

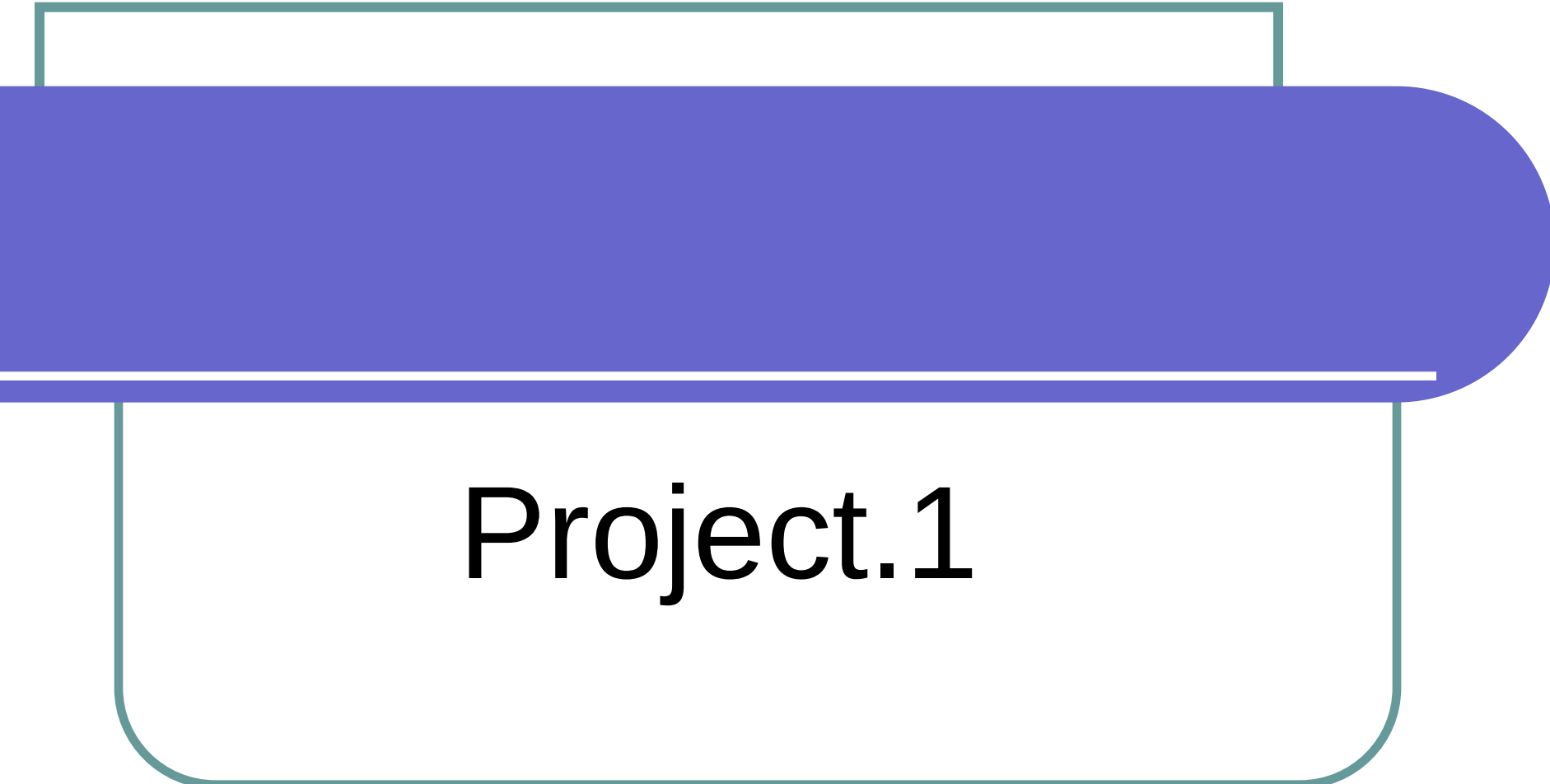
模範解答



Project.0

【略】

※ プログラムなし・・・

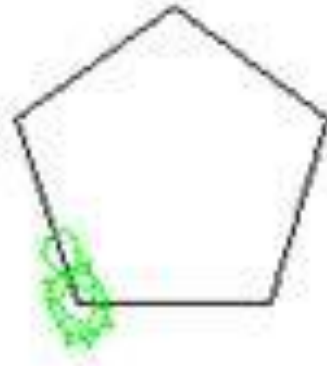


Project.1

【基本問題】

練習問題

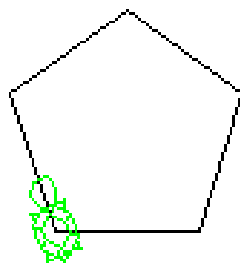
- 五角形を書くプログラム (Pentagon.java) を作りなさい



- 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



File Speed



// タートルを動かす処理

void start() {

// 平らにする

lt(18);

// 五角形を描く

fd(50);

rt(72);

fd(50);

rt(72);

fd(50);

rt(72);

fd(50);

rt(72);

fd(50);

rt(72);

}

練習問題

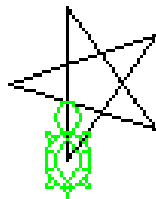
- 星を描くプログラム (Star.java) を作りなさい



- 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



File Speed



```
void start() {
```

```
    //星を描く
```

```
    fd(50);
```

```
    rt(144);
```

```
    fd(50);
```

```
    rt(144);
```

```
    fd(50);
```

```
    rt(144);
```

```
    fd(50);
```

```
    rt(144);
```

```
    fd(50);
```

```
    rt(144);
```

