

Javaプログラミング実習

11. 反復処理 2

株式会社ジードライブ

今回学ぶこと

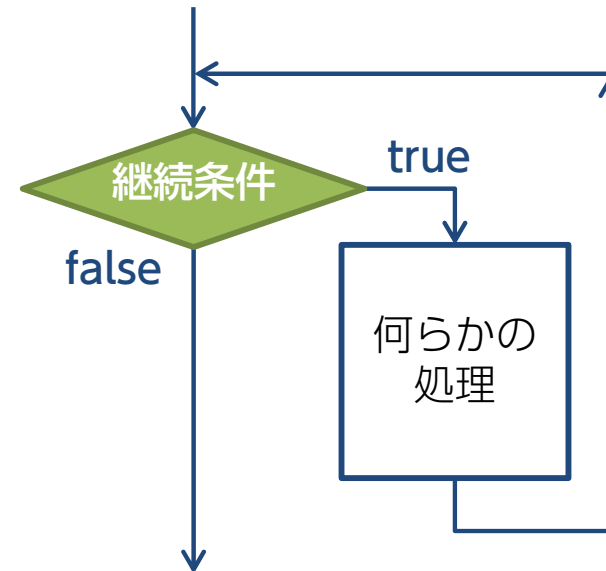
- while/do-while文を使った反復構造の記述方法
- break/continueの意味と使い方

while文

- while文による反復構造では、継続条件式を最初に評価する
 - { }内の処理が一度も実行されずに、反復処理が終了することもある

書式

```
while (継続条件式) {  
    // 何らかの処理  
}
```



while文の使用例

- 1から10まで表示する

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

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

3 6 9 12 15 18

do-while文

- while文に似た構文として、do-while文がある
 - whileと異なり、{ } 内の処理が必ず一度は実行される

書式

```
do {  
    // 何らかの処理  
} while (継続条件式);
```

例

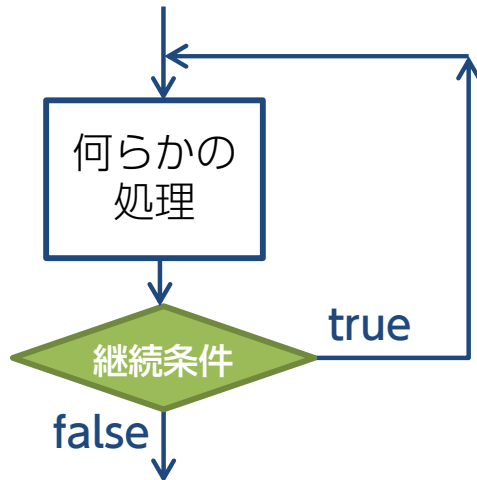
```
int a = 40;  
do {  
    System.out.print(a + " ");  
    a++;  
} while (a <= 10);
```

