 제어 구조

## ■ 순차구조

- 순차적(sequential)으로 코드가 실행되는 것
- '시작'부터 '종료'까지 각 단계가 순서대로 실행되는 순차 구조
- 가장 단순한 구조이지만 실제로 많이 사용되는 구조임

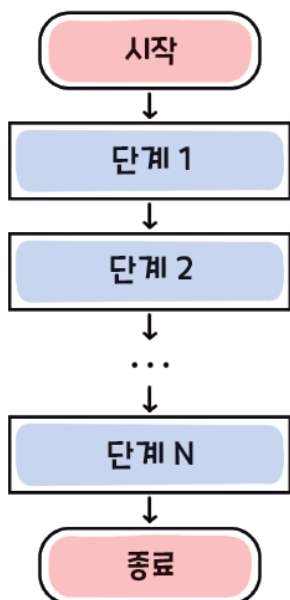


그림 5-1 물의 흐름에 비유되는 순차 구조

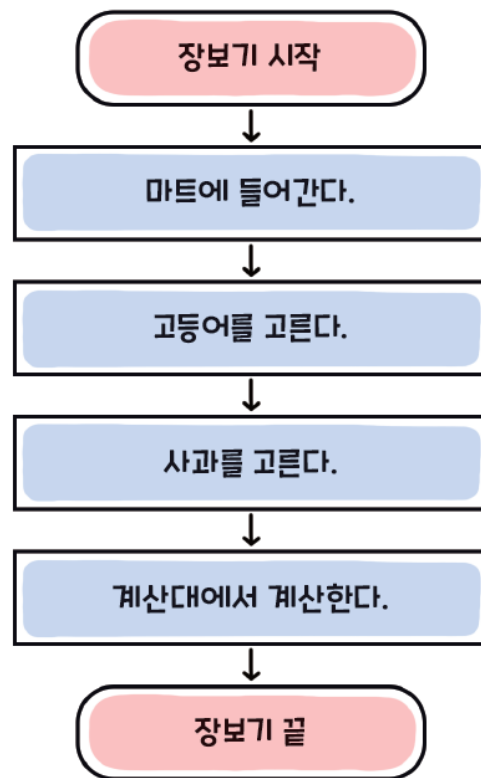


그림 5-2 순차 구조의 예

# 1. 기본 제어 구조

## ■ 선택구조

- 두 가지 중에서 선택한 쪽으로 코드가 실행되는 것
- 조건이 있을 때 둘 중에서 선택하는 선택 구조로 실제 프로그램에서 많이 볼 수 있음

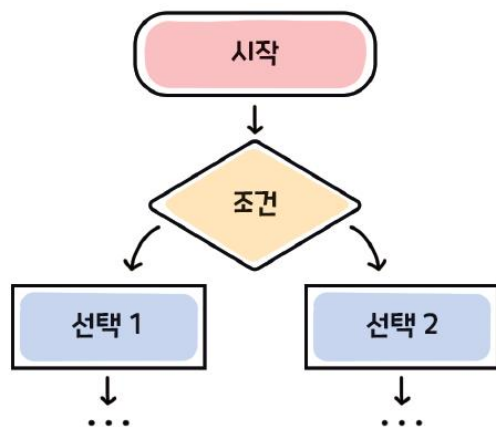


그림 5-3 갈림길에 비유되는 선택 구조

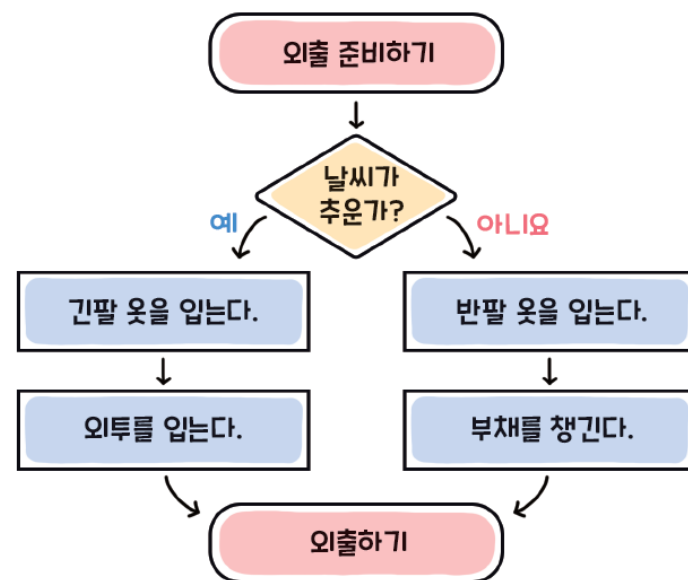


그림 5-4 선택 구조의 예

# 1. 기본 제어 구조

## ■ 반복구조

- 동일한 코드가 반복적으로 여러 번 실행되는 것

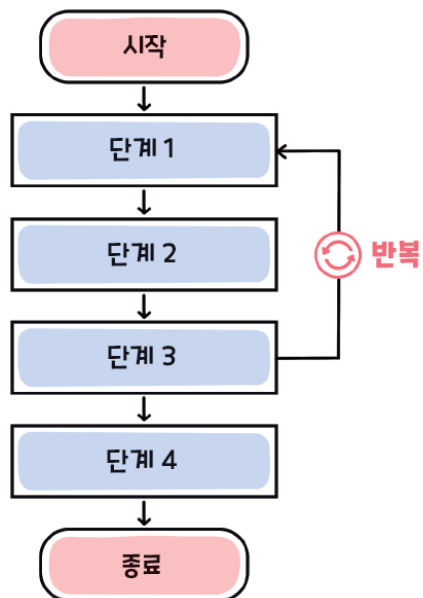


그림 5-5 방글방글 도는 강아지에 비유되는 반복 구조

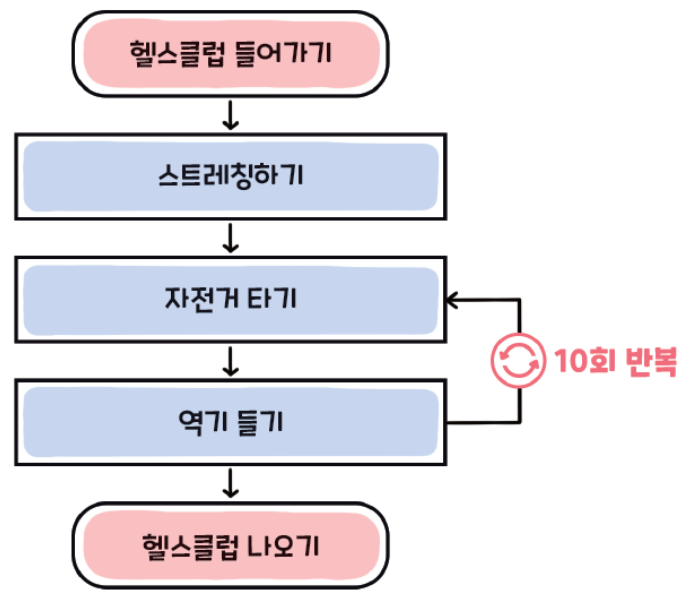


그림 5-6 반복 구조의 예



# 1. 기본 제어 구조

## 확인문제

다음 빈칸에 들어갈 내용이 바르게 짝지어진 것을 고르시오.

- 특정 단계가 반복적으로 실행되는 구조는  이다.
- 두 가지 중에서 선택한 쪽으로 실행되는 구조는  이다.

① ㉠ 선택 구조, ㉡ 순차 구조

② ㉠ 순차 구조, ㉡ 반복 구조

