
コレクション

プログラミング演習Ⅱ (4)

中村, 小松, 菊池

宿題

- 3.1 時刻情報を整数に変換する toInt() メソッドをオーバーライドした Hourly2.pde を書け.

実行結果

```
a = 1:12:34
```

```
a = 4354 [sec]
```

- 3.2 文字列を元に時刻情報をインスタンス化するコンストラクタ Hourly3.pde を書け.
 - 例) `Hourly a = new Hourly("1:12:34");`
- 3.3 値が10以下の時は先頭に0を付ける様に表示する Hourly4.pde を書け.

実行結果

```
a = 01:12:34
```

```
b = 02:48:42
```

```
a+b = 04:01:16
```

宿題3.1 解答例 (一部)

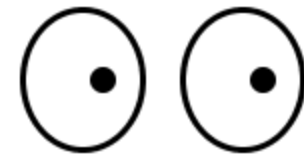
```
■ Hourly2.pde
1. void draw(){
2.     Hourly a = new
   Hourly(1, 12, 34);
3.     Hourly b = new
   Hourly(2, 48, 42);
4.     println("a = " + a.toInt());
5.     noLoop();
6. }
7. class Hourly extends
   Timely{
8.     int hour;
9.     Hourly(int h, int m, int s){
10.         hour = h;
11.         min = m; sec = s;
12.     }
13.     int toInt(){
14.         return(hour * 60*60 +
   super.toInt());
15.     }
16. }
```

1. 静的メソッド(復習)

■ DrawEye1.pde

```
1. void setup() {  
2.   size( 400, 400 );  
3. }  
4. void draw() {  
5.   background( 255 );  
6.   strokeWeight( 3 );  
7.   fill( 255, 255, 255 );  
8.   ellipse( 160, 200, 60, 70 );  
9.   fill( 0, 0, 0 );  
10.  if (160-10 > mouseX) {  
11.    ellipse(160 - 10, 200, 10, 10);  
12.  } else if (160+10 < mouseX) {  
13.    ellipse( 160+10, 200, 10, 10 );  
14.  } else {  
15.    ellipse( mouseX, 200, 10, 10 );  
16.  }
```

```
17. fill( 255, 255, 255 );  
18. ellipse( 240, 200, 60, 70 );  
19. fill( 0, 0, 0 );  
20. if (240-10 > mouseX) {  
21.   ellipse(240 - 10, 200, 10, 10);  
22. } else if (240+10 < mouseX) {  
23.   ellipse( 240+10, 200, 10, 10 );  
24. } else {  
25.   ellipse( mouseX, 200, 10, 10 );  
26. }  
27. }
```



欠点:

