

初級プログラミング I

第4回 プログラミング演習(1)

中谷 祐介

第4回の構成

◆ 第1講 補足事項

- これまでに説明していない項目を理解する.
- 演習課題の進め方について理解する.

◆ 第2講 ~ 第4講 プログラミング演習

- 「@CODE ROOM」において、プログラミングの演習を行う.

第4回 プログラミング演習(1)

第1講 補足事項


第1講の学習目標

- ◆ これまでに説明していない補足事項について理解する:
 - 数学関数
 - `printf` メソッドによる出力
- ◆ 演習課題の進め方について理解する.

これまでの復習

- ◆ 第1回 コンピュータとプログラミング
 - ハードウェア, ソフトウェア, プログラミング
 - プログラミング環境「@CODE ROOM」でのプログラミング方法
 - ◆ 第2回 プログラミングの初歩
 - ソースファイルの記述
 - データの入出力
 - 変数と四則演算
 - ◆ 第3回 条件分岐
 - 条件式の記述
 - `if` 文による分岐, `switch` 文による分岐
- ➡ これまでの知識を用いて, Java によるプログラミングの演習を行う.

数学関数 (1)

- ◆ 第2回第4講において、基本的な演算として、算術演算子(+, -, *, /, %)を用いた四則演算の方法を習得した.
 - ◆ 場合によっては、より複雑な計算が必要な場合が考えられる:
 - 平方根, 累乗
 - 三角関数, 対数関数
 - 円周率 π
- 
- ◆ Java では、Math クラスというツールを利用することで、数学関数、数学定数を利用できる:
 - Math クラス内では、数学関数をメソッドとして提供.

数学関数 (2)

数学関数

| メソッド | 意味 |
|---|--------------------------|
| Math.abs(<i>int</i> x) | 絶対値 ($ x $) |
| Math.abs(<i>double</i> x) | |
| Math.sqrt(<i>double</i> x) | 平方根 (\sqrt{x}) |
| Math.pow(<i>double</i> x, <i>double</i> y) | 累乗 (x^y) |
| Math.sin(<i>double</i> x) | 正弦 ($\sin x$) |
| Math.cos(<i>double</i> x) | 余弦 ($\cos x$) |
| Math.tan(<i>double</i> x) | 正接 ($\tan x$) |
| Math.toDegrees(<i>double</i> x) | ラジアンから度への変換 |
| Math.toRadians(<i>double</i> x) | 度からラジアンへの変換 |
| Math.exp(<i>double</i> x) | 自然対数の底 e の累乗 (e^x) |
| Math.log(<i>double</i> x) | 自然対数 ($\log x$) |
| Math.log10(<i>double</i> x) | 常用対数 ($\log_{10} x$) |

数学定数

| 定数 | 意味 | 値 |
|---------|------------|-----------------|
| Math.E | 自然対数の底 e | 2.7182818284... |
| Math.PI | 円周率 π | 3.1415926535... |

◆ Math クラスの各種メソッドの利用例

```
double x = 3.0;
double a = Math.pow(x, 2.0);
double b = Math.sqrt(x);
double c = Math.sin(Math.toRadians(30.0));

System.out.println(" a : " + a);
System.out.println(" b : " + b);
System.out.println(" c : " + c);

System.out.println("PI : " + Math.PI);
```

a : 9.0
b : 1.7320508075688772
c : 0.49999999999999994
PI : 3.141592653589793

← x^2
← \sqrt{x}
← $\sin(30^\circ)$

← 円周率 π

- $\sin(30^\circ)$ は「0.5」。計算結果には誤差が含まれている。

printf メソッド (1)

[Sample_04_02.java]

```
System.out.printf("書式指定子を含む文字列", データならび)
```

- ◆ 書式指定子に従って、データを出力するメソッド.
- ◆ printf メソッドによるデータの出力の例 (1)

```
int a = 123;  
System.out.printf("%d¥n", a);  
double x = a / 10.0;  
System.out.printf("%f¥n", x);
```

```
123←  
12.300000←
```

- 「 " " 」で囲まれた部分を出力する.
- 「%d」の部分が「,」以降の変数 **a** の内容に置き換わる.
- 「%f」の部分が「,」以降の変数 **x** の内容に置き換わる.
- 「%d」を**書式指定子**という.
- 出力する変数の型により、書式指定子が異なる.