③ ㉠ 선택 구조, ㉡ 반복 구조

④ ㉠ 반복 구조, ㉡ 선택 구조

정답

Click!

## 2. 조건문의 필요성

### ■ 실생활에서 조건문이 활용되는 경우

- 운전면허 필기시험에서 70점 이상이면 합격, 70점 미만이면 불합격인 경우
- 비밀번호를 누르고 건물에 출입하는 경우
- 마트에서 카드로 계산하는 경우
- 지하철을 타기 위해 교통카드를 사용하는 경우 등

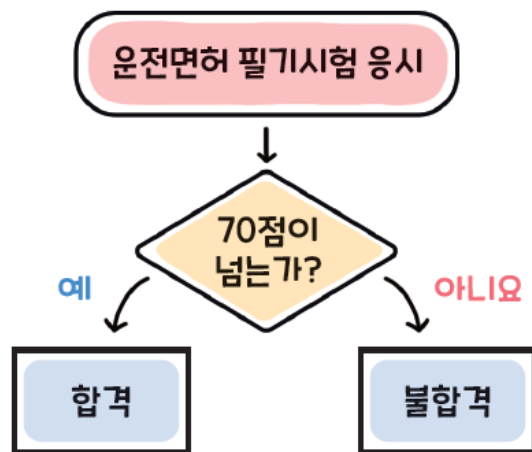


그림 5-7 운전면허 필기시험 채점의 흐름

## 2. 조건문의 필요성

### 확인문제

마트에서 치약을 구매하기 위해 카드로 결제할 때 정상적으로 결제되는 경우와 그렇지 않은 경우를 흐름도로 작성하시오.

### 정답



Click!

# Section 02

## if문

# 1. 기본 if문

## ■ if(조건식)

- 참일 때는 어떤 작업을 수행하고 거짓일 때는 아무것도 하지 않는 가장 단순한 if문의 형식

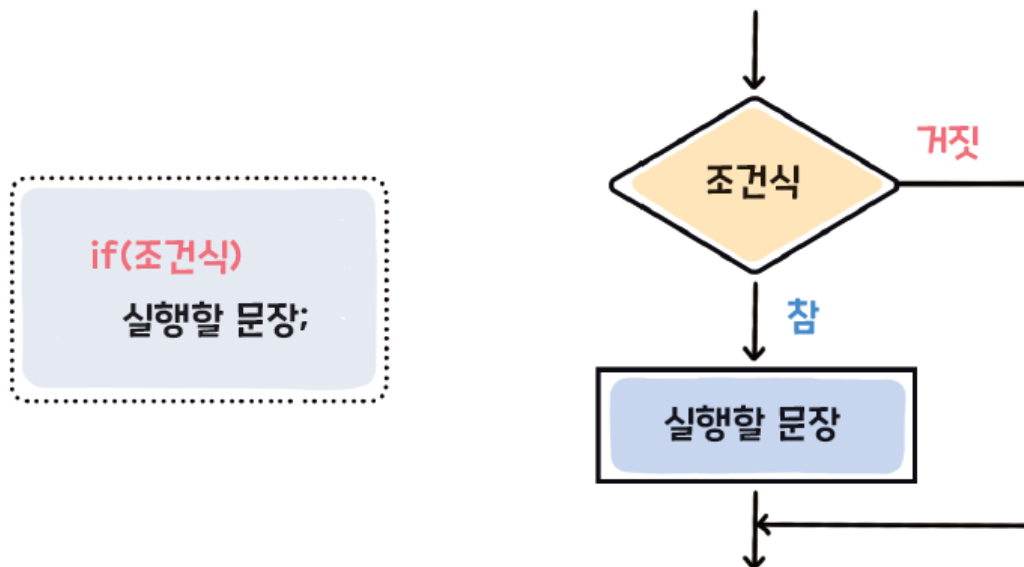


그림 5-8 기본 if문의 형식

# 1. 기본 if문

## ■ if(조건식)

코드 5-1

```
01 public class Code05_01 {  
02     public static void main(String[] args) {  
03         int num = 99;  
04  
05         if (num < 100)  
06             System.out.println("100보다 작습니다.");  
07     }  
08 }
```

100보다 작습니다.