(1) モジュール化

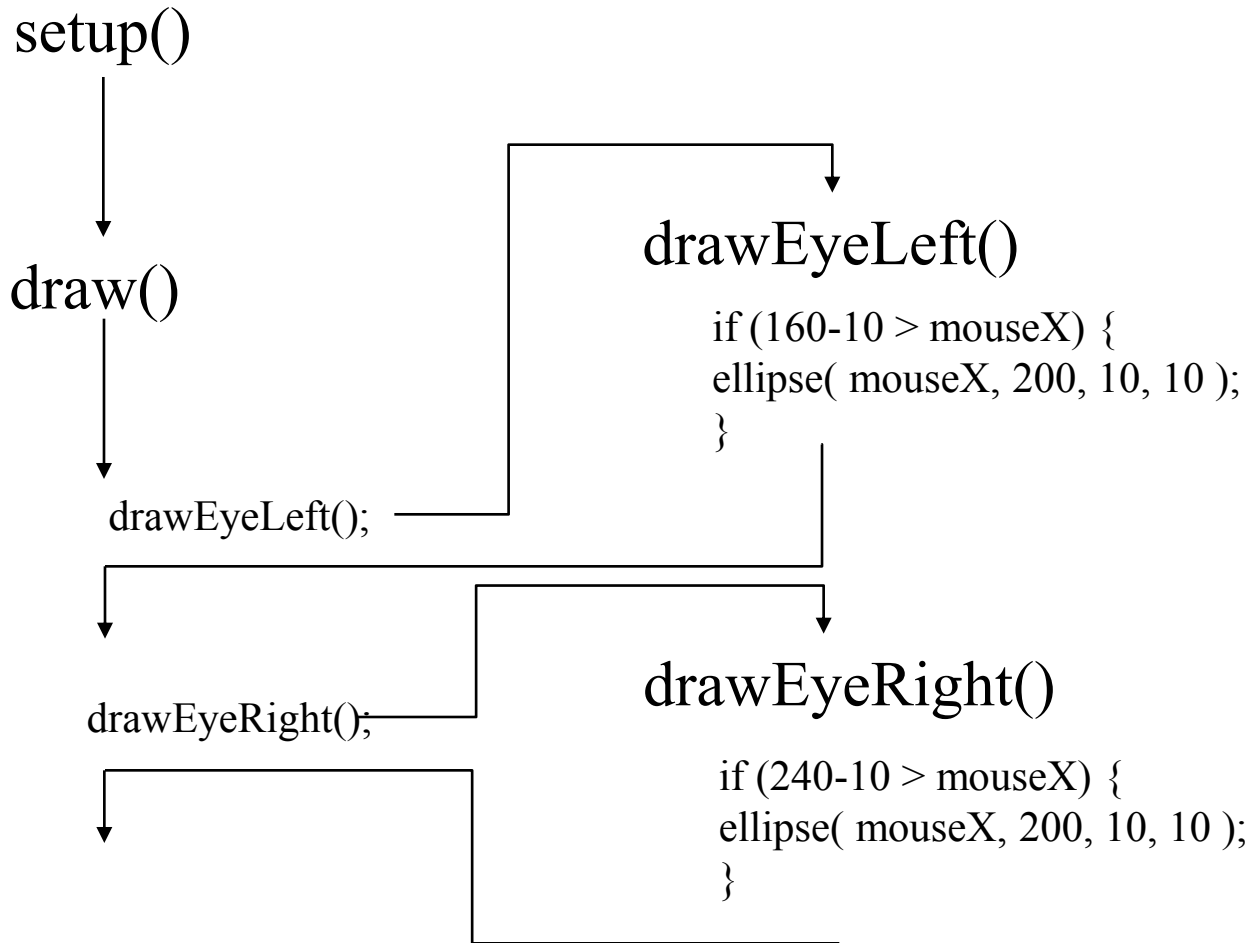
```
■ DrawEye2.pde
1.   void setup() {
2.     size( 400, 400 );
3.   }
4.   void drawEyeLeft(){
5.     strokeWeight( 3 );
6.     fill( 255, 255, 255 );
7.     ellipse( 160, 200, 60, 70 );
8.     fill( 0, 0, 0 );
9.     if (160-10 > mouseX) {
10.      ellipse(160 - 10, 200, 10, 10);
11.    } else if (160+10 < mouseX) {
12.      ellipse( 160+10, 200, 10, 10 );
13.    } else {
14.      ellipse( mouseX, 200, 10, 10 );
15.    }
16.  }

17.  void drawEyeRight(){
18.    fill( 255, 255, 255 );
19.    ellipse( 240, 200, 60, 70 );
20.    fill( 0, 0, 0 );
21.    if (240-10 > mouseX) {
22.      ellipse(240 - 10, 200, 10, 10);
23.    } else if (240+10 < mouseX) {
24.      ellipse( 240+10, 200, 10, 10 );
25.    } else {
26.      ellipse( mouseX, 200, 10, 10 );
27.    }
28.  }

29.  void draw() {
30.    background( 255 );
31.    drawEyeLeft();
32.    drawEyeRight();
33.  }
```

欠点：同じことを繰り返している()

プログラムの流れ



(2)