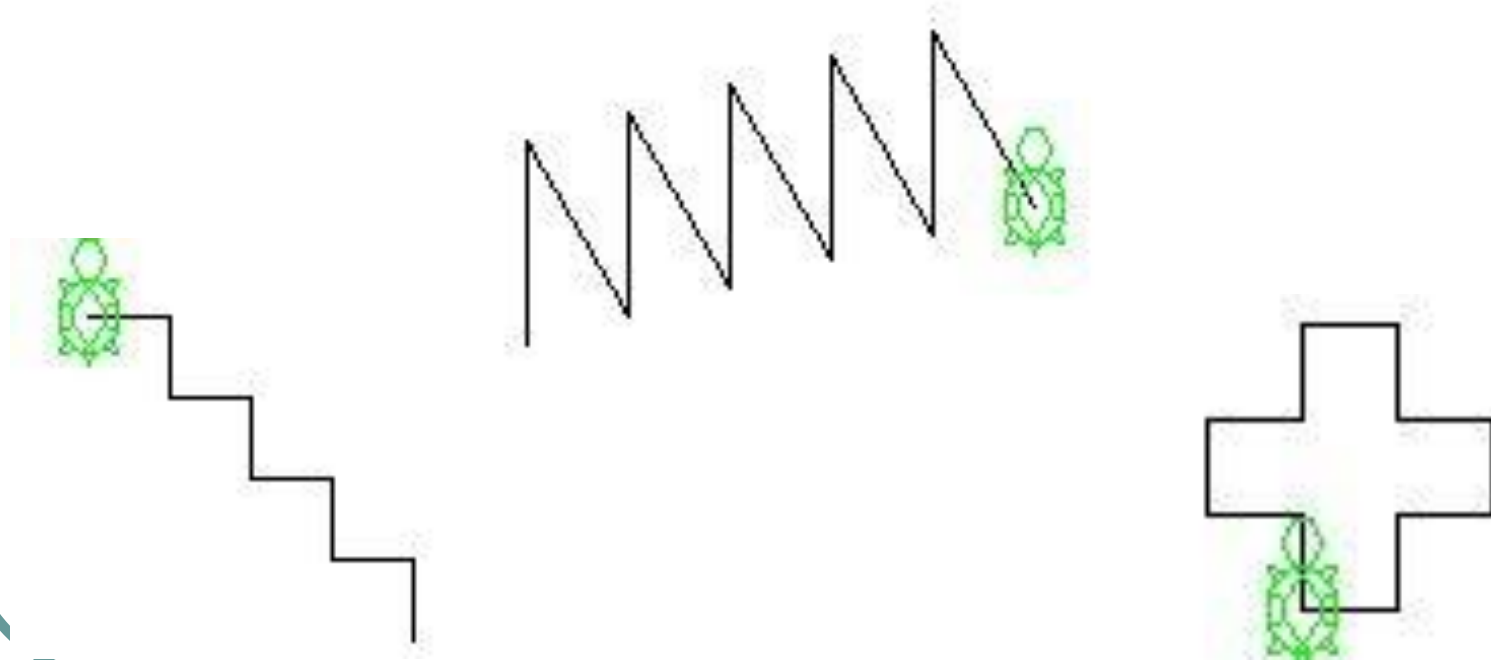
```
}
```

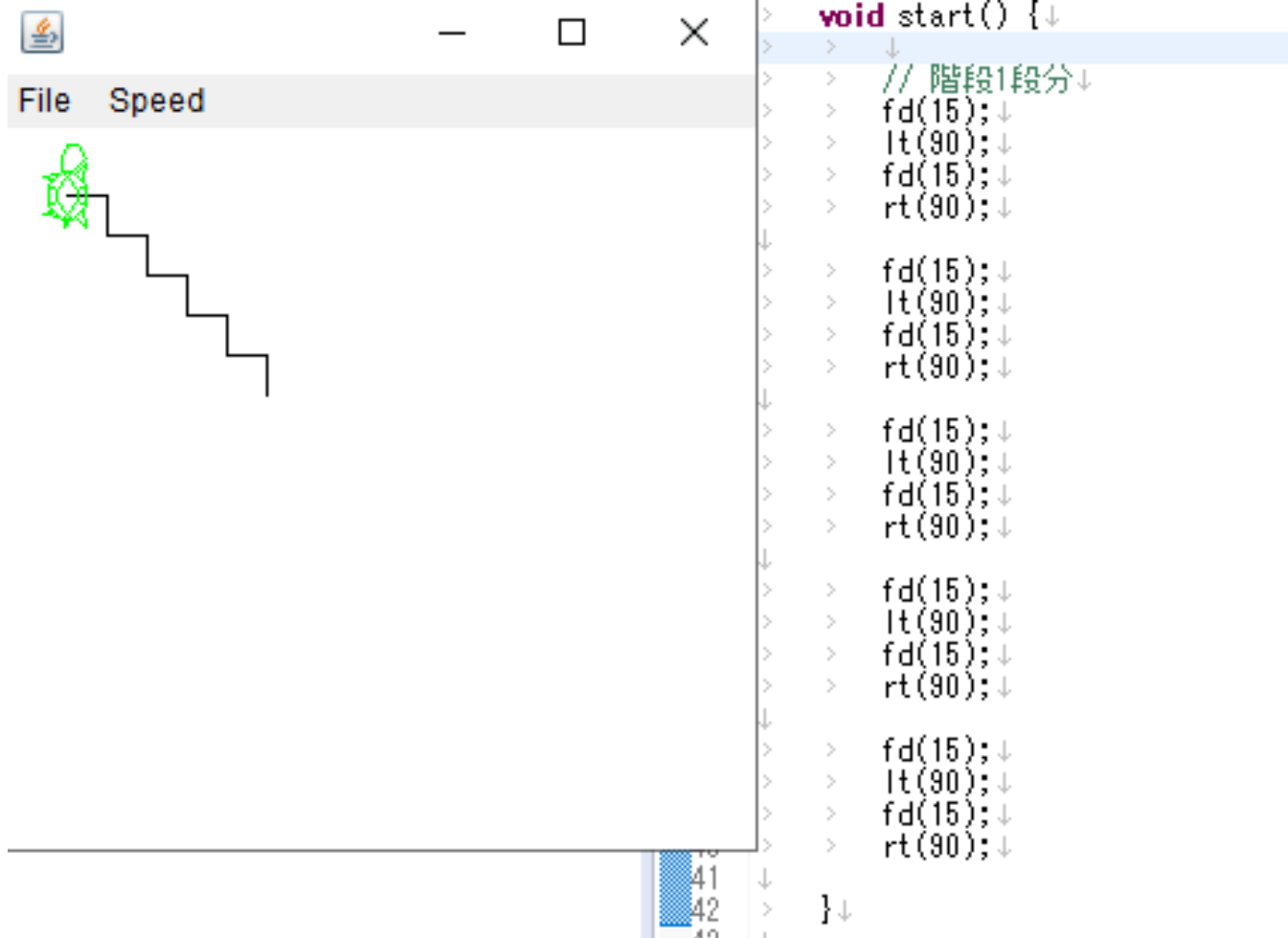
```
}
```

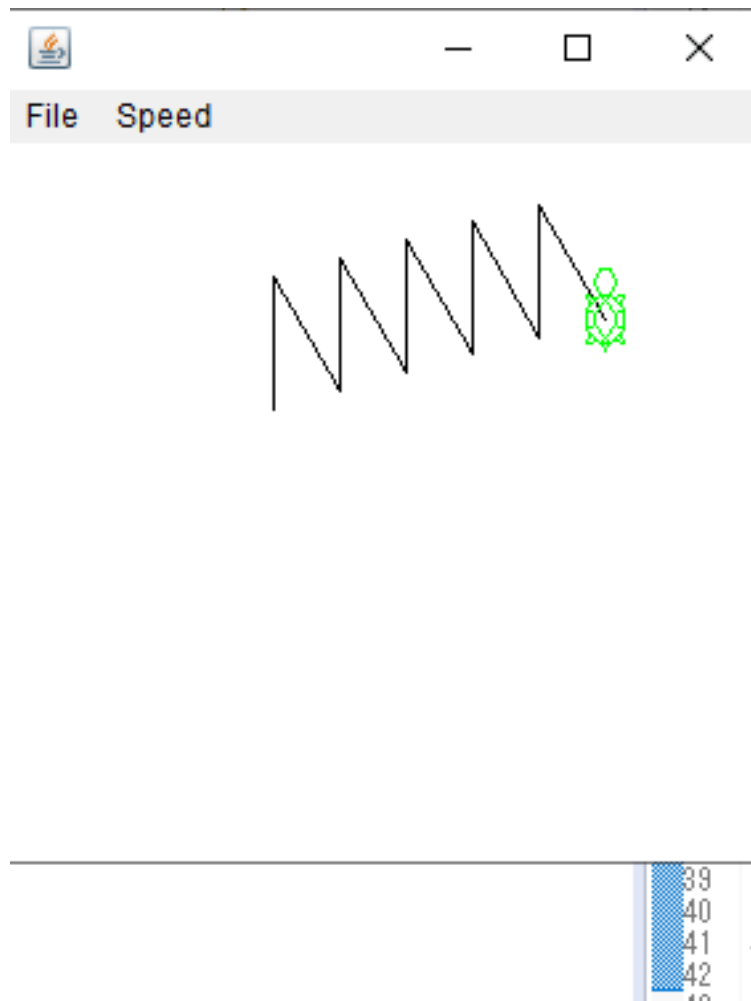
【発展問題】

練習問題

- 練習問題 2.1
 - 色々な図形を描いてみましょう①







// タートルを動かす処理↓

void start() {↓

> // ギザ1段分↓

> fd(50);↓

> rt(150);↓

> fd(50);↓

> lt(150);↓

> fd(50);↓

> rt(150);↓

> fd(50);↓

> lt(150);↓

> fd(50);↓

> rt(150);↓

> fd(50);↓

> lt(150);↓

> fd(50);↓

> rt(150);↓

> fd(50);↓

> lt(150);↓

> fd(50);↓

> rt(150);↓

> fd(50);↓

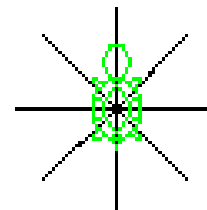
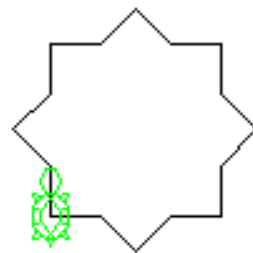
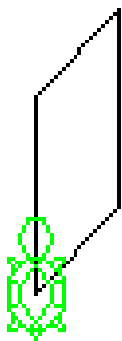
> lt(150);↓

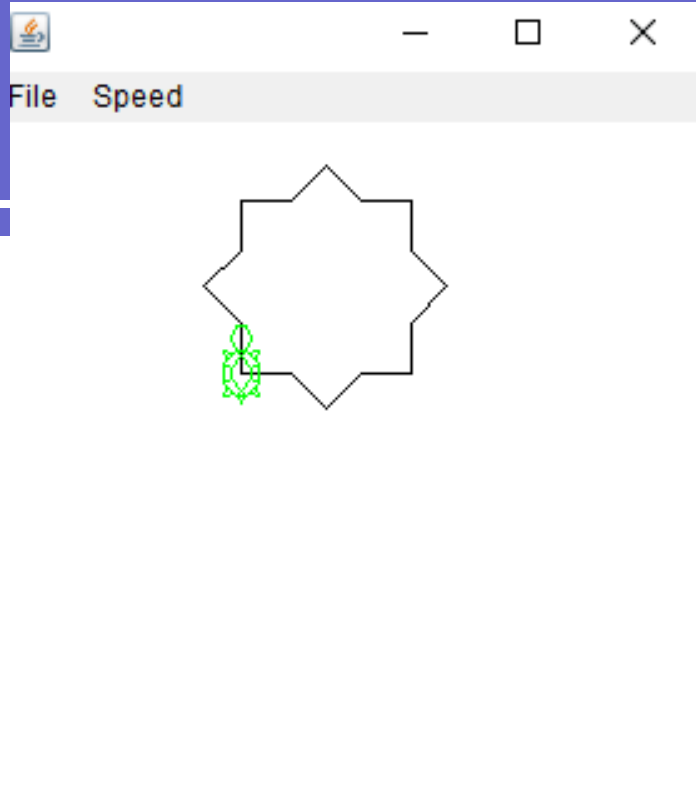
}↓

```
// タートルを動かす処理↓
void start() {↓
> // 十字1角分↓
> fd(20);↓
> lt(90);↓
> fd(20);↓
> rt(90);↓
> fd(20);↓
> rt(90);↓
> fd(20);↓
> lt(90);↓
> fd(20);↓
> rt(90);↓
> fd(20);↓
> rt(90);↓
> fd(20);↓
> lt(90);↓
> fd(20);↓
> rt(90);↓
> fd(20);↓
> rt(90);↓
> fd(20);↓
> lt(90);↓
> fd(20);↓
> rt(90);↓
> fd(20);↓
> rt(90);↓
}↓
```

練習問題

- 練習問題 2.1
 - 色々な図形を描いてみましょう②





```
// タートルを動かす処理↓
```

```
void start() {↓
```

```
> // 金平糖 1辺分↓
```

```
> fd(20);↓
```

```
> lt(45);↓
```

```
> fd(20);↓
```

```
> rt(90);↓
```

```
> fd(20);↓
```

```
> lt(45);↓
```

```
> fd(20);↓
```

```
> rt(90);↓
```

```
> fd(20);↓
```

```
> lt(45);↓
```

```
> fd(20);↓
```

```
> rt(90);↓
```

```
> fd(20);↓
```

```
> lt(45);↓
```

```
> fd(20);↓
```

```
> rt(90);↓
```

```
> fd(20);↓
```

```
> lt(45);↓
```

```
> fd(20);↓
```

```
> rt(90);↓
```

```
> fd(20);↓
```

```
> lt(45);↓
```

```
> fd(20);↓
```

```
> rt(90);↓
```

```
> > fd(20);↓
```

```
> > lt(45);↓
```

```
> > fd(20);↓
```

```
> > rt(90);↓
```

```
> > fd(20);↓
```

```
> > lt(45);↓
```

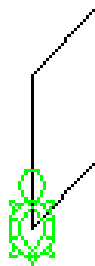
```
> > fd(20);↓
```

```
> > rt(90);↓
```

```
↓  
> }↓
```



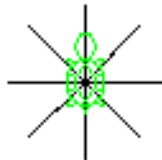
File Speed



```
↓> // タートルを動かす処理↓
> void start() {↓
↓> > // 平行四辺形↓
> > fd(50);↓
> > rt(45);↓
> > fd(30);↓
> > rt(135);↓
↓> > fd(50);↓
> > rt(45);↓
> > fd(30);↓
> > rt(135);↓
↓> }↓
↓> }
```



File Speed



// タートルを動かす処理↓

void start() {↓

> // アスタリスク ループ1↓

> fd(30);↓

> bk(30);↓

> rt(45);↓

> fd(30);↓

> bk(30);↓

> rt(45);↓

> fd(30);↓

> bk(30);↓

> rt(45);↓

> fd(30);↓

> bk(30);↓

> rt(45);↓

> fd(30);↓

> bk(30);↓

> rt(45);↓

> fd(30);↓

> bk(30);↓

> rt(45);↓

> fd(30);↓

> bk(30);↓

> rt(45);↓

> fd(30);↓

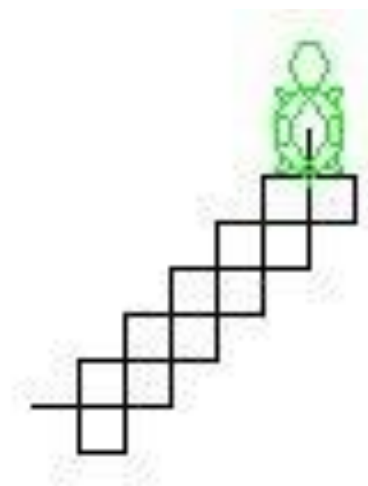
> bk(30);↓

> rt(45);↓

}↓

練習問題

- 練習問題 2.1
 - 色々な図形を描いてみましょう③



```
void start() {↓
```

```
> fd(10); ↓
```

```
> fd(20); ↓
> fd(20); ↓
```

 $\tau(30); \downarrow$

```
> rt(90); ↓
```

```
> bk(20); ↓
+ 11/00; ↓
```

```
> fd(10): ↓
```

```
> fd(20); ↓
```

```
> lt(90); ↓
```

```
> rt(90); ↓
```

```
> BK(20); ↓
> It(90); ↓
```

```
> fd(10); ↓
```

```
> fd(20); ↓
```

```
> fit(90); ↓
```

```
> rt(90); ↓
```

```
> bk(20); ↓
```

