



# ji-3. 条件分岐と場合分け

(Java プログラミング入門)

URL: <https://www.kkaneko.jp/cc/ji/index.html>

金子邦彦



# 内容



例題 1. 平方根の計算

if 文, else 文, 条件分岐, 比較演算

例題 2. 定形郵便物の料金

例題 3. 2 次方程式

多分岐

例題 4. 直線と原点の距離

比較演算と論理演算の組み合わせ

# 目標

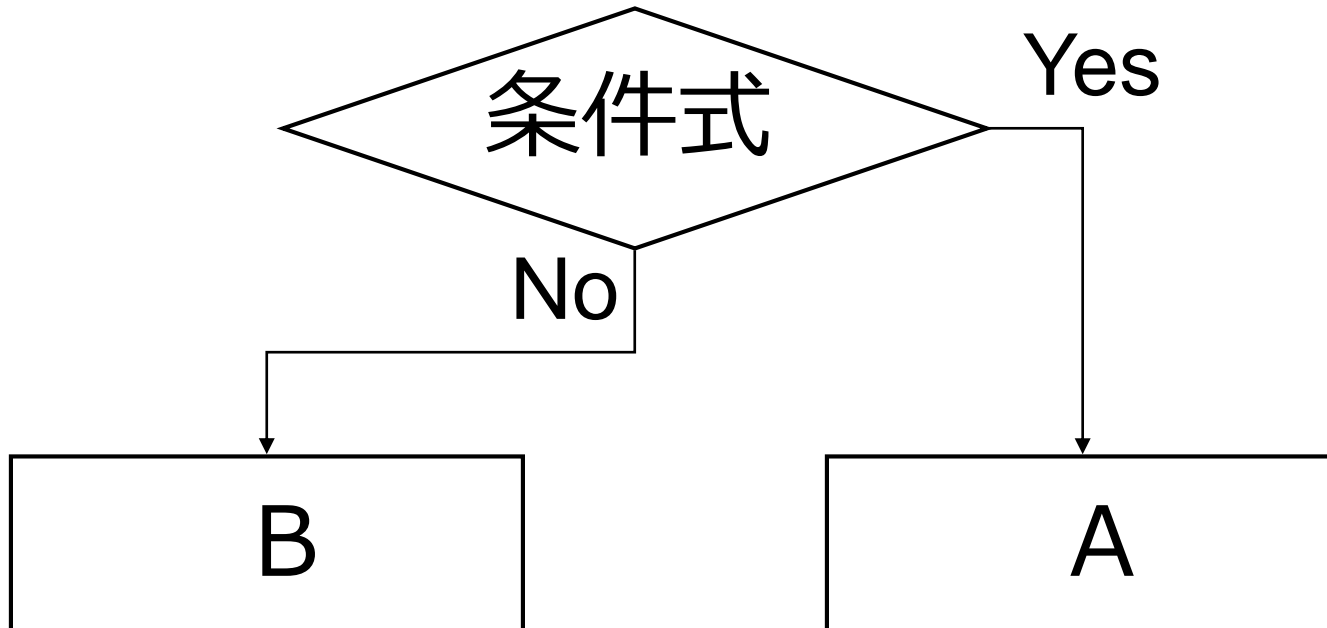


- **条件分岐 (if 文)** を使って, より役に立つプログラムを作ることができるようになる.
- **比較演算 (<, <=, >, >=, =, <>)** の使い方を理解する
- 前回習った**四則演算, Java の標準ライブラリ (指数関数, 対数関数)** も使う

# 条件分岐



条件分岐では、**条件式が成り立てばAを、成り立たなければBを実行**というように分岐する





- プログラミングを行えるオンラインのサービス

<https://www.onlinegdb.com>

- ウェブブラウザを使う

- たくさんの言語を扱うことができる

Python3, Java, C/C++, C#, JavaScript,  
R, アセンブリ言語, SQL など

- オンラインなので、「秘密にしたいプログラム」  
を扱うには十分な注意が必要

# Online GDB で Java を動かす手順



① ウェブブラウザを起動する

② 次の URL を開く

<https://www.onlinegdb.com>

A screenshot of a web browser's address bar. The address bar is a light gray rectangle with a magnifying glass icon on the left. Inside the bar, the text "https://www.onlinegdb.com" is displayed in a dark gray font. Below the address bar, there is a thin horizontal line and a larger, empty light gray rectangular area.