Code05\_01.java

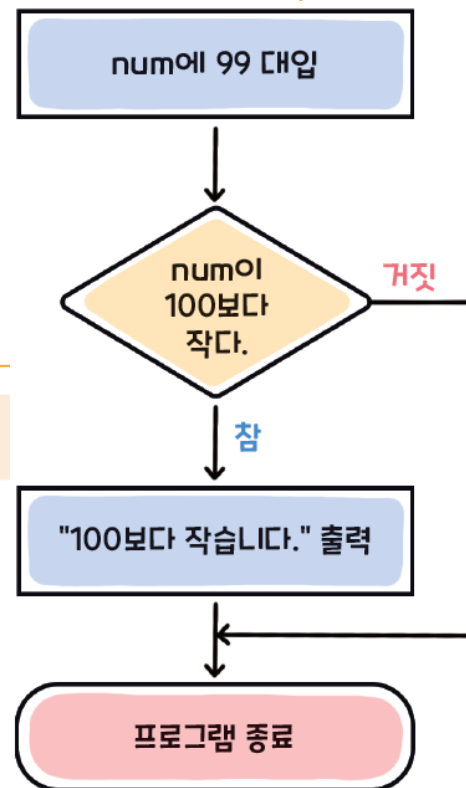


그림 5-9 [코드 5-1]의 실행 과정

# 1. 기본 if문

## ■ if(조건식)

- 실행할 문장이 두 개인 경우

코드 5-2

Code05\_02.java

```
01  public class Code05_02 {  
02      public static void main(String[] args) {  
03          int num = 99;  
04  
05          if (num < 100) {  
06              System.out.print("100보다 ");  
07              System.out.print("작습니다.");  
08          }  
09  
10      }  
11  }
```

100보다 작습니다.

## 2. if~else문

### ■ if~else문

- 조건이 참이면 문장 1을 실행하고, 조건이 거짓이면 문장 2를 실행하는 경우

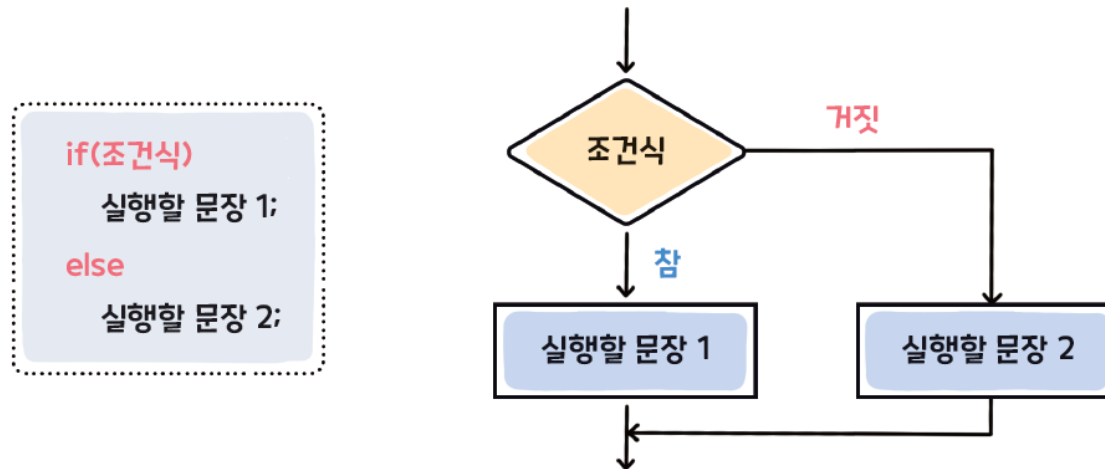


그림 5-10 if~else문의 형식



## 2. if~else문

### ■ if~else문

코드 5-3

```
01 public class Code05_03 {  
02     public static void main(String[] args) {  
03         int num = 200;  
04  
05         if (num < 100)  
06             System.out.println("100보다 작군요.");  
07         else  
08             System.out.println("100보다 크군요.");  
09  
10     }  
11 }
```

Code05\_03.java

100보다 크군요.

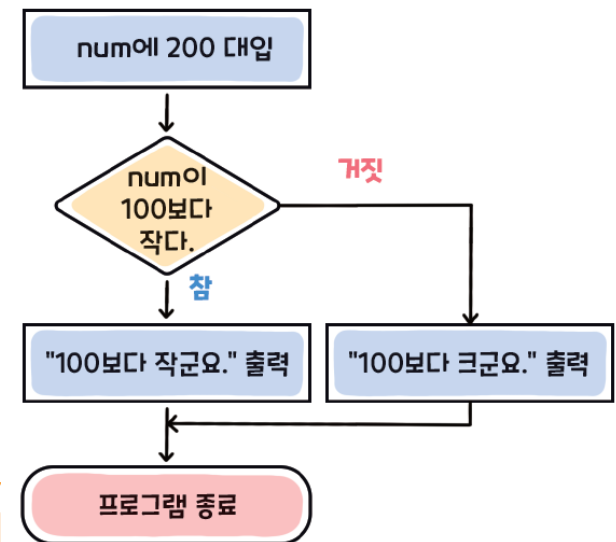


그림 5-11 [코드 5-3]의 실행 과정

[코드 5-3]에서는 if문 및 else문 이후에 실행할 문장이 한 개라 중괄호를 생략했지만, 다음과 같이 중괄호를 넣어도 됩니다.

```
if (num < 100) {  
    System.out.println("100보다 작군요.");  
} else {  
    System.out.println("100보다 크군요.");  
}
```



## 2. if~else문

### ■ if~else문

- 조건이 참일 때 실행할 문장이 여러 개이고 거짓일 때 실행할 문장도 여러 개인 경우

코드 5-4

Code05\_04.java

```
01 public class Code05_04 {  
02     public static void main(String[] args) {  
03         int num = 200;  
04  
05         if (num < 100) {  
06             System.out.println("100보다 작군요.");  
07             System.out.println("여기는 참입니다.");  
08         } else {  
09             System.out.println("100보다 크군요.");  
10             System.out.println("여기는 거짓입니다.");  
11         }  
12  
13         System.out.println("프로그램 끝!");  
14     }  
15 }
```

100보다 크군요.  
여기는 거짓입니다.  
프로그램 끝!

