

プログラミング入門1

第4回

繰り返し(forループ)

授業開始前に

ログオンして待機して
ください

不要ファイルの掃除

- 前回デスクトップにファイルをダウンロードした場合、次のものを削除してください
 - week03.zip
- デスクトップにファイルを置きすぎると、コンピュータをシャットダウンできなくなります

授業を始めます

前回の課題は

うまくできましたか？

ソースプログラムの清書

もし、こんなふうになってしまったら

```
public class Example {  
  
    public static void main(String[] args) {  
        System.out.println("Hello, world!");  
        System.out.println("Hello, everybody!");  
    }  
}
```


↓
メニューから「ソース→フォーマット」
あるいは Ctrl + Shift + f

```
public class Example {  
  
    public static void main(String[] args) {  
        System.out.println("Hello, world!");  
        System.out.println("Hello, everybody!");  
    }  
}
```

復習 前回のテーマ

- ユーザの入力がある対話的プログラム
 - コンソールウィンドウを介して、プログラムと対話する
- ユーザ入力によって処理の内容を変える

復習 入力を行うプログラムの骨格

```
package [パッケージ名 (j1.Lesson03など)];  
  
import java.io.*;  
  
public class [クラス名 (Helloなど)] {  
    public static void main(String[] args) throws IOException {  
  
        BufferedReader reader =  
            new BufferedReader(new InputStreamReader(System.in));  
        // 以下にプログラムを書く  
  
  
  
    }  
}
```

復習 コンソールに入力された整数を 取得する命令

```
int input = Integer.parseInt(reader.readLine());
```

↑ ↓ どちらでもOK

```
int input;  
input = Integer.parseInt(reader.readLine());
```

この命令の実行する内容

1. コンソールからのユーザの入力を待つ(Enterキーが押されるまで)
2. 入力された整数データをint型変数inputに代入
3. 完了

復習 コンソールに入力された実数を 取得する命令

```
double input = Double.parseDouble(reader.readLine());
```

この命令の実行する内容

1. コンソールからのユーザの入力を待つ(Enterキーが押されるまで)
2. 入力された実数データをdouble型変数inputに代入
3. 完了

復習 if 文 (if statement)

```
if ( a > 0 )  
    b = a * 3;
```

- aの値が0より大きいときだけ代入文 `b = a * 3;` が実行される。
- `a > 0` は条件式とよばれる
 - 条件式はboolean型の値(trueかfalseのどちらか)をとる。
 - aの値が0より大きいとき条件式 `a > 0` の値はtrueとなり、そうでないときfalseとなる。
- この「>」は比較演算子と呼ばれる。

復習 if-else 文 (if-else statement)

```
if ( a > 0 )  
    b = a * 3;  
else  
    b = 0;
```

aの値が0より大きいときだけ代入文 $b = a * 3;$ が実行され、
そうでない場合は代入文 $b = 0$ が実行される

復習 ブロックで命令を束ねる

```
if ( a > 0 ) {  
    b = a * 3;  
    d = b + 8;  
} else  
    b = 5;
```

```
if ( a > 0 ) {  
    b = a * 3;  
    d = b + 8;  
} else {  
    b = 5;  
    d = 6;  
}
```

復習 if文の連鎖

```
if (chanceOfRain > 100)
    System.out.println("降水確率は0~100でなければなりません。");
else if (n < 0)
    System.out.println("降水確率は0~100でなければなりません。");
else if (chanceOfRain >= 50)
    System.out.println("傘を忘れずに。");
else
    System.out.println("傘はいりません。");
```

復習終わり

思い出しましたね。

今週のテーマ

- 繰り返し実行のための便利な書き方
- for 文を用いる

for 文を用いた繰り返し実行

変数 `i` の値が 0 から

10 未満の間は

`i` の値を1ずつ加算しながら

命令文を実行する。

```
for (int i = 0 ; i < 10 ; i ++ )  
    System.out.println("i の値は" + i);
```

