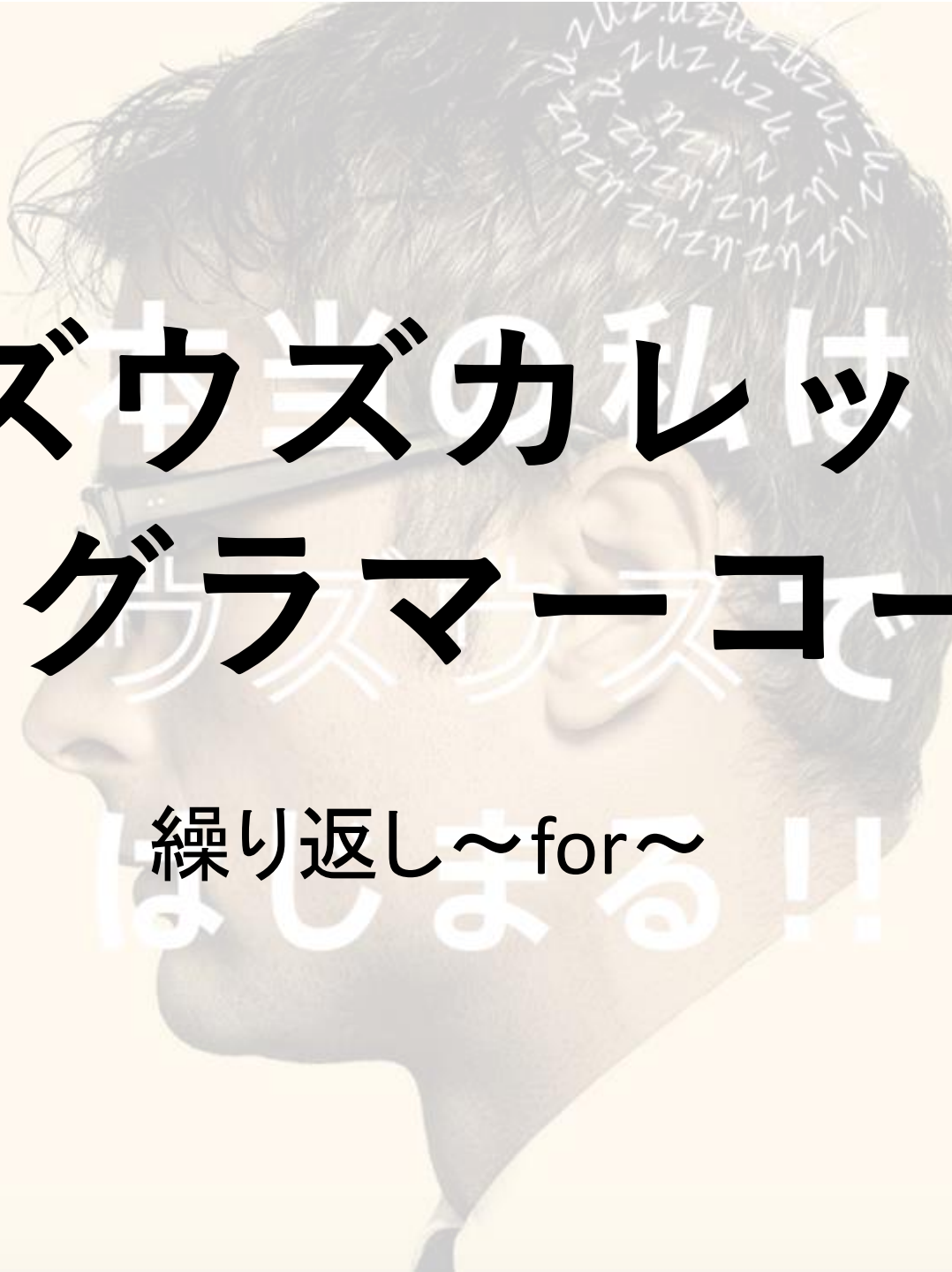


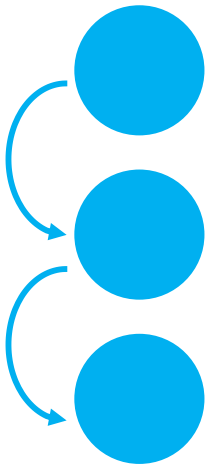
ウズウズカレッジ
プログラマーコース

繰り返し～for～

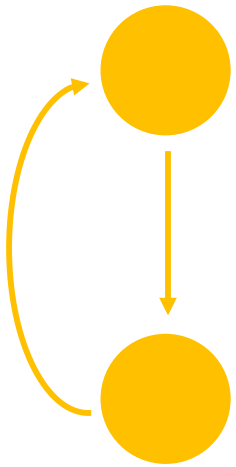
構造化プログラミング

最適に組み合わせることで
品質の高いプログラムを書くことができる

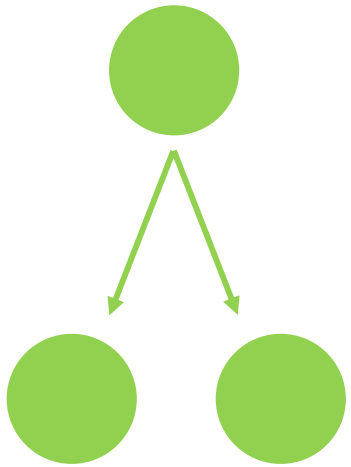
順次



繰り返し



条件分岐

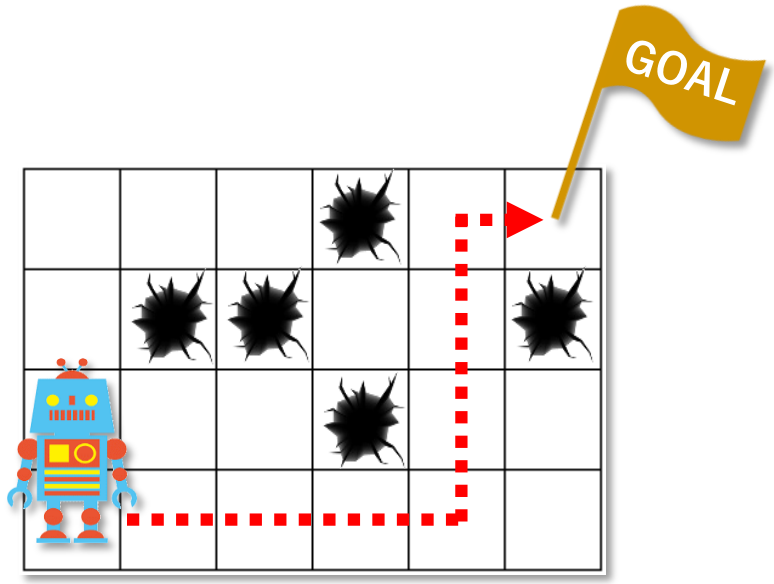


≪構造化プログラミング≫

□これまでプログラムはソースコード上の命令文を上から順番に処理していくものだと説明してきましたが、**制御構文**というものを使うと処理の流れを自在に組むことが可能になります。

□ソースコード上の命令文を上から順に実行していく処理方法のことを「**順次**」と言います。制御構文を扱えるようになると順次以外にも「**繰り返し**」「**条件分岐**」といったプログラムの流れを作ることができるようになります。