 $f_d(10) = 1$

```
> fd(20); ↓
```

```
> lt(90); ↓
```

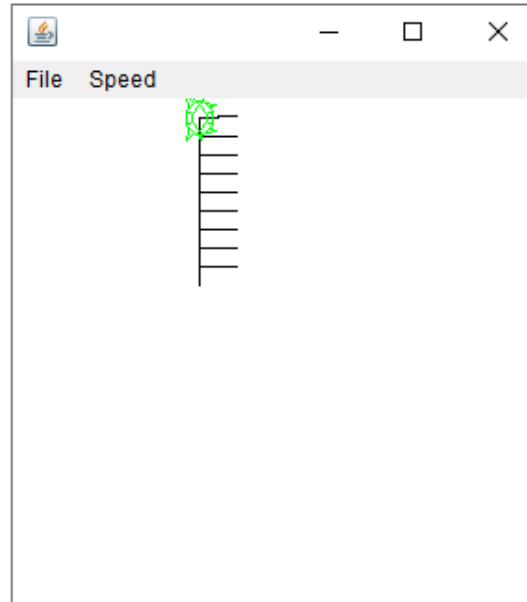
```
> rt(90); ↓
```

```
> bk(20); ↓
> lt(90); ↓
```

```
> fd(10):↓
```

```
> fd(20); ↓
```

```
> lt(90); ↓
```



```
// タートルを動かす処理↓
```

```
void start() {↓
```

```
> //四角形の帯↓
```

```
> rt(90);↓
```

```
> fd(20);↓
```

```
> rt(90);↓
```

```
> fd(10);↓
```

```
> rt(90);↓
```

```
> fd(10);↓
```

```
> rt(90);↓
```

```
> fd(20);↓
```

```
> rt(90);↓
```

```
> fd(20);↓
```

```
> rt(90);↓
```

```
> fd(10);↓
```

```
> rt(90);↓
```

```
> fd(10);↓
```

```
> rt(90);↓
```

```
> fd(20);↓
```

```
> rt(90);↓
```

```
> fd(20);↓
```

```
> rt(90);↓
```

```
> fd(10);↓
```

```
> rt(90);↓
```

```
> fd(10);↓
```

```
> rt(90);↓
```

```
> fd(20);↓
```

```
> rt(90);↓
```

```
> fd(20);↓
```

```
> rt(90);↓
```

```
> fd(10);↓
```

```
> rt(90);↓
```

```
> fd(10);↓
```

```
> rt(90);↓
```

```
> fd(20);↓
```

```
> rt(90);↓
```

```
> fd(20);↓
```

```
> rt(90);↓
```

```
> fd(10);↓
```

```
> rt(90);↓
```

```
> fd(10);↓
```

```
> rt(90);↓
```

```
> fd(20);↓
```

```
> rt(90);↓
```

```
> fd(20);↓
```

```
> rt(90);↓
```

```
> fd(10);↓
```

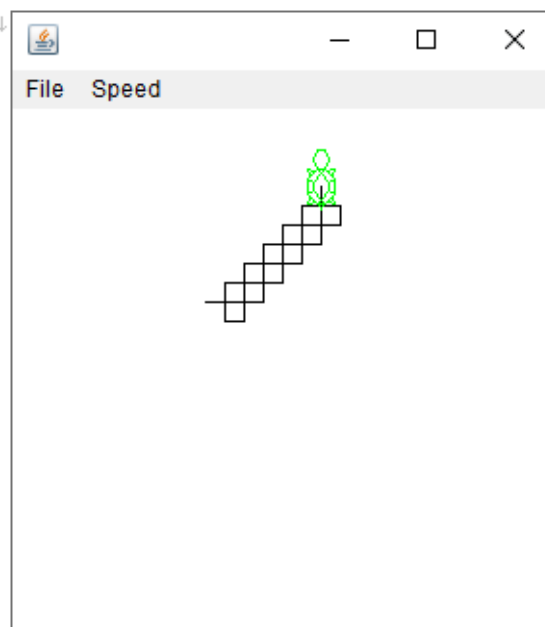
```
> rt(90);↓
```

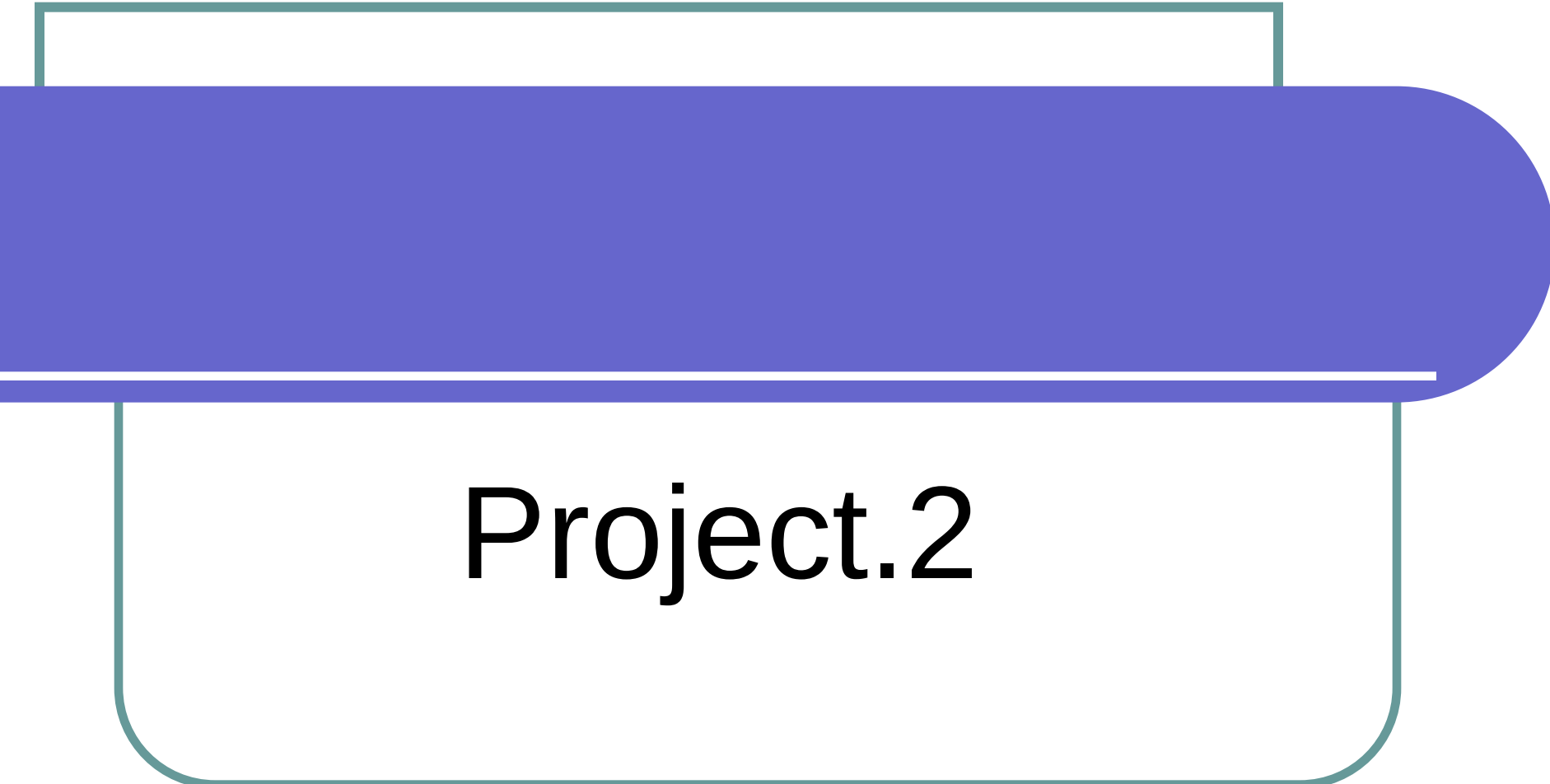
```
> fd(10);↓
```

```
> rt(90);↓
```

```
> fd(20);↓
```

```
}↓
```





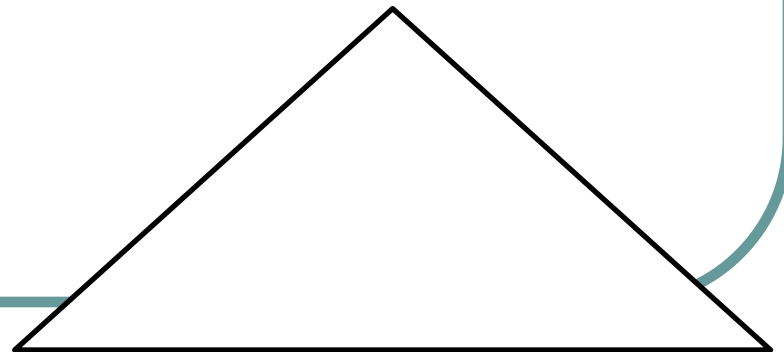
Project.2

【基本問題】

練習問題

- 変数「length」にユーザが値を入力して
自由なサイズの直角二等辺三角形を描くプログラム (InputTriangle.java) を作りなさい
 - 実行すると、「1辺の長さを入力してください」と表示
 - ユーザーが数字を入力するとその大きさの図形を描く
 - 適切なコメントを入力して「読みやすいプログラム」にする
 - 図形の向き(角度)も揃える

例) Example4.java
を参考にしよう！



Speed



// タートルを動かす処理↓

public void start() {↓

> **double** length; // 1辺の長さを記憶しておくための変数↓

> print("1辺の長さを入力してください");↓

> length = input(); // 1辺の長さに入力された値を代入する↓

> // 直角二等辺三角形を書く↓

> rt(45);↓

> fd(length);↓

> rt(90);↓

> fd(length);↓

> rt(135);↓

> fd(length*Math.sqrt(2)); // 1:1:ルート2なので↓

}↓

ランナー 問題 コンソール

InputTriangle [Java アプリケーション] C:\Program Files\Java\jre1.8.0_73\bin\javaw.exe (20:

