

# 初級プログラミング I

## 第3回 条件分岐

中谷 祐介

# 第3回の構成

## ◆ 第1講 if 文による分岐 (1)

- Java における条件分岐について, 基本事項を理解する.

## ◆ 第2講 条件式

- 条件分岐を行うときの条件式の記述について, 基本事項を理解する.

## ◆ 第3講 if 文による分岐 (2)

- 条件分岐を用いた簡単なプログラムが作成できるようになる.

## ◆ 第4講 switch 文による分岐

- 条件分岐のうち, switch 文による分岐の方法を理解する.

## 第3回 条件分岐

# 第1講 `if` 文による分岐(1)

# 第1講の学習目標

- ◆ 第2回の内容について復習する:
  - プログラムの流れ
  - データの入出力
  - 演算
- ◆ Java における条件分岐について, 基本事項を理解する:
  - 条件分岐
  - if 文

## 練習問題2-3（第2回の復習）

### 練習問題2-3

標準入力に三角形の底辺と高さをこの順に整数値で入力すると、三角形の面積を計算し、その結果を標準出力に出力するプログラムを作成してください。

三角形の面積：底辺 × 高さ ÷ 2

```
1 import java.util.Scanner;
2
3 public class Practice_02_03 {
4     public static void main(String[] args) {
5         Scanner scan = new Scanner(System.in);
6
7         int b = scan.nextInt(); // 底辺の入力
8         int h = scan.nextInt(); // 高さの入力
9
10        double area = b * h / 2.0; // 面積の計算
11
12        System.out.println(area); // 結果の出力
13    }
14 }
```

Practice\_02\_03.java

# プログラムの流れ

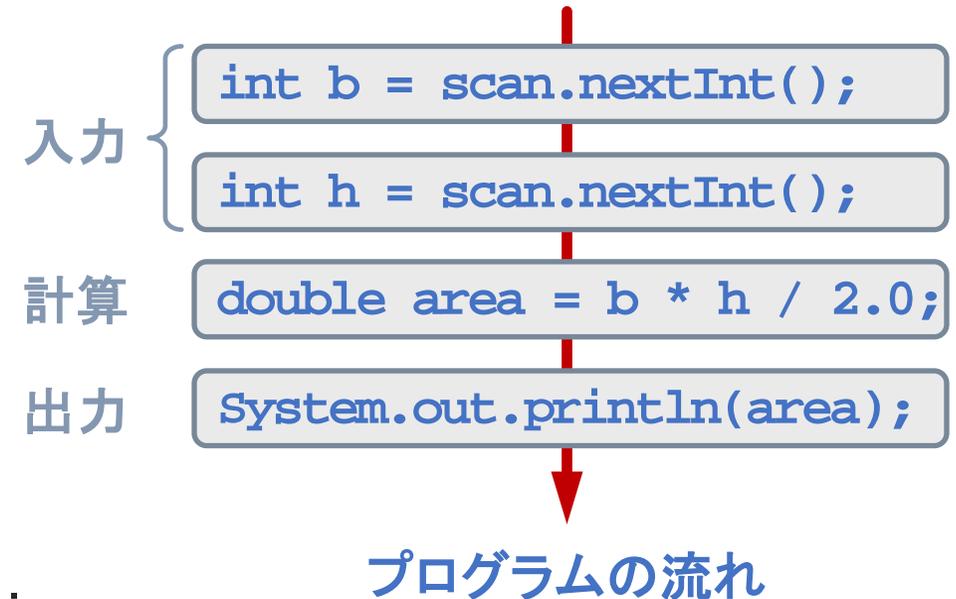
- ◆ プログラムは, 原則として上から下に向かって, 記述した順に処理を行う.

```
int b = scan.nextInt();  
int h = scan.nextInt();  
  
double area = b * h / 2.0;  
  
System.out.println(area);
```

- プログラムとして実現できることが限定される.

- ◆ プログラムの流れを制御する方法を学ぶことで, 実現できる範囲が広がる.

- 条件分岐 — ある条件に従って, その後の処理を変える.
- 繰り返し — 同様のことを繰り返し処理する.



## 例題3-1

### 例題3-1

標準入力に正の整数値  $n$  を入力すると, その整数値が3桁以上であるかどうかを判定し, 3桁以上であれば「yes」を, 2桁以下であれば「no」を標準出力に出力するプログラムを作成してください.

- ◆ 入力した値により, 処理が異なる:
    - 3桁以上の場合 : 「yes」と出力.
    - そうでない場合 : 「no」と出力.
- 
- ◆ 単純に, 上から下に順に処理を行うだけでは実現できない.
  - ◆ ある条件に基づいて, その後の処理を変える:
    - 「例題3-1」の場合の条件は, 「整数値  $n$  が3桁以上」.
- 
- ◆ **条件分岐**により実現する.

# 条件分岐

- ◆ **条件分岐**とは、ある条件に従って、その後の処理を変える制御構造のことである。
- ◆ 条件分岐を実現する方法として、以下を習得する：
  - **if 文** — ある条件に従って、二方向に分岐する。
  - **switch 文** — 整数値に基づいて、多方向に分岐する。

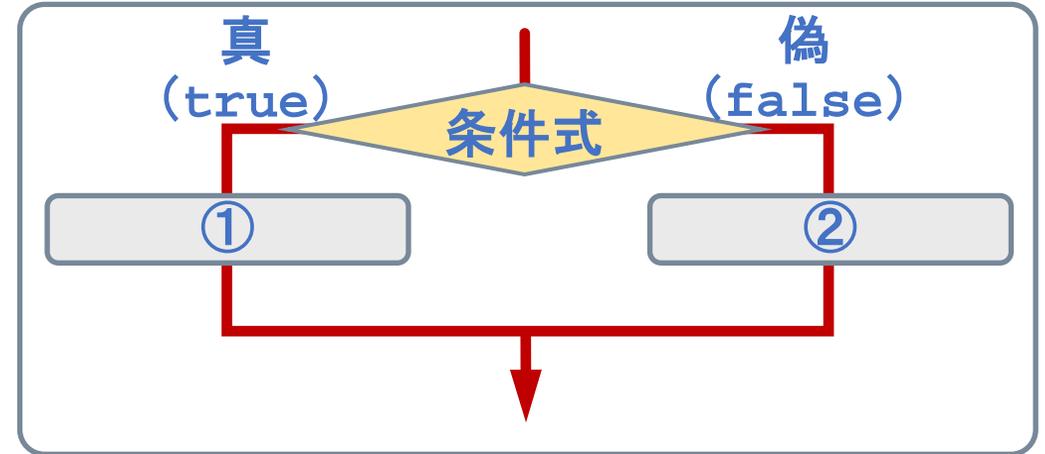
```
if (条件式) {  
    //実行したいこと1  
} else {  
    //実行したいこと2  
}
```

```
switch (整数式) {  
    case 値1:  
        //実行したいこと1  
        break;  
    case 値2:  
        //実行したいこと2  
        break;  
        ⋮  
    default:  
        //実行したいことn  
}
```