□「順次」「繰り返し」「条件分岐」の３つを最適に組み合わせることで誰が見ても理解のしやすい、シンプルでロジカルなアルゴリズムが書けると言われており、このような考え方でプログラムを組み立てる手法のことを**構造化プログラミング**と言います。



順次

「1マス前に進め」
「1マス前に進め」
「1マス前に進め」
「1マス前に進め」
「左に向きを変えよ」
「1マス前に進め」
「1マス前に進め」
「1マス前に進め」
「右に向きを変えよ」
「1マス前に進め」

繰り返し

「1マス前に進め」**を4回繰り返せ**
「左に向きを変えよ」
「1マス前に進め」**を3回繰り返せ**
「右に向きを変えよ」
「1マス前に進め」

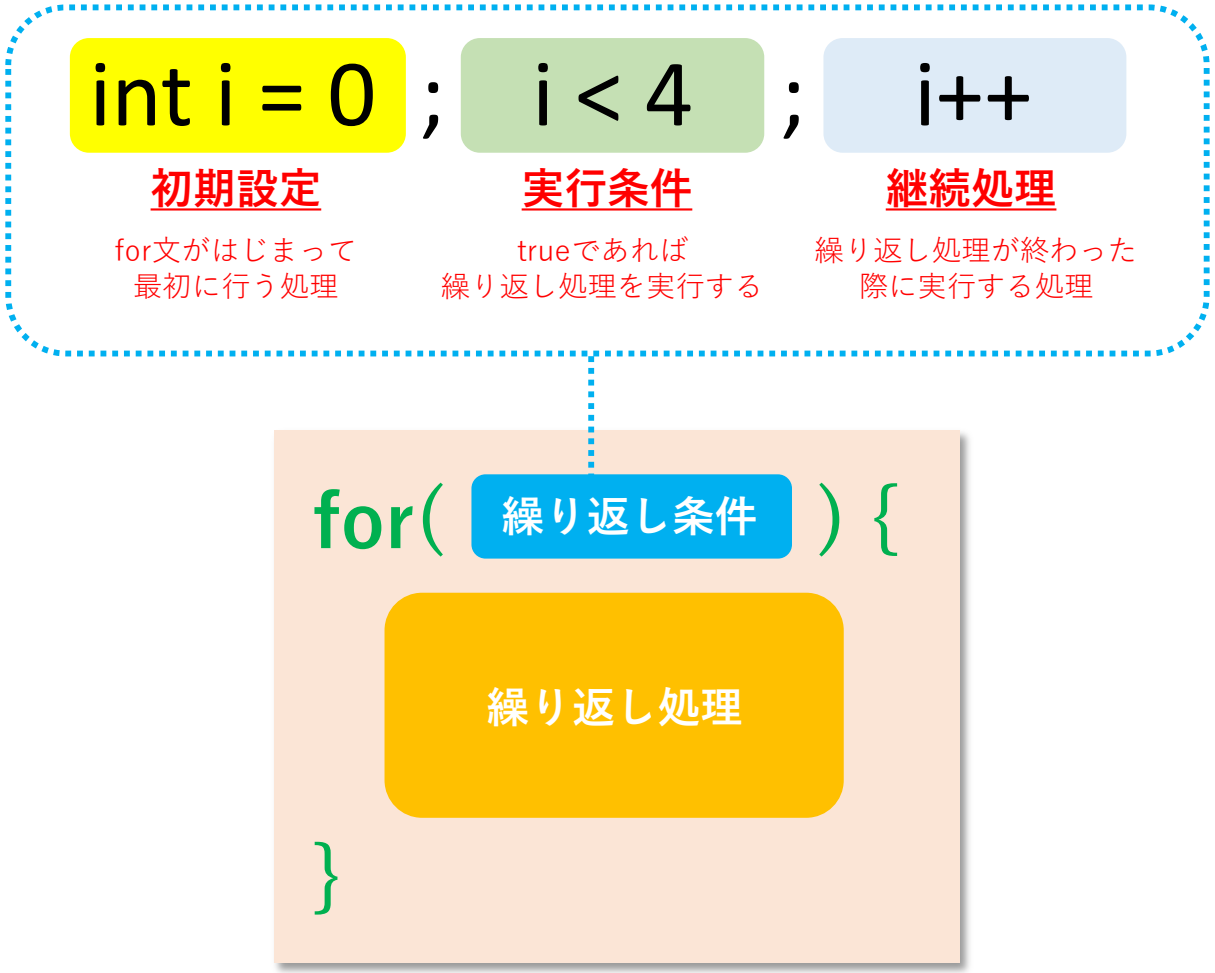
順次に比べて処理が
グッとコンパクトに！

≪繰り返し≫

□ 構造化プログラミングにおける「繰り返し」とは「**条件を満たす限り**同じ処理を反復して実行する」という意味を指します。
書こうと思えばどんなプログラムもすべて順次で書けるのですが、繰り返しを使うことで処理をグッとシンプルで効率的なものにすることが可能になります。