Turtle Version: 1.0.5 (2007/12/21)↓

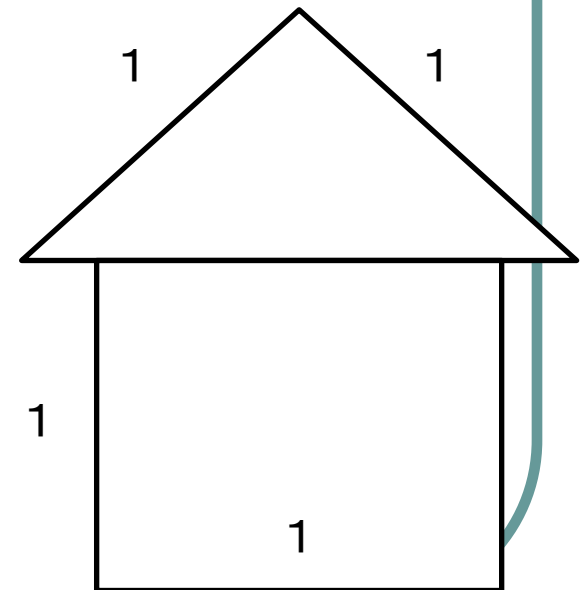
1辺の長さを入力してください↓

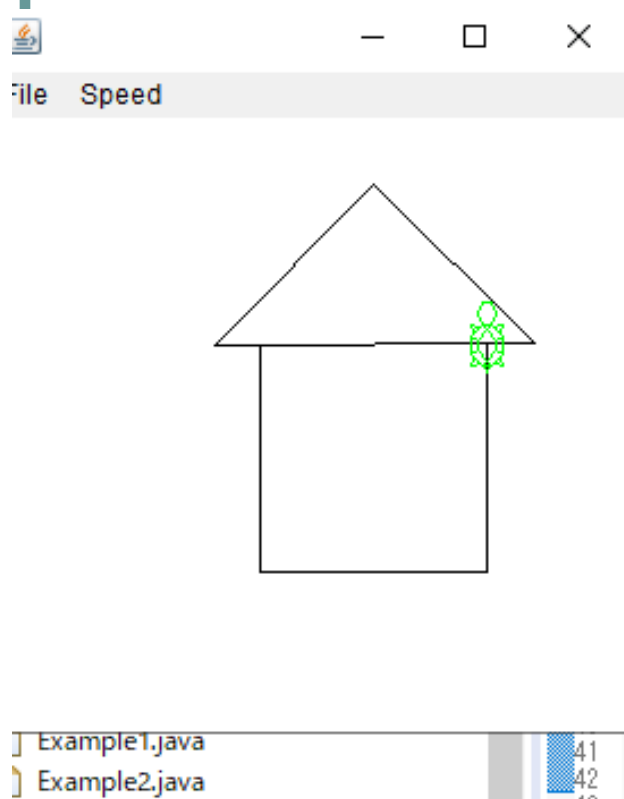
100↓

ト・パッケージ)

練習問題

- 変数「length」にユーザが値を入力して
自由なサイズの家を書くプログラム
(InputHouseEx.java) を作りなさい
 - 屋根は直角二等辺三角形
 - 壁と窓枠は正方形
 - 壁1辺の長さ = 3角形の等辺の長さ





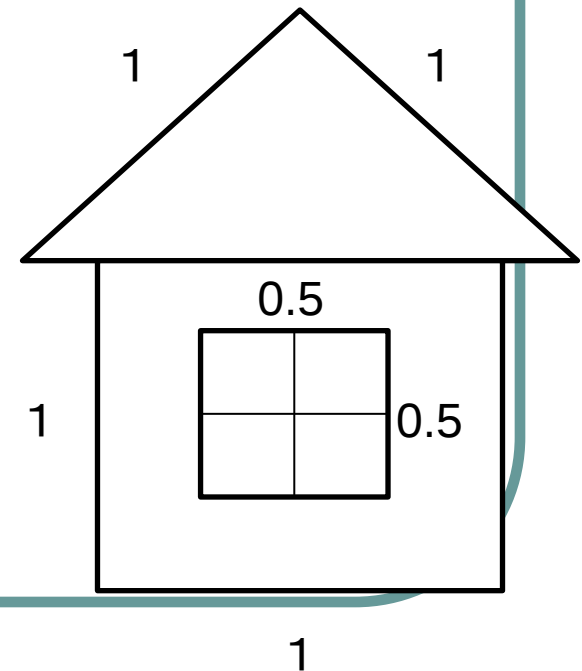
```
> double length;// 1辺の長さを記憶しておくための変数↓
> print("1辺の長さを入力してください");↓
> length = input();// 1辺の長さに入力された値を代入する↓
> // 直角二等辺三角形を描く↓
> rt(45);↓
> fd(length);↓
> rt(90);↓
> fd(length);↓
> rt(135);↓
> fd(length*Math.sqrt(2));//1:1:ルート2なので↓
> //移動↓
> bk((length*Math.sqrt(2)-length)/2);//もしくは length*(Math.sqrt(2)-1)/2↓
> //正方形を描く(1辺は省略)↓
> lt(90);//向きの調整↓
> fd(length);↓
> lt(90);↓
> fd(length);↓
> lt(90);↓
> fd(length);↓
}↓
```

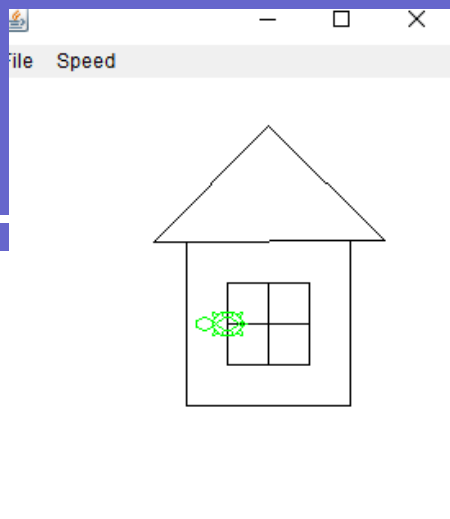
※ 入力画面略

【発展問題】

練習問題

- 家を書くプログラムを改良して窓付きの家 (InputHouseEx1.java) を作りなさい
 - 壁1辺の長さ = 窓枠1辺の長さ $\times 2$
 - 窓枠の中はさらに4つの正方形
 - 描き終わると以下のメッセージを表示
「1辺●●の大きさの家を書きました」





```
public void start() {
    > double length; // 1辺の長さを記憶しておくための変数↓
    > print("1辺の長さを入力してください");
    > length = input(); // 1辺の長さに入力された値を代入する↓

    > // 直角二等辺三角形を描く↓
    > rt(45);
    > fd(length);
    > rt(90);
    > fd(length);
    > rt(135);
    > fd(length*Math.sqrt(2)); // 1:1:ルート2なので↓

    > //移動↓
    > bk((length*Math.sqrt(2)-length)/2); // もしくは length*(Math.sqrt(2)-1)/2↓

    > //正方形を描く(1辺は省略)↓
    > lt(90); // 向きの調整↓

    > fd(length);
    > lt(90);
    > fd(length);
    > lt(90);
    > fd(length);

    > //移動と向きの調整ペンの上げ下げ↓
    > bk(length/4);
    > lt(90);
    > up();
    > fd(length/4);
    > down();

    > /*窓(外枠)を描く↓
    > *もしくは int windowLength=length/2; などと新たに変数を利用しても良い↓
    > */
    > fd(length/2);
    > lt(90);
    > fd(length/2);
    > lt(90);
    > fd(length/2);
    > lt(90);
    > fd(length/2);
    > lt(90);

    > //窓(内枠)を描く↓
    > fd(length/4);
    > lt(90);
    > fd(length/2);
    > lt(90);
    > fd(length/4);
    > lt(90);
    > fd(length/4);
    > lt(90);
    > fd(length/2);

    > print("1辺"+length+"の大きさの家を描きました");
    > }
}
```

ランナー 問題 コンソール

InputTriangle [Java アプリケーション] C:\Program Files\Java\jre1.8.0_73\bin\javaw.exe (2020/05/29 17:20:44)

Turtle Version: 1.0.5 (2007/12/21)

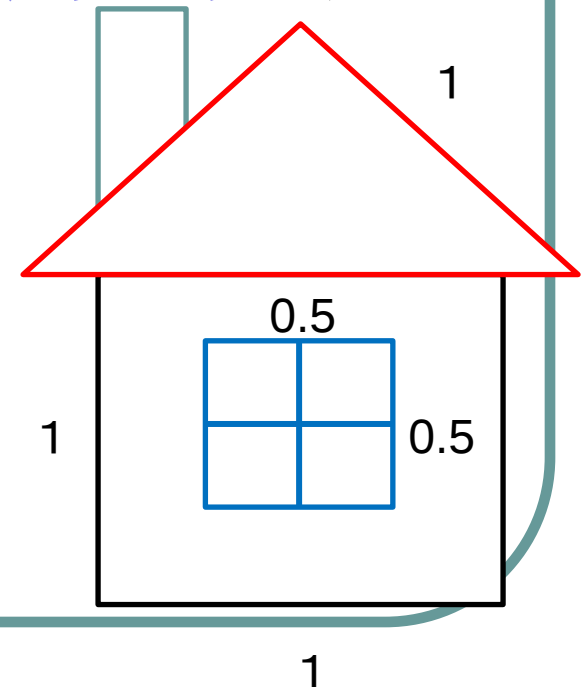
1辺の長さを入力してください

100

1辺100.0の大きさの家を描きました

練習問題

- 家を書くプログラム改良して色と煙突付きの家(InputHouseEx2.java)に改良しなさい
 - 煙突を追加(※高さは三角形、左辺は壁と同じx座標)
 - 色をつけなさい(屋根は赤、窓は青、煙突は灰色)



```
// タートルを動かす処理↓
public void start() {↓

> int length; // 1辺の長さを記憶しておくための変数↓
> int windowlength; // 1辺の長さを記憶しておくための変数↓

> print("どんな大きさの家を書きますか?");↓
> length = input(); // 1辺の長さに入力された値を代入する↓

> // 屋根を書く↓
> color(java.awt.Color, red);↓
> lt(90); // 90度左に回る↓
> fd((Math.sqrt(2) * length) - length) / 2; // 屋根のちょびっと部分を進む

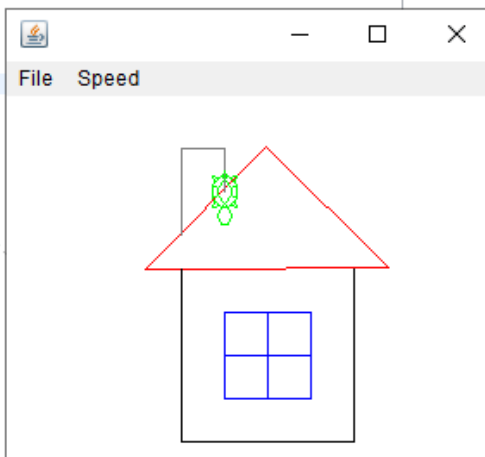
> rt(135); // 135度右に回る↓
> fd(length); // length歩前に進む↓
> rt(90);↓
> fd(length);↓
> rt(135);↓
> fd((Math.sqrt(2) * length) + length) / 2;↓

> // 家本体を書く↓
> color(java.awt.Color, black);↓
> lt(90);↓
> fd(length);↓
> lt(90);↓
> fd(length);↓
> lt(90);↓
> fd(length);↓

> // 窓を書く位置まで移動する↓
> up(); // カメの持っているペンを上げる↓
> lt(180);↓
> fd(length / 4);↓
> rt(90);↓
> fd(length / 4);↓
> down(); // ペンを下げる↓

> // 窓の長さを設定する (2で割って半分の長さにする) ↓
> windowlength = length / 2;↓

> // 窓の本体を書く↓
> color(java.awt.Color, blue);↓
> fd(windowlength);↓
> lt(90);↓
> fd(windowlength);↓
> lt(90);↓
> fd(windowlength);↓
> lt(90);↓
> fd(windowlength);↓
```



```
62 > // 煙突を書く位置まで移動する↓
63 > up(); // カメの持っているペンを上げる↓
64 > fd(length / 4);↓
65 > rt(90);↓
66 > fd(length / 2);↓
67 > fd((Math.sqrt(2) * length) - length) / 2;↓
68 > down(); // ペンを下げる↓
69 > // 煙突の本体を書く↓
70 > color(java.awt.Color, gray);↓
71 > fd(length/2);↓
72 > rt(90);↓
73 > fd(length/4);↓
74 > rt(90);↓
75 > fd(length/4);↓
76 >
77 > print(length + "の大きさの家を書きました");↓
78 }↓
79 ↓
80 ↓
81 ↓
82 ↓
83 ↓
84 ↓
85 ↓
86 ↓
87 ↓
88 ↓
89 ↓
90 ↓
91 ↓
92 ↓
93 ↓
94 ↓
95 }
```