# if文 (1)

◆ if文は、ある条件により、その後の処理を分岐する文。

```
if (条件式) {  
    条件式が「真」のとき実行する文ー①  
} else {  
    条件式が「偽」のとき実行する文ー②  
}
```



if文の処理の流れ

◆ if文は、( ) の中に条件式を記述する。

- 条件式が「真」(**true**)のときに実行する文を条件式のアとの **{ }** 内に記述する。
- 条件式が「偽」(**false**)のときに実行する文を **else** のアとの **{ }** 内に記述する。

◆ if文は、条件式が成立するかどうかで、その後の処理を分岐することができる。

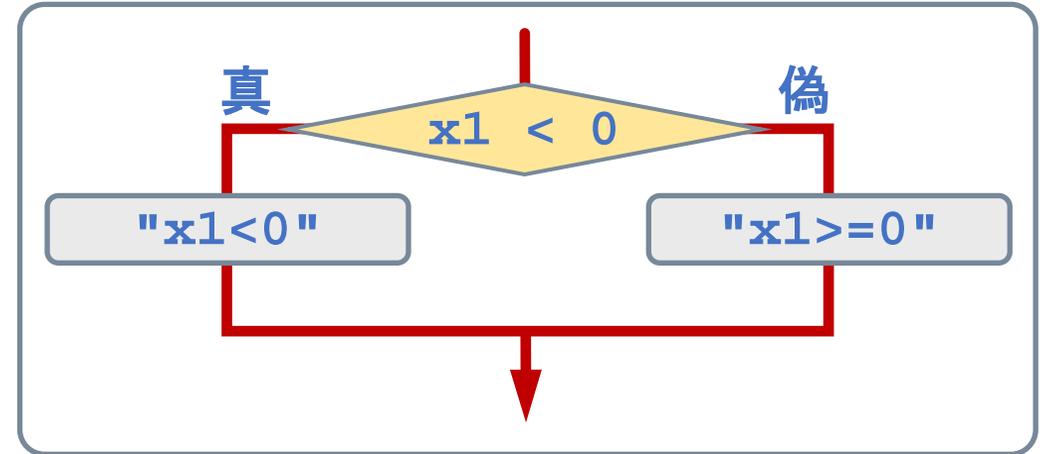
## ◆ if文の使用例1

```
int x1 = -5;

if (x1 < 0) {
    System.out.println("x1<0");
} else {
    System.out.println("x1>=0");
}
```

x1<0 ←

- 上記のプログラムの流れ
  1. 変数 x には「-5」が格納されている.
  2. if文の条件式「x1 < 0」が成立する.
  3. したがって、標準出力には「x1<0」と表示される.



「if文の使用例1」の処理の流れ

## ◆ if文の使用例2

```
int x2 = *;  
  
if (x2 < 0)  
    System.out.println("x2<0");  
else  
    System.out.println("x2>=0");
```

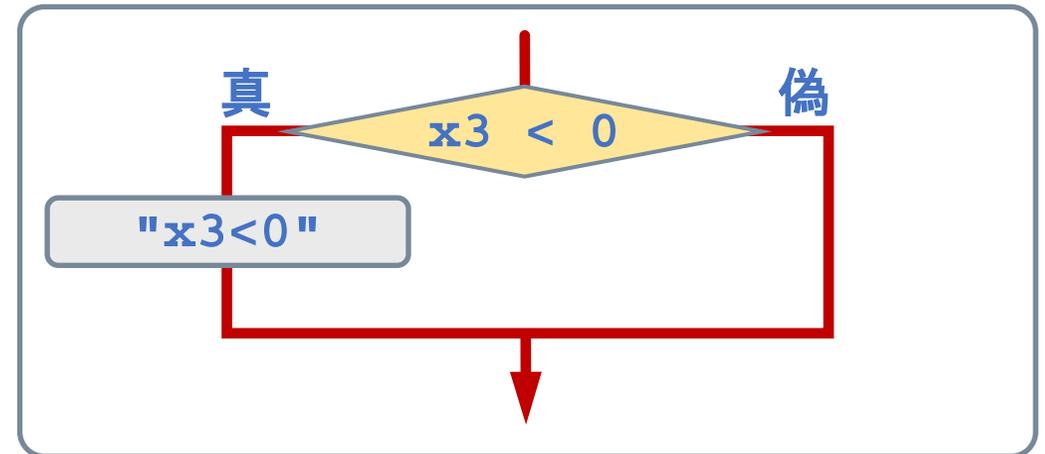
(注) 「\*」の部分には, 具体的な整数値を記述する.

- {} の中で実行する文が1つだけのときは, {} は省略できる.

## ◆ if文の使用例3

```
int x3 = *;  
  
if (x3 < 0) {  
    System.out.println("x3<0");  
}
```

- else 以下は省略できる.



「if文の使用例3」の処理の流れ

## 例題3-1（再掲）

### 例題3-1

標準入力に正の整数値  $n$  を入力すると、その整数値が3桁以上であるかどうかを判定し、3桁以上であれば「yes」を、2桁以下であれば「no」を標準出力に出力するプログラムを作成してください。

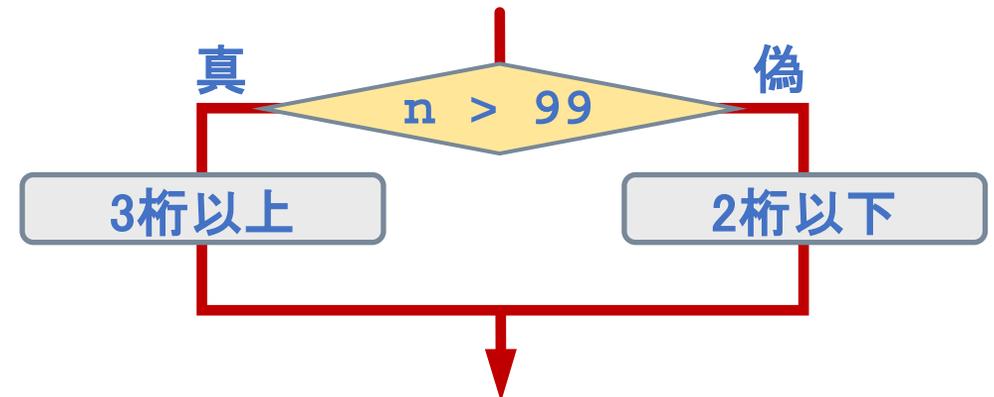
#### ◆ 「整数値 $n$ が3桁以上」という条件は？

- int 型の変数  $n$  が「99」より大きい。  $\Rightarrow n > 99$
- int 型の変数  $n$  が「100」以上。  $\Rightarrow n \geq 100$