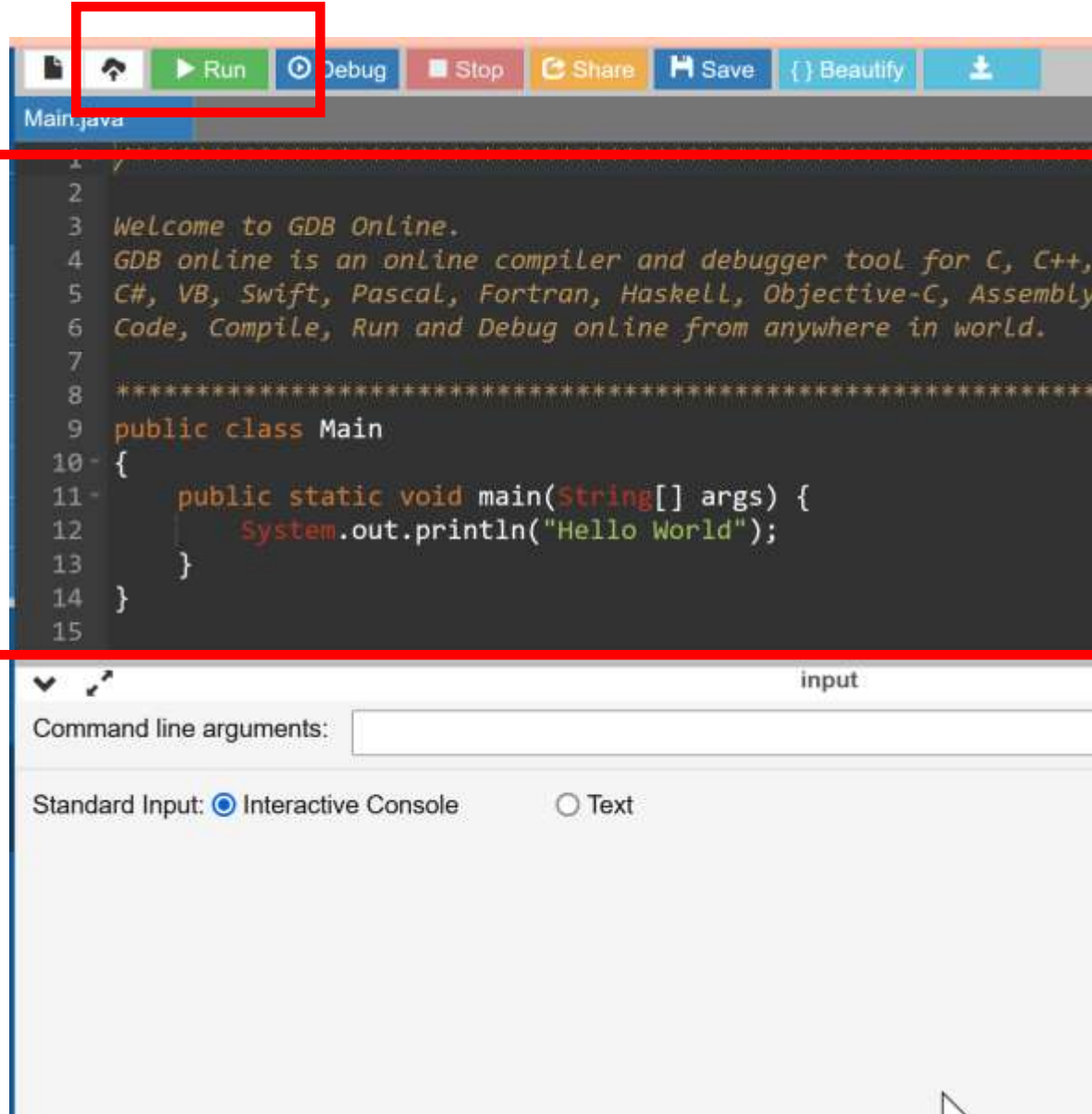
### ③ 「Language」 のところで、「Java」 を選ぶ

The screenshot shows the GDB Online web interface. At the top, there is a navigation bar with buttons for Run, Debug, Stop, Share, Save, Beautify, and a Language dropdown menu. The Language dropdown menu is open, showing a list of programming languages. The 'Java' option is highlighted in blue. The main area of the interface displays a C program with a 'Hello World' message.

```
1 - /*****  
2  
3 Welcome to GDB OnLine.  
4 GDB online is an online compiler and debugger tool for C, C++, Python  
5 C#, VB, Perl, Swift, Prolog, Javascript, Pascal, HTML, CSS, JS  
6 Code, Compile, Run and Debug online from anywhere in world.  
7  
8 *****/  
9 #include <stdio.h>  
10  
11 int main()  
12 {  
13     printf("Hello World");  
14  
15     return 0;  
16 }  
17
```