□繰り返しの制御構文として主要なものは以下の2つです。

- **for文**
- **while文**（＋ **do-while文**）



≪for文≫

□for文は繰り返し構文の1つです。
繰り返しは「**条件を満たす限り**同じ処理を反復して実行する」というものだと説明したとおり、for文の構造は「繰り返し条件」「繰り返し処理」の2つに大きく分けられます。
以下がfor文のざっくりとした構造になります。

for(繰り返し条件) { 繰り返し処理 }

□for文の ( ) 内に記述する繰り返し条件は、**『;』で3つのパートに分けられています。**

左のパートは**初期設定**と呼ばれており、for文の**開始時に1度だけ行う処理**を記述します。
基本的には繰り返すかの判断で使用する変数（この変数のことを**カウンタ変数**と言う）の宣言及び初期化が行われます。

真ん中のパートは**実行条件**と呼ばれており、**この部分がtrueである限り繰り返し処理が実行されます。**

右のパートは**継続処理**と呼ばれており、**繰り返し処理が終わった際に実行する処理**が記述されます。

～for文のしくみ～

▼Sample1_10_1.java（6～21行目）

```
int loopCount = 0 ; //何回目のループかを表す数値（初期値0）
↵
System.out.println("▼[開始]for文");↵
↵
for( int i = 0 ; i < 4 ; i++ ){↵
^    System.out.println("  ▼[開始]1回分の繰り返し処理");↵
^    ↵
^    //loopCountを1上げる↵
^    loopCount += 1 ;  ↵
^    ↵
^    //loopCountおよびカウンタ変数iの表示↵
^    System.out.println("    loopCount：" + loopCount);↵
^    System.out.println("    i（カウンタ変数）：" + i);↵
^    ↵
^    System.out.println("  ▲[終了]1回分の繰り返し処理");↵
}↵
↵
System.out.println("▲[終了]for文");↵
↵
```

▼実行後のコマンドプロンプト

```
C:¥WorkSpace>java Sample1_10_1
▼[開始]for文
  ▼[開始]1回分の繰り返し処理
    loopCount：1
    i（カウンタ変数）：0
  ▲[終了]1回分の繰り返し処理
  ▼[開始]1回分の繰り返し処理
    loopCount：2
    i（カウンタ変数）：1
  ▲[終了]1回分の繰り返し処理
  ▼[開始]1回分の繰り返し処理
    loopCount：3
    i（カウンタ変数）：2
  ▲[終了]1回分の繰り返し処理
  ▼[開始]1回分の繰り返し処理
    loopCount：4
    i（カウンタ変数）：3
  ▲[終了]1回分の繰り返し処理
▲[終了]for文
```

for(繰り返し条件) { 繰り返し処理 }

～for文

繰り返し条件

int i = 0 ; i < 4 ; i++

初期設定
for文がはじまって最初に行う処理

実行条件
trueであれば繰り返し処理を実行する

継続処理
繰り返し処理が終わった際に実行する処理

```
for( int i = 0 ; i < 4 ; i++ ){
    System.out.println("    ▼[開始]1回分の繰り返し処理");

    //loopCountを1上げる
    loopCount += 1 ;

    //loopCountおよびカウンタ変数iの表示
    System.out.println("      loopCount : " + loopCount);
    System.out.println("      i（カウンタ変数） : " + i);

    System.out.println("    ▲[終了]1回分の繰り返し処理");
}
```

繰り返し処理

```
▼[開始]for文
    i（カウンタ変数） : 0
  ▲[終了]1回分の繰り返し処理
  ▼[開始]1回分の繰り返し処理
    loopCount : 2
    i（カウンタ変数） : 1
  ▲[終了]1回分の繰り返し処理
  ▼[開始]1回分の繰り返し処理
    loopCount : 3
    i（カウンタ変数） : 2
  ▲[終了]1回分の繰り返し処理
  ▼[開始]1回分の繰り返し処理
    loopCount : 4
    i（カウンタ変数） : 3
  ▲[終了]1回分の繰り返し処理
▲[終了]for文
```

～for文のしくみ～

▼Sample1_10_1.java（6～21行目）

```
int loopCount = 0 ; //何回目のループかを表す数値（初期値0）
↵
System.out.println("▼[開始]for文");↵
↵
for( int i = 0 ; i < 4 ; i++ ){↵
^    System.out.println("  ▼[開始]1回分の繰り返し処理");↵
^    ↵
^    //loopCountを1上げる↵
^    loopCount += 1 ;  ↵
^    ↵
^    //loopCountおよびカウンタ変数iの表示↵
^    System.out.println("    loopCount : " + loopCount);↵
^    System.out.println("    i（カウンタ変数） : " + i);↵
^    ↵
^    System.out.println("  ▲[終了]1回分の繰り返し処理");↵
}↵
↵
System.out.println("▲[終了]for文");↵
```