実行結果

```
i の値は0  
i の値は1  
i の値は2  
i の値は3  
i の値は4  
i の値は5  
i の値は6  
i の値は7  
i の値は8  
i の値は9
```


for 文の構造

for (<初期化式> ; <条件式> ; <ステップを進める式>)
文

```
for (int i = 0 ; i < 10 ; i ++ )  
System.out.println("iの値は" + i);
```

for 文の初期化式

- 初期化式はfor文を開始するときに1度だけ実行される
- 通常、ループに使用する変数を宣言と同時に初期化する
- この変数はループ制御変数とよばれる

```
for (int i = 0 ; i < 10 ; i++ )  
    System.out.println("iの値は" + i);
```

注意

- ループ制御変数は、どのような名前でもよい。
 - よく使われる変数名は i, j, k など。
- 初期化式で宣言された変数はfor文の中でのみ有効。

```
for (int i = 0; i < 10; i++)  
    System.out.println("iの値は" + i);  
  
// エラー。ここで変数 i は使用できない。  
System.out.println("for文が終わったあとのiの値は" + i);
```

- # 注意
- ループ制御変数は、for 文の外で宣言することができる。
 - その場合は for 文の中でも外でも使える。

```
int i;  
for (i = 0; i < 10; i++)  
    System.out.println("iの値は" + i);  
  
// エラーではない。iの値は10になるはず。  
System.out.println("for文が終わったあとのiの値は" + i);
```

以下の例は変数のスコープ、寿命を考える良い例だが、ここでは説明しない。興味に火がついてしまった人は参考書を参照。

```
int i=0;  
for (int i = 0; i < 10; i++)  
    System.out.println("iの値は" + i);  
  
// エラーではない。iの値は0になるはず。  
System.out.println("for文が終わったあとのiの値は" + i);
```

for 文の条件式

- 毎回最初に条件式が true か false かチェックされる
- true であればその回を実行する
- false であれば、その回は実行されず、for 文の繰り返し実行自体も打ち切られる

```
for (int i = 0 ; i < 10 ; i ++ )  
    System.out.println("i の値は" + i );
```

for 文のステップを進める式

- ループの制御変数を操作する
- 繰り返しの**毎回最後**に実行される

```
for (int i = 0 ; i < 10 ; i ++ )  
    System.out.println("iの値は" + i);
```

初期化式を比べてみよう

```
for (int i = 0 ; i < 10 ; i++ )  
    System.out.println("iの値は" + i);
```

iの値は0
iの値は1
iの値は2
iの値は3
iの値は4
iの値は5
iの値は6
iの値は7
iの値は8
iの値は9

```
for (int i = 1 ; i < 10 ; i++ )  
    System.out.println("iの値は" + i);
```

iの値は1
iの値は2
iの値は3
iの値は4
iの値は5
iの値は6
iの値は7
iの値は8
iの値は9

条件式を比べてみよう

```
for (int i = 0 ; i < 10 ; i++ )  
    System.out.println("iの値は" + i);
```

i の値は0
i の値は1
i の値は2
i の値は3
i の値は4
i の値は5
i の値は6
i の値は7
i の値は8
i の値は9

```
for (int i = 0 ; i <= 10 ; i++ )  
    System.out.println("iの値は" + i);
```

i の値は0
i の値は1
i の値は2
i の値は3
i の値は4
i の値は5
i の値は6
i の値は7
i の値は8
i の値は9
i の値は10

ステップを進める式を比べてみよう

```
for (int i = 0 ; i < 10 ; i++)  
    System.out.println("iの値は" + i);
```

iの値は0
iの値は1
iの値は2
iの値は3
iの値は4
iの値は5
iの値は6
iの値は7
iの値は8
iの値は9

```
for (int i = 0 ; i < 10 ; i+=2)  
    System.out.println("iの値は" + i);
```

iの値は0
iの値は2
iの値は4
iの値は6
iの値は8

カウントダウンと比べてみよう

```
for (int i = 0 ; i < 10 ; i++ )  
    System.out.println("iの値は" + i);
```

iの値は0
iの値は1
iの値は2
iの値は3
iの値は4
iの値は5
iの値は6
iの値は7
iの値は8
iの値は9

```
for (int i = 9 ; i >= 0 ; i-- )  
    System.out.println("iの値は" + i);
```

iの値は9
iの値は8
iの値は7
iの値は6
iの値は5
iの値は4
iの値は3
iの値は2
iの値は1
iの値は0

for 文で繰り返される命令文

```
for (int i = 0 ; i < 10 ; i++ )  
    System.out.println("iの値は" + i);
```