# 実行ボタン



エディタ画面

プログラムを  
書き換えること  
ができる



# if, else

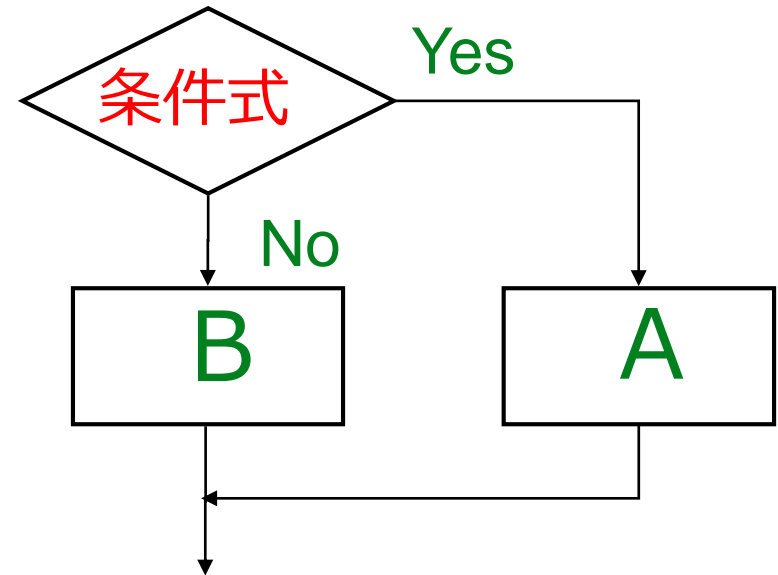


**if, else** を用いて, 「**条件式**」が**成り立てばA**を、**成り立たなければB**を実行するようなプログラムを書くことができる。

```
if ( 条件式 ) {  
    文;  
    文;  
    ...  
} else {  
    文;  
    文;  
    ...  
}
```

A

B



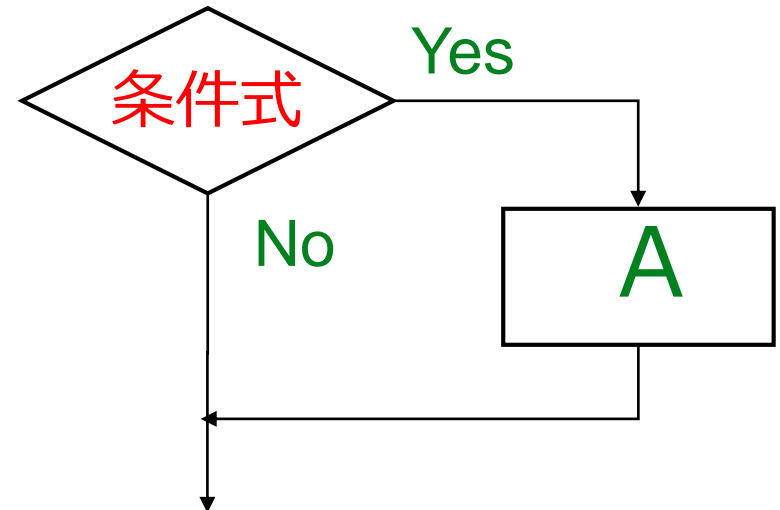
# if のみ



- **if のみ**を書いて, **else** を書かないこともできる
- 「**条件式**」が**成り立つ**ときに限り**A**を実行

```
if ( 条件式 ) {  
  文;  
  文;  
  ...  
}
```

A



# 比較演算



**比較演算**は、**条件式**を書くときなどに使用

<	左辺が右辺より <b>小さい</b>
<=	左辺が右辺 <b>以下</b>
>	左辺が右辺より <b>大きい</b>
>=	左辺が右辺 <b>以上</b>
=	左辺が右辺と <b>等しい</b>
<>	左辺が右辺と <b>等しくない</b>