▼実行後のコマンドプロンプト

```
C:¥WorkSpace>java Sample1_10_1
▼[開始]for文
  ▼[開始]1回分の繰り返し処理
    loopCount : 1
    i（カウンタ変数） : 0
  ▲[終了]1回分の繰り返し処理
  ▼[開始]1回分の繰り返し処理
    loopCount : 2
    i（カウンタ変数） : 1
  ▲[終了]1回分の繰り返し処理
  ▼[開始]1回分の繰り返し処理
    loopCount : 3
    i（カウンタ変数） : 2
  ▲[終了]1回分の繰り返し処理
  ▼[開始]1回分の繰り返し処理
    loopCount : 4
    i（カウンタ変数） : 3
  ▲[終了]1回分の繰り返し処理
▲[終了]for文
```

メモリ上

～for文

繰り返し条件

int i = 0 ; i < 4 ; i++

初期設定
for文がはじまって最初に行う処理

実行条件
trueであれば繰り返し処理を実行する

継続処理
繰り返し処理が終わった際に実行する処理

▼Sample1_10_1.java（6～21行目）

```
int loopCount = 0 ; //何回目のループかを表す数

System.out.println("▼[開始]for文");

for( int i = 0 ; i < 4 ; i++ ){
    System.out.println("    ▼[開始]1回分の繰り返し処理");

    //loopCountを1上げる
    loopCount += 1 ;

    //loopCountおよびカウンタ変数iの表示
    System.out.println("      loopCount : " + loopCount);
    System.out.println("      i（カウンタ変数） : " + i);

    System.out.println("    ▲[終了]1回分の繰り返し処理");
}

System.out.println("▲[終了]for文");
```

コンプト

```
a Sample1_10_1

)繰り返し処理

    i（カウンタ変数） : 0
  ▲[終了]1回分の繰り返し処理
  ▼[開始]1回分の繰り返し処理
    loopCount : 2
    i（カウンタ変数） : 1
  ▲[終了]1回分の繰り返し処理
  ▼[開始]1回分の繰り返し処理
    loopCount : 3
    i（カウンタ変数） : 2
  ▲[終了]1回分の繰り返し処理
  ▼[開始]1回分の繰り返し処理
    loopCount : 4
    i（カウンタ変数） : 3
  ▲[終了]1回分の繰り返し処理
▲[終了]for文
```

メモリ上

0	0
loopCount	i

～for文

繰り返し条件

int i = 0 ; i < 4 ; i++

初期設定：for文がはじまって最初に行う処理

実行条件：trueであれば繰り返し処理を実行する

継続処理：繰り返し処理が終わった際に実行する処理

▼Sample1_10_1.java（6～21行目）

```
int loopCount = 0 ; //何回目のループかを表す数
↵
System.out.println("▼[開始]for文");↵
↵
for( int i = 0 ; i < 4 ; i++ ){↵
    System.out.println("    ▼[開始]1回分の繰り返し処理");↵
    ↵
    //loopCountを1上げる↵
    loopCount += 1 ;  ↵
    ↵
    //loopCountおよびカウンタ変数iの表示
    System.out.println("    loopCount : " +
    System.out.println("    i（カウンタ変数） :
    ↵
    System.out.println("    ▲[終了]1回分の繰り返し処理");↵
}↵
↵
System.out.println("▲[終了]for文");↵
```

i < 4

true

繰り返し処理を実行

コンプト

```
a Sample1_10_1

)繰り返し処理

    i（カウンタ変数）：0
  ▲[終了]1回分の繰り返し処理
  ▼[開始]1回分の繰り返し処理
    loopCount：2
    i（カウンタ変数）：1
  ▲[終了]1回分の繰り返し処理
  ▼[開始]1回分の繰り返し処理
    loopCount：3
    i（カウンタ変数）：2
  ▲[終了]1回分の繰り返し処理
  ▼[開始]1回分の繰り返し処理
    loopCount：4
    i（カウンタ変数）：3
  ▲[終了]1回分の繰り返し処理
▲[終了]for文
```

メモリ上

0	0
loopCount	i

～for文のしくみ～

▼Sample1_10_1.java（6～21行目）

```
int loopCount = 0 ; //何回目のループかを表す数値（初期値0）

System.out.println("▼[開始]for文");

for( int i = 0 ; i < 4 ; i++ ){
    System.out.println("    ▼[開始]1回分の繰り返し処理");

    //loopCountを1上げる
    loopCount += 1 ;

    //loopCountおよびカウンタ変数iの表示
    System.out.println("      loopCount : " + loopCount);
    System.out.println("      i（カウンタ変数） : " + i);

    System.out.println("    ▲[終了]1回分の繰り返し処理");
}

System.out.println("▲[終了]for文");
```

1回目

繰り返し処理

▼実行後のコマンドプロンプト

```
C:¥WorkSpace>java Sample1_10_1
▼[開始]for文
    ▼[開始]1回分の繰り返し処理
      loopCount : 1
      i（カウンタ変数） : 0
    ▲[終了]1回分の繰り返し処理
    ▼[開始]1回分の繰り返し処理
      loopCount : 2
      i（カウンタ変数） : 1
    ▲[終了]1回分の繰り返し処理
    ▼[開始]1回分の繰り返し処理
      loopCount : 3
      i（カウンタ変数） : 2
    ▲[終了]1回分の繰り返し処理
    ▼[開始]1回分の繰り返し処理
      loopCount : 4
      i（カウンタ変数） : 3
    ▲[終了]1回分の繰り返し処理
▲[終了]for文
```

メモリ上

1	0
loopCount	i

～for文

繰り返し条件

int i = 0 ; i < 4 ; i++

初期設定：for文がはじまって最初に行う処理

実行条件：trueであれば繰り返し処理を実行する

継続処理：繰り返し処理が終わった際に実行する処理

▼Sample1_10_1.java（6～21行目）

```
int loopCount = 0 ; //何回目のループかを表す数

System.out.println("▼[開始]for文");

for( int i = 0 ; i < 4 ; i++ ){
    System.out.println("    ▼[開始]1回分の繰り返し処理");

    //loopCountを1上げる
    loopCount += 1 ;

    //loopCountおよびカウンタ変数iの表示
    System.out.println("    loopCount : " + loopCount);
    System.out.println("    i（カウンタ変数） : " + i);

    System.out.println("    ▲[終了]1回分の繰り返し処理");
}

System.out.println("▲[終了]for文");
```