複数の命令文を繰り返したいときはブロックにする

```
for (int i = 0 ; i < 10 ; i++ ){  
    System.out.print("iの値は");  
    System.out.println(i);  
}
```

ユーザが指定した回数だけ繰り返す

```
BufferedReader reader
    = new BufferedReader(new InputStreamReader(System.in));
int input = Integer.parseInt(reader.readLine());

for (int i = 0; i < input; i++) {
    (命令文の列)
}
```

変数がきてもよい

for文の中で計算する 1+2+ ... +10

```
int total = 0;
for (int i = 1; i <= 10; i++) {
    total += i;
}
System.out.println("入力の合計は" + total);
```

作業用の変数totalはループの外でも参照するので宣言もループの外です

10回の入力の総和

```
BufferedReader reader
    = new BufferedReader(new InputStreamReader(System.in));
int total = 0;
for (int i = 1; i <= 10; i++) {
    System.out.print(i + "回目の入力");
    int input = Integer.parseInt(reader.readLine());
    total += input;
}
System.out.println("入力の合計は" + total);
```

一緒に試してみよう

- for文を使った繰り返しのプログラムを書く

演習の準備

- 今回の演習で使うテストドライバをいつものようにインストールします。
- 一緒に作業をしますので、指示に従ってやってください。

テストドライバの導入

1. プロジェクト「java20XX」にある「test」の左側の「+」をクリック
2. ツリーが展開されるので「install-libraries.xml」を右クリック
3. 「実行(R)」にマウスカーソルを合わせる
4. 「1 Ant ビルド」をクリック
5. 「コンソール」タブに"BUILD SUCCESSFUL"と表示されれば成功
6. eclipseの画面でプロジェクト「java20XX」を右クリック
7. メニューが表示されるので、「更新」あるいは「最新表示」をクリック
8. [week04.zip](#) をデスクトップなどにダウンロード
9. eclipseの画面でプロジェクト「java20XX」を右クリック
10. メニューから「インポート(I)」を選択
11. 「インポート」ウィンドウが表示されるので、「Zip ファイル」を選択
12. 「次へ(N)」をクリック
13. 宛先フォルダー(L): が「java20XX」になっていることを確認
14. From archive file: の右側にある「参照(R)...」あるいは「ブラウズ(R)...」をクリック
15. ファイルダイアログが表示されるので、ダイアログ内に表示されたダウンロードしたファイルをダブルクリック
16. 前の画面に戻るので、From archive file: のエリアに正しいパスが入力されていることを確認
17. フォルダ「/」の左にチェックがついていることを確認 (ついていなければチェックボックスをクリック)
18. 「警告を出さずに既存リソースを上書き」にチェックがついていることを確認 (上書きしたくないファイルがある場合はチェックを外す)
19. 「終了 (F)」をクリック

テストドライバの導入に成功すると

- プロジェクト「java20XX」の中の「test」というフォルダに「j1.lesson04.xml」という名前のファイルが作成される。
- このファイルには今週使用するテスト一式が記述されている。

パッケージ

- 今回は `j1.lesson04` というパッケージを作成する。
- パッケージの作成は毎回の演習の最初に行うことを原則とする。

パッケージの作成

1. 「java20XX」プロジェクトの左側にある「+」をクリック
2. ツリーが展開されるので、「src」の上で右クリック
3. マウスカーソルを「新規」に合わせる
4. 「パッケージ」をクリック
5. パッケージ名は `j1.lesson04` と入力する。(以下いつも通り)

クラスの作成

1. 先ほど作成したパッケージ「j1.lesson04」の上で右クリック
2. マウスカーソルを「新規」に合わせる
3. 「クラス」をクリック
4. クラス名は LoopPrint とする。

以下のようにプログラムの骨格を書いて
保存(自動コンパイル) Ctrl+S

```
package j1.Lesson04;
import java.io.*;

public class LoopPrint {
    public static void main(String[] args) throws IOException {

    }
}
```

ここで骨格テスト

1. testフォルダの中にあるj1.lesson04.xml を右クリック
2. 「実行」の中にある「1 Ant ビルド」を選択
3. LoopPrintに対する骨格テストを実行