# 比較演算の例



```
if (x >= 0) {  
    System.out.printf("sqrt(x) = %8.3f\n", Math.sqrt(x));  
} else {  
    System.out.printf("out of range, x = %8.3f\n", x);  
}
```

条件式  $x \geq 0$  は, 「 $x$  が 0 以上」の意味

# 平方根を求める



- 次は、平方根を求めるプログラム
- 「-5」を与えると、NaN と表示される

これは正常動作. そもそも  $x$  の値がマイナスになるのが**想定外** → 次ページ以降で改良版を示す

```
1 import java.lang.Math;
2 import java.util.Scanner;
3 public class Main
4 {
5     public static void main(String[] args) {
6         double x;
7         Scanner s = new Scanner(System.in);
8         System.out.println("Please Enter x =");
9         x = s.nextDouble();
10        System.out.printf("sqrt(x) = %8.3f\n", Math.sqrt(x));
11    }
12 }
13
```

```
▼ ↗ 🔍
Please Enter x =
-5
sqrt(x) =      NaN
```

# 例題 1. 平方根の計算



## 浮動小数点数データを読み込んで、平方根の計算と表示を行うプログラム

- 但し、**負の数の場合**には、「計算できないこと」を意味するメッセージを表示する

正または0のとき： `sqrt()` を使って計算できる

負のとき： 「計算できないこと」と、確認のために `x` の値も表示

- **負であるかどうかによって条件分岐を行う**ために `if` 文を使う



```
import java.lang.Math;
import java.util.Scanner;
public class Main
```

```
{
    public static void main(String[] args) {
```

**条件式**

```
    double x;
    Scanner s = new Scanner(System.in);
    System.out.println("Please Enter x =");
    x = s.nextDouble();
```

**条件が成り立つ場合に実行される部分**

```
    if (x >= 0) {
```

```
        System.out.printf("sqrt(x) = %8.3f¥n", Math.sqrt(x));
```

```
    } else {
```

```
        System.out.printf("out of range, x = %8.3f¥n", x);
```

```
    }
```

**条件が成り立たない場合に実行される部分**

```
}
```

```
}
```



## 実行結果の例

$x = 9$  のときは、**平方根**が表示される

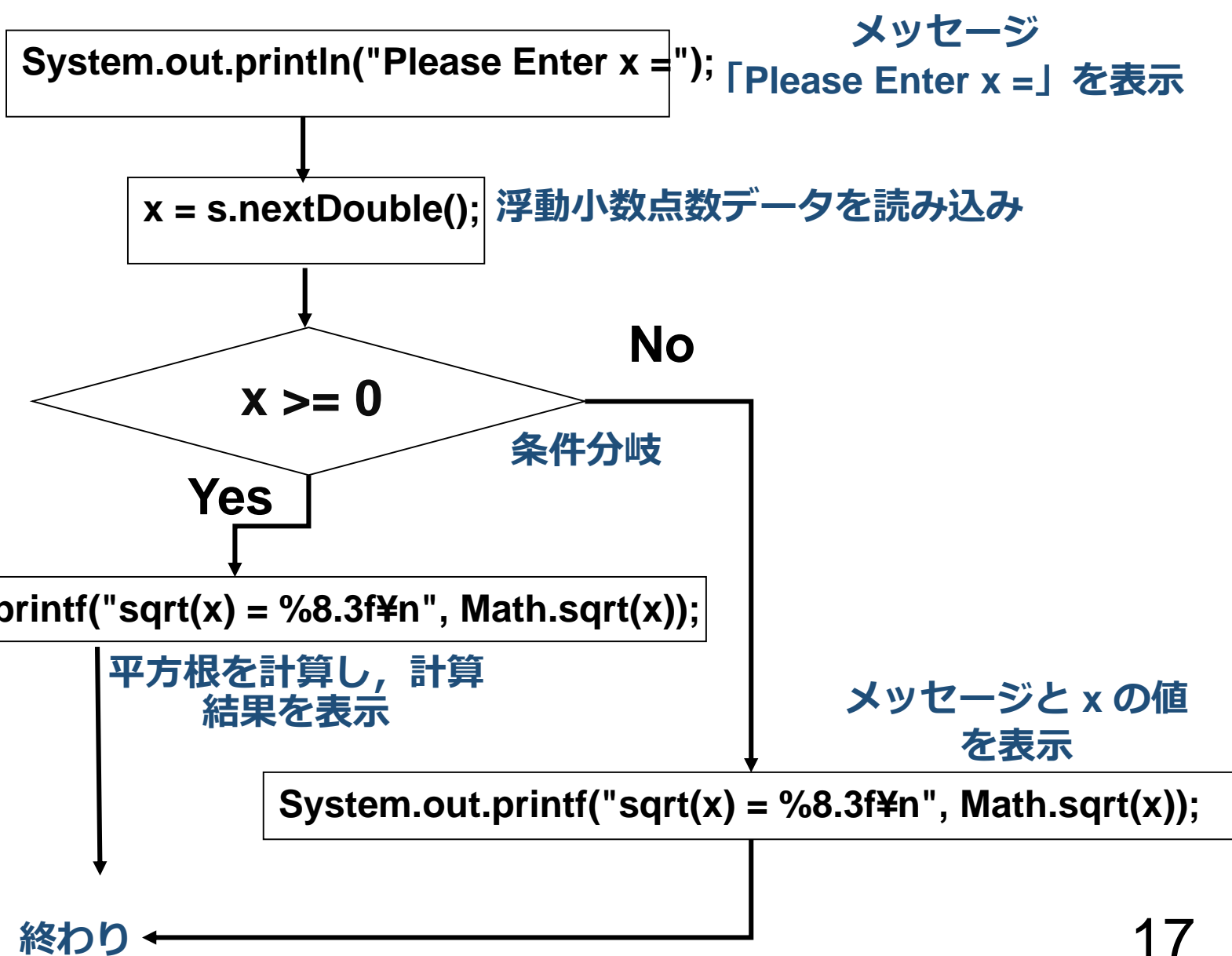
```
Please Enter x =  
9  
sqrt(x) = 3.000
```

