

Javaプログラミング実習

09. 反復処理 1

株式会社ジードライブ

はじめに

- プログラムを記述していると、同じような記述を何度も繰り返さなければならない場面に遭遇する
 - 連続した数を入力する
 - 配列内の値を連続で処理する

今回学ぶこと

- for文を使用した反復処理の記述方法

for文

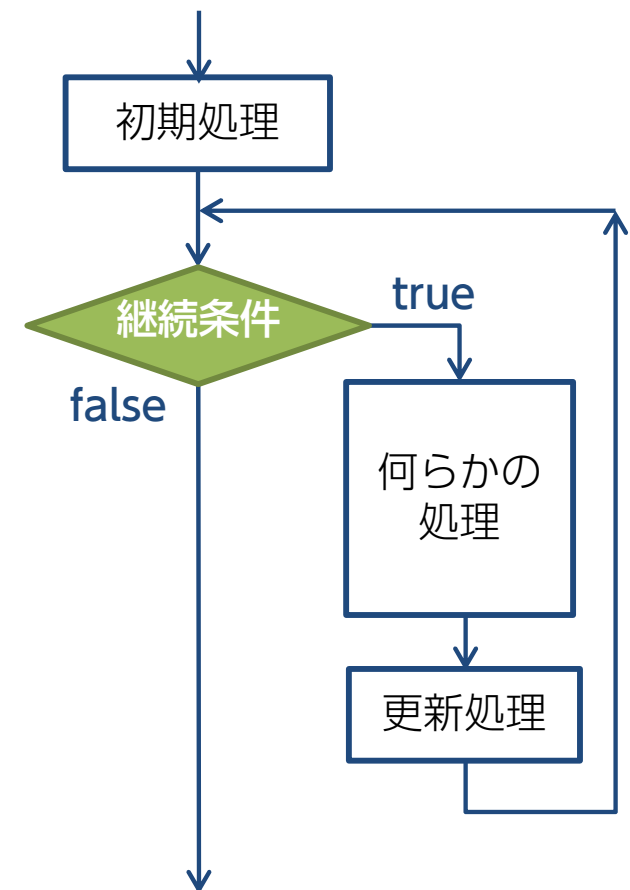
- 反復構造はfor文で作ることができる

書式

```
for (初期処理; 継続条件式; 更新処理) {  
    // 何らかの処理  
}
```

- ①最初に初期処理を実行する
- ②継続条件式が `true` なら `{ }` 内に進み、
`false` なら反復処理を終える
- ③ `{ }` 内の処理を実行する
- ④更新処理を実行する
- ⑤②に戻る

```
for (①初期処理; ②継続条件式; ④更新処理) {  
    // ③何らかの処理  
}
```



for文の使用例

- 1から10まで表示する

```
for (int i = 1; i <= 10; i++) {  
    System.out.print(i + " ");  
}
```

1 2 3 4 5 6 7 8 9 10

- 1～20の範囲で3の倍数を表示する

```
for (int i = 1; i <= 20; i++) {  
    if(i % 3 == 0){  
        System.out.print(i + " ");  
    }  
}
```

3 6 9 12 15 18

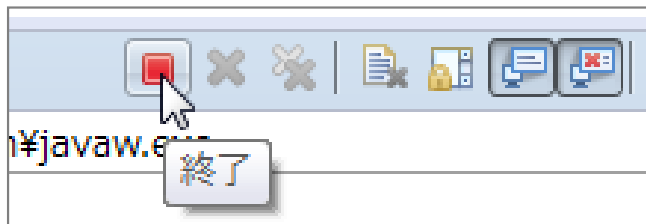
無限ループ

- 永久に継続条件式がtrueになってしまう反復処理を無限ループと呼ぶ

例：変数*i* が10より大きくなならないので、ループを抜けられない

```
for (int i = 1; i <= 10; i--) {  
    System.out.print(i + " ");  
}
```

- 無限ループに入ってしまったプログラムはコンソールビューから強制終了する



練習

- 練習09-1
- 練習09-2
- 練習09-3
- 練習09-4
- 練習09-5

配列の処理

- 配列.**length** で配列内の要素数を取得し、for文で使用する

例：fruits.length で、要素数 3 が取得される

```
String[] fruits = {"りんご", "バナナ", "ぶどう"};  
for (int i = 0; i < fruits.length; i++) {  
    System.out.println(fruits[i]);  
}
```

りんご
バナナ
ぶどう

配列の処理：拡張for文

- 配列内の全ての要素を順番に処理していくような場合、拡張for文が利用できる

書式：1周ごとに、配列内の要素が順番に取り出され、変数に代入される

```
for (型 変数名 : 配列名) {  
    // 何らかの処理  
}
```

例

```
String[] fruits = {"りんご", "バナナ", "ぶどう"};  
for (String item : fruits) {  
    System.out.print(item + " ");  
}
```

りんご バナナ ぶどう

練習

- 練習09-6