プログラミング入門1

第3回

条件分岐

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
  - week02.zip

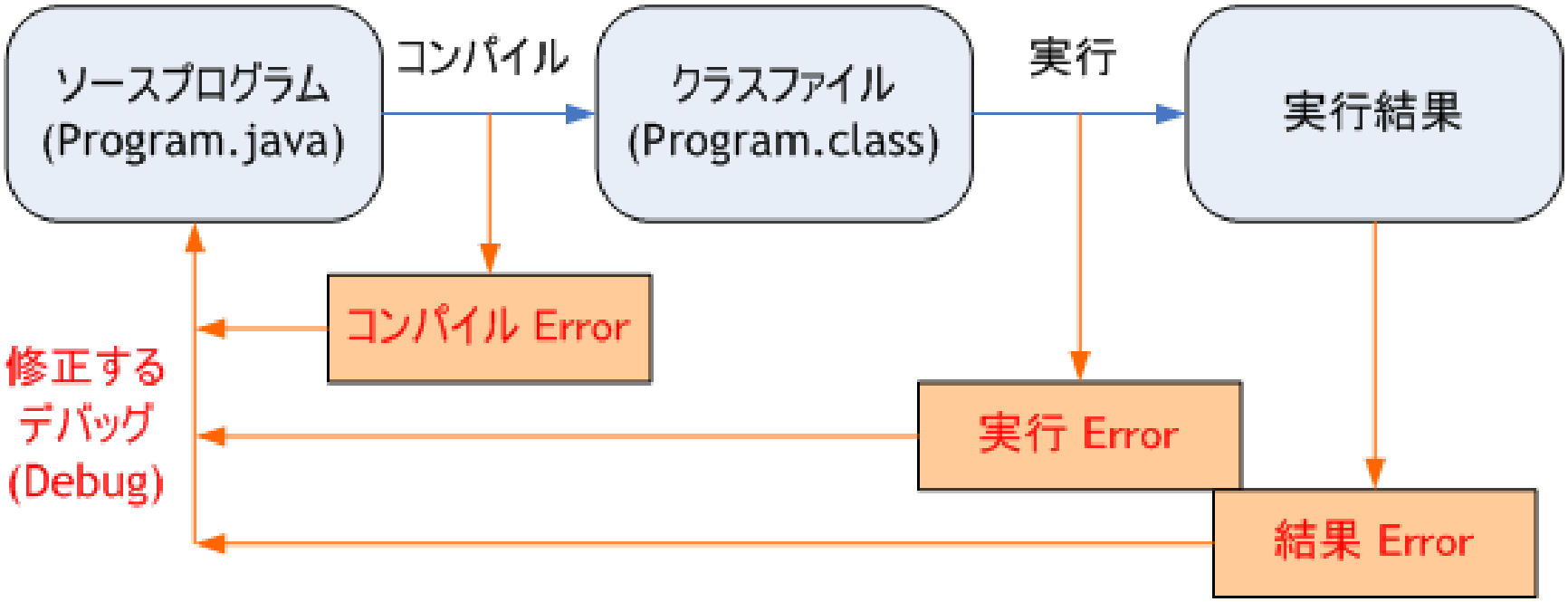
- デスクトップにファイルを置きすぎると、コンピュータをシャットダウンできなくなります

授業を始めます

前回の課題は

うまくできましたか？

復習：プログラムを書く・実行する
もう分かってますね



復習:プログラムの書き方

```
public class プログラムの名前 {
    public static void main(String[] args) {
        System.out.println("Hello!");

        ここに命令をいくつ並べてもよい

    }
}
```

復習　算術演算のデータ
データには型がある

- 整数型(int)

  `0,  1,  3,  563,  -32,  -1024`

- 実数型

  `1.0, 3.14, -73.5, 18.(=18.0), .5`(=0.5)

