2018 年 5 月 17 日（木）実施

反復構造のプログラム

1)　0 回以上の繰り返しのプログラム

for 文

Java 言語で 0 回以上の繰り返しのプログラムを実現するための文として，for 文と while 文とが用意されている。for 文の構文は次の様になる。

for （ 初期化処理; 継続条件式; 更新処理 ） 文

先ず，初期化処理を実行し，続いて継続条件式を評価する。ここで，継続条件式が true であれば，文を実行する。次に更新処理を実行した上で，再度，継続条件式を評価する，という繰り返しを行う。継続条件式が true でなければ，文を実行せず，for 文から抜け出す。最初から継続条件式が true でなければ，全く文を実行しないので，0 回以上の繰り返しと呼ばれる。

for 文は通常，**特定の処理を決まった回数繰り返す目的**で用いられる。

while 文

while 文の構文は次の様になる。

while （ 継続条件式 ） 文

先ず，継続条件式を評価し，true であれば文を実行し，再度，継続条件式を評価する，という繰り返しを行う。継続条件式が true でなければ，文を実行せず，while 文から抜け出す。最初から継続条件式が true でなければ，全く文を実行しないので，0 回以上の繰り返しと呼ばれる。

while 文は通常，**処理を特定の状態になるまで繰り返す目的**で用いられる。

2)　1 回以上の繰り返しのプログラム

do 文

Java 言語で 1 回以上の繰り返しのプログラムを実現するための文として，do 文が用意されている。do 文の構文は次の様になる。

do 文 while （ 継続条件式 ）;

先ず文を実行し，続いて継続条件式を評価し，true であれば再度文を実行し，継続条件式を評価する，という繰り返しを行う。継続条件式が true でなくなれば，文を実行せず，do 文から抜け出す。最初から継続条件式が true でなくとも，先ず文を実行するので，1 回以上の繰り返しと呼ばれる。

do 文は通常，**ある処理を実施し，その処理を特定の状態になるまで繰り返す目的**で用いられる。

例題 1　（for 文を用いた特定の処理の決まった回数の繰り返し）

次のプログラムは，標準入力装置から入力された実数データの入力時点での合計を求めて表示

し，最後に合計をデータ数で割って平均値を求め，表示させるものである。これを入力して，実行せよ。なお，データは 5 個分入力する必要がある。ここで，クラス名は Sample5_1，ソースファイル名は Sample5_1.java とする。

```
import java.util.Scanner;


public class Sample5_1 {

    public static void main(String[] args) {
        // TODO 自動生成されたメソッド・スタブ
        final int NUM=5;
        double data, sum=0.0, ave;
        Scanner sc = new Scanner(System.in);

        for (int i=1; i<=NUM; i++) {
            System.out.print(i + "番目の実数データを入力してください： ");
            data = sc.nextDouble();

            sum = sum + data;

            System.out.println(" ここまでのデータの合計は " + sum);
        }

        ave = sum / NUM;

        System.out.println(" データの平均値： " + ave);
    }

}
```

【解説】

1. final 修飾子を付けて宣言した変数は値を変更出来ない。
2. 変数 sum を宣言する際に 0.0 と初期化しているが，これはデータを足しこんで合計を求めるための工夫である。sum = sum + data; は右辺で計算した結果を左辺の sum に代入し直す。
3. i++は変数 i の値を１だけ増やす。++は**増分演算子**と呼ばれる。なお，i は繰り返しの制御のために用いられることから，**ループ変数**と呼ばれる。
4. print は括弧内に与えた値を標準出力装置に出力するメソッドである。(println では表示後に改行されるが，print では改行されない。)
5. nextDouble()は標準入力装置から入力される実数を読み込む際に用いるメソッドである。