1回目

コンプト

```
a Sample1_10_1

)繰り返し処理

    i（カウンタ変数） : 0
  ▲[終了]1回分の繰り返し処理
  ▼[開始]1回分の繰り返し処理
    loopCount : 2
    i（カウンタ変数） : 1
  ▲[終了]1回分の繰り返し処理
  ▼[開始]1回分の繰り返し処理
    loopCount : 3
    i（カウンタ変数） : 2
  ▲[終了]1回分の繰り返し処理
  ▼[開始]1回分の繰り返し処理
    loopCount : 4
    i（カウンタ変数） : 3
  ▲[終了]1回分の繰り返し処理
▲[終了]for文
```

メモリ上

loopCount	i
1	1

～for文

▼Sample1_10_1.java（6～21行目）

```
int loopCount = 0 ; //何回目のループかを表す数

System.out.println("▼[開始]for文");

for( int i = 0 ; i < 4 ; i++ ){
    System.out.println("    ▼[開始]1回分の繰り返し処理");

    //loopCountを1上げる
    loopCount += 1 ;

    //loopCountおよびカウンタ変数iの表示
    System.out.println("    loopCount : " +
    System.out.println("    i（カウンタ変数） :
    System.out.println("    ▲[終了]1回分の繰り返し処理");
}

System.out.println("▲[終了]for文");
```

1回目

繰り返し条件

int i = 0	;	i < 4	;	i++
初期設定		実行条件		継続処理
for文がはじまって最初に行う処理		trueであれば繰り返し処理を実行する		繰り返し処理が終わった際に実行する処理

i < 4

↓

true

↓

繰り返し処理を実行

コンプト

```
a Sample1_10_1

)繰り返し処理

    i（カウンタ変数） : 0
▲[終了]1回分の繰り返し処理
▼[開始]1回分の繰り返し処理
    loopCount : 2
    i（カウンタ変数） : 1
▲[終了]1回分の繰り返し処理
▼[開始]1回分の繰り返し処理
    loopCount : 3
    i（カウンタ変数） : 2
▲[終了]1回分の繰り返し処理
▼[開始]1回分の繰り返し処理
    loopCount : 4
    i（カウンタ変数） : 3
▲[終了]1回分の繰り返し処理
▲[終了]for文
```

メモリ上

1	1
loopCount	i

～for文のしくみ～

▼Sample1_10_1.java（6～21行目）

```
int loopCount = 0 ; //何回目のループかを表す数値（初期値0）

System.out.println("▼[開始]for文");

for( int i = 0 ; i < 4 ; i++ ){
    System.out.println("    ▼[開始]1回分の繰り返し処理");

    //loopCountを1上げる
    loopCount += 1 ;

    //loopCountおよびカウンタ変数iの表示
    System.out.println("      loopCount : " + loopCount);
    System.out.println("      i（カウンタ変数） : " + i);

    System.out.println("    ▲[終了]1回分の繰り返し処理");
}

System.out.println("▲[終了]for文");
```

2回目

繰り返し処理

▼実行後のコマンドプロンプト

```
C:¥WorkSpace>java Sample1_10_1
▼[開始]for文
    ▼[開始]1回分の繰り返し処理
      loopCount : 1
      i（カウンタ変数） : 0
    ▲[終了]1回分の繰り返し処理
    ▼[開始]1回分の繰り返し処理
      loopCount : 2
      i（カウンタ変数） : 1
    ▲[終了]1回分の繰り返し処理
    ▼[開始]1回分の繰り返し処理
      loopCount : 3
      i（カウンタ変数） : 2
    ▲[終了]1回分の繰り返し処理
    ▼[開始]1回分の繰り返し処理
      loopCount : 4
      i（カウンタ変数） : 3
    ▲[終了]1回分の繰り返し処理
▲[終了]for文
```

メモリ上

2	1
loopCount	i

～for文

繰り返し条件

int i = 0 ; i < 4 ; i++

初期設定
for文がはじまって最初に行う処理

実行条件
trueであれば繰り返し処理を実行する

継続処理
繰り返し処理が終わった際に実行する処理

▼Sample1_10_1.java（6～21行目）

```
int loopCount = 0 ; //何回目のループかを表す数

System.out.println("▼[開始]for文");

for( int i = 0 ; i < 4 ; i++ ){
    System.out.println("    ▼[開始]1回分の繰り返し処理");

    //loopCountを1上げる
    loopCount += 1 ;

    //loopCountおよびカウンタ変数iの表示
    System.out.println("    loopCount : " +
    System.out.println("    i（カウンタ変数） :
    System.out.println("    ▲[終了]1回分の繰り返し処理");
}

System.out.println("▲[終了]for文");
```

2回目

i < 4

true

繰り返し処理を実行

▼コンプト

```
a Sample1_10_1

)繰り返し処理

    i（カウンタ変数） : 0
  ▲[終了]1回分の繰り返し処理
  ▼[開始]1回分の繰り返し処理
    loopCount : 2
    i（カウンタ変数） : 1
  ▲[終了]1回分の繰り返し処理
  ▼[開始]1回分の繰り返し処理
    loopCount : 3
    i（カウンタ変数） : 2
  ▲[終了]1回分の繰り返し処理
  ▼[開始]1回分の繰り返し処理
    loopCount : 4
    i（カウンタ変数） : 3
  ▲[終了]1回分の繰り返し処理
▲[終了]for文
```