  – 小数点つきの数
  – 正確には浮動小数点数（floating point number）と呼ばれる

復習　算術演算の結果

- 整数と整数の演算結果は整数である。
  12 + 34 の結果は整数の46
- 実数と実数の演算結果は実数である。
  4.0 + 1.0 の結果は実数の 5.0
- 整数と実数、あるいは実数と整数の演算結果は実数である。
  7.0 / 3 の結果は実数の2.3333333333333335

復習　変数と型

- 変数は**何かを入れておく箱**のようなもの
  - 変数は名前が必要
  - どんな種類(整数や実数など)のデータを入れる箱であるかを予め決めなければならない。
  - これを型という
- 宣言：　変数を用意すること
  - 変数の型と名前を指定しなければならない
  - **宣言されていない変数は使えない**
- 代入：　変数にデータを入れること
- 参照：　変数に入っているデータを利用すること

宣言：変数を用意する

int型の変数 x を宣言する

```
int x;
```

のようにすると、整数(integer)を入れるための変数 x が用意される。

double型の変数 a を宣言する

```
double a;
```

のようにすると、実数を入れるための変数 a が用意される。

行末のセミコロン（;）を忘れないように。

同じ名前の変数の宣言は2度やってはいけない
ここに示す例はエラーになる

```
……
int x;
int x;
……
```

```
……
int x;
double x;
……
```

変数の宣言と初期化を同時にやってもよい

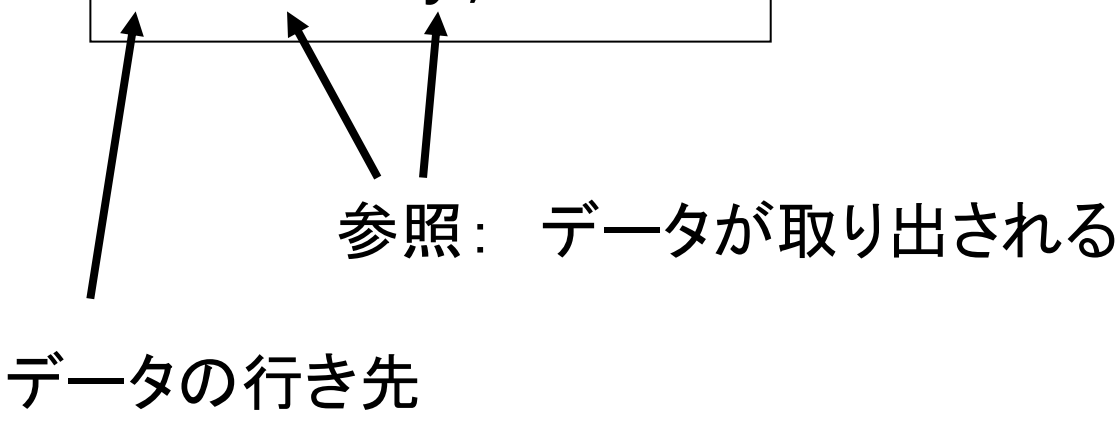
```
int x;
x=100;
```

同等 ⟷

```
int x=100;
```

復習　変数の宣言と参照

```
int x;
int y;
x = 100;
y = 200;
int z;
z = x + y;
```



復習終わり

思い出しましたね。

今回のテーマ

- 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

整数をユーザが入力し、それをそのまま出力する (こんな感じ)

```
package j1.lesson03;

import java.io.*;

public class EchoInt {
  public static void main(String[] args) throws IOException {
    BufferedReader reader =
      new BufferedReader(new InputStreamReader(System.in));
    System.out.print("input:");
    int input = Integer.parseInt(reader.readLine());
    System.out.println(input);
  }
}
```