を用いて統合

■ DrawEye3.pde

```
1. void setup() {
2.   size( 400, 400 );
3. }
4. void drawEye(int x) {
5.   strokeWeight( 3 );
6.   fill( 255, 255, 255 );
7.   ellipse(x, 200, 60, 70 );
8.   fill( 0, 0, 0 );
9.   if (x-10 > mouseX) {
10.    ellipse(x-10, 200, 10, 10
11.    );
12.   } else if (x+10 < mouseX) {
13.    ellipse(x + 10, 200, 10,
14.    10);
```

```
15.   } else {
16.     ellipse(mouseX, 200, 10,
17.     10);
18.   }
19. }
20. void draw() {
21.   background( 255 );
22.   drawEye(160);
23.   drawEye(240);
24. }
```

引数 **x**
(160, 240
が代入)

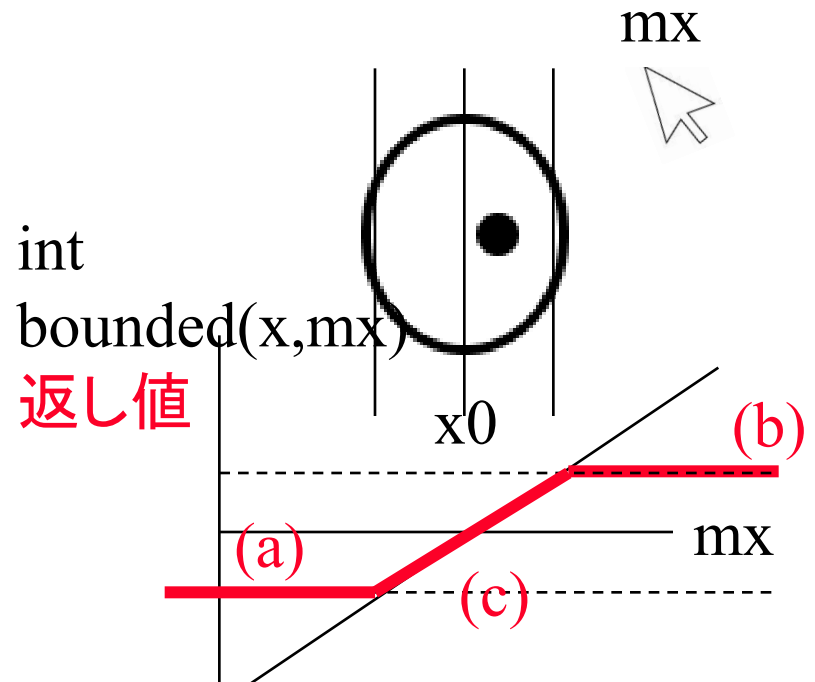
欠点:

関数と返し値

■ DrawEye4.pde

```
1.  int boundedX(int x0, int mx) {
2.    if (x0-10 > mx) {
3.      return(x0 -10); (a)
4.    } else if (x0+10 < mx) {
5.      return(x0 + 10); (b)
6.    } else {
7.      return(mx); (c)
8.    }
9.  }
```

引数



(3)

によって整理された形

ソース

```
■ DrawEye4.pde
1. void setup() {
2.   size( 400, 400 );
3. }
4. int boundedX(int x0, int mx) {
5.   if (x0-10 > mx) {
6.     return(x0 -10);
7.   } else if (x0+10 < mx) {
8.     return(x0 + 10);
9.   } else {
10.    return(mx);
11.  }
12. }

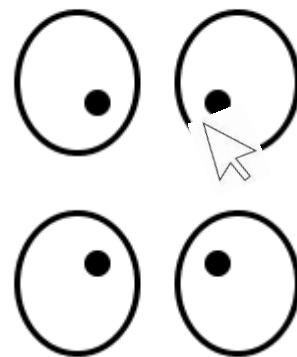
13. void drawEye(int x) {
14.   strokeWeight( 3 );
15.   fill( 255, 255, 255 );
16.   ellipse(x, 200, 60, 70 );
17.   fill( 0, 0, 0 );
18.   int bx = boundedX(x, mouseX);
19.   ellipse(bx, 200, 10, 10 );
20. }

21. void draw() {
22.   background( 255 );
23.   drawEye(160);
24.   drawEye(240);
25. }
```

drawEye()がシンプルになった。boundedXは汎用的

演習1. (静的メソッド)

