

初級プログラミング I

第8回 プログラミング演習(2)

中谷 祐介

第8回の構成

◆ 第1講 補足事項とまとめ

- これまでに説明していない補足事項について理解する.
- 本科目の内容を振り返り, まとめを行う.

◆ 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8回 プログラミング演習(2)

第1講 補足事項とまとめ

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:
 - 文字列に関する補足
 - Scanner クラスに関する補足
- ◆ 本科目の内容を振り返り, まとめを行う.

これまでの復習

◆ 第5回 繰り返し

- `for` 文による繰り返し, `while` 文による繰り返し
- 二重ループ

◆ 第6回 配列

- 配列の宣言, 初期化, 利用
- 二次元配列
- `Scanner` クラス (`hasNextInt` メソッド, `hasNextDouble` メソッド)

◆ 第7回 文字と文字列

- 文字と文字コード
- `String` 型, 文字列の処理を行うメソッド

➡ 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.

文字列の比較 (1)

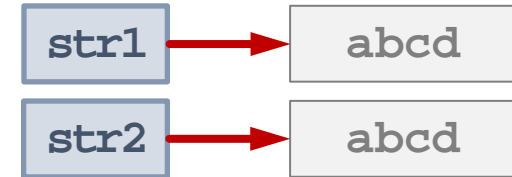
[Sample_08_01.java]

- ◆ 標準入力に入力した2つの文字列の内容が等しいかどうかを判定するプログラム:

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

String str1 = scan.nextLine();
String str2 = scan.nextLine();

if (str1 == str2) {
    System.out.println("等しい");
} else {
    System.out.println("等しくない");
}
```



- String 型は参照型.
- 「str1 == str2」は, 2つの文字列が参照する先が等しいかどうかを判定.
- 実行例

標準入力

abcd
abcd

標準出力

等しくない

文字列の比較 (2)

[Sample_08_01.java]

- ◆ 標準入力に入力した2つの文字列の内容が等しいかどうかを判定するプログラム:

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

String str1 = scan.nextLine();
String str2 = scan.nextLine();

if (str1.equals(str2)) {
    System.out.println("同じ文字列");
} else {
    System.out.println("異なる文字列");
}
```

- equals メソッドは, () 内に指定した文字列と等しいかどうかを判定するメソッド.
- 「str2.equals(str1)」としてもよい.
- 実行例

標準入力

abcd
abcd

標準出力

同じ文字列↵

Scanner クラス (1)

- ◆ Scanner クラスには, 数値をプログラム内に読み込む道具がある:
 - `nextInt` メソッド: 整数値 (`int` 型) を読み込むメソッド.
 - `nextDouble` メソッド: 実数値 (`double` 型) を読み込むメソッド.
- ◆ Scanner クラスには, 次に読み込む数値があるかどうかを調べる道具がある:
 - `hasNextInt` メソッド: 次に読み込む整数値があるかどうかを調べるメソッド.
 - `hasNextDouble` メソッド: 次に読み込む実数値があるかどうかを調べるメソッド.
- ◆ Scanner クラスには, 次に読み込む文字列があるかどうかを調べる道具がある:
 - **`hasNext` メソッド**: 次に読み込む文字列があるかどうかを調べるメソッド.
空白や改行までを, 一つの文字列として認識する.
文字列があれば `true`, なければ `false` となる.
 - **`hasNextLine` メソッド**: 次に読み込む1行の文字列があるかどうかを調べるメソッド.
次の行があれば `true`, なければ `false` となる.

◆ 標準入力に入力した複数の文字列を標準出力に出力するプログラム:

```
Scanner scan = new Scanner(System.in);  
  
while (scan.hasNext()) {  
    String str = scan.next();  
    System.out.println(str);  
}
```

- 「標準入力」欄に文字列が存在する間, 繰り返す.
- 空白や改行までを, 一つの文字列として認識する.

• 実行例

標準入力

```
Tokyo Online University  
Java programming
```

標準出力

```
Tokyo↵  
Online↵  
University↵  
Java↵  
programming↵
```

◆ 標準入力に入力した複数行の文字列を標準出力に出力するプログラム:

```
Scanner scan = new Scanner(System.in);  
  
while (scan.hasNextLine()) {  
    String str = scan.nextLine();  
    System.out.println(str);  
}
```

- 「標準入力」欄に文字列が存在する間, 繰り返す.
- 改行までを, 一つの文字列として認識する.

