

Project.1
「Javaプログラミング入門」

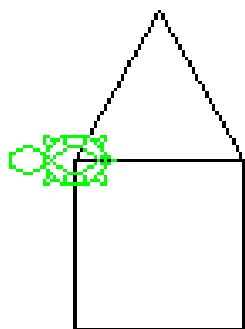
學習事項

学習事項

- プログラムについて
 - 2.2.2.1 ブロック
 - 2.2.2.2 クラス
 - 2.2.2.3 メソッド
- 命令
 - 2.2.2.4 命令
- プログラムの工夫
 - 2.2.2.5 命令の区切り
 - 2.2.2.6 コメント

プログラムについて

Java プログラム



```
⊖ /*
   * 家を書く プログラム
   */
   クラスブロック

public class House extends Turtle {

   // 起動処理
   メソッドブロック1
   public static void main(String[] args) {
       Turtle.startTurtle(new House());
   }

   // タートルを動かす処理
   ⊖ void start() {
       メソッドブロック2

       // 屋根を書く
       rt(30); // 30度右を向く
       fd(50); // 50歩前に進む
       rt(120);
       fd(50);
       rt(120);
       fd(50);

       // 本体を書く
       lt(90);
       fd(50);
       lt(90);
       fd(50);
       lt(90);
       fd(50);
       lt(90);
       fd(50);
       lt(90);
       fd(50);

   }
}
```

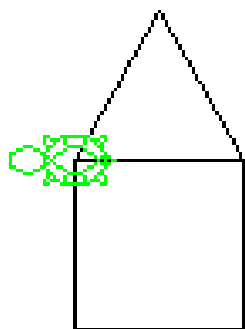
プログラムの解説1

- **ブロック**・・・中括弧{と}で囲まれた部分
 - Javaプログラムは複数のブロックから構成される
 - ブロックの中にブロックを入れることができる
- **クラス**・・・Javaプログラムの一つの単位
 - (本格的なプログラムはクラスが複数になるが)授業では一つだけ
 - **public class [クラス名] extends Turtle{** の括弧から、対応する(ファイルの最後にある)括弧 **}**までを**クラスブロック**
 - Javaのプログラムは全て**クラスブロック**の中に書く

プログラムの解説2

- **メソッド**・・・命令をまとめて一つに束ねたもの
 - (1学期は) **void メソッド名 ()**{から始まるブロックのことを**メソッドブロック**という
 - プログラムの命令は, 全て**メソッドブロック**の中に書く
 - **メソッドブロック**はクラスブロックの中に書く必要がある
 - しばらくすると複数のメソッドを扱うが, 1学期はプログラムは**startメソッド**(**void start()**{})の括弧の中に書くと覚えておけばよい

Java プログラム



```
⊖ /*
   * 家を書く プログラム
   */
   クラスブロック

public class House extends Turtle {

   // 起動処理
   メソッドブロック1
   public static void main(String[] args) {
       Turtle.startTurtle(new House());
   }

   // タートルを動かす処理
   ⊖ void start() {
       メソッドブロック2

       // 屋根を書く
       rt(30); // 30度右を向く
       fd(50); // 50歩前に進む
       rt(120);
       fd(50);
       rt(120);
       fd(50);

       // 本体を書く
       lt(90);
       fd(50);
       lt(90);
       fd(50);
       lt(90);
       fd(50);
       lt(90);
       fd(50);
       lt(90);
       fd(50);

   }
}
```

命令

プログラムの解説3

● 命令

- startメソッドブロックの中にある, rt(〇〇)やfd(〇〇)がタートルに対する指示・命令

- ルール
メソッド内に書かれた命令は,
必ず上から順番に実行される
(**順次実行**)

```
> void start() {↓  
↓  
> // 屋根を書く↓  
> rt(30); // 30度右を向く↓  
> fd(50); // 50歩前に進む↓  
↓  
> rt(120);↓  
> fd(50);↓  
> rt(120);↓  
> fd(50);↓  
↓  
> // 本体を書く↓  
> lt(90);↓  
> fd(50);↓  
> lt(90);↓  
> fd(50);↓  
> lt(90);↓  
> fd(50);↓  
> lt(90);↓  
> fd(50);↓  
> }↓  
,
```

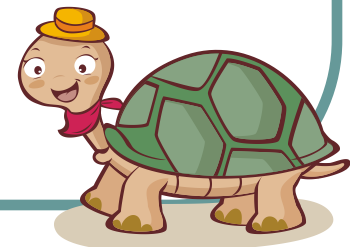


プログラムの解説4

- タートルへの基本命令

- rt(〇〇) タートルを〇〇度右に回らせる
- lt (〇〇) タートルを〇〇度左に回らせる
- fd(〇〇) タートルを〇〇歩前に進ませる
- bk(〇〇) タートルを〇〇歩後に進ませる