## 2. if~else문

### ■ if~else문

- 조건이 참일 때 실행할 문장이 여러 개이고 거짓일 때 실행할 문장도 여러 개인 경우

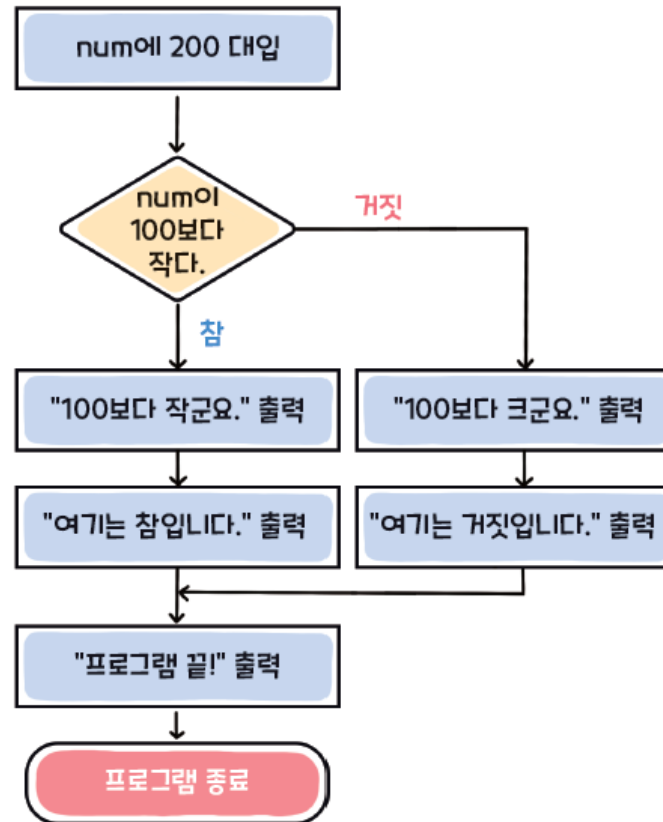


그림 5-12 [코드 5-4]의 실행 과정

## 2. if~else문

### ■ if~else문

- 사용자가 입력한 숫자가 짝수인지 홀수인지 확인하는 예제

코드 5-5

Code05\_05.java

```
01 import java.util.Scanner;
02 public class Code05_05 {
03     public static void main(String[] args) {
04         Scanner s = new Scanner(System.in);
05         int num;
06
07         System.out.print("숫자 입력 ==> ");
08         num = s.nextInt();
09
10         if (num%2 == 0)
11             System.out.println("짝수입니다.");
12         else
13             System.out.println("홀수입니다.");
14
15         s.close();
16     }
17 }
```

숫자 입력 ==> 1773 ● ————— 사용자가 입력  
홀수입니다.

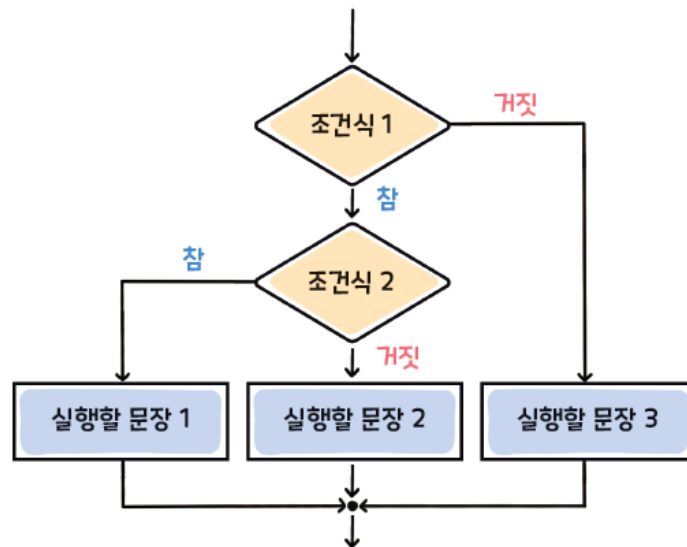
### 3. 중첩 if문

#### ■ 중첩 if문

- 조건을 검사하는 과정이 두 번 이상인 경우에 사용함
- if문 안에 또 다른 if문이 있는 것
- 조건식 1이 참이면 조건식 2를 수행
  - 조건식 2도 참이면 문장 1을 실행함. 그러나 조건식 2가 거짓이면 문장 2가 실행됨
- 조건식 1이 거짓이면 조건식 2를 건너뛰고 문장 3이 실행됨

```
if(조건식 1) {  
    if(조건식 2)  
        실행할 문장 1;  
    else  
        실행할 문장 2;  
}  
else  
    실행할 문장 3;
```

그림 5-13 중첩 if문의 형식



## 3. 중첩 if문

### ■ 중첩 if문

코드 5-6

Code05\_06.java

```
01 import java.util.Scanner;
02 public class Code05_06 {
03     public static void main(String[] args) {
04         Scanner s = new Scanner(System.in);
05         int num;
06
07         System.out.print("숫자 입력 ==> ");
08         num = s.nextInt();
09
10         if (num > 100) {
11             if (num < 1000) {
12                 System.out.println("100보다 크고 1000보다 작군요.");
13             } else {
14                 System.out.println("와~ 1000보다 크군요.");
15             }
16         } else {
17             System.out.println("음~ 100보다 작군요.");
18         }
19
20         s.close();
21     }
22 }
```

숫자를 입력 ==> 150 ● 사용자 입력  
100보다 크고 1000보다 작군요.

