【実習６】

繰り返し処理　for 命令で繰り返し処理をします。

１　ソースコード

テンプレート Hop020 の「//【実習 6】繰り返し処理」の箇所へ次のソースを追加入力します。

```
//i が 8 より小さい間、i に 1 ずつ加算して繰り返す
for(i = 0; i < 8; i++)
{
    num1 = num1 * 3;
}
showDialog(this, "【実習 6】", "答えは " + num1);
```

２　実行結果

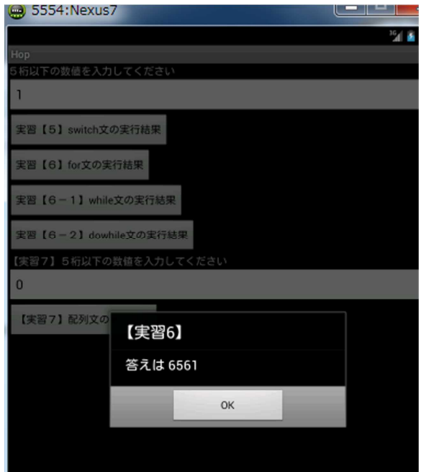
入力数値が 1 の場合、「答えは 6561 」が表示されます。

３　解説

i が 8 より小さい間、i に 1 ずつ加算して num1 = num1 * 3 を繰り返す処理です。

i++　というのは「i = i + 1」と同じ意味です。ここで繰り返しの回数を数えています。中括弧内の処理を一回実行されるたびに、ここでは「i に 1 を加算する」という処理が実行され、その結果、i が繰り返した回数をカウントしています。

なお、num1 の設定は、テンプレートのソースで行っており、新規に設定する必要はありません。また、変数 num1 には、携帯キーから入力された数値が保存されています。



3.1 for 文

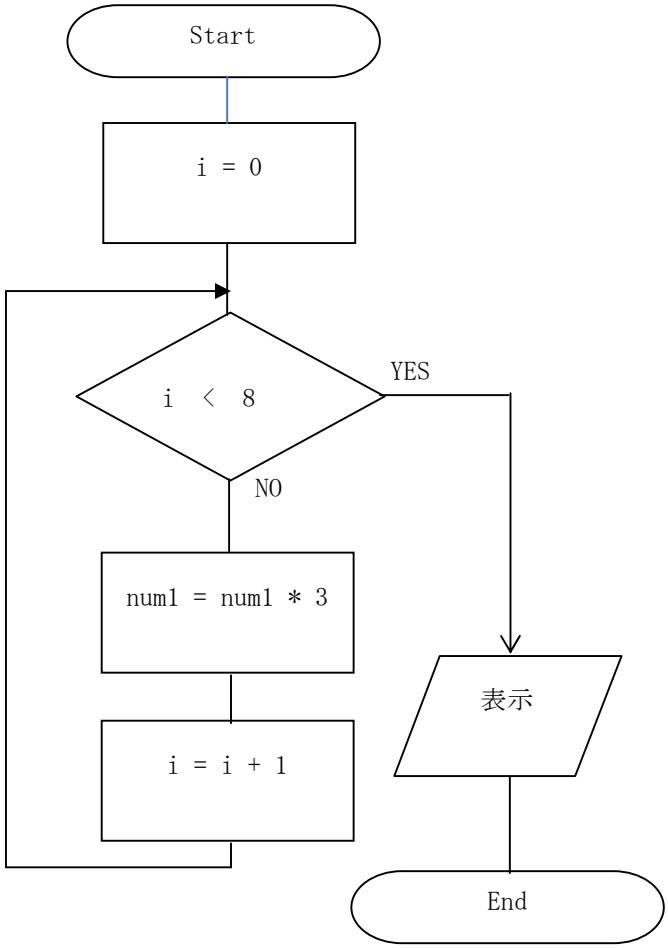
```
for （初期化； 条件； 更新） {
    文（複数可）
}
```