- up() タートルがペンを上げる(軌跡を書かなくなる)
- down() タートルがペンを下げる(軌跡を書く)



プログラムの解説5

● 命令の区切り

- Javaのプログラムでは命令と命令の区切りに必ず「;」(セミicolon)を入れる
- これを入れないと、コンピュータがどこで命令が区切られているのか、理解できない
- ちなみにセミicolonが入っていれば、複数の命令を以下のように同じ行に書けるが非推奨！

悪い例) lt(30); fd(50);

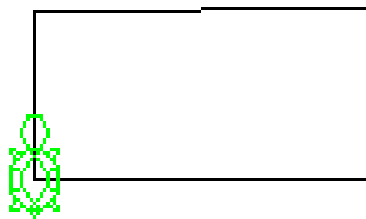
Square.java
を見てみよう！



例題：1 (Rectangle.java)

例題: 1

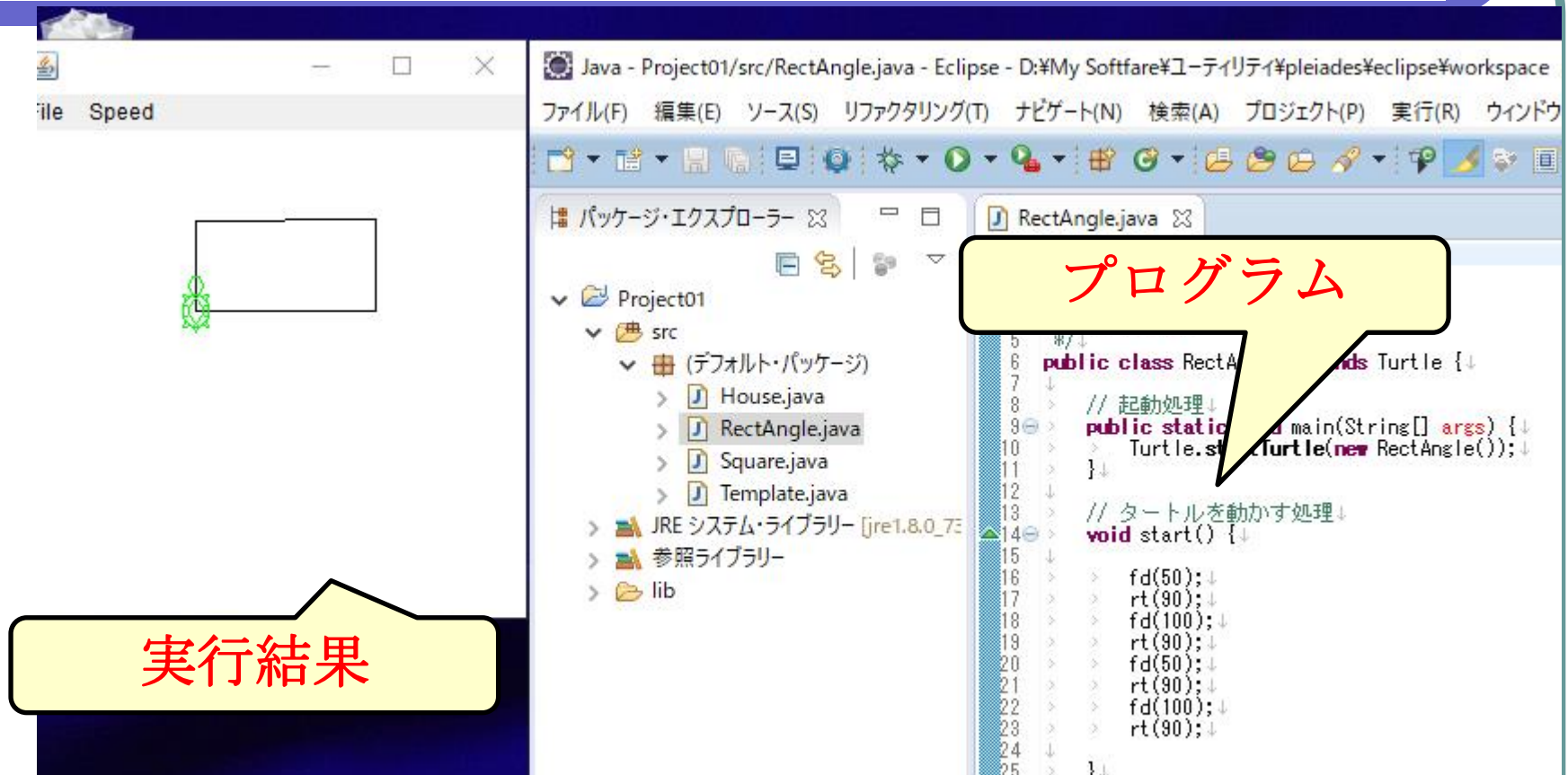
- 長方形を書くプログラム (Rectangle.java) を作りなさい



- 多少直線が曲がっているのは仕様

次ページの答えを見る前に自分で考えてみよう!