# 3. 중첩 if문

## ■ 중첩 if문

### ■ 학점 처리 예제

→ 점수를 입력받아 90점 이상은 A, 80점 이상은 B, 70점 이상은 C, 60점 이상은 D, 그 외는 F로 처리하는 프로그램

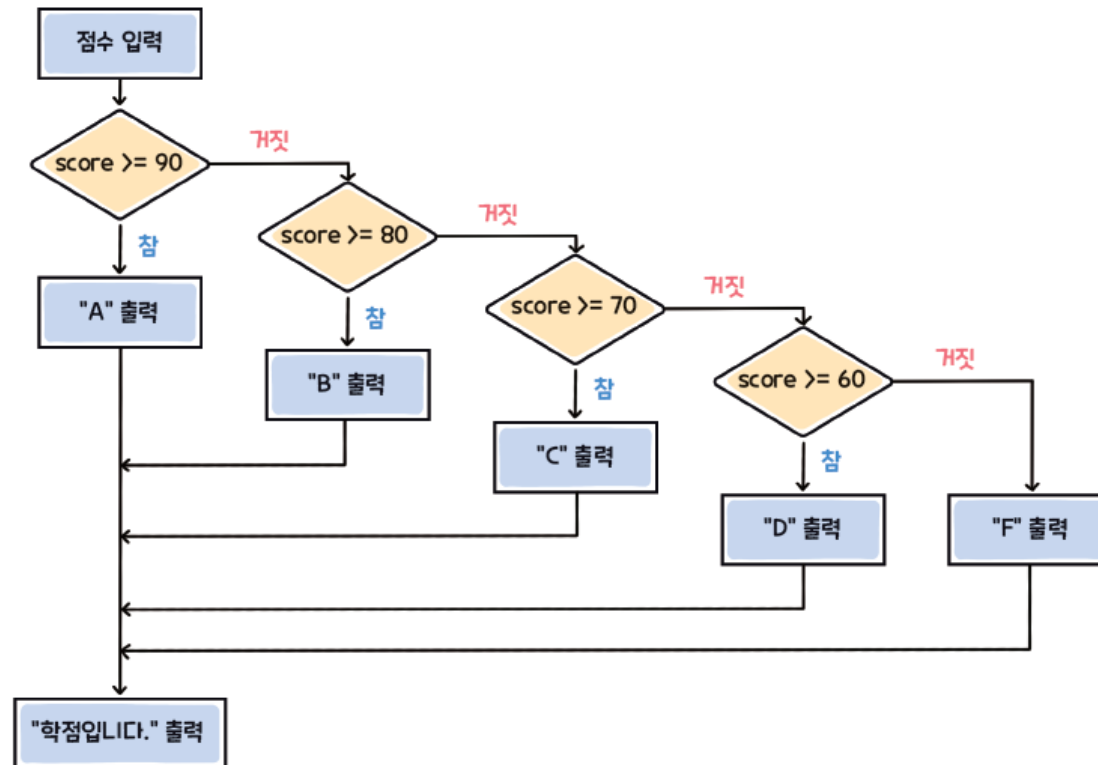


그림 5-14 [코드 5-7]의 실행 과정

# 3. 중첩 if문

## ■ 중첩 if문

코드 5-7

Code05\_07.java

```
01 import java.util.Scanner;
02 public class Code05_07 {
03     public static void main(String[] args) {
04         Scanner s = new Scanner(System.in);
05         int score;
06
07         System.out.print("점수 입력 ==> ");
08         score = s.nextInt();
09
10         if (score >= 90)
11             System.out.print("A");
12         else
13             if (score >= 80)
14                 System.out.print("B");
15             else
16                 if (score >= 70)
17                     System.out.print("C");
18                 else
19                     if (score >= 60)
20                         System.out.print("D");
21                     else
22                         System.out.print("F");
23
24         System.out.println(" 학점입니다.");
25         s.close();
26     }
27 }
```

점수 입력 ==> 77 ● ————— 사용자가 입력  
C 학점입니다.



## 4. else if문

### ■ else if문

- 중첩 if문은 if~else문 안에 또 if~else문이 들어 있어 복잡해 보임
- 그래서 중첩 if문을 사용할 때는中间的 else~if를 else if문으로 줄여서 사용함
- 학점 처리 예제([코드 5-7])를 else if문으로 변형하기

## 4. else if문

### ■ else if문

코드 5-8

Code05\_08.java

```
01 import java.util.Scanner;
02 public class Code05_08 {
03     public static void main(String[] args) {
04         Scanner s = new Scanner(System.in);
05         int score;
06
07         System.out.print("점수 입력 ==> ");
08         score = s.nextInt();
09
10         if (score >= 90)
11             System.out.print("A");
12         else if (score >= 80)
13             System.out.print("B");
14         else if (score >= 70)
15             System.out.print("C");
16         else if (score >= 60)
17             System.out.print("D");
18         else
19             System.out.print("F");
20
21         System.out.println(" 학점입니다.");
22         s.close();
23     }
24 }
```

점수 입력 ==> 77 ● ————— 사용자가 입력  
C 학점입니다.

## 4. else if문

### 확인문제

다음 빈칸에 들어갈 알맞은 말을 [보기]에서 고르시오.

- ① 자바에서 가장 기본적인 조건문은 이다.
- ② 조건식이 참일 때 수행할 문장과 거짓일 때 수행할 문장이 다른 경우에는 을 사용해야 한다.
- ③ 조건문에서 if~else if~else if~else와 같이 if문이 여러 번 중복되는 것을 이라고 한다.

### [보기]

- |            |            |         |
|------------|------------|---------|
| ㉠ if문      | ㉡ if~else문 | ㉢ for문  |
| ㉣ else~if문 | ㉤ 중첩 if문   | ㉥ 조건문   |
| ㉦ 반복문      | ㉧ 순차 구조    | ㉨ 선택 구조 |

### 정답

Click!

# [LAB] PC방 이용자의 나이 검사하기



밤 10시가 넘으면 청소년은 PC방을 이용할 수 없습니다.

나이를 입력하여 밤 10시에 PC방 출입이 가능한지 확인하는 프로그램을 만들어 봅시다.

## 실행 결과

나이 입력 ==> 22 ● — 사용자 입력  
즐거운 시간 보내세요.^^  
협조 감사합니다.



# [LAB] PC방 이용자의 나이 검사하기



1. Lab05\_01.java 파일을 만들고, 키보드로 값을 입력받기 위해 Scanner 클래스를 사용할 수 있도록 준비하기

```
import java.util.Scanner;           // 코드 첫 행에 입력
Scanner s = new Scanner(System.in); // main() 메서드 내 첫 행에 입력
s.close();                          // main() 메서드 내 마지막 행에 입력
```

2. 나이를 입력받을 변수를 선언하고 나이를 입력받기

```
int age;
System.out.print("나이 입력 ==> ");
age = s.nextInt();
```

3. 입력된 나이가 18세 미만이면 집으로 돌아가게 하는 조건문 작성하기. 그리고 청소년 이든 아니든 협조 감사문을 출력함

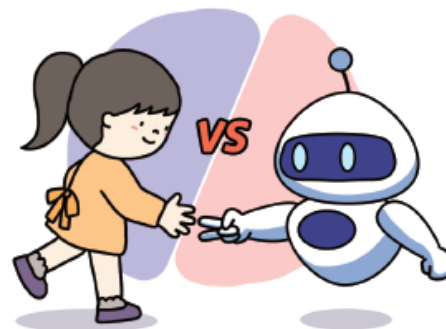
```
if (age >= 18) {
    System.out.println("즐거운 시간 보내세요^^");
} else {
    System.out.println("집에 갈 시간이네요!");
}
System.out.println("협조 감사합니다.");
```

# [LAB] 컴퓨터와 가위바위보하기



컴퓨터와 가위바위보를 하는 프로그램을 만들어봅시다.  
이미 잘 알고 있겠지만 가위바위보의 규칙은 다음과 같습니다.  
사용자를 기준으로 컴퓨터가 내는 것에 따라 승패를 결정하면 됩니다.