プログラム続き

ランナー 問題 コンソール

InputHouseEx2 [Java アプリケーション] C:\Program Files\Java\jre1.8.0_73\bin\
Turtle Version: 1.0.5 (2007/12/21)
どんな大きさの家を書きますか?
100
100の大きさの家を書きました



Project.3

【基本問題】

演習問題

- 初級 (OmikujiEX.java)

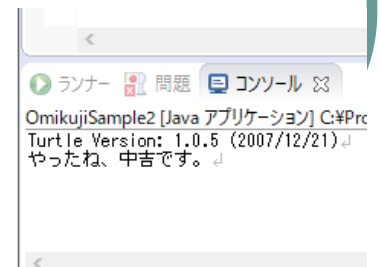
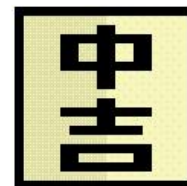
- 実行するたびに異なるおみくじの結果 (画像) が表示されるようにしなさい

- 中級 (OmikujiEX2.java)

- おみくじの結果 (画像) に加えてコンソールに一言気のきいたコメントを表示させなさい (print文を使う)

- 上級 (OmikujiEX3.java)

- 大吉, 中吉, 小吉, 凶がでる確率を10%, 40%, 40%, 10%になるように調整しなさい



初級

```
// タートルを動かす処理
public void start() {
    > // 前処理
    > hide();// 亀を隠す
    > window.size(220, 250);// 表示ウィンドウのサイズ調整

    > // おみくじを引く
    > int omikugiNumber;
    > omikugiNumber = random(4);

    > // おみくじの結果を表示する

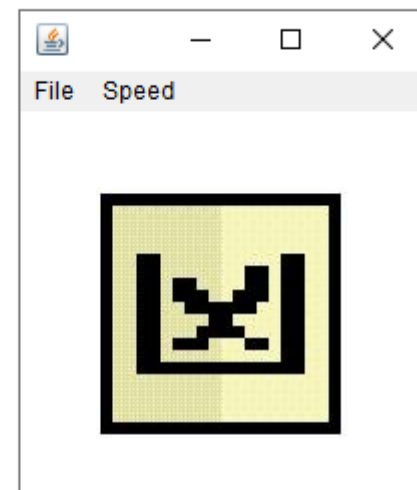
    > if (omikugiNumber == 1) {
    >     > new ImageTurtle("daikichi.jpg");// daikichi.jpgを表示
    > }

    > if (omikugiNumber == 2) {
    >     > new ImageTurtle("chuukichi.jpg");// chuukichi.jpgを表示
    > }

    > if (omikugiNumber == 3) {
    >     > new ImageTurtle("shoukichi.jpg");// shoukichi.jpgを表示
    > }

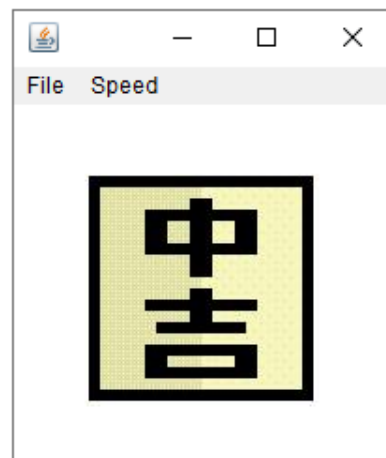
    > if (omikugiNumber == 0) {
    >     > new ImageTurtle("kyou.jpg");// kyou.jpgを表示
    > }

    > // プログラムの最後に必ず1回update命令を実行すること
    > update();
}
}
```



中級

```
10 <
11 > // タートルを動かす処理<
12 > public void start() {<
13 <
14 > > // 前処理<
15 > > hide();// 亀を隠す<
16 > > window.size(220, 250);// 表示ウィンドウのサイズ調整<
17 <
18 > > // おみくじを引く<
19 > > int omikujNumber;<
20 > > omikujNumber = random(4);<
21 <
22 > > // おみくじの結果を表示する(コメント表示編)<
23 <
24 > > if (omikujNumber == 0) {<
25 > > > print("おめでとう、大吉です!");<
26 > > > new ImageTurtle("daikichi.jpg");// daikichi.jpgを表示<
27 > > } else if (omikujNumber == 1) {<
28 > > > print("やったね、中吉です.");<
29 > > > new ImageTurtle("chuukichi.jpg");// chuukichi.jpgを表示<
30 > > } else if (omikujNumber == 2) {<
31 > > > print("小吉です。まずまずですね.");<
32 > > > new ImageTurtle("shoukichi.jpg");// shoukichi.jpgを表示<
33 > > } else if (omikujNumber == 3) {<
34 > > > print("...凶です.");<
35 > > > new ImageTurtle("kyou.jpg");// kyou.jpgを表示<
36 > > } else {<
37 > > > print("予想GUYです.");<
38 > > }<
39 <
40 > > // プログラムの最後に必ず1回update命令を実行すること<
41 > > update();<
42 <
43 <
44 }<
45 <
```



ランナー 問題 コンソール

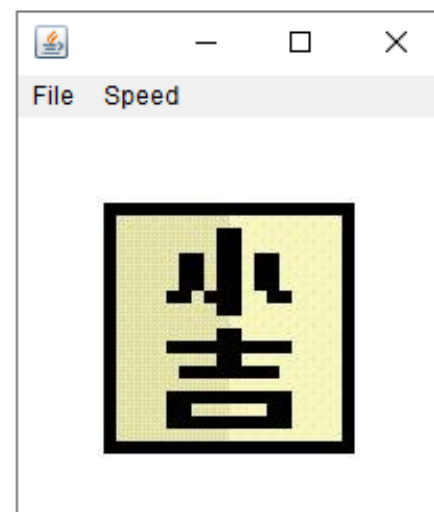
OmikujSample2 [Java アプリケーション] C:\Program Files\Java\jre1.8.0_73\bin\javaw.exe (2020/07/06 17:53:50)

Turtle Version: 1.0.5 (2007/12/21)

やったね、中吉です。

上級

```
9 > }  
10  
11 // タートルを動かす処理  
12 public void start() {  
13  
14 > // 前処理  
15 > hide();// 亀を隠す  
16 > window.size(220, 250);// 表示ウィンドウのサイズ調整  
17  
18 > // おみくじを引く  
19 > int omikujNumber;  
20 > omikujNumber = random(10);  
21  
22 > // おみくじの結果を表示する（確率変動編）  
23  
24 > if (omikujNumber == 0) {  
25 > > print("おめでとう、大吉です!");  
26 > > new ImageTurtle("daikichi.jpg");// daikichi.jpgを表示  
27 > > } else if (1 <= omikujNumber && omikujNumber <= 4) {  
28 > > > print("やったね、中吉です。");  
29 > > > new ImageTurtle("chuukichi.jpg");// chuukichi.jpgを表示  
30 > > > } else if (5 <= omikujNumber && omikujNumber <= 8) {  
31 > > > > print("小吉です。まずまずですね。");  
32 > > > > new ImageTurtle("shoukichi.jpg");// shoukichi.jpgを表示  
33 > > > } else if (omikujNumber == 9) {  
34 > > > > print("…凶です。");  
35 > > > > new ImageTurtle("kyou.jpg");// kyou.jpgを表示  
36 > > > } else {  
37 > > > > print("予想GUYです。");  
38 > > > }  
39  
40 > // プログラムの最後に必ず1回update命令を実行すること  
41 > > update();  
42 > }  
43  
44 }  
45
```



ランナー 問題 コンソール

OmikujSample3 [Java アプリケーション] C:\Program Files\Java\jre1.8.0_73\bin\javaw.exe (2020/07/06 17:56:39)

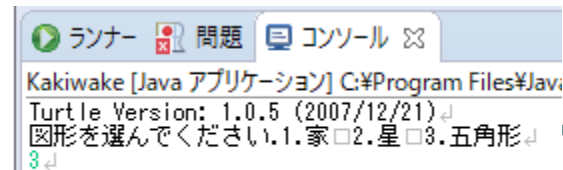
Turtle Version: 1.0.5 (2007/12/21)

小吉です。まずまずですね。

【発展問題】

練習問題

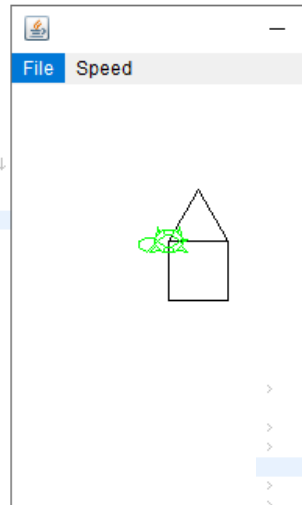
- 絵の描き分けプログラム(Kakiwake.java)
ユーザの入力に応じて、タートルが書く絵(1.家と2.星と3.五角形)を変えるプログラムを作りなさい
 - 大まかな処理の流れ
 - 「1:家、2:星、3: 五角形 を描きます。数字を入力してください」と表示
 - ユーザが1 or 2 or 3を入力
 - 該当の図形を表示
 - ※その他が入力された場合は「無効な値です」と表示



```

12 > // タートルを動かす処理↓
13 > public void start() {↓
14 > ↓
15 > > print("図形を選んでください.1.家□2.星□3.五角形");↓
16 > > int zukei = input();↓
17 > ↓
18 > > if (zukei == 1) {↓
19 > ↓
20 > > > int length; // 1辺の長さ↓
21 > > > int rightAngle; // 直角↓
22 > ↓
23 > > > length = random(100); // 1辺の長さを0-99までのランダム値に設定する↓
24 > > > rightAngle = 90; // 直角を90度に設定する↓
25 > ↓
26 > > > // 屋根を書く↓
27 > > > rt(30); // 30度右に回る↓
28 > > > fd(length); // x歩前に進む↓
29 > > > rt(120);↓
30 > > > fd(length);↓
31 > > > rt(120);↓
32 > > > fd(length);↓
33 > ↓
34 > > > // 本体を書く↓
35 > > > lt(rightAngle);↓
36 > > > fd(length);↓
37 > > > lt(rightAngle);↓
38 > > > fd(length);↓
39 > > > lt(rightAngle);↓
40 > > > fd(length);↓
41 > > > lt(rightAngle);↓
42 > > > fd(length);↓
43 > > } else if (zukei == 2) {↓
44 > ↓
45 > > > int length; // 1辺の長さ↓
46 > > > length = random(100); // 1辺の長さを0-99までのランダム値に設定する↓
47 > ↓
48 > > > // 星を描く↓
49 > > > rt(144);↓
50 > > > fd(length);↓
51 > > > rt(144);↓
52 > > > fd(length);↓
53 > > > rt(144);↓
54 > > > fd(length);↓
55 > > > rt(144);↓
56 > > > fd(length);↓
57 > > > rt(144);↓
58 > > > fd(length);↓
59 > ↓
60 > > } else if (zukei == 3) {↓
61 > ↓
62 > > > int length; // 1辺の長さ↓
63 > > > length = random(100); // 1辺の長さを0-99までのランダム値に設定する↓
64 > ↓
65 > > > // 地面を平らにする↓
66 > > > rt(90);↓
67 > ↓
68 > > > // 五角形を書く↓
69 > > > fd(length);↓
70 > > > lt(72);↓