40

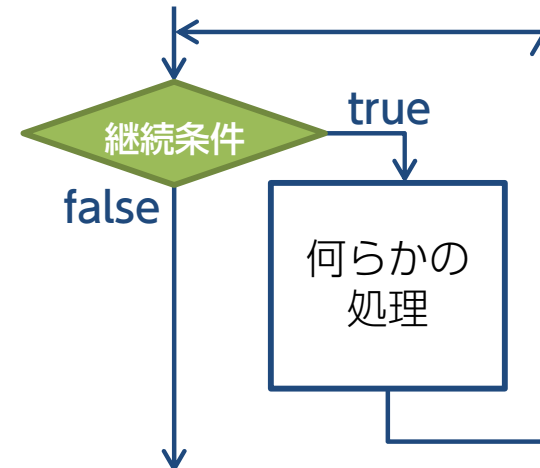
do-whileとwhile

- do-while文では、{ } 内の処理が継続条件より前にあるため、必ず一度は実行される

do-while文



while文



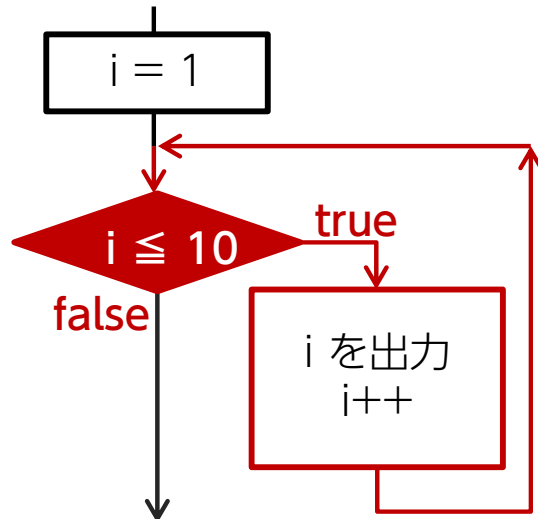
練習

- 練習 11-1

for文との比較

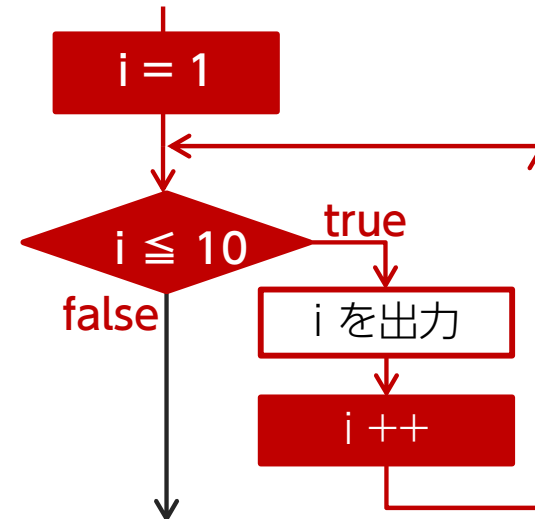
while文

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



for文

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



while文は構文内に初期処理(`i = 1`)や更新処理(`i++`)を含まない

for文との比較

- for文は繰り返す回数が決まっている場合に、while文は回数が決まっていない場合に利用されることが多い

例

```
Scanner scanner = new Scanner(System.in);
boolean flag = true;
while (flag) {
    System.out.println("何かしらの処理");

    System.out.print("再実行しますか? y/n >> ");
    if (scanner.nextLine().equals("n")) {
        System.out.println("終了します");
        flag = false;
    }
}
scanner.close();
```

練習

- 練習11-2

break

- **break**は、ループやswitchの処理をその時点で中断し、ブロックを抜ける

例：ぶどうが何番目か数える（ぶどうが見つかったら処理を抜ける）

```
String[] fruits = {"りんご", "バナナ", "ぶどう", "みかん"};
for (int i = 0; i < fruits.length; i++) {
    if (fruits[i].equals("ぶどう")) {
        int order = i + 1;
        System.out.println("ぶどうは" + order + "番目");
        break;
    }
}
```

ぶどうは3番目

breakとネスト

- 反復構造がネスト(入れ子)になっている場合、breakを使うと、ネストを1段階だけ抜ける

```
while ( ... ) { // ①
    for ( ... ) { // ②
        if ( ... ) break; // ②は抜けるが①は抜けない
    }
}
```