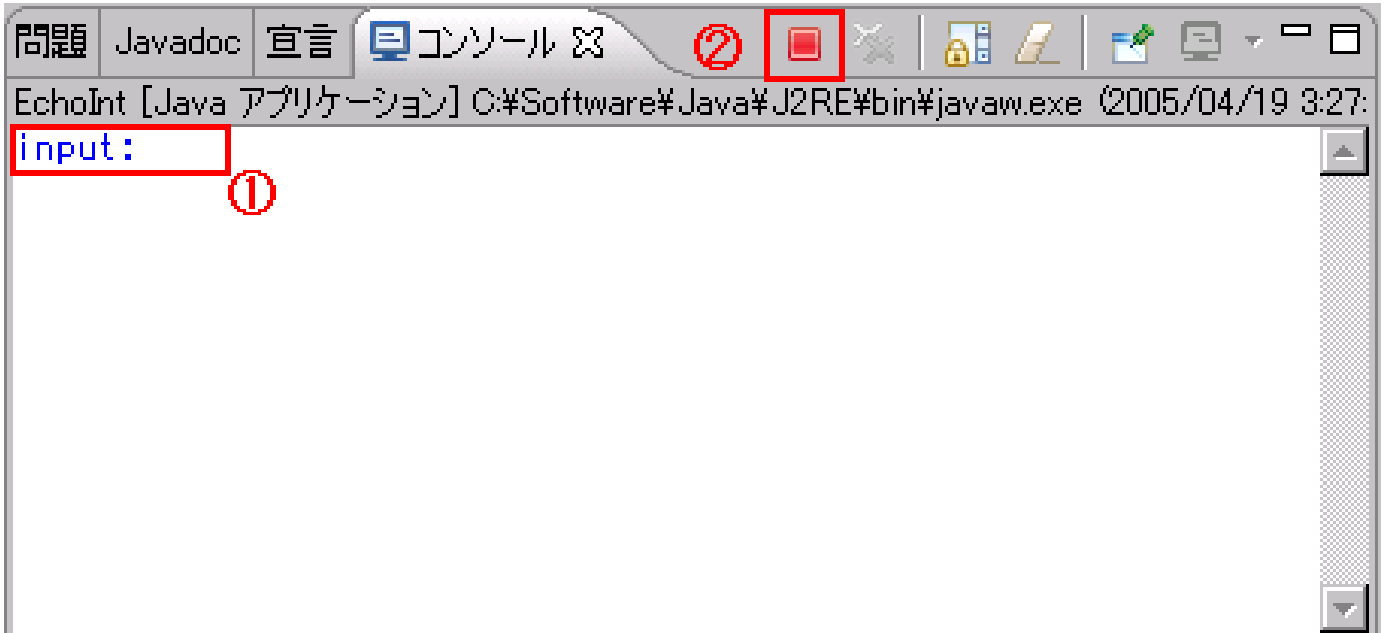
実行すると:　プログラムは一時的に停止して、

コンソール画面へのユーザの入力を待っている。

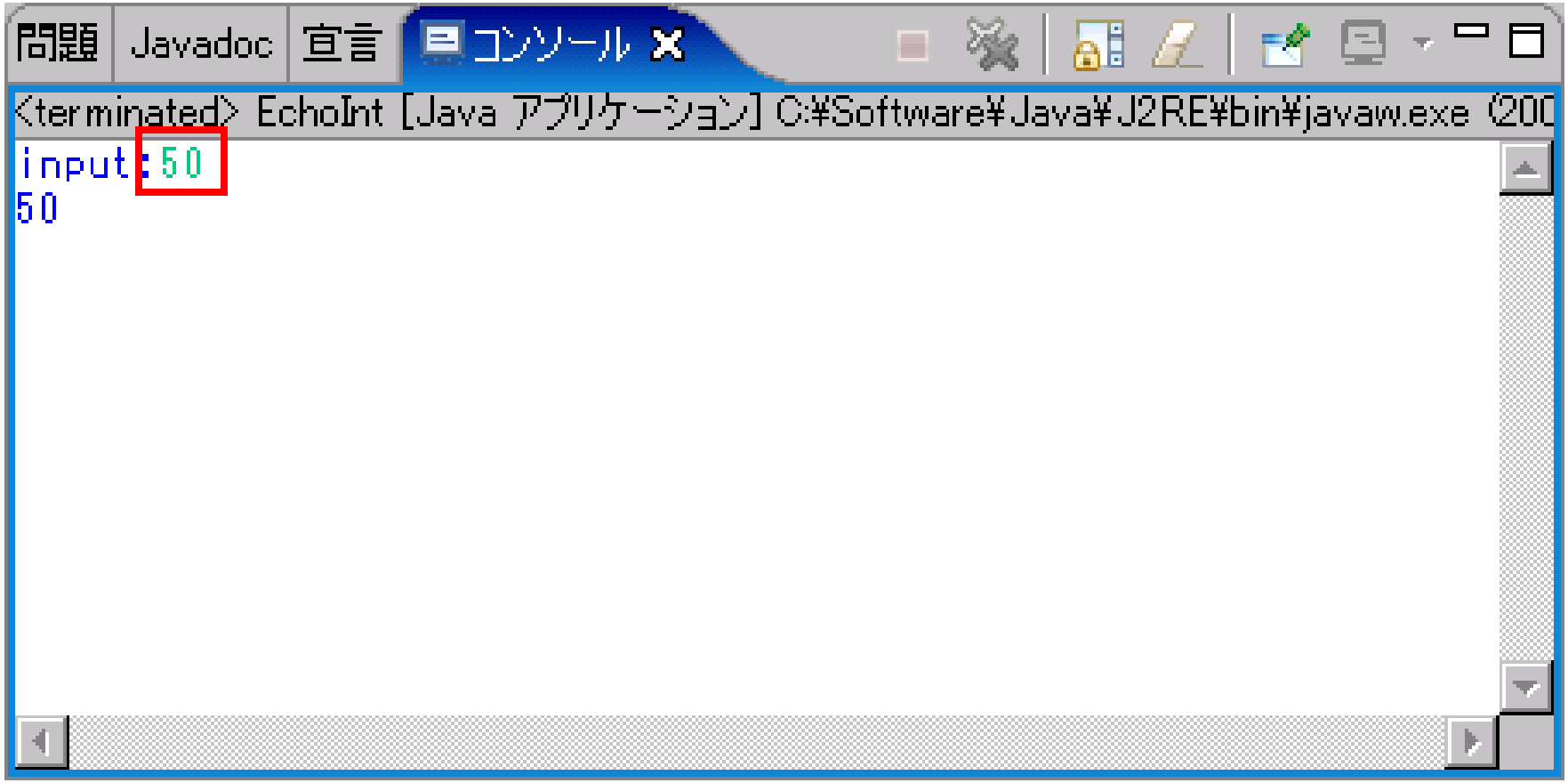
```
package j1.lesson03;

import java.io.*;

public class EchoInt {
  public static void main(String[] args) throws IOException {
    BufferedReader reader =
      new BufferedReader(new InputStreamReader(System.in));
    System.out.print("input:");
    int input = Integer.parseInt(reader.readLine());
    System.out.println(input);
  }
}
```



"input:" と表示されている部分の少し右あたりをマウスでクリックする。
ここで整数を入力しEnterキーを押すとプログラムが再開される。



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


  }
}
```

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

次の例をトレースしてみよう

```
package j1.lesson03;
import java.io.*;

public class EchoTwoInt {
  public static void main(String[] args) throws IOException {
    BufferedReader reader
      = new BufferedReader(new InputStreamReader(System.in));
    System.out.print("input (a):");
    int a = Integer.parseInt(reader.readLine());
    System.out.print("input (b):");
    int b = Integer.parseInt(reader.readLine());
    System.out.println("aの値は" + a);
    System.out.println("bの値は" + b);
  }
}
```

入力のための準備

```
package j1.lesson03;
import java.io.*;

public class EchoTwoInt {
  public static void main(String[] args) throws IOException {
    BufferedReader reader
      = new BufferedReader(new InputStreamReader(System.in));
    System.out.print("input (a):");
    int a = Integer.parseInt(reader.readLine());
    System.out.print("input (b):");
    int b = Integer.parseInt(reader.readLine());
    System.out.println("aの値は" + a);
    System.out.println("bの値は" + b);
  }
}
```

「入力のための準備」は種を明かせば BufferedReader型変数readerの宣言と初期化 でも今は「準備」という理解でよい