解答例



演習課題の画像を提出するときは
「実行結果」と「プログラム」
両方が写るようにすること！

プログラミングの工夫

プログラムの解説6

● 文字

- プログラム内は**全て半角文字**で記述し、大文字と小文字に注意
(※コメント部分は日本語OK)
- **ファイル名**も**全て半角文字**で、ファイル名先頭と単語の**区切りに大文字**を使う

例

Square.java

BigHouse.java

プログラムの解説7

● コメント

- コンピュータにプログラム(命令)として考慮されず, 読み飛ばされる部分
- 人間が読めないとメンテナンスに困るためコメントをつける
- 例えばファイルの先頭には, プログラムのタイトル, 名前や日付を書く

範囲指定コメント

- **/*から*/**までコメントになります

行コメント

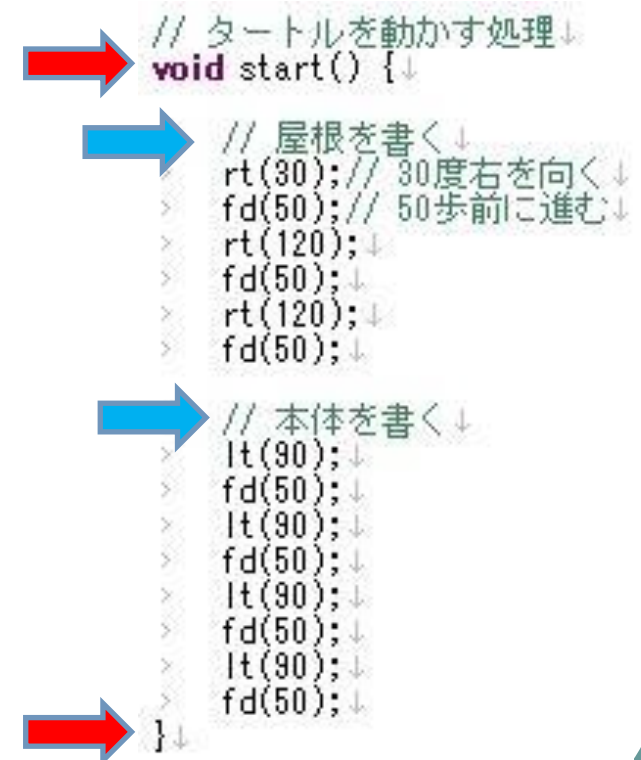
- **//**から行の終わりまでコメントになります

プログラムの解説8

● インデント

- プログラムに空白を挿入して意味のまとまりごとに字下げを行うこと
- 空白はスペースキーではなくtabキーを使った方がきれいに入る

始まりの文字の位置で意味を変えている



```
// タートルを動かす処理↓  
void start() {  
    // 屋根を書く↓  
    rt(30); // 30度右を向く↓  
    > fd(50); // 50歩前に進む↓  
    > rt(120);  
    > fd(50);  
    > rt(120);  
    > fd(50);  
    // 本体を書く↓  
    > lt(90);  
    > fd(50);  
    > lt(90);  
    > fd(50);  
    > lt(90);  
    > fd(50);  
    > lt(90);  
    > fd(50);  
}
```

The diagram illustrates the use of indentation in a Java program. A red arrow points to the start of the `void start() {` line. A blue arrow points to the first line of the roof-drawing block (`// 屋根を書く↓`). Another blue arrow points to the first line of the body-drawing block (`// 本体を書く↓`). A final red arrow points to the closing brace `}` of the `start` method.