$x = -5$  のときは、**計算できない**という  
メッセージと、 $x$  の値が表示される

```
Please Enter x =  
-5  
out of range, x = -5.000
```

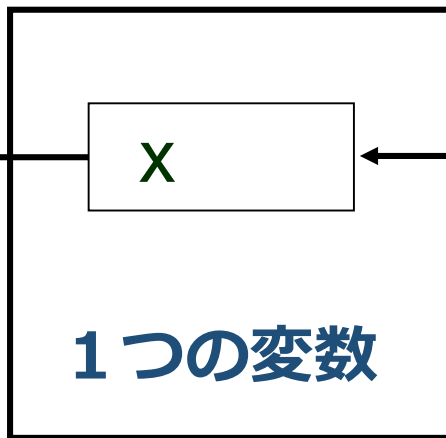


# プログラム実行順





## メモリ



①

```
x = s.nextDouble();
```

浮動小数点数  
データを読み込み

③

```
System.out.printf("sqrt(x) = %8.3f¥n", Math.sqrt(x));
```

計算と表示 ( $x \geq 0$  のとき)



## 例題 2. 定形郵便物の料金

- キーボードから数値を読み込んで,

25 以下なら	「84 Yen」
25 より大きくて50以下なら	「94 Yen」
50より大きいなら	「Too heavy」

と表示するプログラムを作る

- **多分岐**を行うために, **if 文**を使う



```
import java.lang.Math;
import java.util.Scanner;
public class Main
{
    public static void main(String[] args) {
        double x;
        Scanner s = new Scanner(System.in);
        System.out.println("juuryou?");
        x = s.nextDouble();
        if (x <= 25) {
            System.out.println("84 Yen");
        } else if (x <= 50 ) {
            System.out.println("94 Yen");
        } else {
            System.out.println("Too heavy");
        }
    }
}
```