#### ◆ 「例題3-1」の条件分岐は、次のとおり:

```
if (n > 99) {  
    // 3桁以上の場合の処理  
} else {  
    // 2桁以下の場合の処理  
}
```

- 最初の行は、「 $n \geq 100$ 」でもよい。



「例題3-1」の条件分岐

# 「例題3-1」のプログラム例

Example\_03\_01.c

```
1 import java.util.Scanner;
2
3 public class Example_03_01 {
4     public static void main(String[] args) {
5         Scanner scan = new Scanner(System.in);
6
7         int n = scan.nextInt();
8
9         if (n > 99) {
10            System.out.println("yes");
11        } else {
12            System.out.println("no");
13        }
14    }
15 }
```

## ◆ 実行例

標準入力

1234

標準出力

yes↵

標準入力

12

標準出力

no↵

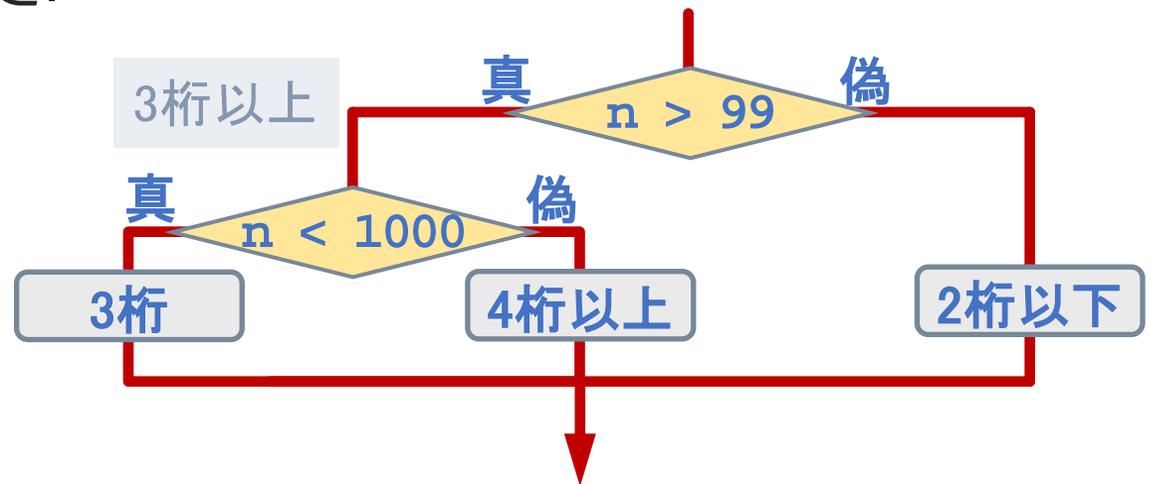
## 例題3-2

### 例題3-2

標準入力に正の整数値  $n$  を入力すると, その整数値が3桁であるかどうかを判定し, 3桁であれば「yes」を, そうでなければ「no」を標準出力に出力するプログラムを作成してください.

◆ 「例題3-1」で「整数値  $n$  が3桁以上」を判定.

```
if (n > 99) {  
    // 3桁以上の場合の処理  
    if (n < 1000) {  
        // 3桁の場合の処理  
    } else {  
        // 4桁以上の場合の処理  
    }  
} else {  
    // 2桁以下の場合の処理  
}
```



「例題3-2」の条件分岐

- 変数  $n$  が3桁以上 ( $n > 99$ ) の状態で,  $n$  が「1000」より小さければ,  $n$  は3桁.

## 「例題3-2」のプログラム例

Example\_03\_02.c

```
1 import java.util.Scanner;
2
3 public class Example_03_02 {
4     public static void main(String[] args) {
5         Scanner scan = new Scanner(System.in);
6
7         int n = scan.nextInt();
8
9         if (n > 99) {
10             if (n < 1000) {
11                 System.out.println("yes");
12             } else {
13                 System.out.println("no");
14             }
15         } else {
16             System.out.println("no");
17         }
18     }
19 }
```

### ◆ 実行例

標準入力

123

標準出力

yes←

標準入力

1234

標準出力

no←

# 第1講のまとめ

- ◆ 第2回の内容について復習した:
  - プログラムの流れ
  - データの入出力
  - 演算
- ◆ Java における条件分岐について, 基本事項を理解した:
  - 条件分岐
  - if 文

## 第3回 条件分岐

# 第1講 `if` 文による分岐(1)

終わり

## 第3回 条件分岐

# 第2講 条件式

## 第2講の学習目標

- ◆ 条件分岐を行うときの条件式の記述について、基本事項を理解する：
  - 等価演算子と関係演算子
  - 論理演算子
    - 論理積
    - 論理和
    - 否定

## 例題3-2（再掲）

### 例題3-2

標準入力に正の整数値  $n$  を入力すると、その整数値が3桁であるかどうかを判定し、3桁であれば「yes」を、そうでなければ「no」を標準出力に出力するプログラムを作成してください。

- ◆ 不等号「>」、「<」を使用して、条件式を記述した:

```
if (n > 99) {  
    if (n < 1000) {  
        System.out.println("yes");  
    } else {  
        System.out.println("no");  
    }  
} else {  
    System.out.println("no");  
}
```

- ◆ 条件式を記述するために必要な、**等価演算子**、**関係演算子**、**論理演算子**を理解する。

# 等価演算子と関係演算子 (1)

◆ 条件式を記述するための演算子として、以下がある:

- 等価演算子 : `==`, `!=`
- 関係演算子 : `<`, `>`, `<=`, `>=`

◆ 2つの値の関係式が成立するか否かで演算結果を導く:

- 関係式が成立するとき ⇒ 演算結果は「`true`」(真)
- 関係式が成立しないとき ⇒ 演算結果は「`false`」(偽)

◆ 演算の例

- `3 < 5` ⇒ 「`true`」(真)
- `4 == 8` ⇒ 「`false`」(偽)
- `i <= 5` ⇒ `i` の値により「`true`」(真)または「`false`」(偽)

◆ 「`true`」, 「`false`」は真偽値で、変数の型としては `boolean` 型.

等価演算子と関係演算子

演算子	例	意味
<code>==</code>	<code>x == y</code>	等しい
<code>!=</code>	<code>x != y</code>	等しくない
<code>&lt;</code>	<code>x &lt; y</code>	より小さい
<code>&gt;</code>	<code>x &gt; y</code>	より大きい
<code>&lt;=</code>	<code>x &lt;= y</code>	以下
<code>&gt;=</code>	<code>x &gt;= y</code>	以上

## ◆ 等価演算子, 関係演算子の使用例

```
int a = 10;
System.out.println(a <= 20); // 「a <= 20」は成立する.
System.out.println(a > 12); // 「a > 12」は成立しない.
System.out.println(a != 15); // 「a != 15」は成立する.

true
false
true
```

- 等価演算子, 関係演算子を用いた演算の結果は「true」(真)または「false」(偽)となる.

## ◆ 「a と b の値が等しい」という関係式を, 「a = b」と記述しない!

- 「a = b」は, b の値を a に代入するという意味.
- 「a == b」とする.