• 実行例

標準入力

```
Tokyo Online University  
Java programming
```

標準出力

```
Tokyo Online University↵  
Java programming↵
```

例題8-1

例題8-1

標準入力に1行以上の文字列を入力すると、入力した各行の文字列の文字数を標準出力に1行ずつ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_08_01 {
4     public static void main(String[] args) {
5         Scanner scan = new Scanner(System.in);
6
7         while (scan.hasNextLine()) {
8             String str = scan.nextLine();
9             int n = str.length();
10            System.out.println(n);
11        }
12    }
13 }
```

Example_08_01.java

◆ 実行例

標準入力

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

標準出力

6↵
10↵
9↵

プログラミング演習

- ◆ 第5回～第8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.

本科目のまとめ

回	タイトル
1	コンピュータとプログラミング
2	プログラミングの初歩
3	条件分岐
4	プログラミング演習(1)
5	繰り返し
6	配列
7	文字と文字列
8	プログラミング演習(2)

第1回 コンピュータとプログラミング

講	項目	内容
1	ハードウェア	CPU, 記憶装置, 入出力装置, 通信制御装置
2	ソフトウェア	基本ソフトウェア, 応用ソフトウェア, ミドルウェア
3	プログラミング	プログラミング言語の種類と歴史, Java の概要
4	プログラミング環境	プログラミング環境「@CODE ROOM」の利用方法, プログラミングの手順(ソースファイルの編集, コンパイル, 実行)

第2回 プログラミングの初歩

講	項目	内容
1	ソースファイル	Java のソースファイルの形式, プログラムの記述(半角文字の使用, 文末は「;」, コメントの記述, インデント)
2	データの出力	println メソッド, print メソッド, 改行文字(\n), 文字列の連結
3	変数	型(int 型, double 型, 等), 宣言, 変数名, 値の代入
4	データの入力	Scanner クラス(nextInt メソッド, nextDouble メソッド), 整数値・実数値の入力
4	四則演算	算術演算子(+, -, *, /, %), 演算結果, 優先順位

第3回 条件分岐

講	項目	内容
1	条件分岐	if 文
2	条件式	等価演算子(==, !=), 関係演算子(<, >, <=, >=), 論理演算子(&&, , !)
3	条件分岐	if 文の例題, else if 文
4	条件分岐	switch 文

第4回 プログラミング演習(1)

講	項目	内容
1	数学関数・ 数学定数	Math クラス
1	データの出力	printf メソッド, 書式指定子(%d, %f, 等)

第5回 繰り返し

講	項目	内容
1	繰り返し	for 文
1	便利な演算子	複合代入演算子(+=, -=, *=, /=, %=, 等), インクリメント・デクリメント演算子(++ , --)
2	繰り返し	while 文, do - while 文
3	繰り返し	繰り返しの例題
3	演算	キャスト演算子
4	繰り返し	二重ループ
4	流れの制御	break 文, continue 文

第6回 配列

講	項目	内容
1	配列	宣言, 初期化, 利用, 要素数
2	配列	配列の要素の最大値
2	変数	final 変数
3	データの入力	Scanner クラス (hasNextInt メソッド, hasNextDouble メソッド)
3	流れの制御	break 文, continue 文を使用した例題
4	配列	二次元配列 (宣言, 初期化, 処理, 要素数)

第7回 文字と文字列

講	項目	内容
1	文字	char 型, 文字コード, 入出力
2	文字列	String 型, 宣言, 初期化, 入出力, 長さ
2	データ	基本型(int 型, double 型, char 型, 等), 参照型(String 型)
3	文字列	String 型のメソッド(length, charAt, equals, indexOf, toUpperCase, substring, toCharArray)
4	文字列	文字列の配列
4	データの入力	コマンドライン引数, 文字列から数値への変換

第8回 プログラミング演習(2)

講	項目	内容
1	文字列の比較	String 型の equals メソッド
1	データの入力	Scanner クラス (hasNext メソッド, hasNextLine メソッド)

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:
 - 文字列に関する補足
 - Scanner クラスに関する補足
- ◆ 本科目の内容を振り返り, まとめを行った.

第8回 プログラミング演習(2)

第1講 補足事項とまとめ

終わり