メモリ上

loopCount	i
2	2

～for文のしくみ～

▼Sample1_10_1.java（6～21行目）

```
int loopCount = 0 ; //何回目のループかを表す数値（初期値0）

System.out.println("▼[開始]for文");

for( int i = 0 ; i < 4 ; i++ ){
    System.out.println("    ▼[開始]1回分の繰り返し処理");

    //loopCountを1上げる
    loopCount += 1 ;

    //loopCountおよびカウンタ変数iの表示
    System.out.println("      loopCount : " + loopCount);
    System.out.println("      i（カウンタ変数） : " + i);

    System.out.println("    ▲[終了]1回分の繰り返し処理");
}

System.out.println("▲[終了]for文");
```

3回目

繰り返し処理

▼実行後のコマンドプロンプト

```
C:¥WorkSpace>java Sample1_10_1
▼[開始]for文
    ▼[開始]1回分の繰り返し処理
      loopCount : 1
      i（カウンタ変数） : 0
    ▲[終了]1回分の繰り返し処理
    ▼[開始]1回分の繰り返し処理
      loopCount : 2
      i（カウンタ変数） : 1
    ▲[終了]1回分の繰り返し処理
    ▼[開始]1回分の繰り返し処理
      loopCount : 3
      i（カウンタ変数） : 2
    ▲[終了]1回分の繰り返し処理
    ▼[開始]1回分の繰り返し処理
      loopCount : 4
      i（カウンタ変数） : 3
    ▲[終了]1回分の繰り返し処理
▲[終了]for文
```

メモリ上

3	2
loopCount	i

～for文

繰り返し条件

int i = 0 ; i < 4 ; i++

初期設定
for文がはじまって最初に行う処理

実行条件
trueであれば繰り返し処理を実行する

継続処理
繰り返し処理が終わった際に実行する処理

▼Sample1_10_1.java（6～21行目）

```
int loopCount = 0 ; //何回目のループかを表す数

System.out.println("▼[開始]for文");

for( int i = 0 ; i < 4 ; i++ ){
    System.out.println("    ▼[開始]1回分の繰り返し処理");

    //loopCountを1上げる
    loopCount += 1 ;

    //loopCountおよびカウンタ変数iの表示
    System.out.println("    loopCount : " +
    System.out.println("    i（カウンタ変数） :
    System.out.println("    ▲[終了]1回分の繰り返し処理");
}

System.out.println("▲[終了]for文");
```

3回目

i < 4

true

繰り返し処理を実行

▼コンプト

```
a Sample1_10_1

)繰り返し処理

    i（カウンタ変数） : 0
  ▲[終了]1回分の繰り返し処理
  ▼[開始]1回分の繰り返し処理
    loopCount : 2
    i（カウンタ変数） : 1
  ▲[終了]1回分の繰り返し処理
  ▼[開始]1回分の繰り返し処理
    loopCount : 3
    i（カウンタ変数） : 2
  ▲[終了]1回分の繰り返し処理
  ▼[開始]1回分の繰り返し処理
    loopCount : 4
    i（カウンタ変数） : 3
  ▲[終了]1回分の繰り返し処理
▲[終了]for文
```

メモリ上

3	3
loopCount	i

～for文のしくみ～

▼Sample1_10_1.java（6～21行目）

```
int loopCount = 0 ; //何回目のループかを表す数値（初期値0）

System.out.println("▼[開始]for文");

for( int i = 0 ; i < 4 ; i++ ){
    System.out.println("    ▼[開始]1回分の繰り返し処理");

    //loopCountを1上げる
    loopCount += 1 ;

    //loopCountおよびカウンタ変数iの表示
    System.out.println("      loopCount : " + loopCount);
    System.out.println("      i（カウンタ変数） : " + i);

    System.out.println("    ▲[終了]1回分の繰り返し処理");
}

System.out.println("▲[終了]for文");
```

4回目

繰り返し処理

▼実行後のコマンドプロンプト

```
C:¥WorkSpace>java Sample1_10_1
▼[開始]for文
    ▼[開始]1回分の繰り返し処理
      loopCount : 1
      i（カウンタ変数） : 0
    ▲[終了]1回分の繰り返し処理
    ▼[開始]1回分の繰り返し処理
      loopCount : 2
      i（カウンタ変数） : 1
    ▲[終了]1回分の繰り返し処理
    ▼[開始]1回分の繰り返し処理
      loopCount : 3
      i（カウンタ変数） : 2
    ▲[終了]1回分の繰り返し処理
    ▼[開始]1回分の繰り返し処理
      loopCount : 4
      i（カウンタ変数） : 3
    ▲[終了]1回分の繰り返し処理
▲[終了]for文
```

メモリ上

4	3
loopCount	i

～for文

繰り返し条件

int i = 0 ; i < 4 ; i++

初期設定
for文がはじまって最初に行う処理

実行条件
trueであれば繰り返し処理を実行する

継続処理
繰り返し処理が終わった際に実行する処理

▼Sample1_10_1.java（6～21行目）

```
int loopCount = 0 ; //何回目のループかを表す数

System.out.println("▼[開始]for文");

for( int i = 0 ; i < 4 ; i++ ){
    System.out.println("    ▼[開始]1回分の繰り返し処理");

    //loopCountを1上げる
    loopCount += 1 ;

    //loopCountおよびカウンタ変数iの表示
    System.out.println("    loopCount : " +
    System.out.println("    i（カウンタ変数） :
    System.out.println("    ▲[終了]1回分の繰り返し処理");
}

System.out.println("▲[終了]for文");
```

4回目

i < 4
↓
false
↓
繰り返し処理を終了

▼コンプト

```
Sample1_10_1
の繰り返し処理
    i（カウンタ変数） : 0
  ▲[終了]1回分の繰り返し処理
  ▼[開始]1回分の繰り返し処理
    loopCount : 2
    i（カウンタ変数） : 1
  ▲[終了]1回分の繰り返し処理
  ▼[開始]1回分の繰り返し処理
    loopCount : 3
    i（カウンタ変数） : 2
  ▲[終了]1回分の繰り返し処理
  ▼[開始]1回分の繰り返し処理
    loopCount : 4
    i（カウンタ変数） : 3
  ▲[終了]1回分の繰り返し処理
▲[終了]for文
```