■トレース

No	num1	i	su1

No	num1	i	

■流れ図



3.2 while文

前判断の繰り返し処理で、条件がtrueの間、指定した文を繰り返します。

```
while （条件式） {
    文（複数可）
}
```

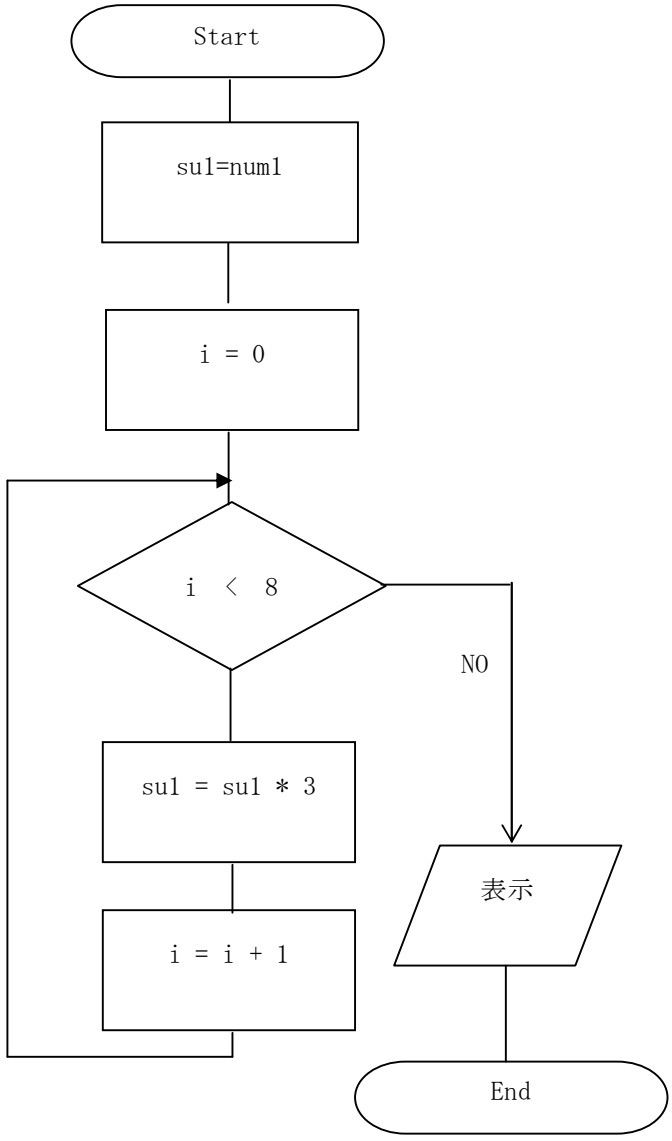
```
i=0;
// iが8より小さい間、iに1ずつ加算して繰り返す
while (i < 8)
{
    num1 = num1*3;
    i++;
}
showDialog(this, "【実習6-1】", "答えは " + num1);
```

■トレース

No	num1	i	su

No	num1	i	su

■流れ図



3.3 do～while文

後判断の繰り返し処理で、条件がtrueの間、指定した文を繰り返します。

```
do  {
    文（複数可）
}  while  （条件式）；
```

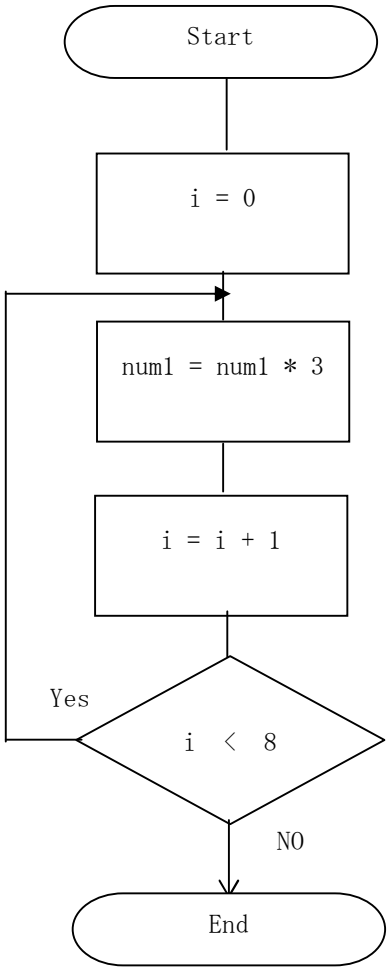
```
i=0;
// iが8より小さい間、iに1ずつ加算して繰り返す
do
{
    num1 = num1 * 3;
    i++;
} while (i < 8);
showDialog(this, "【実習6-2】", "答えは " + num1);
```