- DrawEye4.pdeを基にして, xだけでなくy方向にもマウスを追う目玉を4つ持つプログラム DrawEye5.pde を書け.
 - ヒント: `drawEye(int x, int y)` に拡張する
 - `boundedX()`は変更せずそのまま利用.



静的メソッドのまとめ

■ 静的メソッドの役割

□(1) 一般化

» プログラムの分割

□(2) 引数による 重複を統合

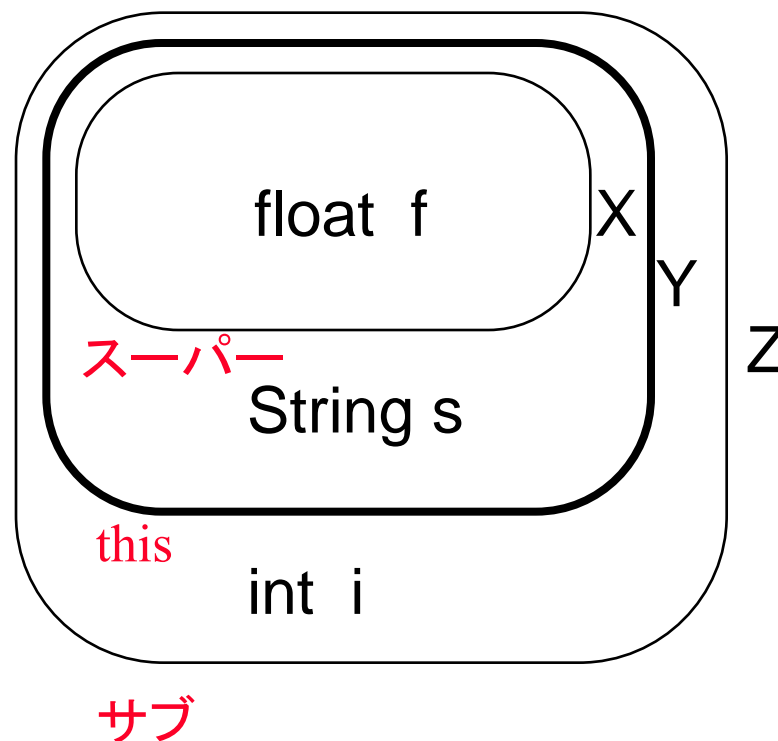
» 冗長な部分を一般化

□(3) 返回值によって 重複を整理

» 関数を実現

2. 継承の役割 (復習)

- サブクラス
 - ZはYの クラス
 - YはZの クラス
- 継承 (inheritance)
 - f, sなどはサブクラスに引き継がれる
 - ソースの再利用



Ballクラス

■ Ball.pde

```
1. class Ball {
2.     int x;
3.     float y;
4.     float speed;
5.     float gravity = 0.1;
6.     int w = 48;
7.     int life = 0;
8.     Ball(int tempX, int tempY) {
9.         x = tempX;
10.        y = tempY;
11.        speed = 0;
12.        life = 0;
13.    }
14.    Ball() {
15.    }
16.    void move() {
17.        speed = speed + gravity;
18.        y = y + speed;
19.        if (y > height) {
20.            speed = speed * -0.8;
21.            y = height;
22.        }
23.    }
24.    boolean isFinished() {
25.        if (life == 0) {
26.            x = mouseX;
27.            y = mouseY;
28.            speed = 0;
29.            life = 255;
30.            return(true);
31.        } else {
32.            return false;
33.        }
34.    }
35.    void display() {
36.        if (life == 0) return;
37.        --life;
38.        fill(0, life);
39.        ellipse(x, y, w, w);
40.    }
41. }
```