```



```

> } else if (zukei == 3) {↓
> > > int length; // 1辺の長さ↓
> > > length = random(100); // 1辺の長さを0-99までのランダム値に設定する↓
> > > // 地面を平らにする↓
> > > rt(90);↓
> > > // 五角形を書く↓
> > > fd(length);↓
> > > lt(72);↓
> > > fd(length);↓
> > > lt(72);↓
> > > fd(length);↓
> > > lt(72);↓
> > > fd(length);↓
> > > lt(72);↓
> > > fd(length);↓
> > > lt(72);↓
> } else {↓
> > > print("不正な値です");↓
> }↓

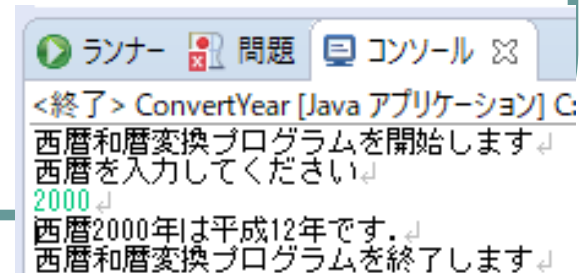
```

プログラム続き

練習問題

- 西暦和暦変換プログラム(ConvertYear.java)
ユーザが西暦を入力すると、変換処理を行い、和暦を出力するプログラムを作りなさい
 - 大まかな処理の流れ
 - ①「西暦を入力してください」と表示
 - ② ユーザが西暦を入力
 - ③「西暦xxxx年は令和〇〇年です」等と結果を表示
 - ※和暦は「昭和(63年まで)、平成(30年まで)、令和」のみでその他は「無効な値です」と表示
 - ※元年は1年表記

【追記】問題文ミス！



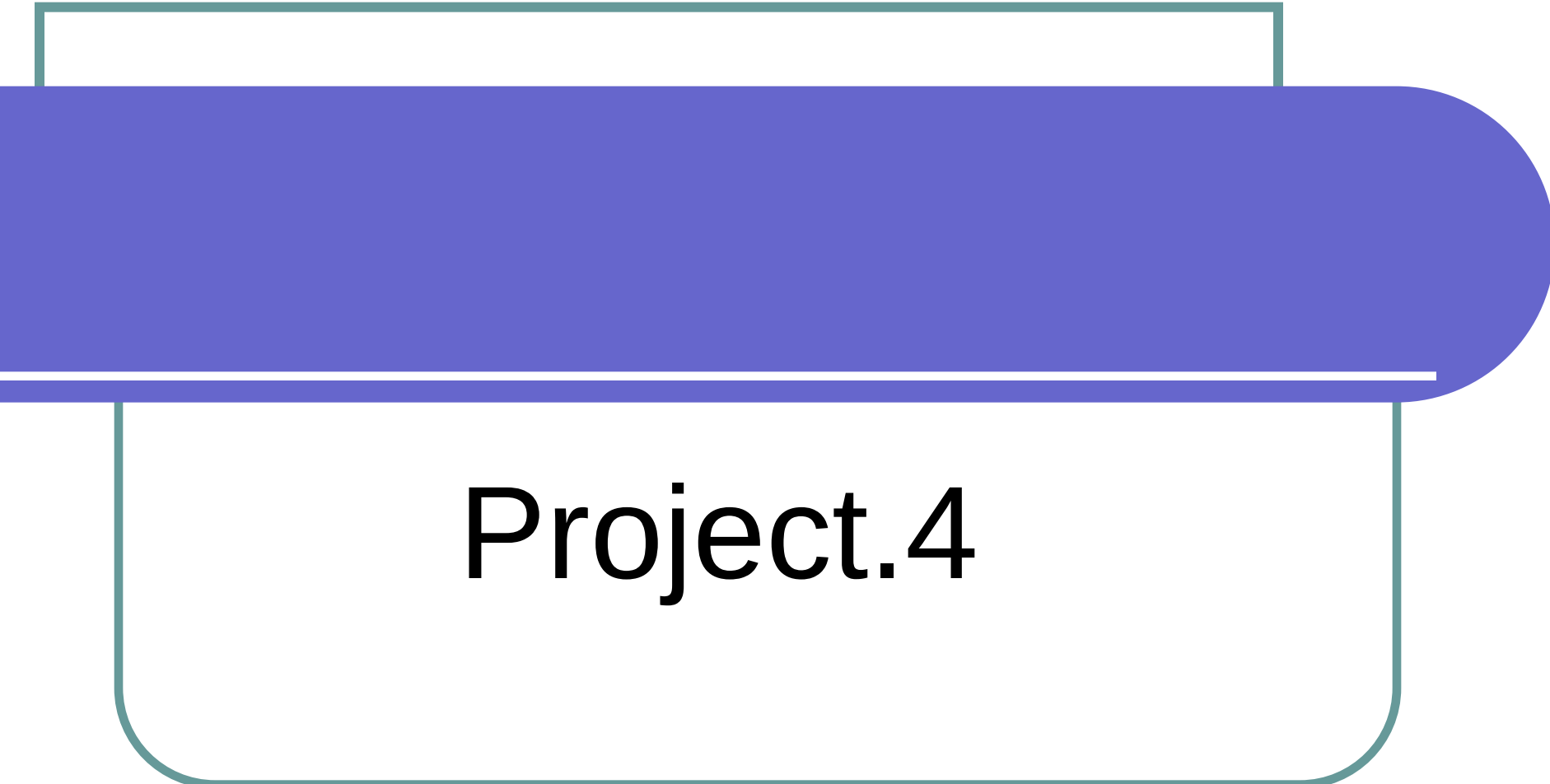
```
<終了> ConvertYear [Java アプリケーション] C:  
西暦和暦変換プログラムを開始します↓  
西暦を入力してください↓  
2000↓  
西暦2000年は平成12年です。↓  
西暦和暦変換プログラムを終了します↓
```

おまけ！

```
11 >
12 > // タートルを動かす処理↓
13 > public void start() {↓
14 >
15 >     hide();↓
16 >
17 >     // アプリケーションの開始を知らせる↓
18 >     print("西暦和暦変換プログラムを開始します");↓
19 >
20 >     print("西暦を入力してください");↓
21 >     int year = input();↓
22 >
23 >     String japaneseYear;↓
24 >
25 >     // 西暦を和暦に変換する↓
26 >     if (year >= 2021) {↓
27 >         print("未来のことは分かりません");↓
28 >     } else if (year >= 2019) {↓
29 >         japaneseYear = "令和" + (year - 2018);↓
30 >         // 変換結果を出力する↓
31 >         print("西暦" + year + "年は" + japaneseYear + "年です.");↓
32 >     } else if (year >= 1989) {↓
33 >         japaneseYear = "平成" + (year - 1988);↓
34 >         // 変換結果を出力する↓
35 >         print("西暦" + year + "年は" + japaneseYear + "年です.");↓
36 >     } else if (year >= 1926) {↓
37 >         japaneseYear = "昭和" + (year - 1925);↓
38 >         // 変換結果を出力する↓
39 >         print("西暦" + year + "年は" + japaneseYear + "年です.");↓
40 >     } else if (year >= 1912) {↓
41 >         japaneseYear = "大正" + (year - 1911);↓
42 >         // 変換結果を出力する↓
43 >         print("西暦" + year + "年は" + japaneseYear + "年です.");↓
44 >     } else {↓
45 >         print("そんな前の時代の人は知りません");↓
46 >     }↓
47 >
48 >     // アプリケーションの終了を知らせる↓
49 >     print("西暦和暦変換プログラムを終了します");↓
50 >     System.exit(0);↓
51 >
52 > }↓
53 >
```

ランナー 問題 コンソール

```
<終了> ConvertYear [Java アプリケーション] C:\Program Files\Java\jre1.8.0_73\bin\javaw.exe (
Turtle Version: 1.0.5 (2007/12/21)
西暦和暦変換プログラムを開始します
西暦を入力してください
2020
西暦2020年は令和2年です。
西暦和暦変換プログラムを終了します
```



Project.4

【基本問題】

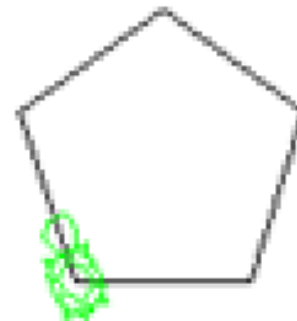
演習課題

- PentagonEX.java

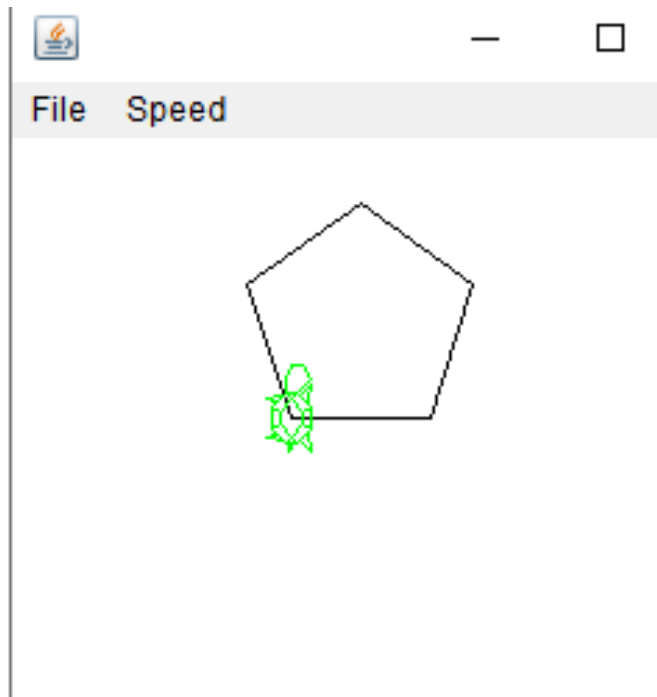
- Pentagon.javaを改良し、繰り返し(while文)を使って正五角形を書くプログラムにしない

ポイント

- 一周したらプログラムを止めるよう改良



```
// タートルを動かす処理↓  
void start() {↓  
  > // 左に傾ける↓  
  > rt(-18);↓  
  > // 五角形を書く↓  
  > int i;↓  
  > i = 0;↓  
  > while (i < 5) {↓  
  >   > fd(50);↓  
  >   > rt(72);↓  
  >   > i++;↓  
  > }↓  
  > // 元に戻る↓  
  > rt(18);↓  
}↓
```