■トレース

No	num1	i	

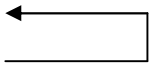
No	num1	i	

■流れ図



４ continue文

繰り返し内の処理を飛ばし、ブロックの先頭位置に戻って次の処理を続けます。

例
```
for(int i=1; i<=10; i++){
    if(i==res) continue;
}
```


５ break文

ループを中止します。【実習５】条件分岐２を参照してください。

【演習】

■次のプログラムを作成してください。

プロジェクト名: HopExPro（既存のプロジェクト）

アプリケーション名： HopEx050（既存のアプリケーション）

【演習61】

次の実行例のように、for 文を使って入力した数だけ * を出力するコードを作成してください。

実行例

```
【実習 61】入力した整数分の*表示
5 *****
```

変数 su1 を表示する。

・変数の型、名前は適宜使用してください。

・変数 num1 にキーボードから入力された値が保存されています。なお、int num1;はテンプレートのプログラム中で設定済です。

■ヒント

```
String m="";
for (～      ) {
  m=m+"*";
}
```

【演習62】

演習 61 の問題を while 文を使ってプログラミングしてください。

【演習63】

演習 61 の問題を do～while 文を使ってプログラミングしてください。

確認：0 を入力すると、演習 61、62 はなにも表示されませんが、演習 63 は表示されます。トレースで確認してください。

【演習64】

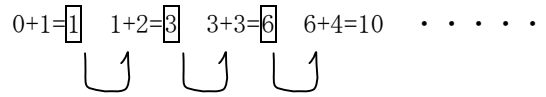
次の実行例のように for 文を使って、1 から num1 までの数の合計を計算して表示するプログラムを作成してください。

実行例

```
【実習 64】1 から 10 までの合計値
55
```

変数 num1 を表示する。

ヒント：



■次のプロジェクトとプログラムを作成してください。

プロジェクト名 ： HopPro???? (????：年組席) (既存)

アプリケーション名： HopEx060View

①Hop010View のソースをコピーして、Hop010View の２箇所の文字を、HopEx060View に変更してください。

②テンプレート(Hop010.java) の①の箇所に HopEx060View を入力してください。

③Hop010.java を実行してください。

【演習65】

次の実行例のように for 文を使って ＊ で三角形の模様を描くプログラムを作成してください。

実行例

```
*
**
***
****
*****
******
*******
********
```

■ヒント：多重の繰り返し処理

```
int x = 60;
int y = 60;
int p;
int q;
for (p=1; p<__; p++)
{
    for (q=0; q<__; q++)
    {
        canvas.drawText ("*", x, y,～);
        x = x + 10;  // x座標のカウント
    }
    y = y + 20;   // y座標のカウント
    x = 60;
}
```

【演習66】

次の実行例のように九九の内容を出力するプログラムを作成してください。

実行例

```
＊＊＊＊＊九九の表＊＊＊＊＊
1  2  3  4  5  6  7  8  9
2  4  6  8 10 12 14 16 18
3  6  9 12 15 18 21 24 27
4  8 12 16 20 24 28 32 36
5 10 15 20 25 30 35 40 45
6 12 18 24 30 36 42 48 54
7 14 21 28 35 42 49 56 63
```

ヒント：多重の繰り返し処理

```
x = 60;
y = 300;
canvas.drawText ("＊＊＊～",～);
for (p=1; p<__; p++)
{
    y = y + 30;   // y座標のカウント
    for (q=1; q<__;q++)
    {
        canvas.drawText ((p * q)+"  ",
                          x, y, ～);
        x = x + 40;  // x座標のカウント
    }
    x = 60;
}
```