Bouncingクラス

ソース

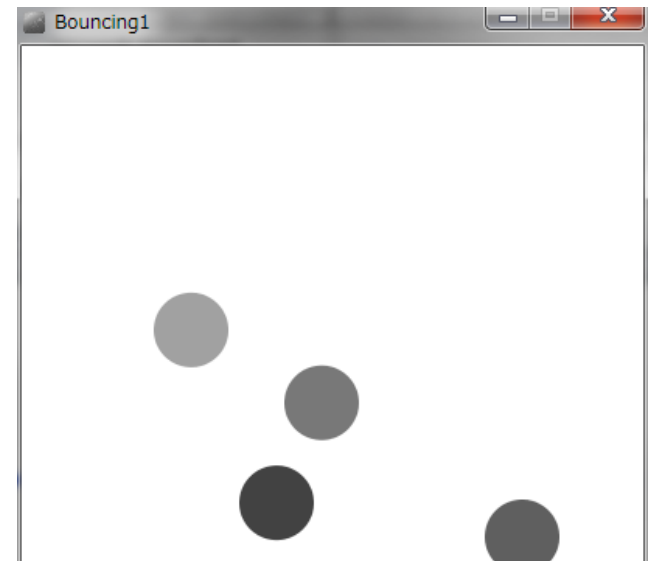
■ Bouncing1.pde

```
1.  Ball balls[] = new Ball[5];

2.  void setup() {
3.    size(400, 400);
4.    smooth();
5.    noStroke();
6.    for (int i = 0; i < balls.length; ++i) {
7.      balls[i] = new Ball(width/2, 0);
8.    }
9.  }

10. void draw() {
11.   background(255);
12.   for (Ball b: balls) {
13.     b.move();
14.     b.display();
15.   }
16. }

17. void mousePressed() {
18.   for (Ball b: balls) {
19.     if(b.isFinished()) return;
20.   }
21. }
```



For-each文

■ 配列(コレクション)の要素を列挙する例

1. `int a[] = new int[]{10,20,30};`

2. `for(int i = 0; i < a.length; ++i){`

3. `println(a[i]);`

4. `}`

10

20

30

5. `for(int b : a){`

6. `println(b);`

7. `}`

10

20

30

継承のやり方 (1/2)

ソース

■ display() のオーバーライド

```
1. class Triangle extends Ball {
2.     Triangle(int x, int y) {
3.         super(x, y);
4.     }
5.     void display() {
6.         if (life == 0) return;
7.         --life;
8.         fill(0, life);
9.         triangle(x, y-w, x+w, y+w, x-w, y+w);
10.    }
11. }
```

```
1. Class Ball {
2.     void display() {
3.         if (life == 0) return;
4.         --life;
5.         fill(0, life);
6.         ellipse(x, y, w, w);
7.     }
8. }
```

Ballのdisplay()メソッド

それ以外のメソッドは手を触れない

継承のやり方 (2/2)