メモリ上

4	4
loopCount	i

～for文のしくみ～

▼Sample1_10_1.java（6～21行目）

```
int loopCount = 0 ; //何回目のループかを表す数値（初期値0）

System.out.println("▼[開始]for文");

for( int i = 0 ; i < 4 ; i++ ){
    System.out.println("  ▼[開始]1回分の繰り返し処理");

    //loopCountを1上げる
    loopCount += 1 ;

    //loopCountおよびカウンタ変数iの表示
    System.out.println("    loopCount : " + loopCount);
    System.out.println("    i（カウンタ変数） : " + i);

    System.out.println("  ▲[終了]1回分の繰り返し処理");
}

System.out.println("▲[終了]for文");
```

計4回

▼実行後のコマンドプロンプト

```
C:¥WorkSpace>java Sample1_10_1
▼[開始]for文
  ▼[開始]1回分の繰り返し処理
    loopCount : 1
    i（カウンタ変数） : 0
  ▲[終了]1回分の繰り返し処理
  ▼[開始]1回分の繰り返し処理
    loopCount : 2
    i（カウンタ変数） : 1
  ▲[終了]1回分の繰り返し処理
  ▼[開始]1回分の繰り返し処理
    loopCount : 3
    i（カウンタ変数） : 2
  ▲[終了]1回分の繰り返し処理
  ▼[開始]1回分の繰り返し処理
    loopCount : 4
    i（カウンタ変数） : 3
  ▲[終了]1回分の繰り返し処理
▲[終了]for文
```

メモリ上

4	4
loopCount	i

～for文のしくみ～

▼Sample1_10_1.java（6～21行目）

```
int loopCount = 0 ; //何回目のループかを表す数値（初期値0）
↵
System.out.println("▼[開始]for文");↵
↵
for( int i = 0 ; i < 4 ; i++ ){↵
    System.out.println("  ▼[開始]1回分の繰り返し処理");↵
    ↵
    //loopCountを1上げる↵
    loopCount += 1 ;  ↵
    ↵
    //loopCountおよびカウンタ変数iの表示↵
    System.out.println("    loopCount：" + loopCount);↵
    System.out.println("    i（カウンタ変数）：" + i);↵
    ↵
    System.out.println("  ▲[終了]1回分の繰り返し処理");↵
}↵
↵
System.out.println("▲[終了]for文");↵
```