**x <= 25 が成り立つ  
場合に実行される部分**

**x <= 25 が成り立たず,  
x <= 50 が成り立つ  
場合に実行される部分**

**x <= 25 も成り立たず,  
x <= 50 も成り立たない  
場合に実行される部分**

# 定形郵便物の料金



## 実行結果の例

```
juuryou?  
20  
84 Yen
```

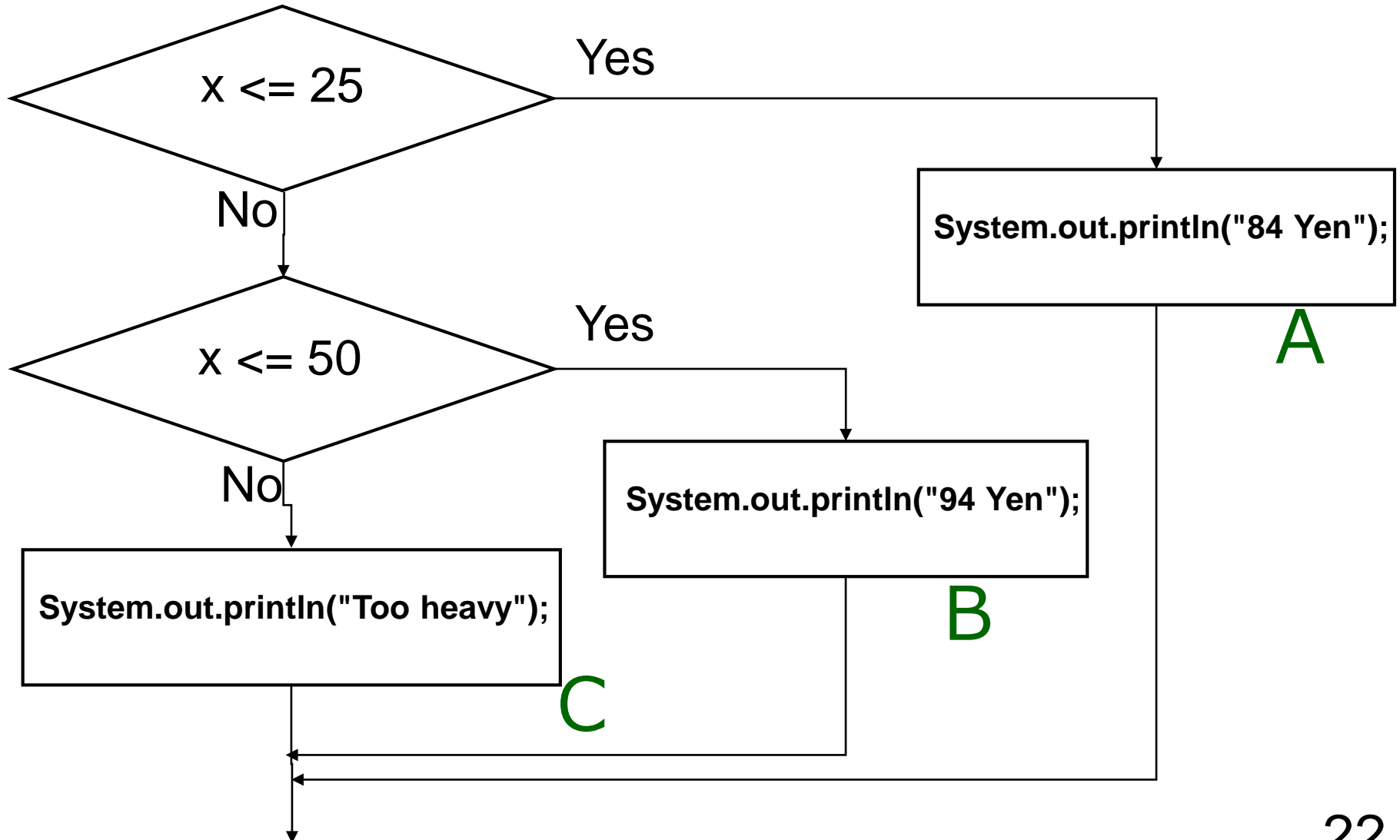
```
juuryou?  
40  
94 Yen
```

```
juuryou?  
60  
Too heavy
```

# 定形郵便物の料金



A, B, C のうちどれか 1 つを実行





```
import java.lang.Math;
import java.util.Scanner;
public class Main
{
    public static void main(String[] args) {
        double x;
        Scanner s = new Scanner(System.in);
        System.out.println("juuryou?");
        x = s.nextDouble();
        if (x <= 25) {
            System.out.println("84 Yen");
        } else if (x <= 50 ) {
            System.out.println("94 Yen");
        } else {
            System.out.println("Too heavy");
        }
    }
}
```

このうちどれか1つ  
が実行される

## 例題 3. 2 次方程式



- 2 次方程式  $ax^2 + by + c = 0$  の解を求めるプログラムを作る
  - 重解, 虚数解も正しく求めるために, 判別式  $b^2 - 4ac$  の値で条件分岐する
  - 「 $a=0$ 」の場合は考えない