骨格テストのエラーメッセージ

メッセージ	詳細
クラスが存在しません	j1.lesson04 に LoopPrint クラスが存在していない。 パッケージやクラス名を確認
クラスがpublicで宣言されていません	class の前に public の指定がない。
mainメソッドが存在しません	public static void main(String[]) が存在していない。 mainという名前やString[]の部分を確認
mainメソッドがpublicで宣言されていません	mainメソッドを作る際に public が抜けている。
mainメソッドがstaticで宣言されていません	mainメソッドを作る際に static が抜けている。
mainメソッドがvoidで宣言されていません	mainメソッドを作る際に void 以外を指定している。
mainメソッドにthrows IOExceptionの指定がありません	mainメソッドを作る際にthrows IOExceptionの指定 を行ってない。 40

mainメソッドの中身を書く このプログラムの意味をよく考えよう

```
BufferedReader reader =
    new BufferedReader(new InputStreamReader(System.in));
System.out.println("繰り返す回数を入力してください:");
int count = Integer.parseInt(reader.readLine());

// 0回未満の繰り返しはエラー
if (count < 0) {
    System.out.println("繰り返す回数には0以上を入れてください。");
} else {
    // i が 0 から count - 1 まで繰り返す
    for (int i = 0; i < count; i++) {
        System.out.println("Hello!");
    }
}
```

プログラムの実行

1. 「LoopPrint.java」の上で右クリック
2. メニューが表示されるので、「実行(R)」にマウスカーソルを合わせる
3. 「1 Java アプリケーション」をクリック
4. コンソールでプログラムのメッセージにしたがって入力をしてみよう。
5. 結果に納得できただろうか。

機能テスト

1. Jtafウィンドウで「LoopPrintに対する機能テスト」選択し実行する
2. エラーが出たらメッセージをよく読みソースプログラムを直して、今までのステップを繰り返す

コンソール入力には準備が必要

```
package j1.lesson04;
import java.io.*;
public class LoopPrint {
    public static void main(String[] args) throws IOException {
        BufferedReader reader =
            new BufferedReader(new InputStreamReader(System.in));
        System.out.println("繰り返す回数を入力してください:");
        int count = Integer.parseInt(reader.readLine());

        // 0回未満の繰り返しはエラー
        if (count < 0) {
            System.out.println("繰り返す回数には0以上を入れてください。");
        }
        else {
            // i が 0 から count - 1 まで繰り返す
            for (int i = 0; i < count; i++) {
                System.out.println("Hello!");
            }
        }
    }
}
```

ユーザに整数の入力を促す

```
package j1.lesson04;
import java.io.*;
public class LoopPrint {
    public static void main(String[] args) throws IOException {
        BufferedReader reader =
            new BufferedReader(new InputStreamReader(System.in));
        System.out.println("繰り返す回数を入力してください:");
        int count = Integer.parseInt(reader.readLine());

        // 0回未満の繰り返しはエラー
        if (count < 0) {
            System.out.println("繰り返す回数には0以上を入れてください。");
        }
        else {
            // i が 0 から count - 1 まで繰り返す
            for (int i = 0; i < count; i++) {
                System.out.println("Hello!");
            }
        }
    }
}
```

int 型の変数 count を宣言 ユーザの入力した整数を count に代入

```
package j1.lesson04;
import java.io.*;
public class LoopPrint {
    public static void main(String[] args) throws IOException {
        BufferedReader reader =
            new BufferedReader(new InputStreamReader(System.in));
        System.out.println("繰り返す回数を入力してください:");
        int count = Integer.parseInt(reader.readLine());

        // 0回未満の繰り返しはエラー
        if (count < 0) {
            System.out.println("繰り返す回数には0以上を入れてください。");
        }
        else {
            // i が 0 から count - 1 まで繰り返す
            for (int i = 0; i < count; i++) {
                System.out.println("Hello!");
            }
        }
    }
}
```

