**第5章　繰り返し**

**この章では JAVAの繰り返し処理について説明します。**

**繰り返し**

繰り返し処理は、指定した回数や条件式に当てはまる間、ブロック内の処理を繰り返し行います。繰り返しには、**while文**や**for文**といった構文が使われます。以下でそれぞれについて説明します。

**1.while文**

while文は、何かの条件が成立している間、処理を繰り返すという構文です。

* while (条件式) {
* (繰り返し実行したい処理)
* }

while文では、最初に条件式の確認を行います。結果がtureなら「繰り返し実行したい処理」を実行し、処理が終わると、また、条件式の確認に戻ってきます。

このようにして繰り返しが行われ、条件がfalseになった時点でwhile文を抜けて、次のステップへ進みます。

**例**

[JSample5\_1.java]

* package JSample;
* public class JSample5\_1 {
* public static void main(String[] args) {
* int i = 1;
* while (i < 5) {
* System.out.println(i);
* i++;
* }
* }
* }

**実行結果**  
1  
2  
3  
4

この例では、変数（i）が5未満である限り(条件式)、7行目と8行目の処理が繰り返し実行されます。

まず、5行目で i には1が代入されています。

次に6行目の条件式「i が5未満の時」という条件を確認し、当てはまるためブロック内の 「i を出力する処理(7行目)」、と「i の値を1増やす」という処理を行います。

その後6行目の条件の確認に戻り、条件に当てはまる場合はブロック内の処理を行うというループを繰り返します。

**2.do/while文**

while文のもう一つの構文として**do/while文**があります。

whilw分は、処理を実行する前に条件の確認を行いましたが、

**do/while文**は、処理を実行した後に条件の確認を行います。

まず、「処理」を実行して、それから条件を確認するため、少なくとも1回は「処理」を実行する点がwhile文と異なります。

* do {
* (繰り返し実行したい処理)
* }
* while (条件式);

**例1**

[JSample5\_2.java]

* package JSample;
* public class JSample5\_2 {
* public static void main(String[] args) {
* int i = 1;
* do {
* System.out.println(i);
* i++;
* }while (i < 5);
* }
* }

**実行結果**

1  
2  
3  
4

まず、4行目で i には1が代入されています。

次に6～8行目のブロック内の 「i を出力する処理(7行目)」、と「i の値を1増やす」という処理を行います。

その後9行目の条件の確認し、条件に当てはまる場合はブロック内の処理を行うというループを繰り返します。

**例2**

[JSample5\_3.java]

* package JSample;
* public class JSample5\_3 {
* public static void main(String[] args) {
* int i = 10;
* do {
* System.out.println(i);
* i++;
* }while (i < 5);
* }
* }

**実行結果**

10

**3.for文**

for文は、「回数」を指定することで、「繰り返したい処理」を指定した「回数」だけ実行する構文です。

for文は次章で学習する配列などでよく用いられるため

* for ([初期化式]; [条件式]; [加算式]) {
* (繰り返し実行したい処理)
* }

初期化式：変数の宣言と初期化

条件式：ループ処理を続ける条件

加算式：ループ処理が1回終わるごとに実行される内容

**例**

[JSample5\_4.java]

* package JSample;
* public class JSample5\_3 {
* public static void main(String[] args) {
* for (int i = 1; i <= 5; i = i + 1) {
* System.out.println(i);
* }
* }
* }

**実行結果**

1

2

3

4

5

この例では5行目が繰り返し処理を行う条件を示しています。

まず、初期化式「int i = 1;」 は、変数 i を宣言し、i に1を代入し初期値を設定しています。  
条件式「 i <= 5」は、変数 i が5になるまでループを続ける、という条件を設定しています。

加算式「 i = i + 1」は、5行目から7行目のブロックの処理が1回終わるごとに i の値を1増やすという処理が行われます。

**4.for-each文**

for文の少し違った使い方として拡張forループ(for-eachループ)という使い方ができるようになりました。

これはfor文のように、繰り返しを行う条件式がありません。対象となる配列やコレクションと呼ばれる要素の数だけ繰り返しを行います。

* for (データ型 変数名: コレクション) {
* (繰り返し実行したい処理)
* }

**例**

[JSample5\_5.java]

* package JSample;
* public class JSample5\_4 {
* public static void main(String[] args) {
* String[] cars = {"Toyota", "Honda", "Subaru", "Nissan"};
* for (String i : cars) {
* System.out.println(i);
* }
* }
* }

**実行結果**

Toyota

Honda

Subaru

Nissan

5行目は、配列で [String型] の [car] という配列で、要素は("Toyota", "Honda", "Subaru", "Nissan")の4つです。詳しくいは次章で学習します。

6行目のfor-each文で、配列 [car]から、String型の要素を順番に取り出し、変数 i に代入します。

そして7行目の出力処理を配列 [car]の要素の数だけ行います。

**練習**

**問題1**[JEx5\_1.java]

1から30までで3の倍数のみを出力するプログラムを作成せよ。 ただし、while文を使用すること。

**問題2**[JEx5\_2.java]

forループで、実行結果が 1\*2\*3...\*nになるような文を作ってください。

* 例（入力変数n=10の時）：
* 1!=1
* 2!=1\*2
* 3!=1\*2\*3
* 4!=1\*2\*3\*4
* 5!=1\*2\*3\*4\*5
* 6!=1\*2\*3\*4\*5\*6
* 7!=1\*2\*3\*4\*5\*6\*7
* 8!=1\*2\*3\*4\*5\*6\*7\*8
* 9!=1\*2\*3\*4\*5\*6\*7\*8\*9
* 10!=1\*2\*3\*4\*5\*6\*7\*8\*9\*10
* 10!=3628800