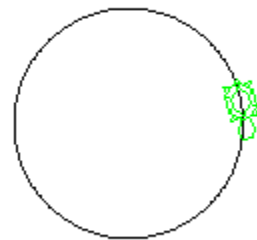


演習課題

- CircleEX.java
 - Circle.java（円を描くプログラム）を1周したら亀が止まるように改良しなさい

ポイント

- 一周したらプログラムを止めるよう改良



// タートルを動かす処理↓

public void start() ↓

> // 円を書く↓

> **int** i = 0;↓

> **while** (i < 360) {↓

> > fd(1);↓

> > rt(1);↓

> > ↓

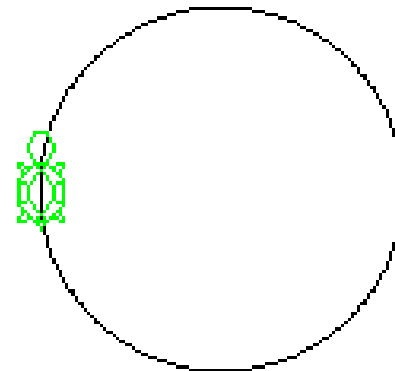
> > i++;↓

> }↓

}↓



File Speed



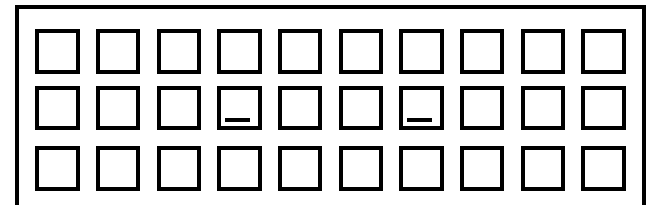
演習課題

- Keyboard.java

- キーボードを描くプログラムを作りなさい

ポイント

- 横10列 縦3列のキー配列
 - キーの大きさ「lengthは10」キー間隔「marginは5」
- キーボードの外枠と印はプログラム済（省略）



```

// タートルを動かす処理↓
public void start() {↓
  > int i; // ループ用1↓
  > int j; // ループ用2↓
  > int k; // ループ用3↓

  > // キーボードのキーを書く↓

  > int length = 10; // キーボードの大きさ↓
  > int margin = 5; // キーの間隔↓

  > // 縦に3回繰り返すためのループ↓
  > i = 1;↓
  > while (i <= 3) {↓

    > > // 横に10回繰り返すためのループ↓
    > > j = 1;↓
    > > while (j <= 10) {↓

      > > > // 四角形を書く↓
      > > > k = 1;↓
      > > > while (k <= 4) {↓
      > > > > fd(length);↓
      > > > > rt(90);↓

      > > > > k++;↓
      > > > }↓

      > > > // 図形を書いたら隣に移動↓
      > > > up();↓
      > > > rt(90);↓
      > > > fd(length);↓
      > > > fd(margin);↓
      > > > lt(90);↓
      > > > down();↓

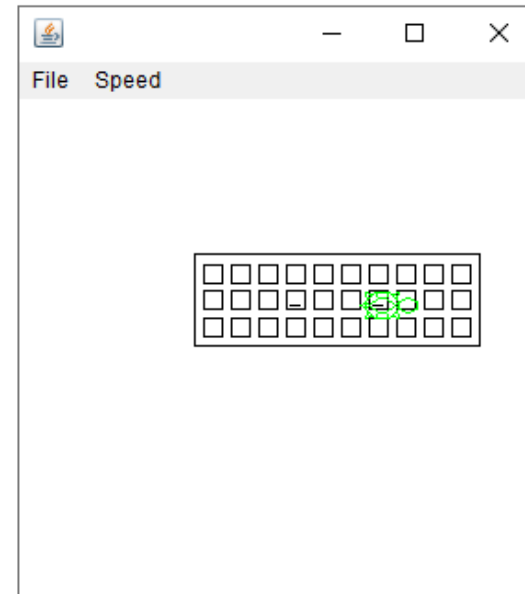
      > > > j++;↓
      > > }↓

      > > // 横一列が終わったら左下に移動↓
      > > up();↓
      > > lt(90);↓
      > > fd((length + margin) * 10);↓
      > > lt(90);↓
      > > fd(length + margin);↓
      > > rt(180);↓
      > > down();↓

      > > i++;↓
      > }↓

  > // キーボードの外枠を書く↓

```



【発展問題】

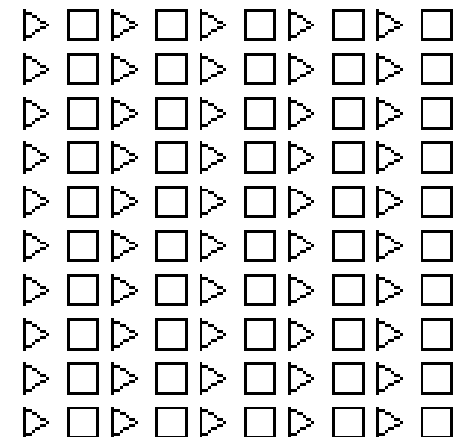
演習課題

- HousePattern.java

- 四角形と三角形を交互に100個描くプログラムを作りなさい

ポイント

- 入れ子を使ったプログラミング
- 5セット × 10回



```
// タートルを動かす処理↓
public void start() {↓

    > int i;// ループ用1↓
    > int j;// ループ用2↓
    > int k;// ループ用3↓
    > int l;// ループ用4↓

    > int length = 10;// 図形の大きさ↓
    > int margin = 5;// 隣の図形との間隔↓

    > // 縦に10回繰り返すためのループ↓
    > i = 1;↓
    > while (i <= 10) {↓

        > // 横に10回繰り返すためのループ↓
        > j = 1;↓
        > while (j <= 10) {↓

            > > if (j % 2 == 0) {↓
            > > // 偶数なら↓
            > > // 四角形を書く↓
            > > k = 1;↓
            > > while (k <= 4) {↓
            > > > fd(length);↓
            > > > rt(90);↓

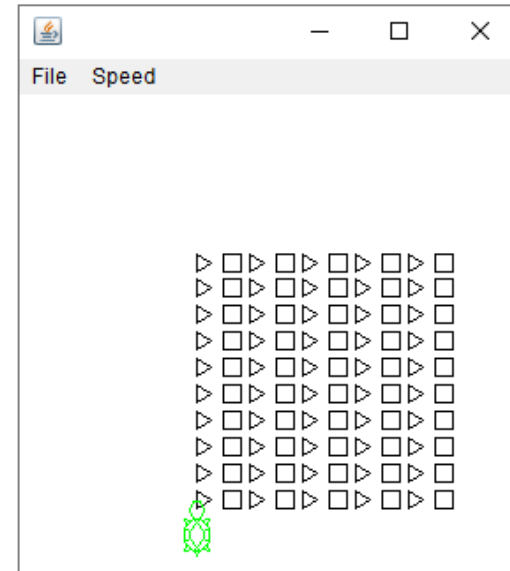
            > > > k++;↓
            > > > }↓
            > > } else {// 奇数なら↓
            > > > // 正三角形を書く↓
            > > > l = 1;↓
            > > > while (l <= 3) {↓
            > > > > fd(length);↓
            > > > > rt(120);↓

            > > > > l++;↓
            > > > > }↓
            > > > }↓
            > > // 図形を書いたら隣に移動↓
            > > up();↓
            > > rt(90);↓
            > > fd(length);↓
            > > fd(margin);↓
            > > lt(90);↓
            > > down();↓

            > > j++;↓
            > > }↓

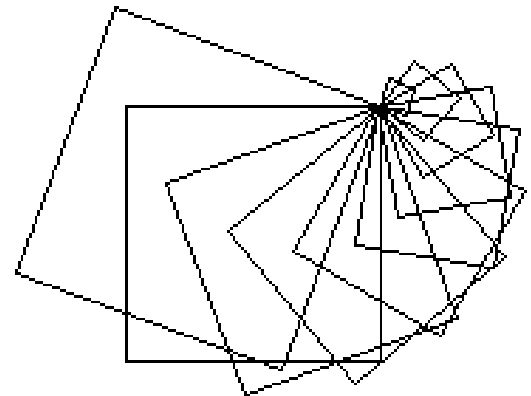
            > // 横一列が終わったら左下に移動↓
            > > up();↓
            > > lt(90);↓
            > > fd((length + margin) * 10);↓
            > > lt(90);↓
            > > fd(length + margin);↓
            > > rt(180);↓
            > > down();↓

            > > i++;↓
            > > }↓
        }↓
    }↓
}
```