count の中身が 0 未満かどうかで分岐

```
package j1.lesson04;
import java.io.*;
public class LoopPrint {
    public static void main(String[] args) throws IOException {
        BufferedReader reader =
            new BufferedReader(new InputStreamReader(System.in));
        System.out.println("繰り返す回数を入力してください:");
        int count = Integer.parseInt(reader.readLine());

        // 0回未満の繰り返しはエラー
        if (count < 0) {
            System.out.println("繰り返す回数には0以上を入れてください。");
        }
        else {
            // i が 0 から count - 1 まで繰り返す
            for (int i = 0; i < count; i++) {
                System.out.println("Hello!");
            }
        }
    }
}
```

count の中身が 0 未満なら

```
package j1.lesson04;
import java.io.*;
public class LoopPrint {
    public static void main(String[] args) throws IOException {
        BufferedReader reader =
            new BufferedReader(new InputStreamReader(System.in));
        System.out.println("繰り返す回数を入力してください:");
        int count = Integer.parseInt(reader.readLine());

        // 0回未満の繰り返しはエラー
        if (count < 0) {
            System.out.println("繰り返す回数には0以上を入れてください。");
        }
        else {
            // i が 0 から count - 1 まで繰り返す
            for (int i = 0; i < count; i++) {
                System.out.println("Hello!");
            }
        }
    }
}
```


count の中身が 0 以上なら

```
package j1.lesson04;
import java.io.*;
public class LoopPrint {
    public static void main(String[] args) throws IOException {
        BufferedReader reader =
            new BufferedReader(new InputStreamReader(System.in));
        System.out.println("繰り返す回数を入力してください:");
        int count = Integer.parseInt(reader.readLine());

        // 0回未満の繰り返しはエラー
        if (count < 0) {
            System.out.println("繰り返す回数には0以上を入れてください。");
        }
        else {
            // i が 0 から count - 1 まで繰り返す
            for (int i = 0; i < count; i++) {
                System.out.println("Hello!");
            }
        }
    }
}
```

階乗の計算を行うプログラム

1. 先ほど作成したパッケージ「j1.lesson04」の上で右クリック
2. マウスマウスカーソルを「新規」に合わせる
3. 「クラス」をクリック
4. クラス名は Factorial とする
5. 先と同じ手順でクラスを作成する

以下のようにプログラムの骨格を書いて
保存(自動コンパイル) Ctrl+S

```
package j1.lesson04;
import java.io.*;

public class Factorial {
    public static void main(String[] args) throws IOException {

    }
}
```

骨格テストは pr02factorial-skel.xml

mainメソッドの中身を書く

```
public class Factorial {
    public static void main(String[] args) throws IOException {
        BufferedReader reader =
            new BufferedReader(new InputStreamReader(System.in));
        System.out.println("整数を入力:");
        int n = Integer.parseInt(reader.readLine());
        if (n < 0) {
            System.out.println("未定義です");
        }
        else {
            // 乗算の単位元は 1
            int factorial = 1;
            for (int i = n; i >= 1; i--) {
                factorial *= i;
            }
            System.out.println(n + "の階乗は" + factorial);
        }
    }
}
```

Factorialの実行

- 暗算でできる程度のテストはすぐにやってみよ

$$3! = 6$$

$$4! = 24$$

$$5! = 120$$

機能テスト

- Factorialに対す機能テストを実行する
- これがうまく行ったらFactorialの解説をします

Factorial の解説

```
public class Factorial {
    public static void main(String[] args) throws IOException {
        BufferedReader reader =
            new BufferedReader(new InputStreamReader(System.in));
        System.out.println("整数を入力:");
        int n = Integer.parseInt(reader.readLine());
        if (n < 0) {
            System.out.println("未定義です");
        }
        else {
            // 乗算の単位元は 1
            int factorial = 1;
            for (int i = n; i >= 1; i--) {
                factorial *= i;
            }
            System.out.println(n + "の階乗は" + factorial);
        }
    }
}
```

コンソール入力の準備

```
public class Factorial {
    public static void main(String[] args) throws IOException {
        BufferedReader reader =
            new BufferedReader(new InputStreamReader(System.in));
        System.out.println("整数を入力:");
        int n = Integer.parseInt(reader.readLine());
        if (n < 0) {
            System.out.println("未定義です");
        }
        else {
            // 乗算の単位元は 1
            int factorial = 1;
            for (int i = n; i >= 1; i--) {
                factorial *= i;
            }
            System.out.println(n + "の階乗は" + factorial);
        }
    }
}
```