## 等価演算子と関係演算子 (3)

### ◆ 「例題3-2」の場合 (整数値が3桁であるかどうかの判定)

```
if (n >= 100) {  
    if (n <= 999) {  
        System.out.println("yes");  
    }  
}
```

- 「>=」(以上), 「<=」(以下)を使用して実現した例.
- 「n が 100 以上」の状態, 「n が 999 以下」であることを判定.

### ◆ 「n が 100 以上 999 以下」という条件式を別の表現で実現したい:

```
if (100 <= n <= 999) {  
    System.out.println("yes");  
}
```

← 間違い!

- 論理演算子の利用により実現できる.

# 論理演算子

- ◆ 論理演算子は、等価・関係演算子を用いて記述した条件式を組み合わせたたり、否定したりするために使用する演算子である。
- ◆ 論理演算子を使用することで、より複雑な条件式を記述することができる。
- ◆ 論理演算には、論理積(かつ)、論理和(または)、否定がある。

論理演算子

演算子	例	意味
&&	A && B	論理積(かつ)
	A    B	論理和(または)
!	!A	否定

## ◆ 論理演算の例:

- 論理積 — `a >= 10 && b <= 20`
- 論理和 — `a < 10 || b <= 20`
- 否定 — `!(a != 10)`

## ◆ 論理積 (A && B)

- A かつ B
- 条件式 A, B の結果がともに「true」(真) のとき ⇒ 「A && B」は「true」(真)  
それ以外 のとき ⇒ 「A && B」は「false」(偽)
- 論理積演算の例

```
int a = 10, b = 15;  
System.out.println(a >= 10 && b <= 20); // 「a >= 10」, 「b <= 20」ともに「true」  
System.out.println(a == 5 && b >= 10); // 「b >= 10」のみ「true」
```

```
true  
false
```

### 論理積

A	B	A && B
true	true	true
true	false	false
false	true	false
false	false	false

## ◆ 論理和 (A || B)

- A または B
- 条件式 A, B のどちらか一方でも「true」(真) のとき ⇒ 「A || B」は「true」(真)  
それ以外(ともに「false」(偽)) のとき ⇒ 「A || B」は「false」(偽)
- 論理和演算の例

```
int a = 10, b = 15;  
System.out.println(a < 10 || b <= 20); // 「b <= 20」のみ「true」  
System.out.println(a != 10 || b > 20); // 「a != 10」, 「b > 20」ともに「false」
```

```
true  
false
```

### 論理和

A	B	A && B
true	true	true
true	false	true
false	true	true
false	false	false

## ◆ 否定 (!A)

- A ではない
- 条件式 A が「true」(真)のとき ⇒ 「!A」は「false」(偽),  
条件式 A が「false」(偽)のとき ⇒ 「!A」は「true」(真)
- 否定演算の例

```
int a = 10, b = 15;  
System.out.println(!(a!=10));           // 「a!=10」は「false」  
System.out.println(!(a<10 || b<=20));  // 「a<10 || b<=20」は「true」  
true  
false
```

### 否定

A	!A
true	false
false	true

## 「例題3-2」のプログラム例（論理演算子使用版）

### 例題3-2

標準入力に正の整数値  $n$  を入力すると、その整数値が3桁であるかどうかを判定し、3桁であれば「yes」を、そうでなければ「no」を標準出力に出力するプログラムを作成してください。

```
1 import java.util.Scanner;
2
3 public class Example_03_02_2 {
4     public static void main(String[] args) {
5         Scanner scan = new Scanner(System.in);
6
7         int n = scan.nextInt();
8
9         if ((100 <= n) && (n <= 999)) { // n が 100 以上 999 以下
10            System.out.println("yes");
11        } else {
12            System.out.println("no");
13        }
14    }
15 }
```

Example\_03\_02\_2.java

## 第2講のまとめ

- ◆ 条件分岐を行うときの条件式の記述について, 基本事項を理解した:
  - 等価演算子と関係演算子
  - 論理演算子
    - 論理積
    - 論理和
    - 否定

## 第3回 条件分岐

# 第2講 条件式

終わり

## 第3回 条件分岐

### 第3講 `if` 文による分岐(2)

## 第3講の学習目標

- ◆ 条件分岐を用いた簡単なプログラムが作成できるようになる:
  - 例題を用いた条件分岐のプログラム作成
- ◆ Java における条件分岐について, 基本事項を理解する:
  - `else if` 文

## 例題3-3 (1)

### 例題3-3

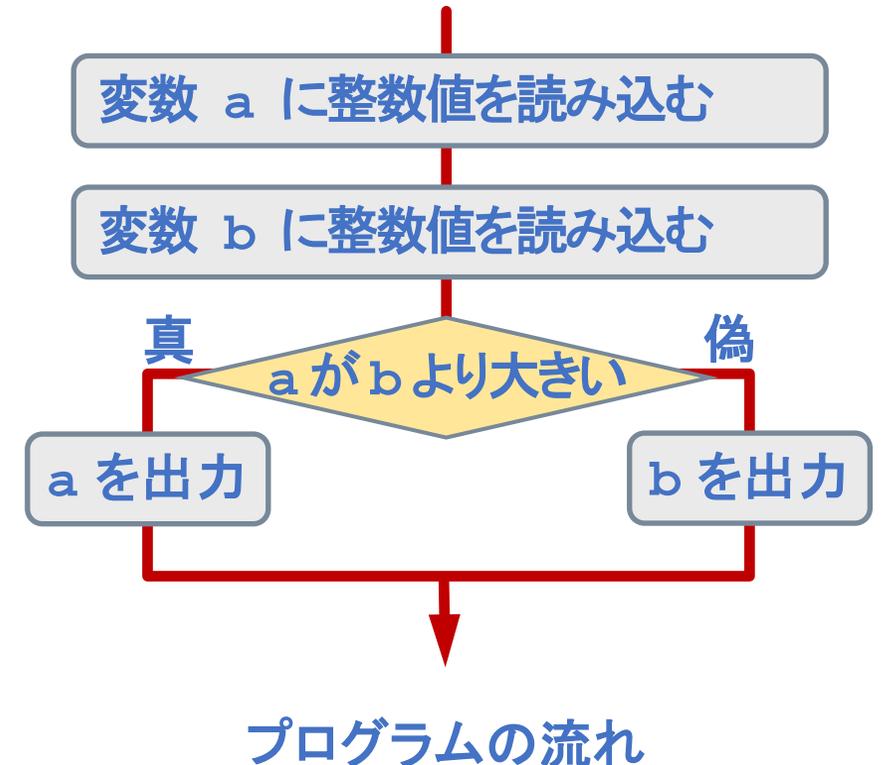
標準入力に2つの整数値を入力すると、大きい方の整数値を標準出力に出力するプログラムを作成してください。