House.java
を見てみよう！

演習課題

演習について

- 演習課題の解答について

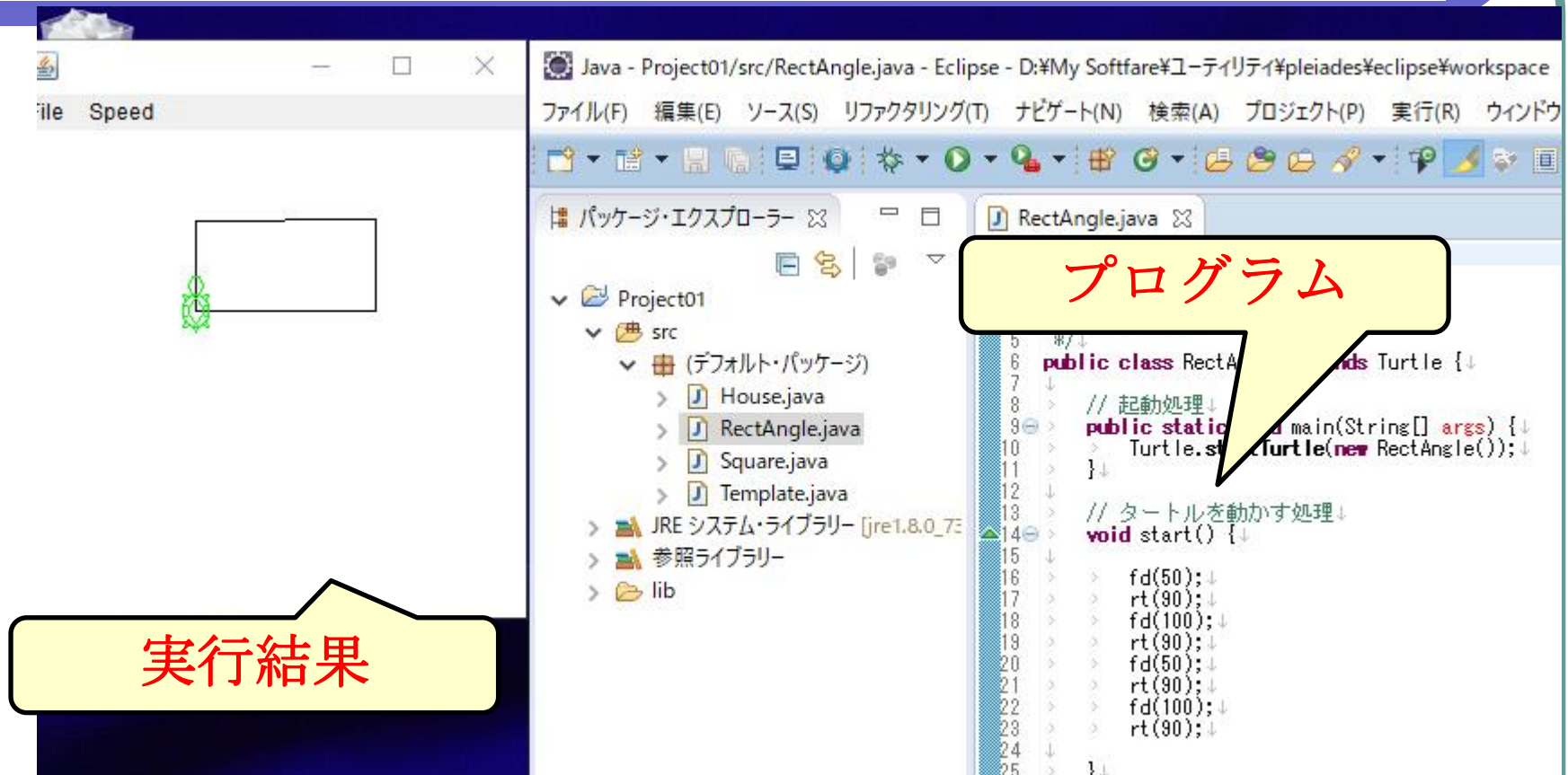
- 「実行結果」と「プログラム」の両方を見せる

- 課題用プログラムの作成方法

- **Template.java**を右クリック→コピー
- その場で右クリック→貼り付け
- 名前の競合→(該当の演習課題の)名前を付ける

授業用サイトに
コピー方法
載せました

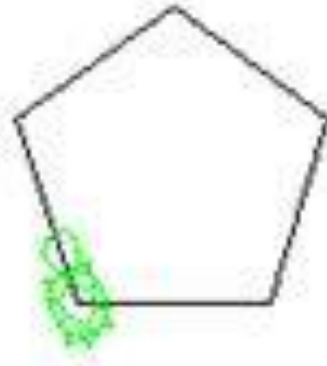
解答例



演習課題を見せるときは
「実行結果」と「プログラム」
両方を提示すること！

練習問題

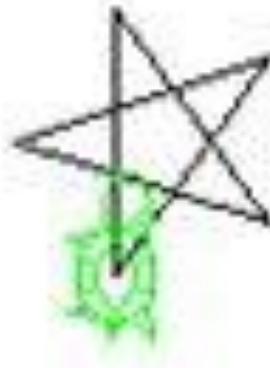
- 五角形を書くプログラム (Pentagon.java) を作りなさい



- Template.javaをコピーして作成する
- 図形の向き(角度)も揃える
- プログラムには適切なコメントを挿入してみよう

練習問題

- 星を描くプログラム (Star.java) を作りなさい



- Template.javaをコピーして作成する
- 図形の向き(角度)も揃える
- プログラムには適切なコメントを挿入してみよう

次回予告

次回予告

- Project.1 「はじめてのJavaプログラミング」
- Project.2 「コンピュータに情報を記憶させてみよう(変数)」

クリアファイルを
提出して帰ること！

以上

NEXT
TIME

おつかれさまでした。
それでは、また次回！！