```
BufferedReader reader =
  new BufferedReader(new InputStreamReader(System.in));
```

したがって、どちらでもOK

右辺の new …については後で勉強する

```
BufferedReader reader;
reader = new BufferedReader(new InputStreamReader(System.in));
```

コンソールに"input (a):"という文字列を出力し
コンソールからの整数の入力をint型変数aに代入する

```
package j1.lesson03;
import java.io.*;

public class EchoTwoInt {
  public static void main(String[] args) throws IOException {
    BufferedReader reader
      = new BufferedReader(new InputStreamReader(System.in));
    System.out.print("input (a):");
    int a = Integer.parseInt(reader.readLine());
    System.out.print("input (b):");
    int b = Integer.parseInt(reader.readLine());
    System.out.println("aの値は" + a);
    System.out.println("bの値は" + b);
  }
}
```

コンソールに"input (b):"という文字列を出力し
コンソールからの整数の入力をint型変数bに代入する

```
package j1.lesson03;
import java.io.*;

public class EchoTwoInt {
  public static void main(String[] args) throws IOException {
    BufferedReader reader
      = new BufferedReader(new InputStreamReader(System.in));
    System.out.print("input (a):");
    int a = Integer.parseInt(reader.readLine());
    System.out.print("input (b):");
    int b = Integer.parseInt(reader.readLine());
    System.out.println("aの値は" + a);
    System.out.println("bの値は" + b);
  }
}
```

int型変数a, b に格納されている値を出力

```
package j1.lesson03;
import java.io.*;

public class EchoTwoInt {
  public static void main(String[] args) throws IOException {
    BufferedReader reader
      = new BufferedReader(new InputStreamReader(System.in));
    System.out.print("input (a):");
    int a = Integer.parseInt(reader.readLine());
    System.out.print("input (b):");
    int b = Integer.parseInt(reader.readLine());
    System.out.println("aの値は" + a);
    System.out.println("bの値は" + b);
  }
}
```

コンソールでの実行画面
区別のためユーザの入力部分を斜体にしてある

```
input (a):100
input (b):200
aの値は100
bの値は200
```

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

実数型の入力、出力の例をトレースしてみよう

```
package j1.lesson03;
import java.io.*;

public class EchoDouble {
  public static void main(String[] args) throws IOException {
    BufferedReader reader
      = new BufferedReader(new InputStreamReader(System.in));
    System.out.print("実数をひとつ入力してください:");
    double input = Double.parseDouble(reader.readLine());
    System.out.println("入力された実数は" + input);
  }
}
```

入力のための準備

```
package j1.lesson03;
import java.io.*;

public class EchoDouble {
  public static void main(String[] args) throws IOException {
    BufferedReader reader
      = new BufferedReader(new InputStreamReader(System.in));
    System.out.print("実数をひとつ入力してください:");
    double input = Double.parseDouble(reader.readLine());
    System.out.println("入力された実数は" + input);
  }
}
```

コンソールに "実数をひとつ入力してください: " と出力して
コンソールからの実数の入力をdouble型変数inputに代入する

```
package j1.lesson03;
import java.io.*;

public class EchoDouble {
  public static void main(String[] args) throws IOException {
    BufferedReader reader
      = new BufferedReader(new InputStreamReader(System.in));
    System.out.print("実数をひとつ入力してください:");
    double input = Double.parseDouble(reader.readLine());
    System.out.println("入力された実数は" + input);
  }
}
```

double型変数inputに格納されている実数データをコンソールに出力する