- ◆ 2つの整数値を変数  $a$ ,  $b$  に読み込むことにする。
- ◆ 「プログラムの流れ」に沿ってプログラムを作成：

```
Scanner scan = new Scanner(System.in);

int a = scan.nextInt();
int b = scan.nextInt();

if (a > b) {
    System.out.println(a);
} else {
    System.out.println(b);
}
```



## 例題3-3 (2)

### 例題3-3

標準入力に2つの整数値を入力すると、大きい方の整数値を標準出力に出力するプログラムを作成してください。

```
1 import java.util.Scanner;
2
3 public class Example_03_03 {
4     public static void main(String[] args) {
5         Scanner scan = new Scanner(System.in);
6
7         int a = scan.nextInt();
8         int b = scan.nextInt();
9
10        if (a > b) {
11            System.out.println(a);
12        } else {
13            System.out.println(b);
14        }
15    }
16 }
```

Example\_03\_03.java

#### ◆ 実行例

標準入力

12  
34

標準出力

34↵

標準入力

22  
22

標準出力

22↵

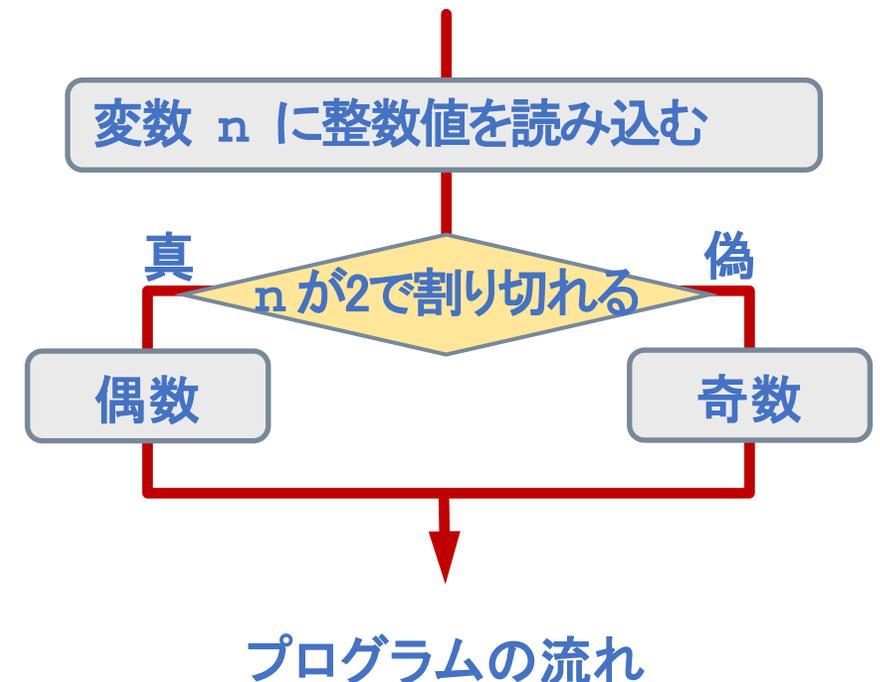
## 例題3-4 (1)

### 例題3-4

標準入力に整数値を入力すると、その整数値が「奇数」か「偶数」かを判定し、その結果を標準出力に出力するプログラムを作成してください。

- ◆ 整数値を変数  $n$  に読み込むことにする。
- ◆ 「奇数」か「偶数」かの判定
  - $n$  が 2 で割り切れる ⇒ 偶数
  - $n$  が 2 で割り切れない ⇒ 奇数
- ◆ 「プログラムの流れ」に沿ってプログラムを作成：

```
int a = scan.nextInt();  
  
if (n%2 == 0) {  
    System.out.println("偶数");  
} else {  
    System.out.println("奇数");  
}
```



## 例題3-4 (2)

### 例題3-4

標準入力に整数値を入力すると、その整数値が「奇数」か「偶数」かを判定し、その結果を標準出力に出力するプログラムを作成してください。

```
1 import java.util.Scanner;
2
3 public class Example_03_04 {
4     public static void main(String[] args) {
5         Scanner scan = new Scanner(System.in);
6
7         int n = scan.nextInt();
8
9         if (n%2 == 0) {
10            System.out.println("偶数");
11        } else {
12            System.out.println("奇数");
13        }
14    }
15 }
```

Example\_03\_04.java

#### ◆ 実行例

標準入力

123

標準出力

奇数←

標準入力

1234

標準出力

偶数←

## 例題3-5

### 例題3-5

標準入力に正の整数値  $n$  (9桁以下とする)を入力すると, その整数値が「1桁」, 「2桁」, 「3桁」, 「4桁以上」のいずれであるかを判定し, その結果を標準出力に出力するプログラムを作成してください.

◆ 右の表に従って条件式を記述すると…

- 1以上10未満 :  $n < 10$
- 10以上100未満 :  $10 \leq n \ \&\& \ n < 100$
- 100以上1,000未満 :  $100 \leq n \ \&\& \ n < 1000$
- 1,000以上 :  $1000 \leq n$

◆ 上記の条件式を用いて `if` 文で実現できそう.

「例題3-5」の出力

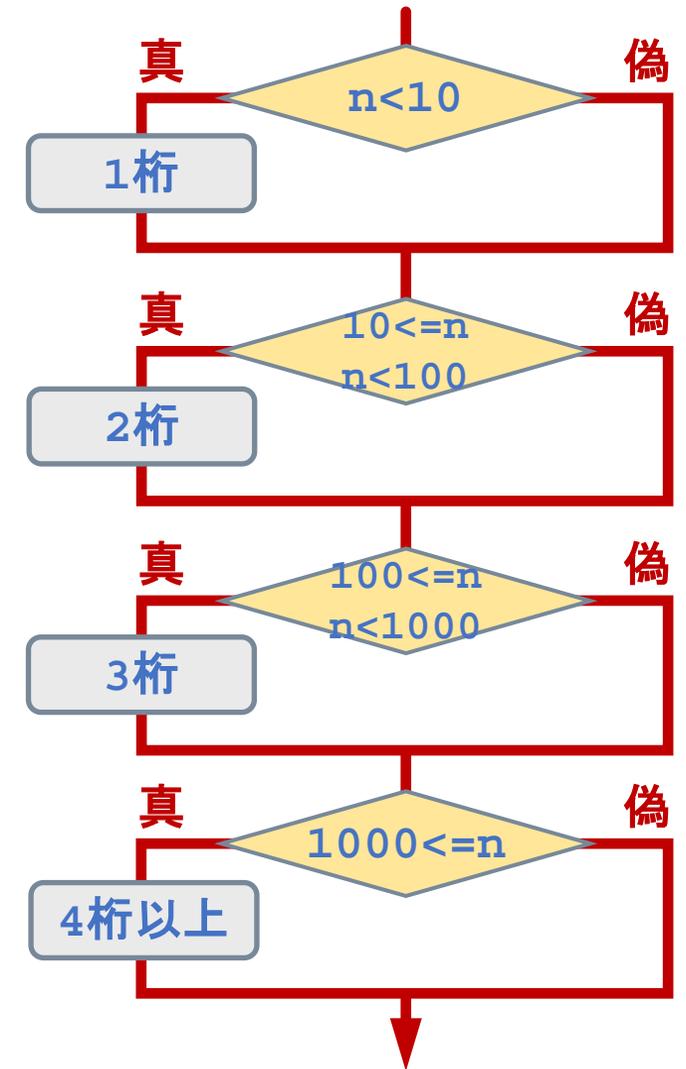
整数値 $n$	出力
1以上10未満	1桁
10以上100未満	2桁
100以上1,000未満	3桁
1,000以上	4桁以上