書式指定子

| 書式指定子 | 出力する値の型 |
|-------|--------------|
| %c | char (文字) |
| %d | int (整数値) |
| %f | double (実数値) |
| %s | String (文字列) |

printf メソッド (2)

[Sample_04_02.java]

◆ printf メソッドによるデータの出力の例 (2)

```
int b = 5;  
double y = Math.sqrt(5.0);  
System.out.printf("b:%d, y:%f\n", b, y);
```

b:5, y:2.236068

- 変数の型に応じて、書式指定子を使い分ける。
- 「%d」、 「%f」がそれぞれ変数 **b**, **y** の内容に置き換わる。

◆ printf メソッドによるデータの出力の例 (3)

```
System.out.printf("東京通信大学\n情報マネジメント学部\n");  
System.out.printf("初級プログラミングI\n");
```

東京通信大学
情報マネジメント学部
初級プログラミングI

- 文字列のみを出力する場合は、「 " "」のあとの「,」以降は不要。

printf メソッド (3)

[Sample_04_03.java]

- ◆ printf メソッドでは, 出力する数値の桁数を指定することができる.
- ◆ 桁数を指定しての出力の例 (1)

```
int a = 4;  
int b = 1234;  
System.out.printf("a:%4d\n", a);  
System.out.printf("b:%4d\n", b);
```

```
a:    4  
b:1234
```

- 「%4d」と指定すると, 整数値を4桁で出力.
- 出力する整数値が4桁未満の場合は, 空白を出力し, 右詰めで出力.

◆ 桁数を指定しての出力の例 (2)

```
double x = Math.sqrt(2.0);  
System.out.println("x:" + x);  
System.out.printf("x:%f\n", x);  
System.out.printf("x:%8.5f\n", x);  
System.out.printf("x:%.2f\n", x);
```

← $\sqrt{2}$

```
x:1.4142135623730951  
x:1.414214  
x: 1.41421  
x:1.41↵
```

- 「%f」の場合, 小数点以下6桁まで出力.
- 「%8.5f」と指定すると, 小数点を含めて8桁で, そのうち小数部分を5桁で出力.
整数部分が桁数に満たない場合は, 空白を出力し, 右詰めで出力.
- 「%.2f」と指定すると, 整数部分に関係なく, 小数部分を2桁で出力.

例題4-1

例題4-1

標準入力に円の半径 r に相当する実数値を入力すると、半径 r の円の円周の長さを計算し、計算結果を小数点以下3桁まで標準出力に出力するプログラムを作成してください。

$$\text{円周の長さ} = 2\pi r$$

```
1 import java.util.Scanner;
2
3 public class Example_04_01 {
4     public static void main(String[] args) {
5         Scanner scan = new Scanner(System.in);
6
7         double r = scan.nextDouble();
8         double l = 2*Math.PI*r;
9         System.out.printf("%.3f¥n", l);
10    }
11 }
```

Example_04_01.java

◆ 実行例

標準入力

2.5

標準出力

15.708↵

プログラミング演習

- ◆ 第1回～第4回の内容を含んだ演習課題を出題する.
- ◆ 「@CODE ROOM」を確認し, 演習課題に取り組む.
 - 各課題には「レベル1」～「レベル3」を設定している.
 - 各課題のプログラムを作成する. その際, 各課題に示されている要件を満たすようにプログラムを作成すること.
 - プログラムの作成後は, コンパイル, 実行を行う.
 - 正しく動作していれば, 「採点」(提出)を行う.
- ◆ 「@CODE ROOM」の利用方法は, 第1回第4講を復習すること.
- ◆ 良い成績を目指すには, なるべく多くの課題に取り組むこと.

第1講のまとめ

- ◆ これまでに説明していない補足事項について理解した:
 - 数学関数
 - `printf` メソッドによる出力
- ◆ 演習課題の進め方について理解した.

第4回 プログラミング演習(1)

第1講 補足事項

終わり