

4.1 繰り返し文

Java の構文に繰り返し文と呼ばれるものがあります。Java の繰り返し文には、for 文、while 文、do-while 文の 3 つがあります。

4.2 for 文

for 文の概略は以下のように記されます。

`for(A;B;C){D}`

- ①A の処理がおこなわれます。
- ②B が真の場合に D の処理がおこなわれます。B が偽の場合には for 文は終了します。
- ③C の処理がおこなわれます。
- ④ (=②) B が真の場合に D の処理がおこなわれます。B が偽の場合には for 文は終了します。
- ⑤ (=③) C の処理がおこなわれます。

以下②③の繰り返し

繰り返し文の注意として、C の処理により有限回の処理の後に B の条件が偽となることが挙げられます。そうしないとプログラムがいつまでも終了しない無限ループに陥ってしまいます。もし eclipse 上で無限ループするプログラムを実行してしまった場合は Ctrl+C でプログラムを強制終了できます。

Sample014

```
public class Sample014{
public static void main(String[] args) {
for(int i=1;i<=5;i++){
System.out.println(“繰り返しています。”);
}
System.out.println(“繰り返しが終わりました。”);
}
}
```

【line3】

for 文では繰り返す回数をカウントするために変数を用います。Sample014 では変数 *i* を用いています。for に続く () のはじめに int 型の変数 *i* の宣言と 1 の代入を同時におこないます (初期化)。続いて *i* ≤ 5 という条件判断に移ります。今 *i* == 1 なので {} 内の処理がおこなわれます。そして *i* ++ の処理がおこなわれ *i* == 2 となります。

これが for 文のはじめの 1 回の処理となります。

2 回目以降は、() 内のはじめの初期化はおこないません。Sample014 では処理を繰り返すたびに *i* がインクリメントされていき、*i* == 6 となった場合に *i* ≤ 5 の条件が偽となるので for 文のブロックから抜けることとなります。

Sample015

```
public class Sample015{
public static void main(String[] args) {
int i,j;
for(i=1,j=0;i<1000;i*=2,j++){
System.out.println("2 の"+j+"乗は"+i+"です。");
}
}
}
```

【line3】

Sample14 では for 文の中で変数 *i* を宣言しましたが、for 文内で宣言した変数は for 文内では使用できません。また for 文の外で宣言した変数と同じ名前を持つ変数の宣言もできません。よって一般的には for 文内で用いる変数も、for 文の外で宣言します。

【line4】

A で複数の変数の初期化をおこなえますし、C の処理も複数記述することができます。

【line5】

Sample015 では各繰り返しの時点での変数 *i* と *j* の値を表示させています。実際に *i* と *j* が変化している様子が確認できるはずです。

Sample016

```
public class Sample016{
public static void main(String[] args) {
int i,j;
for(i=1;i<9;i++){
for(j=1;j<9;j++){
```

```
System.out.print(i*j+"¥t");
}
System.out.print("¥n");
}
}
}
```

繰り返し文や前回勉強した条件判断文では複数の文を埋めこんでネストすることができます。Sample016のようにfor文の中にfor文を使うといった複雑な記述ができます。

【line4】

iが1から9まで繰り返しをおこないます。

【line5】

for文の中にfor文がネストされています。まずiが1のもと、jが1から9までインクリメントされていきます。次にiが2へインクリメントされ、再度jが1から9までインクリメントされていきます。Sample016では最終的に $9 \times 9 = 81$ 回の繰り返しが行われることになります。

【line4】

Sample016はiとjの積を表示させています。またエスケープシーケンス¥tを用いることで水平タブを挿入し、表示に統一感を持たせています。

【line6】

エスケープシーケンス¥nを用いることで、jが1から9まで変化するごとに改行を挿入しています。

4.3 while 文

while文の概略は以下のように記されます。

```
while(B){D}
```

①Bが真の場合にDの処理がおこなわれます。Bが偽の場合にはwhile文は終了します。

② (=①) Bが真の場合にDの処理がおこなわれます。Bが偽の場合にはwhile文は終了します。

以下①の繰り返し

Sample017

```
public class Sample017{
public static void main(String[] args) {
```

```
int =1;
while(i<=-5){
System.out.println(i+"番目の繰り返しです。");
i++;
}
System.out.println("繰り返しが終わりました。");
}
}
```

【line3】

while 文では for 文とは異なり構文の中に変数を初期化する部分が含まれていません。よって変数を構文の外で宣言し値を代入する必要があります。

【line4】

while 文では条件判断文のみを構文内に記述します。この条件が真の場合のみブロック内の処理がおこなわれます。

【line6】

while 文では for 文とは異なり構文の中に 1 回の処理が終わったときの処理を記述できません。よってブロック内で変数をインクリメントするなど有限回で繰り返しが終了するよう変数等を変化させる必要があります。

ここで、Smple017 では i の初期値は 1 で、while 文の条件は $i \leq 5$ でした。では仮に i の初期値が 6 であつたり、条件が $i < 0$ だったりした場合はどうなるでしょうか。この場合は while 文は実行されません。

条件によらず最低でも 1 回は処理をおこないたい場合は do-while 文を用います。

4.4 do-while 文

do-while 文の概略は以下のように記されます。

do{D}while(B)

①D の処理がおこなわれます

②B が真の場合に D の処理がおこなわれます。B が偽の場合には while 文は終了します。

③ (=②) B が真の場合に D の処理がおこなわれます。B が偽の場合には while 文は終了します。

以下②の繰り返し

Sample018

```
public class Sample018{
public static void main(String[] args) {
int i=1;
do{
System.out.println(i+"番目の繰り返しです。");
i++;
} while(i<0)
System.out.println("繰り返しが終わりました。");
}
}
```

【line4】

do-while 文では必ず 1 回はブロック内の処理がおこなわれます。

【line7】

do-while 文では処理が終わった段階で、再度繰り返しをおこなうかの条件判断がおこなわれます。**Sample018** では 1 度ブロック内の処理がおこなわれた時点で $i==2$ となっており、条件は偽となるので繰り返しは終了します。

4.5 演習

4.5.1

100 より小さい素数を表示するプログラムを記述しなさい。

4.5.2

キーボードから整数の入力を複数回受け付け、それらの合計が 100 以上になったら終了するプログラムを記述しなさい。

4.5.3

任意の整数入力 n に対し、コラッツ予想を確認しなさい。コラッツ予想とは

- ・ n が偶数のときは 2 で割る
- ・ n が奇数のときは 3 倍し、1 を足す

以上の操作を繰り返せば必ず 1 に到達するというものである。