# 「例題3-5」の条件分岐

## ◆ 前頁で示した条件式を使用すると…

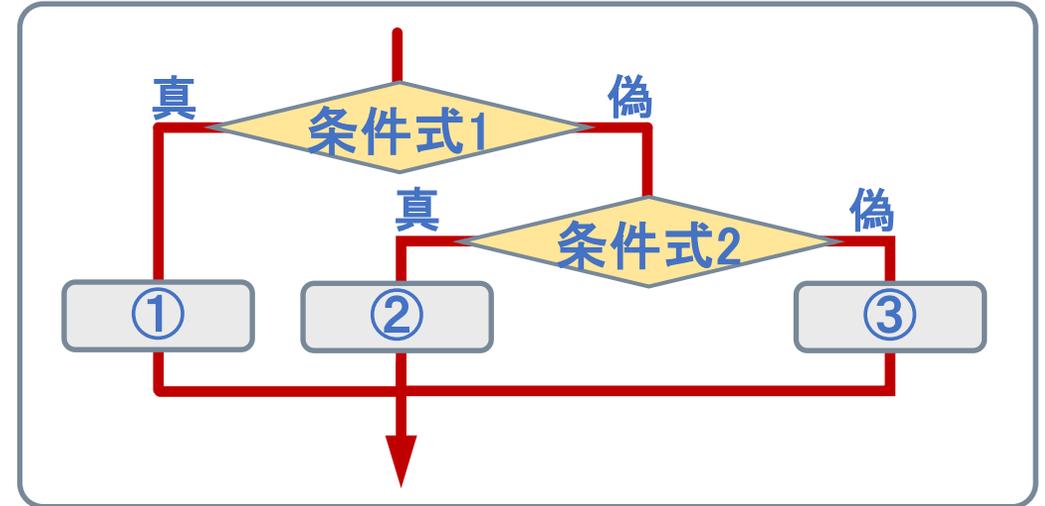
```
if (n<10) {  
    System.out.println("1桁");  
}  
if (10<=n && n<100) {  
    System.out.println("2桁");  
}  
if (100<=n && n<1000) {  
    System.out.println("3桁");  
}  
if (1000<=n) {  
    System.out.println("4桁以上");  
}
```

- このプログラムは、シンプルで間違いではないが無駄がある。
- 常に、4つのif文が実行される。
- 例 変数 n の値が「9」のとき、最初のif文以外は実行不要。



# else if文 (1)

```
if (条件式1) {  
    ①  
}  
else if (条件式2) {  
    ②  
}  
else {  
    ③  
}
```



else if文の処理の流れ

- ◆ 3方向に分岐させて処理を行う.
- ◆ 「else if (条件式) {…}」の部分が必要な回数だけ記述することで, 4方向, 5方向, …,  $n$  方向へと処理を分岐させることができる.

[参考] else if 文は以下と同じ

```
if (条件式1) {  
    ①  
}  
else {  
    if (条件式2) {  
        ②  
    } else {  
        ③  
    }  
}
```

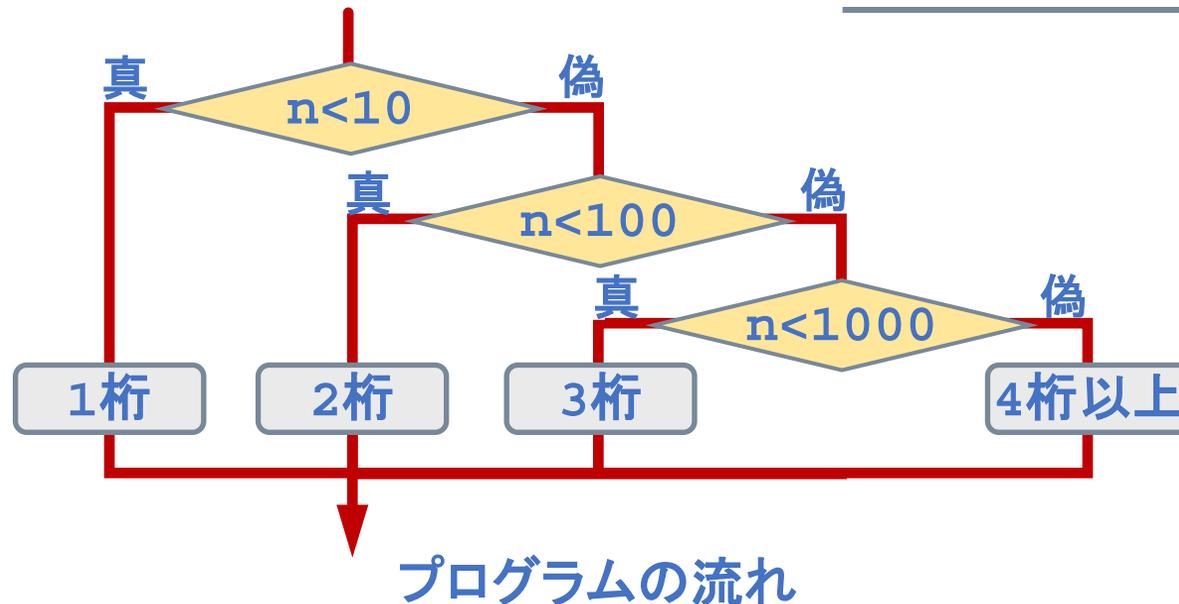
# else if文 (2)

## ◆ 「例題3-5」の場合

```
if (n < 10) {  
    System.out.println("1桁");  
} else if (n < 100) {  
    System.out.println("2桁");  
} else if (n < 1000) {  
    System.out.println("3桁");  
} else {  
    System.out.println("4桁以上");  
}
```

## 「例題3-5」の出力

整数値 $n$	出力
1以上10未満	1桁
10以上100未満	2桁
100以上1,000未満	3桁
1,000以上	4桁以上



# 「例題3-5」のプログラム例 (1)

Example\_03\_05.java

```
1 import java.util.Scanner;
2
3 public class Example_03_05 {
4     public static void main(String[] args) {
5         Scanner scan = new Scanner(System.in);
6
7         int n = scan.nextInt();
8
9         if (n < 10) {
10            System.out.println("1桁");
11        } else if (n < 100) {
12            System.out.println("2桁");
13        } else if (n < 1000) {
14            System.out.println("3桁");
15        } else {
16            System.out.println("4桁以上");
17        }
18    }
19 }
```