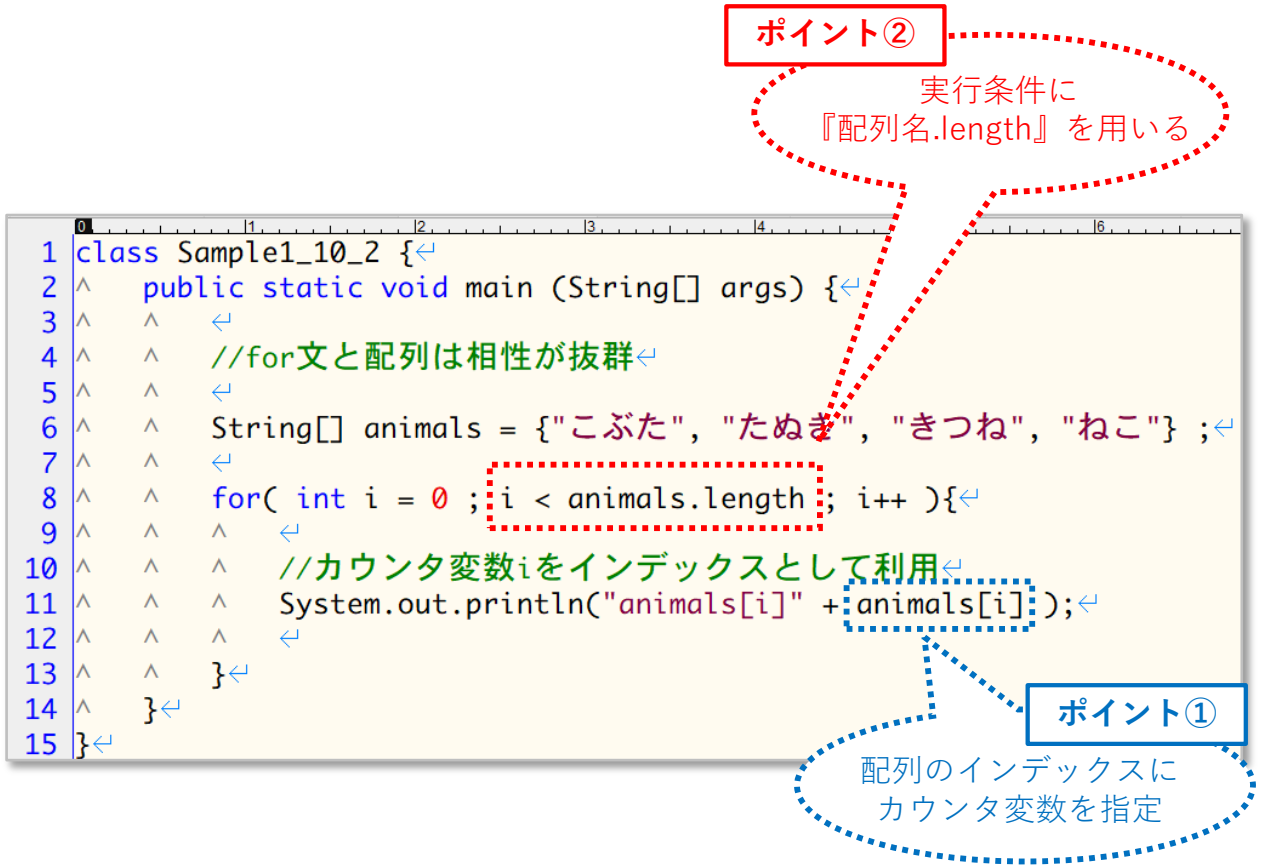
計4回

▼実行後のコマンドプロンプト

```
C:¥WorkSpace>java Sample1_10_1
▼[開始]for文
  ▼[開始]1回分の繰り返し処理
    loopCount：1
    i（カウンタ変数）：0
  ▲[終了]1回分の繰り返し処理
  ▼[開始]1回分の繰り返し処理
    loopCount：2
    i（カウンタ変数）：1
  ▲[終了]1回分の繰り返し処理
  ▼[開始]1回分の繰り返し処理
    loopCount：3
    i（カウンタ変数）：2
  ▲[終了]1回分の繰り返し処理
  ▼[開始]1回分の繰り返し処理
    loopCount：4
    i（カウンタ変数）：3
  ▲[終了]1回分の繰り返し処理
▲[終了]for文
```

メモリ上

4	4
loopCount	i



▼実行後のコマンドプロンプト

```
C:¥WorkSpace>java Sample1_10_2
animals[i] : こぶた
animals[i] : たぬき
animals[i] : きつね
animals[i] : ねこ
```

≪for文と配列≫

□配列の要素をすべてチェックしていくような場面は現場において山ほどあり、こうした場面ではfor文が頻繁に用いられます。

□配列のチェックをfor文で行うポイントは以下の2点です。

＜ポイント①＞
配列のインデックスにカウンタ変数（初期値0）を指定することで、ループ毎に要素の値をインデックス0から順に取得していくことが可能になる

＜ポイント②＞
実行条件に『配列名.length』を用いることで、配列の要素数がいくつであっても必ずその要素数だけきっちり繰り返すループを作ることができる

＜演習：Ex1_10_1＞

```
/*-< 演習：Ex1_10_1 >--------------------------------
コマンドライン引数として数字を1つ受け取り、その数字の数だけ
「＊」を繋げた文字列を表示するプログラムを作ってください。
例：受け取った数字が5→「＊＊＊＊＊」が表示される
-------------------------------------------------*/
class Ex1_10_1 {
    public static void main (String[] args) {

        //コマンドライン引数で好きな数字を1つ受け取る
        int receiveNumber = ________ ;    //(10行目)アンダーバーに適切な処理を埋めてください

        //最終的に表示するString型変数（初期値：""（0文字の文字列））
        String display = "";              //(13行目)変更しないでください

        /*
        **以下にreceiveNumberの数だけ「＊」を繋げた文字列を
        **displayに格納する処理を書いてください。
        **※必ずfor文を使用してください。
        */









        //「＊」を繋げた文字列を表示
        System.out.println(display) ;     //(32行目)変更しないでください

    }
}
```

＜演習：Ex1_10_2＞

```
 2 ↵
 3 以下、どのようなデータが画面に表示されるでしょう？↵
 4 ※プログラムで実行はしないで、紙とペンだけで考えましょう。↵
 5 ↵
 6 ^    String display = "";↵
 7 ^    for(int i = 1 ; i <= 9  ; i++){↵
 8 ^    ^    display = "";↵
 9 ^    ^    for(int j = 1 ; j <= 9  ; j++){↵
10 ^    ^    ^    display = display + i*j + " ";↵
11 ^    ^    }↵
12 ^    ^    System.out.println(display);↵
13 ^    }↵
14 ↵
```

＜演習：Ex1_10_3＞

以下のように00～99までの数字を並べて表示するプログラムを作成してください。

```
00 01 02 03 04 05 06 07 08 09
10 11 12 13 14 15 16 17 18 19
20 21 22 23 24 25 26 27 28 29
30 31 32 33 34 35 36 37 38 39
40 41 42 43 44 45 46 47 48 49
50 51 52 53 54 55 56 57 58 59
60 61 62 63 64 65 66 67 68 69
70 71 72 73 74 75 76 77 78 79
80 81 82 83 84 85 86 87 88 89
90 91 92 93 94 95 96 97 98 99
```

＜演習：Ex1_10_4＞

Ex1_10_4.java

```
/*-< 演習：Ex1_10_4 >--------------------------------
コメントに従って2次元配列からデータを抽出して画面に表示する
プログラムを作ってください。
----------------------------------------------------*/
class Ex1_10_4 {
    public static void main (String[] args) {

        //出力元の2次元配列
        String[][] animals = {
                                {"モコ" ,"トイプードル" ,"4歳" ,"メス"},
                                {"ポチ" ,"シバイヌ□□" ,"6歳" ,"オス"},
                                {"ムギ" ,"パピヨン□□" ,"2歳" ,"オス"},
                                {"ブブ" ,"ブルドッグ□" ,"3歳" ,"メス"},
                                {"シロ" ,"秋田犬□□□" ,"8歳" ,"オス"}
                              };

        /*
        **以下の条件を満たす2次元配列のデータ抽出&表示する処理を書いてください。
        **  - for文を2つ組み合わせる
        **  - 「animals.length」「 animals[i].length」
        **  - 出力結果は以下になる
        **      モコ / トイプードル / 4歳 / メス
        **      ポチ / シバイヌ□□ / 6歳 / オス
        **      ムギ / パピヨン□□ / 2歳 / オス
        **      ブブ / ブルドッグ□ / 3歳 / メス
        **      シロ / 秋田犬□□□ / 8歳 / オス
        */

```

```
C:¥WorkSpace>java Ex1_10_4
モコ / トイプードル / 4歳 / メス
ポチ / シバイヌ / 6歳 / オス
ムギ / パピヨン / 2歳 / オス
ブブ / ブルドッグ / 3歳 / メス
シロ / 秋田犬 / 8歳 / オス
```