나	컴퓨터	결과(나 기준)
가위	가위 바위 보	비김 짐 이김
바위	가위 바위 보	이김 비김 짐
보	가위 바위 보	짐 이김 비김



## 실행 결과

나의 가위/바위/보 ==> 가위 ● — 사용자 입력  
컴퓨터의 가위/바위/보 ==> 가위  
비겼습니다. —.—

## 실행 결과

나의 가위/바위/보 ==> 바위 ● — 사용자 입력  
컴퓨터의 가위/바위/보 ==> 가위  
이겼습니다. ^^

# [LAB] 컴퓨터와 가위바위보하기



1. Lab05\_02.java 파일을 만들고, 키보드로 값을 입력받기 위해 Scanner 클래스를 사용할 수 있도록 준비하기

```
import java.util.Scanner;           // 코드 첫 행에 입력
Scanner s = new Scanner(System.in); // main() 메서드 내 첫 행에 입력
s.close();                          // main() 메서드 내 마지막 행에 입력
```

2. 사용자의 손에 해당하는 변수 myHand, 컴퓨터의 손에 해당하는 변수 comHand를 선언하고, 사용자로부터 가위, 바위, 보 중 하나를 입력받아 myHand에 저장하기

```
String myHand, comHand;
System.out.print("나의 가위/바위/보 ==> ");
myHand = s.next();
```

3. 컴퓨터의 가위, 바위, 보는 랜덤으로 추출하여 comHand에 저장하기

```
String[] hands = {"가위", "바위", "보"};
int rnd = (int) (Math.random() * hands.length);
comHand = hands[rnd];

System.out.println("컴퓨터의 가위/바위/보 ==> " + comHand);
```

# [LAB] 컴퓨터와 가위바위보하기



4. 사용자가 입력한 가위, 바위, 보에 따라 if문을 사용함. 우선 사용자가 가위를 냈을 때 컴퓨터가 가위를 내면 비기고, 컴퓨터가 바위를 내면 지고, 컴퓨터가 보를 내면 이기도록 작성하기

```
if (myHand.equals("가위")) {  
    if (comHand.equals("가위"))  
        System.out.println("비겼습니다. -.-");  
    else if (comHand.equals("바위"))  
        System.out.println("졌습니다. ππ");  
    else if (comHand.equals("보") )  
        System.out.println("이겼습니다. ^^");  
}
```

5. 사용자가 바위를 낸 경우와 보를 낸 경우도 위와 같이 작성하기.

- 가위가 아니라고 판단된 상태에서 바위와 보를 확인해야 하므로 else if문을 사용함

```
else if (myHand.equals("바위")) {  
    ~~~생략~~~  
}  
else if (myHand.equals("보")) {  
    ~~~생략~~~  
}
```

6. 사용자가 입력한 값이 가위, 바위, 보가 아닐 경우에 대비하여 else문을 작성하기

```
else {  
    System.out.println("가위/바위/보 중 하나를 내세요.");  
}
```





## [하나 더 알기] 랜덤 값 추출하기

알 수 없는 임의의 랜덤 값이 필요한 경우가 종종 있는데, 몇 가지 사례를 살펴봅시다.

### • 로또 추천하기

로또는 1~45 중에서 여섯 개의 숫자를 추천하는 복권입니다. 그런데 어떤 숫자가 추천되도록 프로그래밍을 해놓는다면 프로그램을 만든 사람이 항상 1등에 당첨될 것입니다. 이를 방지하려면 숫자를 추천할 때마다 임의의 숫자가 나와야 하는데, 자바에서는 랜덤 값을 추출하는 데 `Math.random()` 메서드를 사용합니다.

```
double rnd = Math.random();  
System.out.println(rnd);
```

0.7591030749604109 ● ——— 랜덤 값

`Math.random()` 메서드를 사용하면 0부터 1 미만의 값, 즉 0.00000000~0.999999999가 `double`형으로 추출되며, 이 값을 1~45로 변경하기 위해 45를 곱한 다음 1을 더하고 정수형인 `int`로 변환합니다.

```
int rnd = (int) (Math.random() * 45 + 1);  
System.out.println(rnd);
```

23 ● ——— 랜덤 값



## [하나 더 알기] 랜덤 값 추출하기

### • 점심 메뉴 고르기

임의로 추출하는 값이 숫자가 아닌 경우에는 어떻게 해야 할까요? 예를 들어 점심 메뉴를 김밥, 라볶이, 돈까스, 짜장면 중에서 임의로 고르는 경우에는 다음과 같이 작성하면 됩니다.

```
String[] foods = {"김밥", "라볶이", "돈까스", "짜장면"};  
int rnd = (int) (Math.random() * foods.length);  
System.out.println(foods[rnd]);
```

돈까스 ●————— 랜덤 값

String[]은 여러 개의 값을 묶어놓은 배열을 의미합니다. 배열의 순서는 0번부터 시작되므로 0번이 김밥, 1번이 라볶이, 2번이 돈까스, 3번이 짜장면입니다. 그리고 foods.length는 foods 배열의 개수를 나타내므로 4이며, 따라서 rnd 값에 0, 1, 2, 3 중 하나가 들어갑니다. 또한 foods[숫자]는 배열에서 해당 숫자 위치의 값을 가리키며, 만약 foods[2]라면 "돈까스"가 됩니다. 배열은 9장에서 자세히 살펴보겠습니다.