## ◆ 実行例

標準入力

12

標準出力

2桁↵

標準入力

12345

標準出力

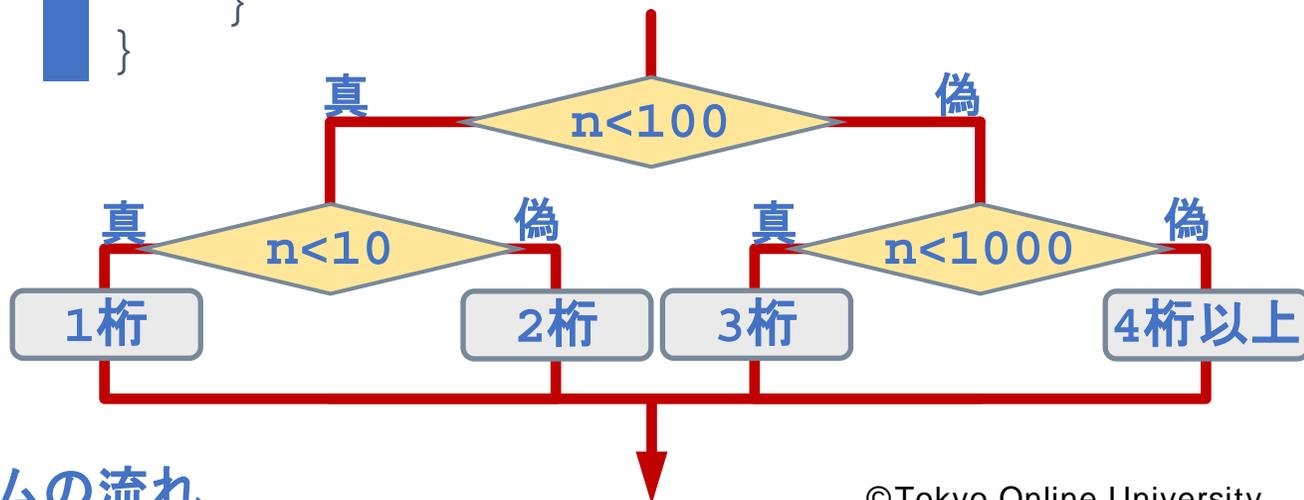
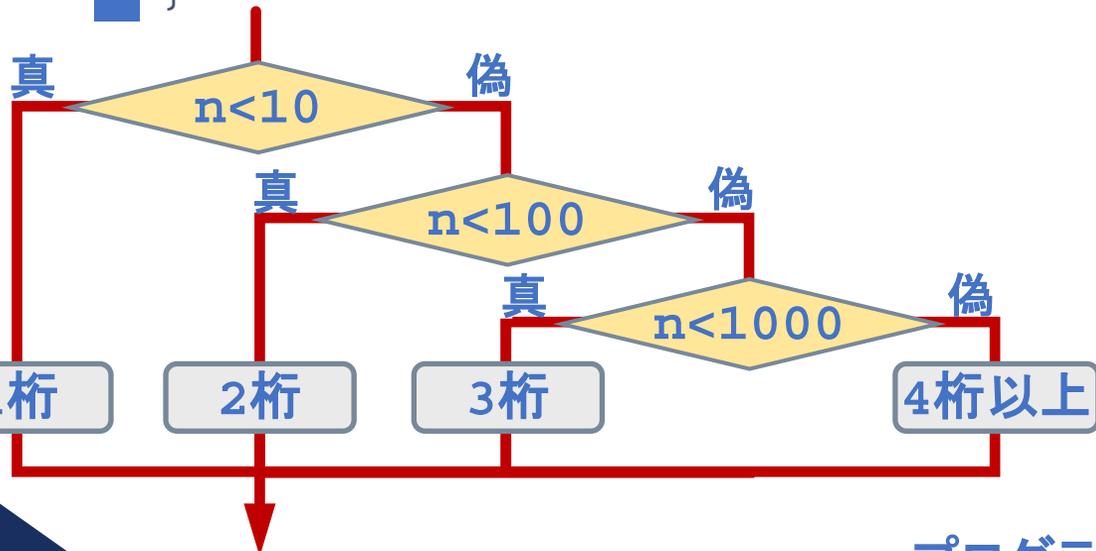
4桁以上↵

# 「例題3-5」のプログラム例（2）

## ◆ どちらも同じことを行う

```
if (n < 10) {  
    System.out.println("1桁");  
} else if (n < 100) {  
    System.out.println("2桁");  
} else if (n < 1000) {  
    System.out.println("3桁");  
} else {  
    System.out.println("4桁以上");  
}
```

```
if (n < 100) {  
    if (n < 10) {  
        System.out.println("1桁");  
    } else {  
        System.out.println("2桁");  
    }  
} else {  
    if (n < 1000) {  
        System.out.println("3桁");  
    } else {  
        System.out.println("4桁以上");  
    }  
}
```



プログラムの流れ

## 第3講のまとめ

- ◆ 条件分岐を用いた簡単なプログラムを作成した:
  - 例題を用いた条件分岐のプログラム作成
- ◆ Java における条件分岐について, 基本事項を理解した:
  - `else if` 文

## 第3回 条件分岐

# 第3講 `if` 文による分岐(2)

終わり

## 第3回 条件分岐

# 第4講 `switch` 文による分岐

## 第4講の学習目標

- ◆ Java における条件分岐について, 基本事項を理解する:
  - switch 文
- ◆ 第3回のまとめとして, これまでに習得した知識を用いたプログラムを作成する:
  - 例題を用いたプログラム作成

## 例題3-6

### 例題3-6

標準入力に何月かを表す整数値(1 ~ 12)を入力すると, 下表に従って季節の名前を標準出力に出力するプログラムを作成してください. なお, 1 ~ 12 以外の整数値が入力された場合は, 「?」を出力してください.

表

月	季節
3月, 4月, 5月	春
6月, 7月, 8月	夏
9月, 10月, 11月	秋
12月, 1月, 2月	冬

- ◆ まず, `if` 文による分岐で実現する.
- ◆ 別の方法として, `switch` 文による方法を理解する.