演習課題

- Shell.java
 - 貝殻を描くプログラムを作りなさい
- ポイント
- 入れ子を使ったプログラミング
 - 「ループ用変数」を変化



```

// タートルを動かす処理↓
public void start() {↓

    > // ループ用変数の設定↓
    > int i;↓
    > int j;↓

    > int length = 10;↓
    > int angle = 90;↓

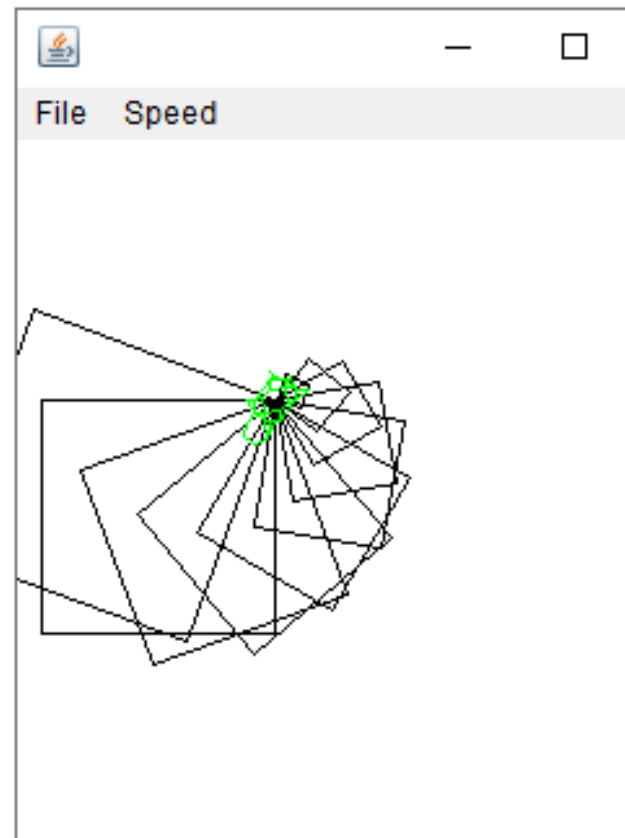
    > //角度の調整↓
    > rt(20);↓

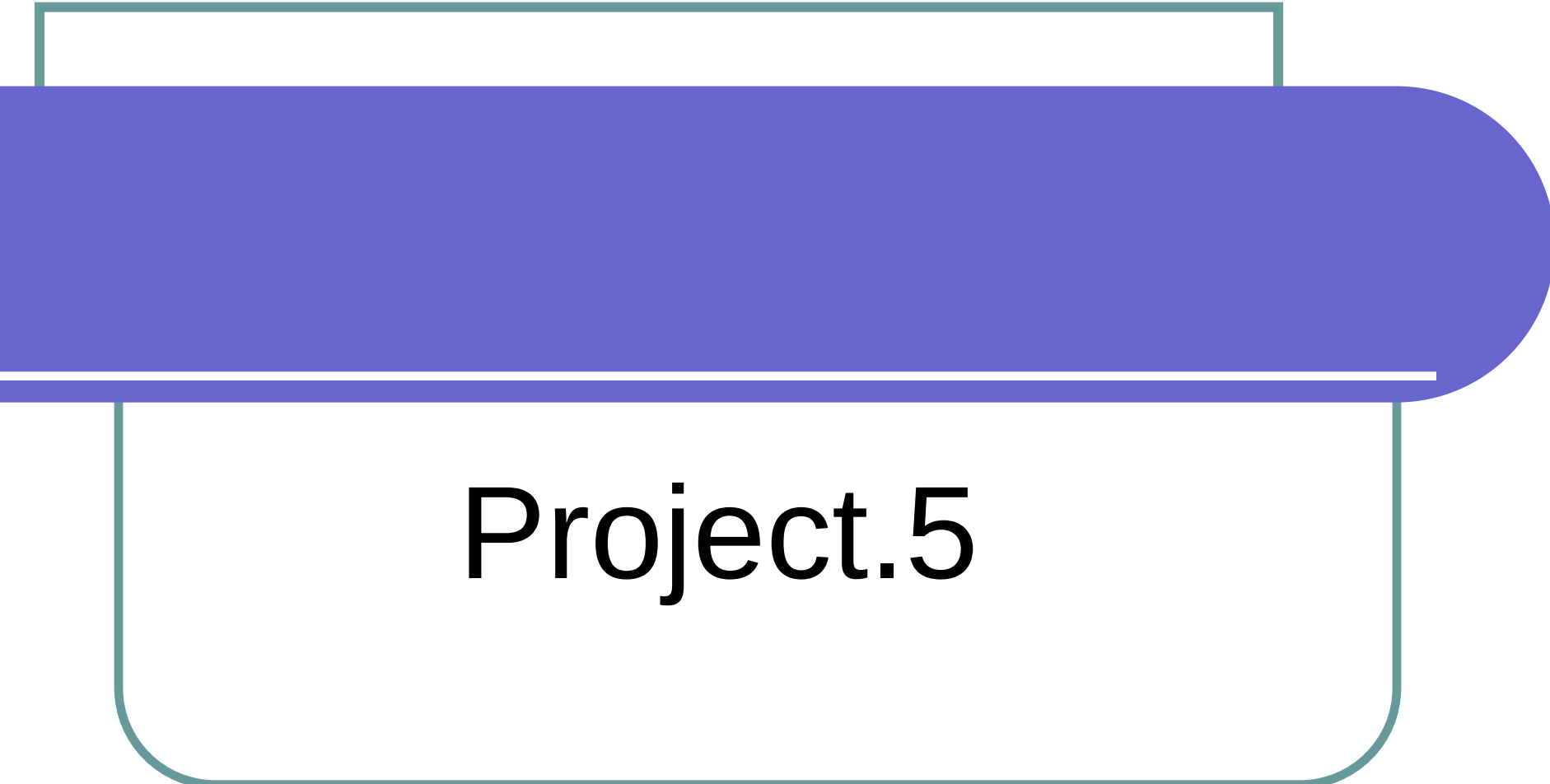
    > //正方形を繰り返し書く↓
    > i = 1;↓
    > while (i <= 10) {↓

        > // 正方形を書く↓
        > j = 1;↓
        > while (j <= 4) {↓
            > fd(length);↓
            > rt(angle);↓
            > j++;↓
        > }↓

        > // 次の正方形の大きさへ↓
        > rt(20);↓
        > length = length + 10;↓
        > i++;↓
    > }↓
}↓

```





Project.5

【基本問題】

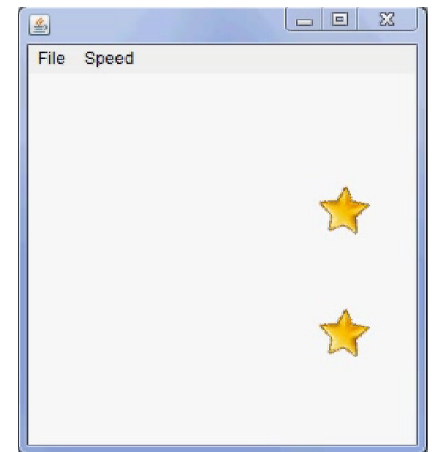
演習課題

- ShootingStarEX.java
 - 流れ星を表示するプログラムを作りなさい

ポイント

- ① 星を回してみる
- ② 座標を使って星を右に動かす
- ③ 一番右端にいったら、左端に戻るようにする
- ④ 星を下方方向にも動かす
- ⑤ 星を増やす(2個でよい)

完成イメージは
Webページに上がっています



座標用変数が
1組(x,y)だと
使い回すため
ミスすることも...

```
// タートルを動かす処理↓
public void start() {↓

    > // カメを隠す↓
    > hide();↓

    > // 星を生成する↓
    > ImageTurtle star = new ImageTurtle("star.png");↓
    > ImageTurtle star2 = new ImageTurtle("star.png");↓

    > int x; // x座標を入れる変数↓
    > int y; // y座標を入れる変数↓

    > //starの初期位置の設定↓
    > star.warp(0, 0);↓
    > star2.warp(0, 50);↓

    // アニメーションループ↓
    while (true) {↓

        > // --- 待つ ---↓
        > sleep(0.1); // 0.1秒↓

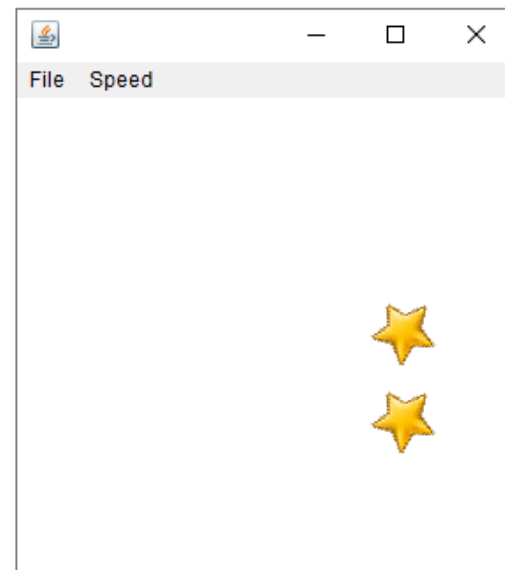
        > // --- starの処理を行う ---↓
        > star.rt(10);↓
        > x = star.getX(); // 家のx座標を取得する↓
        > y = star.getY(); // 家のy座標を取得する↓
        > star.warp(x + 5, y + 3); // 右下に移動↓

        > // --- star2の処理を行う ---↓
        > star2.rt(10);↓
        > x = star2.getX(); // 家のx座標を取得する↓
        > y = star2.getY(); // 家のy座標を取得する↓
        > star2.warp(x + 5, y + 3); // 右下に移動↓

        > // 右端だったら左端にワープ↓
        > if (x >= 300) {↓
        > > star.warp(0, 0);↓
        > > star2.warp(0, 50);↓
        > }↓

        > // --- 再描画する ---↓
        > update();↓
    }↓

}↓
```



発展問題では
座標変数を2種類用意

【発展問題】

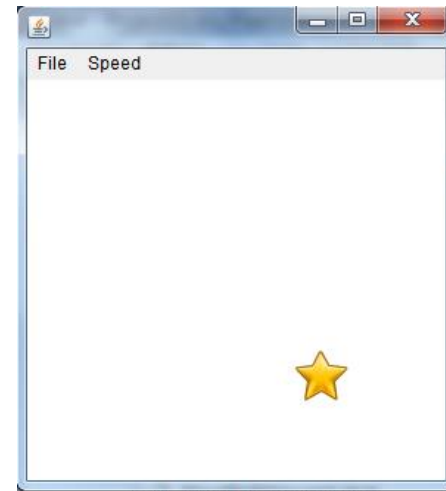
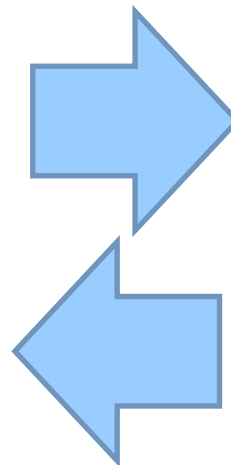
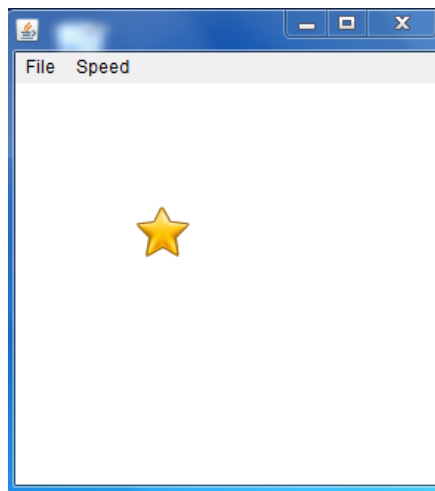
演習課題

- FlashStar.java (発展問題)
 - 星を点滅するプログラムを作りなさい

ポイント

- ① 星を表示する
- ② 座標を上手に使って星を点滅させる
- ③ 星を増やす(2個でよい)

完成イメージは
Webページに上がっています



座標用変数が
2組(x,y)と(x1,y1)
なので
ミスしやすい！

```
// タートルを動かす処理↓
public void start() {↓

    > // カメを隠す↓
    > hide();↓

    > // 星を生成する↓
    > ImageTurtle star = new ImageTurtle("star.png");↓
    > ImageTurtle star2 = new ImageTurtle("star.png");↓

    > int x; // x座標を入れる変数↓
    > int y; // y座標を入れる変数↓
    > int x2; // x座標を入れる変数2↓
    > int y2; // y座標を入れる変数2↓

    > while (true) {↓

        > //星と星2の初期位置の設定↓
        > x = star.getX(); // 星のx座標を取得する↓
        > y = star.getY(); // 星のy座標を取得する↓
        > x2 = star2.getX(); // 星2のx座標を取得する↓
        > y2 = star2.getY(); // 星2のy座標を取得する↓

        > // 待つ↓
        > sleep(0.1);↓

        > // 処理↓
        > > star.warp(100, 100);↓
        > > star2.warp(2000, 2000);↓

        > > if (x == 100 && y == 100) {↓
        > > > star.warp(5000, 5000); // 右下に移動↓

        > > } else {↓
        > > > star.warp(100, 100);↓
        > > }↓

        > > if (x2 == 200 && y2 == 200) {↓
        > > > star2.warp(5000, 5000); // 右下に移動↓

        > > } else {↓
        > > > star2.warp(200, 200);↓
        > > }↓

        > > // 再描画↓
        > > update();↓
        > }↓
    }↓
}
```