```
package j1.lesson03;
import java.io.*;

public class EchoDouble {
  public static void main(String[] args) throws IOException {
    BufferedReader reader
      = new BufferedReader(new InputStreamReader(System.in));
    System.out.print("実数をひとつ入力してください:");
    double input = Double.parseDouble(reader.readLine());
    System.out.println("入力された実数は" + input);
  }
}
```

コンソールでの実行画面
区別のためユーザの入力部分を斜体にしてある

実数をひとつ入力してください：*3.14*
入力された実数は3.14

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。

いろいろな比較演算子

条件式	意味
a == b	aとbが等しければtrue、そうでなければfalse
a != b	aとbが等しくなければtrue、そうでなければfalse
a >= b	aがbより大きいか等しければtrue、そうでなければfalse
a <= b	aがbより小さいか等しければtrue、そうでなければfalse
a > b	aがbより大きいければtrue、そうでなければfalse
a < b	aがbより小さければtrue、そうでなければfalse

例

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

もし (a が 0 以上) ならば、b に a * 3 を代入する

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

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;
}
```

次のプログラムをトレースしてみよう

```
package j1.lesson03;
import java.io.*;

public class Umbrella {
  public static void main(String[] args) throws IOException {
    BufferedReader reader
      = new BufferedReader(new InputStreamReader(System.in));
    System.out.print("降水確率を入力してください:");
    int chanceOfRain = Integer.parseInt(reader.readLine());
    System.out.println("降水確率は" + chanceOfRain + "% です。");
    if (chanceOfRain >= 50)
      System.out.println("傘を忘れずに。");
    else System.out.println("傘はいりません。");
    System.out.println("いってらっしゃい。");
  }
}
```

入力のための準備

```
package j1.lesson03;
import java.io.*;

public class Umbrella {
  public static void main(String[] args) throws IOException {
    BufferedReader reader
      = new BufferedReader(new InputStreamReader(System.in));
    System.out.print("降水確率を入力してください:");
    int chanceOfRain = Integer.parseInt(reader.readLine());
    System.out.println("降水確率は" + chanceOfRain + "% です。");
    if (chanceOfRain >= 50)
      System.out.println("傘を忘れずに。");
    else
      System.out.println("傘はいりません。");
    System.out.println("いってらっしゃい。");
  }
}
```

コンソールに"降水確率を入力してください"と出力し
コンソールから整数の入力をint型変数chanceOfRainに代入

```
package j1.lesson03;
import java.io.*;

public class Umbrella {
  public static void main(String[] args) throws IOException {
    BufferedReader reader
      = new BufferedReader(new InputStreamReader(System.in));
    System.out.print("降水確率を入力してください:");
    int chanceOfRain = Integer.parseInt(reader.readLine());
    System.out.println("降水確率は" + chanceOfRain + "% です。");
    if (chanceOfRain >= 50)
      System.out.println("傘を忘れずに。");
    else
      System.out.println("傘はいりません。");
    System.out.println("いってらっしゃい。");
  }
}
```

int型変数chanceOfRainの値をコンソールに出力する(前後に飾りつき)

```
package j1.lesson03;
import java.io.*;

public class Umbrella {
  public static void main(String[] args) throws IOException {
    BufferedReader reader
      = new BufferedReader(new InputStreamReader(System.in));
    System.out.print("降水確率を入力してください:");
    int chanceOfRain = Integer.parseInt(reader.readLine());
    System.out.println("降水確率は" + chanceOfRain + "% です。");
    if (chanceOfRain >= 50)
      System.out.println("傘を忘れずに。");
    else
      System.out.println("傘はいりません。");
    System.out.println("いってらっしゃい。");
  }
}
```

chanceOfRainの値が50以上なら "傘を忘れずに" そうでなければ"傘はいりません。" を出力

```
package j1.lesson03;
import java.io.*;

public class Umbrella {
  public static void main(String[] args) throws IOException {
    BufferedReader reader
      = new BufferedReader(new InputStreamReader(System.in));
    System.out.print("降水確率を入力してください:");
    int chanceOfRain = Integer.parseInt(reader.readLine());
    System.out.println("降水確率は" + chanceOfRain + "% です。");
    if (chanceOfRain >= 50)
      System.out.println("傘を忘れずに。");
    else
      System.out.println("傘はいりません。");
    System.out.println("いってらっしゃい。");
  }
}
```