例題２　（while 文を用いた特定の状態になるまでの処理の繰り返し）

次のプログラムは，標準入力装置から入力された氏名に対応して，挨拶を表示させるものである。これを入力して，実行せよ。なお，繰り返しを終了するには quit を入力する。ここで，クラス名は Sample5_2，ソースファイル名は Sample5_2.java とする。

```
import java.util.Scanner;


public class Sample5_2 {

    public static void main(String[] args) {
        // TODO 自動生成されたメソッド・スタブ
        String str1="quit";
        Scanner sc = new Scanner(System.in);
        System.out.print("氏名を入力して下さい（終了は quit と入力）: ");
        String str2 = sc.next();

        while (!str2.equals(str1)) {
            System.out.println(" 今日は " + str2 + " さん。");
            System.out.println("");
            System.out.print("続けて氏名を入力して下さい（終了は quit と入力）: ");
            str2 = sc.next();
        }

        System.out.println("プログラムを終了します。");
    }

}
```

【解説】str2.equals(str1)は str2 に格納された文字列と str1 に格納された文字列とを比較し，等しければ true，等しくなければ false を返す。!str2.equals(str1)とすると，文字列が等しければ false，等しくなければ true となるので，標準入力装置から入力された文字列が quit でない限り，{}に挟まれた範囲の処理を繰り返す。

例題３（do 文を用いた特定の状態になるまでの処理の繰り返し）

次のプログラムは，1/2 のべき乗を表示させていき，1/2 の何乗が 10^{-9} より小さくなるかを求めて結果を表示させるものである。これを入力して，実行せよ。ここで，クラス名は Sample5_3，ソースファイル名は Sample5_3.java とする。

```
public class Sample5_3 {

    public static void main(String[] args) {
        // TODO 自動生成されたメソッド・スタブ
        final double EPSILON=1.0e-9;
        int n=0;
        double x=1.0;

        do {
            x = x / 2;

            System.out.printf("1/2 の%d 乗: %.15e%n", ++n, x);
        } while (x>=EPSILON);

        System.out.printf("⇒ 1/2 の%d 乗は%.1e より小さい。%n", n, EPSILON);
    }
```

```
}
```

【解説】

1. 1.0e-9 は指数形式の表記で，10^{-9} を表す。
2. printf は C 言語から移植された，書式を指定して括弧内に与えた値を標準出力装置に出力するメソッドである。二重引用符（″ ″）に続くカンマ（,）で区切られた式の値を，%から始まる**書式指示子**で書式を指定して，出現順に出力する。

```
System.out.printf("1/2 の%d 乗: %.15e%n", ++n, x);
```

ここで，%d は式の値を 10 進数に変換し，%.15e は精度（小数点以下の桁数）を 15 桁として指数形式の 10 進表現に変換する。なお，%e では精度は 6 桁に設定されている。また，%n はその位置で改行を行う指定である。

3. ++n は，先ず変数 n の値を 1 だけ増やしてから，その値が出力される。この場合，++は**前置増分演算子**と呼ばれる。（例題 1 の i++の場合の++は**後置増分演算子**と呼ばれる。）
4. 10^3 (k) に近い値として，$2^{10} = 1024$ がコンピュータ関係ではよく用いられる。また，10^6 (M)，10^9 (G) に近い値としてはそれぞれ，$2^{20} = 1024 \times 1024$，$2^{30} = 1024 \times 1024 \times 1024$ がコンピュータ関係ではよく用いられる。

演習

第４回の教材の例題１で「点数(0～100)を入力して下さい。」というガイドコメントを表示してから，キーボードで入力した点数に基づいて成績を表示するまでの処理を，for 文を用いて 5 回繰り返すプログラムを作成し，実行せよ。ここで，クラス名は Ex5，ソースプログラム名は Ex5.java とする。

```
for (      ;      ;    ) {
    繰り返したい処理
}
```

提出物：

1) 例題 1，例題 2，例題 3 及び演習のプログラムのコンソールへの出力結果をコピーして貼り付けたテキストファイル res5.txt をメールに添付する。
2) 演習のソースプログラムのファイル Ex5.java をメールに添付する。
3) 第 4 回の授業の復習の内容を埋めたファイル Review_4th.txt をメールに添付する。

＊ メールの件名は『プログラミング１　第５回課題』（鍵括弧は要らない）とする。