# Section 03

## switch~case문

# 1. switch~case문의 개요

## ■ 이중 분기와 다중 분기

- 이중 분기: 참 또는 거짓 중에서 하나를 선택하는 조건문(if문)
- 다중 분기: 참 또는 거짓만으로 해결할 수 없고 여러 개 중에서 하나를 선택해야 하는 조건문(switch~case문)

## ■ switch~case문의 형식

```
switch(정수값) {  
  case 정수값 1:  
    실행할 문장 1;  
    break;  
  case 정수값 2:  
    실행할 문장 2;  
    break;  
  default:  
    실행할 문장 3;  
    break;  
}
```

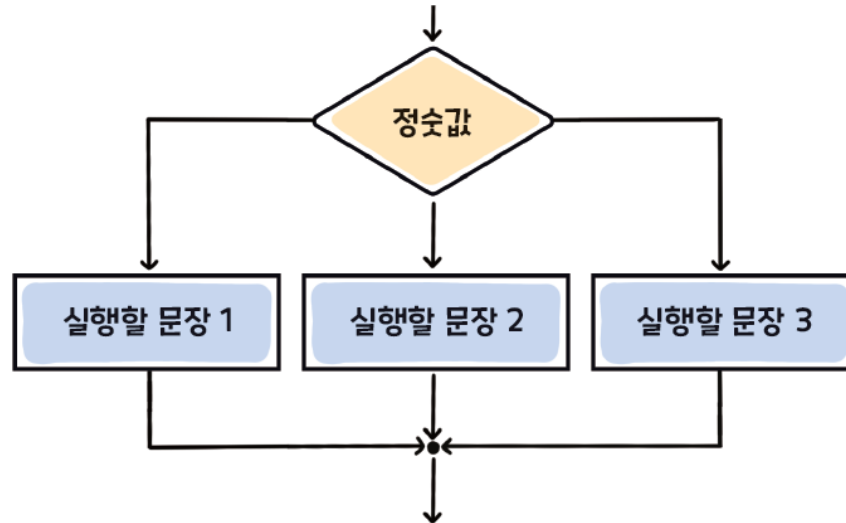


그림 5-15 switch~case문의 형식

# 1. switch~case문의 개요

## ■ switch~case문 예제

코드 5-9

Code05\_09.java

```
01  import java.util.Scanner;
02  public class Code05_09 {
03      public static void main(String[] args) {
04          Scanner s = new Scanner(System.in);
05          int select;
06
07          System.out.print("1~3 중에 선택하세요 : ");
08          select = s.nextInt();
09
10          switch(select) {      // 입력된 select 값에 따라 분기
11              case 1:            // select가 1이면 12행을 실행하고 13행에서 빠져나감
12                  System.out.println("1을 선택했습니다.");
13                  break;
14              case 2:
15                  System.out.println("2를 선택했습니다.");
16                  break;
17              case 3:
18                  System.out.println("3을 선택했습니다.");
19                  break;
20              default:           // select가 1, 2, 3이 아닌 경우 실행
21                  System.out.println("이상한 것을 선택했습니다.");
22          }
23
24          s.close();
25      }
26  }
```

1~3 중에 선택하세요 : 2 ● 사용자 입력  
2를 선택했습니다.

# 1. switch~case문의 개요

## ■ switch~case문 예제

- [코드 5-9]의 흐름

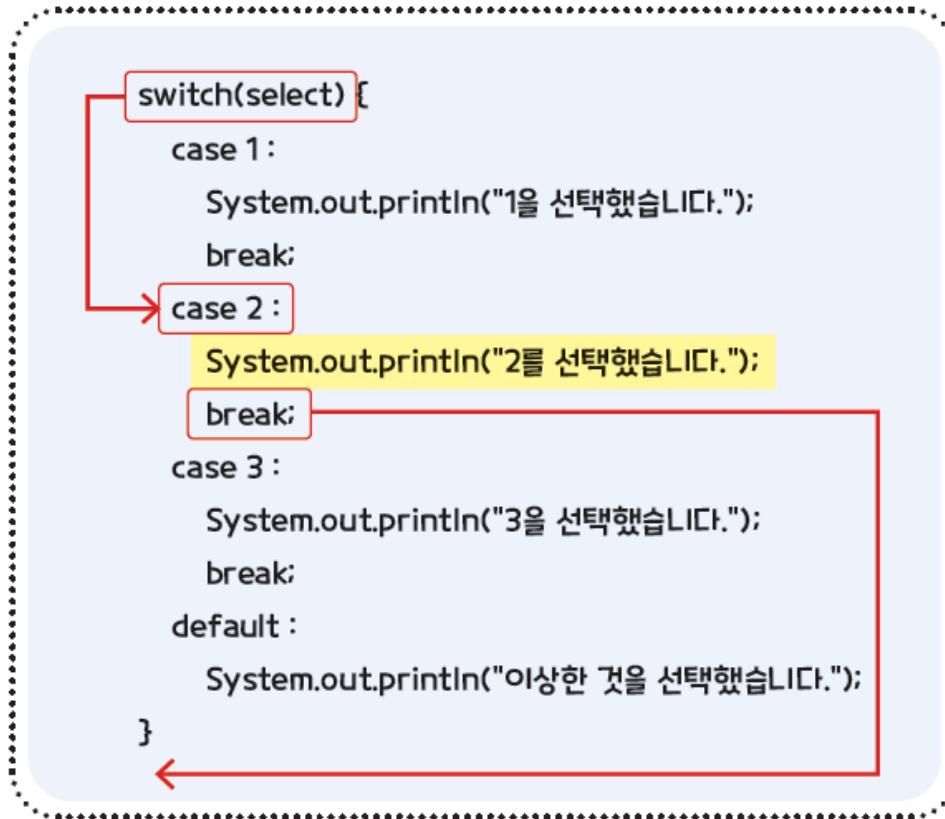


그림 5-16 [코드 5-9]에서 select가 2일 때 switch~case문의 흐름

## 2. switch~case문 사용 시 주의점

### ■ switch~case문을 사용할 때 주의할 점

- 각각의 실행문 끝부분에 반드시 break문을 써야 함
- 그렇지 않으면 switch~case문을 빠져나가지 않고 다음 코드를 계속 수행하기 때문
- [코드 5-9]에서 break문을 모두 뺀 경우

```
switch(select) {  
    case 1:  
        System.out.println("1을 선택했습니다.");  
    case 2:  
        System.out.println("2를 선택했습니다.");  
    case 3:  
        System.out.println("3을 선택했습니다.");  
    default:  
        System.out.println("이상한 것을 선택했습니다.");  
}
```

1~3 중에 선택하세요 : 2 ● ————— 사용자가 입력

2를 선택했습니다.