int型変数chanceOfRainの値に関係なく実行

```
package j1.lesson03;
import java.io.*;

public class Umbrella {
  public static void main(String[] args) throws IOException {
    BufferedReader reader
      = new BufferedReader(new InputStreamReader(System.in));
    System.out.print("降水確率を入力してください:");
    int chanceOfRain = Integer.parseInt(reader.readLine());
    System.out.println("降水確率は" + chanceOfRain + "% です。");
    if (chanceOfRain >= 50)
      System.out.println("傘を忘れずに。");
    else
      System.out.println("傘はいりません。");
    System.out.println("いってらっしゃい。");
  }
}
```

if文の連鎖
100以上の値は警告する

```
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 (chanceOfRain >= 50)
  System.out.println("傘を忘れずに。");
else
  System.out.println("傘はいりません。");
```

if文の連鎖
負の値も警告する

```
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。");
```

一緒に試してみよう

- これから入力、条件分岐を使ったプログラムを作成します。

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. week03.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 zip 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

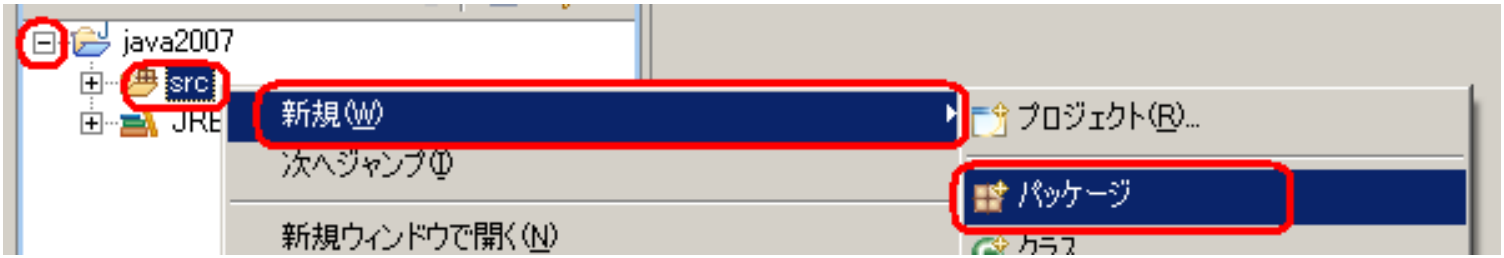
- プロジェクト「java20XX」の中の「test」というフォルダに「j1.lesson03」という名前のファイルが作成される。
- このファイルには今週使用するテスト一式が記述されている。

パッケージ

- 今回は j1.lesson03 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



クラスの作成
クラス = プログラム　と思ってよい

1. 先ほど作成したパッケージ「j1.lesson03」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 Adder とする。

実際のソースファイル

- この教室でパッケージ j1.lesson03 に Adder クラスを作成すると

  U:¥Eclipse3.1¥Windows¥java20XX¥src¥j1¥lesson03¥Adder.java

  というファイルができる

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.lesson03;
import java.io.*;

public class Adder {
  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.lesson03.xml を右クリック
2. 「ファイル」メニューの中にある「Adderに対する骨格テスト」を選択し実行する

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.lesson03 に Adder 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の指定を行っていない。

mainメソッドの中身を書く
このプログラムの意味が分かりますか

```
package j1.lesson03;
import java.io.*;

public class Adder {
  public static void main(String[] args) throws IOException {
    BufferedReader reader
      = new BufferedReader(new InputStreamReader(System.in));
    // 一つ目の整数を入力させる
    System.out.print("一つ目の整数を入力:");
    int input1 = Integer.parseInt(reader.readLine());
    // 二つ目の整数を入力させる
    System.out.print("二つ目の整数を入力:");
    int input2 = Integer.parseInt(reader.readLine());
    System.out.println("合計は:" + (input1 + input2));
  }
}
```