$D = b^2 - 4ac$  とする

1) 判別式  $D > 0$  のとき

$$x = \frac{-b + \sqrt{D}}{2a}, \frac{-b - \sqrt{D}}{2a}$$

**異なる 2 実数解**

2)  $D = 0$  のとき

$$x = -\frac{b}{2a},$$

**重解**

3)  $D < 0$  のとき

$$x = \frac{-b + i\sqrt{-D}}{2a}, \frac{-b - i\sqrt{-D}}{2a}$$

**異なる 2 虚数解**

```

import java.lang.Math;
import java.util.Scanner;
public class Main
{
    public static void main(String[] args) {
        double a, b, c, D;
        Scanner s = new Scanner(System.in);
        System.out.println("Please Enter a =");
        a = s.nextDouble();
        System.out.println("Please Enter b =");
        b = s.nextDouble();
        System.out.println("Please Enter c =");
        c = s.nextDouble();
        D = b * b - 4 * a * c;
        if (D > 0) {
            System.out.printf("x = %8.3f, %8.3f\n",
                (-b + Math.sqrt(D)) / (2 * a),
                (-b - Math.sqrt(D)) / (2 * a));
        } else if (D == 0) {
            System.out.printf("x = %8.3f\n", -b / (2*a));
        } else {
            System.out.printf("x = %8.3f + %8.3f i, %8.3f - %8.3f i\n",
                -b / (2 * a), Math.sqrt(-D) / (2 * a),
                -b / (2 * a), Math.sqrt(-D) / (2 * a));
        }
    }
}

```



**D > 0 のとき**

**D = 0 のとき**

**D < 0 のとき**

# 2次方程式



## 実行結果の例

```
Please Enter a =  
1  
Please Enter b =  
5  
Please Enter c =  
6  
x = -2.000, -3.000
```

```
Please Enter a =  
-2  
Please Enter b =  
-8  
Please Enter c =  
-8  
x = -2.000
```

```
Please Enter a =  
1  
Please Enter b =  
1  
Please Enter c =  
1  
x = -0.500 + 0.866 i, -0.500 - 0.866 i
```