## 「例題3-6」の if 文による条件分岐

- ◆ ここまでに習得した知識でプログラムを記述すると…

```
int m = scan.nextInt();
if (3<=m && m<=5) {
    System.out.println("春");
} else if (6<=m && m<=8) {
    System.out.println("夏");
} else if (9<=m && m<=11) {
    System.out.println("秋");
} else if (m==12 || (1<=m && m<=2)) {
    System.out.println("冬");
} else {
    System.out.println("?");
}
```

- 第3講での記述に準じた方法

```
int m = scan.nextInt();
if (m==3 || m==4 || m==5) {
    System.out.println("春");
} else if (m==6 || m==7 || m==8) {
    System.out.println("夏");
} else if (m==9 || m==10 || m==11) {
    System.out.println("秋");
} else if (m==12 || m==1 || m==2) {
    System.out.println("冬");
} else {
    System.out.println("?");
}
```

- 論理和(または)を用いた方法

# 「例題3-6」のプログラム例 (if 文使用版)

```
1 import java.util.Scanner;
2
3 public class Example_03_06_1 {
4     public static void main(String[] args) {
5         Scanner scan = new Scanner(System.in);
6
7         int m = scan.nextInt();
8         if (m==3 || m==4 || m==5) {
9             System.out.println("春");
10        } else if (m==6 || m==7 || m==8) {
11            System.out.println("夏");
12        } else if (m==9 || m==10 || m==11) {
13            System.out.println("秋");
14        } else if (m==12 || m==1 || m==2) {
15            System.out.println("冬");
16        } else {
17            System.out.println("?");
18        }
19    }
20 }
```

Example\_03\_06\_1.java

## ◆ 実行例

標準入力

12

標準出力

冬←

標準入力

7

標準出力

夏←

標準入力

20

標準出力

?←

## switch文 (1)

- ◆ **switch**文は, 整数値で評価できる式をもとに多方向に分岐する文.
- ◆ switch文はまず「**整数式**」の部分の評価する.
- ◆ **case**の「**値 $i$** 」の中から一致する値を探し, その**case**に記述された文を実行する.
- ◆ 「**値 $i$** 」に一致するものがなければ**default**の部分に記述された文を実行する.
- ◆ **default**以下は省略できる.
- ◆ **break**文は, そこで処理を打ち切ってswitch文を終了させる役割をする.

```
switch (整数式) {  
    case 値1:  
        値1のとき実行する文  
        break;  
    case 値2:  
        値2のとき実行する文  
        break;  
        ⋮  
    case 値 $n$ :  
        値 $n$ のとき実行する文  
        break;  
    default:  
        その他のとき実行する文  
}
```

## ◆ switch文の使用例

```
int n = scan.nextInt();
switch (n) {
    case 1:
        System.out.println("n : 1");
        break;
    case 2:
        System.out.println("n : 2");
    case 3:
    case 4:
        System.out.println("n : 3 or 4");
        break;
    default:
        System.out.println("other number");
}
```

- 変数 n の値 (整数値) で分岐.

## ◆ 実行例1

標準入力

1

標準出力

n : 1↵

## ◆ 実行例2

標準入力

3

標準出力

n : 3 or 4↵

- 複数のcaseに対して, 同じことをさせる.

## ◆ 実行例3

標準入力

2

標準出力

n : 2↵  
n : 3 or 4↵

- break文がないと, 次のcaseに処理が移る.

- ◆ 「整数式」の部分は、整数値で評価できる式を記述できる.

```
int n = scan.nextInt();
switch (n%3) {
    case 0:
        System.out.println("3で割った余り : 0");
        break;
    case 1:
        System.out.println("3で割った余り : 1");
        break;
    case 2:
        System.out.println("3で割った余り : 2");
}
```

- 「 $a\%3$ 」( $a$ を3で割った余り)の結果は、整数値.

## ◆ 実行例

標準入力

16

標準出力

3で割った余り : 1↵

- ◆ 「整数式」の部分は、実数値(double型)は指定できない。

```
double x = scan.nextDouble();
switch (x) {
    case 10.0:
        System.out.println("x : 10.0");
        break;
    case 20.0:
        System.out.println("x : 20.0");
        break;
    default:
        System.out.println("other number");
}
```

- 変数 x は整数値でないため、コンパイルできない。

# 「例題3-6」のプログラム例

Example\_03\_06\_2.java

```
1 import java.util.Scanner;
2
3 public class Example_03_06 {
4     public static void main(String[] args) {
5         Scanner scan = new Scanner(System.in);
6
7         int m = scan.nextInt();
8         switch (m) {
9             case 3:
10            case 4:
11            case 5:
12                System.out.println("春");
13                break;
14            case 6:
15            case 7:
16            case 8:
17                System.out.println("夏");
18                break;
```

```
19            case 9:
20            case 10:
21            case 11:
22                System.out.println("秋");
23                break;
24            case 12:
25            case 1:
26            case 2:
27                System.out.println("冬");
28                break;
29            default:
30                System.out.println("?");
31            }
32        }
33    }
```

◆ 実行例

標準入力

標準出力

12

冬←

## 例題3-7

### 例題3-7

標準入力に華氏温度(°F)を実数値で入力すると、下記の計算式に従って摂氏温度(°C)を実数値で計算し画面上に表示するとともに、計算結果に応じて下記に示す表に従ってメッセージを表示するプログラムを作成してください。

$$C = \frac{5}{9}(F - 32)$$

( $C$ : 摂氏温度,  $F$ : 華氏温度)

#### メッセージの表示

摂氏温度(°C)	メッセージ
10.0°C未満	寒い
10.0°C以上20.0°C未満	涼しい
20.0°C以上30.0°C未満	暖かい
30.0°C以上	暑い

◆ これまでに習得した知識を利用して上記のプログラムを作成する。

- 変数, データの入出力, 四則演算(第2回)
- 条件分岐(第3回)

## 「例題3-7」のプログラムの流れ

◆ 華氏温度(°F), 摂氏温度(°C)の変数を, double型の変数 `f`, `c` とする.

◆ 標準入力から華氏温度(°F)を読み込む:

```
double f = scan.nextDouble();
```

- Scanner クラスの変数 `scan` は, 事前に宣言されているとする.

◆ 華氏温度(°F)を摂氏温度(°C)に変換し, 結果を表示する:

```
double c = 5.0/9.0*(f - 32.0);  
System.out.println("摂氏" + c + "度");
```

- 「5.0」, 「9.0」と記述することでdouble型として認識される.
- 「摂氏〇〇度」と出力する.

◆ 変数 `c` の値に従って, 条件分岐によりメッセージを出力する.

# 「例題3-7」のプログラム例

Example\_03\_07.java

```
1 import java.util.Scanner;
2
3 public class Example_03_07 {
4     public static void main(String[] args) {
5         Scanner scan = new Scanner(System.in);
6
7         double f = scan.nextDouble();
8         double c = 5.0/9.0*(f - 32.0);
9         System.out.println("摂氏" + c + "度");
10
11         if (c < 10.0) {
12             System.out.println("寒い");
13         } else if (c < 20.0) {
14             System.out.println("涼しい");
15         } else if (c < 30.0) {
16             System.out.println("暖かい");
17         } else {
18             System.out.println("暑い");
19         }
20     }
21 }
```

## ◆ 実行例

標準入力

77

標準出力

摂氏25.0度↵  
暖かい↵

標準入力

55

標準出力

摂氏12.77...79度↵  
涼しい↵

## 第4講のまとめ

- ◆ Java における条件分岐について、基本事項を理解した：
  - switch 文
- ◆ 第3回のまとめとして、これまでに習得した知識を用いたプログラムを作成した：
  - 例題を用いたプログラム作成

## 第3回 条件分岐

# 第4講 `switch` 文による分岐

終わり