continue

- **continue**は、{ }内の処理をそこで中断し、次の繰り返し分に処理を進める

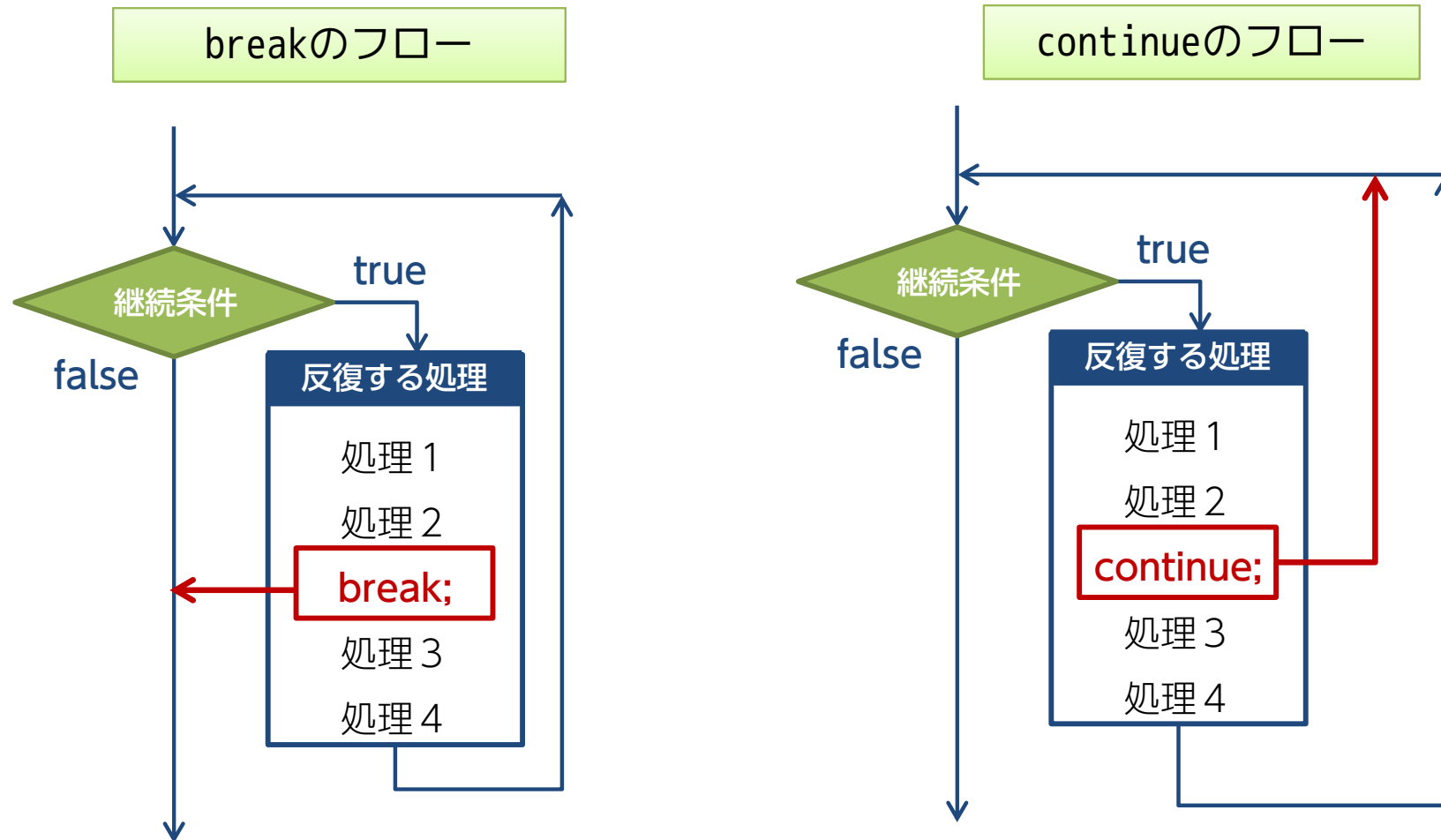
例：リスト内の300円以下の商品数を数える（300より大きい場合はカウントの処理を飛ばす）

```
int totalPrice = 0;
int count = 0;
int[] priceList = {320, 480, 270, 380, 180, 300};
for (int price : priceList) {
    totalPrice += price;
    if (price > 300) continue;
    count++;
}
System.out.println("合計金額は" + totalPrice + "円");
System.out.println("300円以下の商品は" + count + "個");
```

合計金額は1930円
300円以下の商品は3個

breakとcontinue

- **break**はループを抜けるが、**continue**はループを抜けない



練習

- 練習11-3