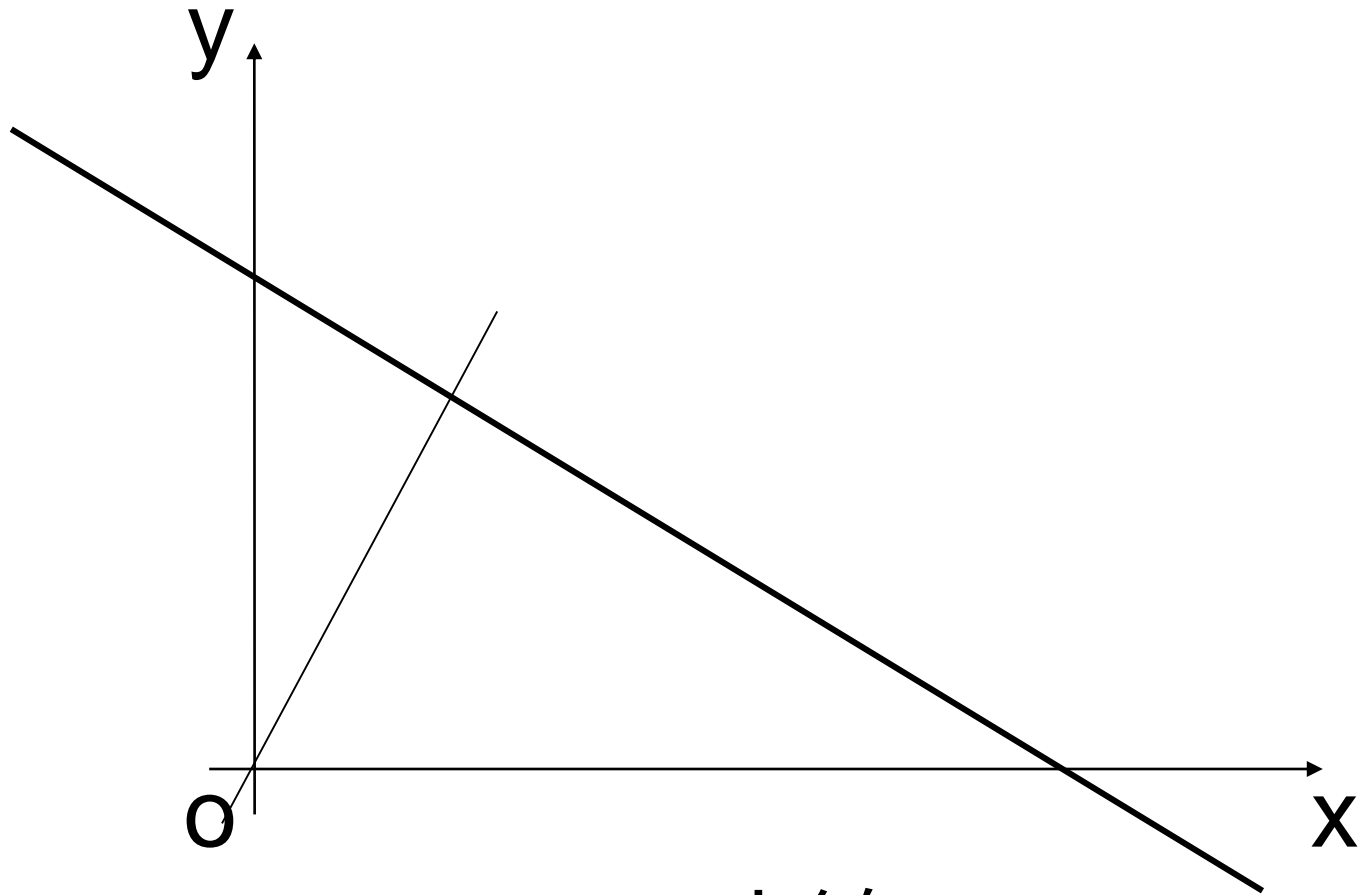
## 例題 4 . 直線と原点の距離



- 平面 :  $ax + by + c = 0$  と原点との距離を求めるプログラムを作成する
- 但し,  $a = 0$  かつ  $b = 0$  のときには計算を行わない

$$d = \frac{|c|}{\sqrt{a^2 + b^2}}$$

# 直線と原点の距離



直線  $ax + by + c = 0$



```
import java.lang.Math;
import java.util.Scanner;
public class Main
{
    public static void main(String[] args) {
        double a, b, c, d;
        Scanner s = new Scanner(System.in);
        System.out.println("Please Enter a =");
        a = s.nextDouble();
        System.out.println("Please Enter b =");
        b = s.nextDouble();
        System.out.println("Please Enter c =");
        c = s.nextDouble();
        if ((a == 0) && (b == 0)) {
            System.out.println("a=0 and b=0!");
        } else {
            d = Math.abs(c) / Math.sqrt(a * a + b * b);
            System.out.printf("d = %8.3f¥n", d);
        }
    }
}
```

**条件が成り立つ場合に  
実行される部分**

**条件が成り立たない  
場合に実行される部分 30**

# 直線と原点の距離



## 実行結果の例

```
Please Enter a =  
1  
Please Enter b =  
1  
Please Enter c =  
1  
d =      0.707
```

```
Please Enter a =  
0  
Please Enter b =  
0  
Please Enter c =  
1  
a=0 and b=0!
```

# 条件式



$(a == 0) \ \&\& \ (b == 0)$

$a=0$  である

$b=0$  である

$a=0$  である

$b=0$  である

かつ



# 論理演算



$A \ \&\& \ B$

A かつ B

$A \ || \ B$

A または B

$!A$

Aでない

真, 偽に関する論理的な演算を行う.

# 演習. 論理式に関する演習



(1) 2次方程式  $ax^2 + by + c = 0$  の解を求めるプログラムを作りなさい

例題3のプログラムを書き換えて、 **$a=0$** の場合にも、**正しく解を求める**ようにしなさい

(2) **平面  $ax + by + cz + d = 0$  と原点の距離**を求めるプログラムを作りなさい



## 演習 (1) で行うこと

- $a = 0$  かつ  $b = 0$  かつ  $c = 0$  のとき  
「すべての  $x$  が解である」と表示
- $a = 0$  かつ  $b = 0$  かつ  $c \neq 0$  のとき  
「解なし」と表示
- $a = 0$  かつ  $b \neq 0$  のとき  
$$x = -c / b$$



## 演習（1）で行うこと

- $a = 0$  かつ  $b \neq 0$  のとき

$$x = -c / b$$

この部分のプログラム例（正解は1つでは無い）

```
else if ((a == 0) && (b != 0)) {  
    System.out.printf("x = %8.3f", -c / b);  
}
```