■ Bouncing2.pde

```
1.   Ball balls[] = new Ball[5];
2.   Triangle tris[] = new Triangle[5];

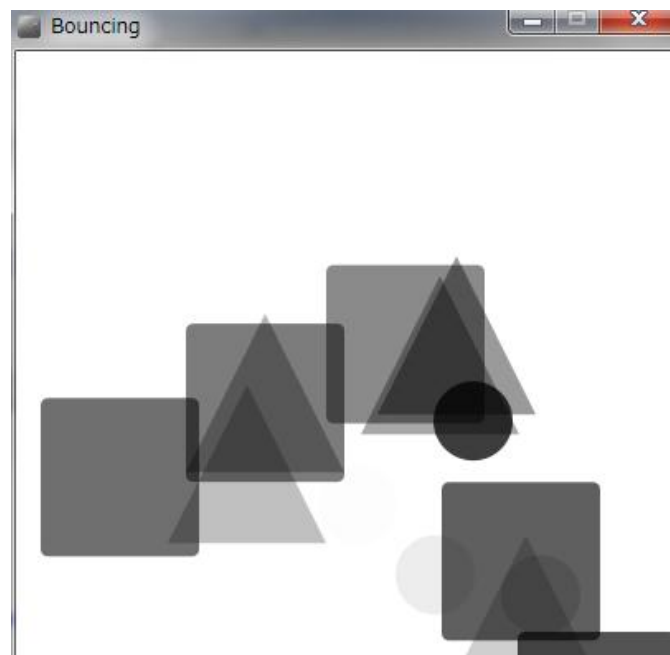
3.   void setup() {
4.     size(400, 400);
5.     smooth();
6.     noStroke();
7.     for (int i = 0; i < balls.length; ++i) {
8.       balls[i] = new Ball(width/2, 0);
9.       tris[i] = new Triangle(width/2, 0);
10.    }
11.  }

12.  void draw() {
13.    background(255);
14.    for (Ball b: balls) {
15.      b.move();
16.      b.display();
17.    }
18.    for (Triangle b: tris) {
19.      b.move();
20.      b.display();
21.    }
22.  }

23.  void mousePressed() {
24.    for (Ball b: balls) {
25.      if (b.isFinished()) return;
26.    }
27.    for (Triangle b: tris) {
28.      if (b.isFinished()) return;
29.    }
30.  }
```

演習2. 継承

- Bouncing2.pdeと Traiangle.pde を基にして, 四角形が弾むクラス Rectangle.pde と Bouncing3.pde を作れ.
 - `rect(a, b, c, d)`: (a,b) の位置に幅 c , 高さ d の四角形を描く



3. Mapクラス

■ MapDic.pde

1. `HashMap<String, Integer> dic = new HashMap<String, Integer>();`
2. `dic.put("one", 1);`
3. `dic.put("two", 2);`
4. `dic.put("three", 3);`

5. `println(dic);`
6. `String s[] = {"two", "one", "two", "three"};`
7. `for(int i = 0; i < s.length; ++i){`
8. `print(dic.get(s[i]));`
9. `}`

```
{two=2, one=1, three=3}
two one two three =
2, 1, 2, 3,
```

Mapクラス仕様

■ コンストラクター

- Map <キー型, 値型> 変数 =
new **HashMap**<キー型, 値型>();
- 参考 **TreeMap**<キー型, 値型>(); 2分木
- 型: String, Integer, Float

■ メソッド

- 値型 **put**(,) キーに値を格納
- 値型 **get**() キーを持つ値を参照

もしIf-Thenを使って書くと

- MapDic.pde

```
1. for(String c : argv){  
2.     if(c.equals("one")){  
3.         print(1 + " ");  
4.     }else if(c.equals("two")){  
5.         print(2 + " ");  
6.     }else if(c.equals("three")){  
7.         print(3 + " ");  
8.     }  
9. }
```



演習3

- MapDic.pde を修正して、英数字から漢数字へ変換するMapDic2.pdeを書け。
 - ヒント: dic.putを書き換える。
 - 実行例)
two one two three = 二 一 二 三
 - 数値を増やしたり他の語を登録したりしてみよ

まとめ

- 静的メソッド `int func(int x)` の、`x` を () とい
い、`return` で返される値を () という。
- 継承を利用することで、() クラスのメ
ソッドを再利用し、変更が必要ならば
() して書き換える。
- Mapクラスは、キー `key`、値 `value` の組を、
_____メソッドで登録し、_____メソッドで参
照する(取出す)。

宿題

- 4.1. Ballクラスを拡張して, 星形が弾むクラスを作れ.
- 4.2 漢数字を好きな言語に翻訳する
NumberTrans.pde を書け.
 - 二 一 二 三 = two one two three