3을 선택했습니다.

이상한 것을 선택했습니다.

## 2. switch~case문 사용 시 주의점

### 확인문제

다음 빈칸에 들어갈 알맞은 말을 고르시오.

switch~case 블록을 빠져나가기 위해 사용하는 구문은  이다.

① if

② for

③ else if

④ break

정답

Click!



# [LAB] 출생 연도에 따른 띠 확인하기



우리나라와 중국, 일본 등에서는 출생 연도에 따라 열두 지지를 상징하는 띠가 부여됩니다.

출생연도를 입력하면 쥐띠, 소띠, 호랑이띠, 토끼띠, 용띠, 뱀띠, 말띠, 양띠, 원숭이띠, 닭띠, 개띠, 돼지띠 중에서 해당하는 띠를 알려주는 프로그램을 만들어봅시다.



## 실행 결과

출생 연도를 입력하세요 ==> 2003 ● ————— 사용자가 입력

양띠

# [LAB] 출생 연도에 따른 띠 확인하기



1. Lab05\_03.java 파일을 만들고, 키보드로 값을 입력받기 위해 Scanner 클래스를 사용할 수 있도록 준비하기

```
import java.util.Scanner;           // 코드 첫 행에 입력
Scanner s = new Scanner(System.in); // main() 메서드 내 첫 행에 입력
s.close();                         // main() 메서드 내 마지막 행에 입력
```

2. 사용자의 손에 해당하는 변수 myHand, 컴퓨터의 손에 해당하는 변수 comHand를 선언하고, 사용자로부터 가위, 바위, 보 중 하나를 입력받아 myHand에 저장하기

```
int year;

System.out.printf("출생 연도를 입력하세요 ==> ");
year = s.nextInt();
```



3. 출생 연도를 12로 나눈 나머지로 띠를 결정하기 위해 switch~case문을 이용함
- 나머지가 0: 원숭이띠, 1: 닭띠, 2: 개띠, 3: 돼지띠, 4: 쥐띠, 5: 소띠, 6: 호랑이띠, 7: 토끼띠, 8: 용띠, 9: 뱀띠, 10: 말띠, 11: 양띠

```
switch(year%12) {  
    case 0: System.out.println("원숭이띠"); break;  
    case 1: System.out.println("닭띠"); break;  
    case 2: System.out.println("개띠"); break;  
    ~~~생략~~~  
    case 11: System.out.println("양띠"); break;  
}
```

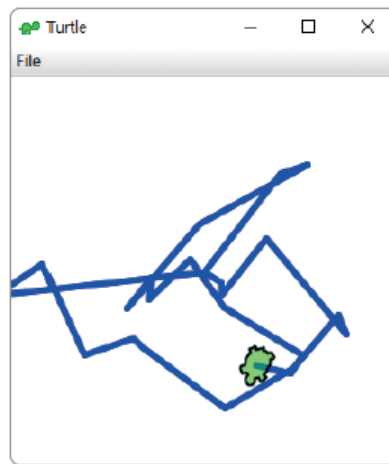
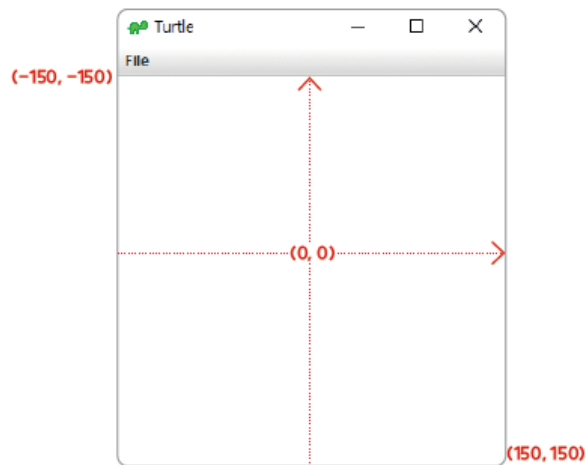
**[실전 예제]**  
**사춘기 거북이**  
**지도하기**

## [실전 예제] 사춘기 거북이 지도하기

**[문제]**

사춘기인 거북이는 언제 어디로 튈지 몰라 그대로 두면 화면 밖으로 나가버릴 수도 있습니다. 그렇다고 해서 너무 통제하는 것도 좋지 않으니 거북이가 화면 안에서 움직이면 그냥 놔두고, 화면 밖으로 나가려고 하면 화면 한가운데에 데려다 놓읍시다.

화면의 크기를 (300, 300)으로 지정하면 화면 중앙이 (0, 0)이므로 거북이의 활동 공간은 가로가 -150~150, 세로가 -150~150입니다. 거북이의 이동 위치가 이 범위 내라면 그대로 두고, 거북이가 범위 밖으로 벗어나면 화면 중앙인 (0, 0)으로 옮깁니다.



# [실전 예제] 사춘기 거북이 지도하기

## [해결]

Ex05\_01.java

```
01  public class Ex05_01 {
02      public static void main(String[] args) {
03          Turtle turtle = new Turtle();
04          int angle, distance, curX, curY;
05
06          turtle.speed(500);
07          turtle.setCanvasSize(300, 300);
08          turtle.width(5);
09          turtle.penColor("blue");
10
11          while(true) {
12              angle = (int)(Math.random()*360);
13              distance = (int)(Math.random()*90+10);
14
15              turtle.right(angle);
16              turtle.forward(distance);
17
18              curX = (int)turtle.getX();
19              curY = (int)turtle.getY();
20
21              if((curX >= -150 && curX <= 150) && (curY >= -150&& curY <= 150))
22                  System.out.println("Good Boy ~");
23              else
24                  turtle.setPosition(0, 0);
25          }
26
27      }
28  }
```

**Thank you!**