プログラムの実行

1. 「Adder.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ウィンドウの「ファイル」メニューの中にある「Adderに対する機能テスト」を選択し実行
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

条件分岐を行うプログラム

1. 先ほど作成したパッケージ「j1.lesson03」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 Sign 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.lesson03;
import java.io.*;

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

  }
}
```

ここで骨格テストを実施する。

変数の初期化と宣言を分けてもよい
以下の2つは同等

```
// 一つ目の整数を入力させる
System.out.print("一つ目の整数を入力:");
int input1 = Integer.parseInt(reader.readLine());
// 二つ目の整数を入力させる
System.out.print("二つ目の整数を入力:");
int input2 = Integer.parseInt(reader.readLine());
```

```
int input1, input2;
// 一つ目の整数を入力させる
System.out.print("一つ目の整数を入力:");
input1 = Integer.parseInt(reader.readLine());
// 二つ目の整数を入力させる
System.out.print("二つ目の整数を入力:");
input2 = Integer.parseInt(reader.readLine());
```

mainメソッドの中身を書く
ひとつの命令文を書く度に保存(自動コンパイル)

```
package j1.lesson03;
import java.io.*;

public class Sign {
  public static void main(String[] args) throws IOException {
    BufferedReader reader
      = new BufferedReader(new InputStreamReader(System.in));
    // 実数を入力させる
    System.out.print("値を入力:");
    double input = Double.parseDouble(reader.readLine());
    if (input == 0) {
      System.out.println("入力された値は0");
    } else if (input > 0) {
      System.out.println("入力された値は正の値");
    } else {
      System.out.println("入力された値は負の値");
    }
  }
}
```

実行してみよう
続けて機能テスト

- コンソールから入力を適切に行い
- 結果が期待通りになるか確認
- コンソール入力をいろいろ変えてプログラムが正しいことを検証しよう。どうすればよいテストになるだろうか。
- 機能テストはその点をきちんとやるために用意されている。

課題

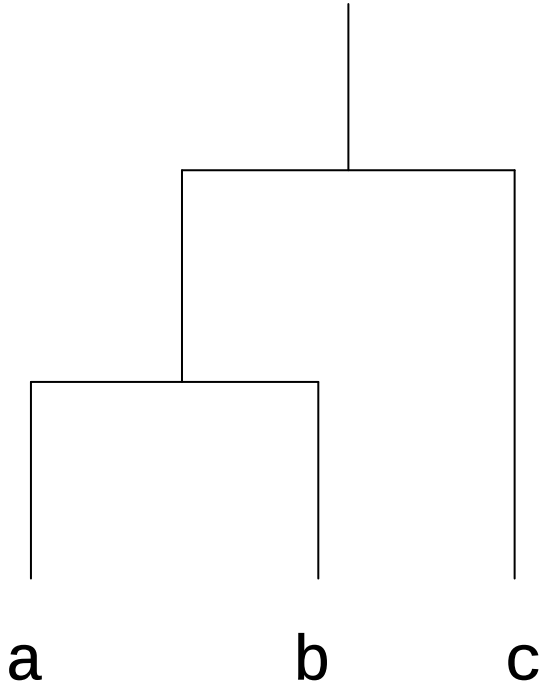
各自のペースで
「第03週目の課題」をやってみよう

1. http://java.cis.k.hosei.ac.jp/ をブラウザで開く
2. 第03回 – Conditional Branch (if-else) - をクリック
3. 第03週目課題をクリック

課題0302のヒント

```
a, b, c に入力

if(a >= b){
    if(a >= c)
        aが最大
    else
        cが最大
}
else if (b >= c)
    bが最大
else
    cが最大
```



課題0303のヒント 芸がないが素朴な書き方

```
if(確率>100){

}else if(確率>=50){

}else if(確率>=20){

}else if(確率>=0){

}else{

}
```