14

Java の覚え書き

繰り返し処理 (for 文)

```
for (int i=0; i< 繰り返す回数 ; i++) {
    繰り返し行いたい処理
}
```

「繰り返す回数」に任意の数字を入れると、その回数だけ処理が行われます。
また変数「i」を参照すると現在の処理回数 (0〜) を取得する事ができます。

（使用例）

```
for (int i=0; i<3; i++) {
      System.out.println((i+1)+ “ 回目の処理です。”) ;
}
```

実行結果

```
１回目の処理です。
２回目の処理です。
３回目の処理です。
```

```
for ( 初期化 ; 条件式 ; 次の一歩 ) {
      繰り返し行いたい処理
}
```

本来の for 構文の構造は左記の意味になっていますが、
for 文では「何回処理を繰り返すか」という考えが重要であるので、
最初は上記の構文を１つの形としてしっかり覚えましょう。

15

Java の覚え書き

変数の有効範囲（スコープ）

Java における変数の有効範囲（スコープ）は、原則として次のルールに従います。

中括弧 {} で括られたブロック中で宣言された変数は、そのブロック内だけで有効。

if 文や switch 文、try 文など、中括弧 {} で括られたブロック内で宣言された変数は、そのブロック内が有効範囲となります。

よって有効範囲を広げたい場合は、そのブロックの外（一つ上の中括弧内）で変数を宣言して使用する必要があります。

```
boolean b = true;
if (b == true) {
    String s = “Hello”;
}
System.out.println(s);
```

変数 S の有効範囲が外れているので
コンパイルエラーになる。

```
boolean b = true;
String s = “”;
if (b == true) {
    s = “Hello”;
}
System.out.println(s);
```

変数 S は有効範囲内なので
コンパイルできる。

16

Java の覚え書き

繰り返し処理（while 文）

while 文は for 文と同じループ処理を行いますが、
for 文との違いは「何回繰り返すか」ではなく、「条件が続く限り繰り返す」です。

ここで条件式の判定が行われる

```
while（条件式）{
    繰り返し行いたい処理
}
```

```
do {
    繰り返し行いたい処理
} while(条件式)
```

ここで条件式の判定が行われる

また while 文と似た動作をする構文に do-while 文というものがあります。
while 文との違いは条件判定の位置が異なるだけで基本動作に違いはありません。

while 文の場合は、条件式の判定が最初に行われるので、「0 回（実行されない）」のケースも含みますが、
do-while 文は、処理の最後に条件式の判定が行われるので、「最低でも 1 回」は実行される事になります。

17

Java の覚え書き

繰り返し処理の制御

break 文 と continue 文

for 文や while 文のループ処理を行っている際、
処理途中でループから抜けたい時やスキップしたい時は、「break 文」「continue 文」を利用します。

```
1:  while(...) {
2:      if ( 条件式 ) {
3:          break;
4:      }
5:      ...
6:  }
```

脱出

while 文を中断して、
ループから抜ける。

```
1:  while(...) {
2:      if ( 条件式 ) {
3:          continue;
4:      }
5:      ...
6:  }
```

Skip

while 文の残りを中断して、
再び繰り返し処理を続ける。

18

Java の覚え書き

文字列を比較する方法

文字列 A.equals(文字列 B);

これまで値（数値）を比較するには比較演算子の「==」を使用してきましたが、
文字列を比較する時は「equals メソッド」を使用します。
比較結果が等しい時は true、等しくない時は false を返します。

（使用例）

```
BufferedReader reader = new BufferedReader(...);
String str = reader.readLine();
if (str.equals(”テスト”)) {
      System.out.println(”変数 str の値と文字列「テスト」は等しい”);
} else {
      System.out.println(”変数 str の値と文字列「テスト」は等しくない”);
}
```