ユーザに整数の入力を促す

```
public class Factorial {
    public static void main(String[] args) throws IOException {
        BufferedReader reader =
            new BufferedReader(new InputStreamReader(System.in));
        System.out.print("整数を入力: ");
        int n = Integer.parseInt(reader.readLine());
        if (n < 0) {
            System.out.println("未定義です");
        }
        else {
            // 乗算の単位元は 1
            int factorial = 1;
            for (int i = n; i >= 1; i--) {
                factorial *= i;
            }
            System.out.println(n + "の階乗は" + factorial);
        }
    }
}
```

int 型の変数 n を宣言 ユーザの入力した整数を n に代入

```
public class Factorial {
    public static void main(String[] args) throws IOException {
        BufferedReader reader =
            new BufferedReader(new InputStreamReader(System.in));
        System.out.println("整数を入力:");
        int n = Integer.parseInt(reader.readLine());
        if (n < 0) {
            System.out.println("未定義です");
        }
        else {
            // 乗算の単位元は 1
            int factorial = 1;
            for (int i = n; i >= 1; i--) {
                factorial *= i;
            }
            System.out.println(n + "の階乗は" + factorial);
        }
    }
}
```

n<0 かどうかで処理を分ける

```
public class Factorial {
    public static void main(String[] args) throws IOException {
        BufferedReader reader =
            new BufferedReader(new InputStreamReader(System.in));
        System.out.println("整数を入力:");
        int n = Integer.parseInt(reader.readLine());
        if (n < 0) {
            System.out.println("未定義です");
        }
        else {
            // 乗算の単位元は 1
            int factorial = 1;
            for (int i = n; i >= 1; i--) {
                factorial *= i;
            }
            System.out.println(n + "の階乗は" + factorial);
        }
    }
}
```

n<0 なら

```
public class Factorial {
    public static void main(String[] args) throws IOException {
        BufferedReader reader =
            new BufferedReader(new InputStreamReader(System.in));
        System.out.println("整数を入力:");
        int n = Integer.parseInt(reader.readLine());
        if (n < 0) {
            System.out.println("未定義です");
        }
        else {
            // 乗算の単位元は 1
            int factorial = 1;
            for (int i = n; i >= 1; i--) {
                factorial *= i;
            }
            System.out.println(n + "の階乗は" + factorial);
        }
    }
}
```

n<0 でないなら

```
public class Factorial {
    public static void main(String[] args) throws IOException {
        BufferedReader reader =
            new BufferedReader(new InputStreamReader(System.in));
        System.out.println("整数を入力:");
        int n = Integer.parseInt(reader.readLine());
        if (n < 0) {
            System.out.println("未定義です");
        }
        else {
            // 乗算の単位元は 1
            int factorial = 1;
            for (int i = n; i >= 1; i--) {
                factorial *= i;
            }
            System.out.println(n + "の階乗は" + factorial);
        }
    }
}
```

カウントダウンによる繰り返し

nが5の場合

```
int factorial = 1;
for (int i = n; i >= 1; i--) {
    factorial *= i;
}
```

i	i >= 1 条件式	factorial 掛け算実行後の値	i --
5	true	5	4
4	true	20	3
3	true	60	2
2	true	120	1
1	true	120	0
0	false	実行しない	

課題

各自のペースで

「第04週目の課題」をやってみよう

1. <http://java.cis.k.hosei.ac.jp/> をブラウザで開く
2. 第04回 – Loops (for) - をクリック
3. 第04週目課題をクリック

ToZero のヒント

count へ入力

```
if (countが正){  
    for文でカウントダウン  
} else if (countが負){  
    for文でカウントアップ  
} else  
    0を出力
```


Average のヒント

```
BufferedReader reader = .....
```

```
...
```

```
//回数を入力
```

```
int n = Integer.parseInt(reader.readLine());
```

```
if(n が 0 以下){
```

```
    警告
```

```
}else{
```

```
    total を初期化
```

```
    for(int i=0; i<n; i++){
```

```
        x に入力
```

```
        total += x;
```

```
    }
```

```
    平均を出力
```

```
}
```