
継承 (インヘリタンス)

プログラミング演習Ⅱ (3)

中村, 小松, 菊池

宿題2.1 解答例

■ ObjectBall7.pde

```
1. class Ball {
2.     int x;
3.     int y;
4.     int w;
5.     int r;
6.     int distance(Ball q){
7.         return((int)sqrt((x - q.x)*(x -
8.         q.x) + (y - q.y)*(y - q.y)));
9.     }
10. }
11. int distance(Ball a, Ball b){
12.     return((int) sqrt((a.x - b.x)*(a.x
13.     - b.x) + (a.y - b.y)*(a.y-b.y)));
14. }
```

インスタンスメソッド

```
1. void draw(){
2.     background(255);
3.     p.x = mouseX; p.y =
4.     mouseY;
5.     p.display();
6.     q.display();
7.     q.r = p.distance(q);
8. }
```

静的メソッド

宿題2.2 解答例

■ 例1)

1. `String s = "1,234,000";`
2. `String t = s.substring(0, 1) + s.substring(2,5) + s.substring(6);`
3. `println(s + " * 2 = " + (int(t) * 2));`

■ 例2)

1. `int head = s.length() % 4;`
2. `String tail = s.substring(0, head);`
3. `for(int i = 1; i < s.length() / 4.0; ++i){`
4. `tail += s.substring(head + (i-1)*4 + 1, head + i*4);`
5. `}`
6. `println(s + " * 2 = " + int(tail) * 2);`

■ 例3)

1. `int x2 = int(s.replaceAll(",", ""));`
2. `println(s + " * 2 = " + x2 * 2);`

宿題2.3 解答例

■ ObjectBall8Degree.pde

```
1.   Ball p, q;
2.   void setup(){
3.       size(200, 200);
4.       fill(255, 0, 0);
5.       p = new Ball(100, 30, 20);
6.       q = new Ball(150, 130, 30);
7.   }
8.   void draw(){
9.       background(255);
10.      p.x = mouseX; p.y = mouseY;
11.      p.display(); q.display();
12.      p.r = int(255.0 * 2/PI *
13.      (abs(degree(p, q))));
13.  }
```

```
1.   float degree(Ball p, Ball q){
2.       if(p.x == p.y){ return(PI/2); }
3.       return(atan( (float)(p.y - q.y) /
4.       (float)(p.x - q.x) ));
4.   }
1.   class Ball{
2.       int x; int y; int w; int r;
3.       Ball(int ax, int ay, int aw){
4.           x = ax; y = ay; w = aw; r = 255;
5.       }
6.       void display(){
7.           fill(255, 0, 0, r);
8.           ellipse(x, y, w, w);
9.       }
10.  }
```

概要

- キーワード

- 1. 静的メソッドの復習

- 2. インスタンスメソッドの復習

- 3. クラスの継承, メソッドのオーバーライド

1. クラス Timely

■ クラスの定義

- Class クラス名 {
 インスタンス変数の宣言;
}
- 例) int min, int sec;

■ オブジェクトの生成 (インスタンス化)

- クラス名 変数 = new クラス名();
- 例) Timely a = new Timely();
 Timely b = new Timely();

```
■ Timely.pde
1. class Timely {
2.     int min;
3.     int sec;
4.     Timely(int m, int s) {
5.         min = m;
6.         sec = s;
7.     }
8. }
```

Timelyの表示メソッド

■ Timely2Str.pde

```
1. void setup(){
2.   size(300, 300);
3. }
4. void draw(){
5.   Timely a = new Timely(11, 23);
6.   println("a = " + a.toStr());
7.   println("a = " + toStr(a));
8.   noLoop();
9. }
```

実行結果

```
a = 11:23
a = 11:23
```

■ Timely.pde

メソッド

```
1. String toStr(Timely a){
2.   return( str(a.min) + ":"
3.         + str(a.sec));
3. }
```

```
4. class Timely{
5.   int min, sec;
6.   Timely(int m, int s){
7.     min = m; sec = s;
8.   }
```

メソッド

```
9.   String toStr(){
10.    return(str(min) + ":" +
11.          str(sec));
11.   }
12. }
```

二つのメソッド

| | | インスタンスメソッド |
|----------|--|--|
| 文法 | [static] 戻り値型 メソッド名(引数) クラスの外側で定義 例) static String toStr (Timely a) | class クラス名{ ...(インスタンス変数など) 戻り値型 メソッド名(引数) } 例) String toStr () |
| 呼出 | 条件なし 例) Timely a = new Timely(); String s = toStr (a) | を事前実行する必要がある。 例) Timely a = new Timely(); String s = a. toStr (); |
| 対象オブジェクト | 引数で指定 | で指定, thisは省略可能 |
| 代表例 | sqrt(3.2) | String.indexOf("X") |

インスタンスメソッドの作り方

■ 静的メソッドを基にする

```
1. static String toStr(Timely a){  
2.   return( str(a.min) + ":" + str(a.sec));  
3. }
```

■ Step

- 1. static 宣言を外し, classの中に移動する.
- 2. 引数を一つ削除(オブジェクトにインスタンス化済みなので)
- 3. 引数をすべてthisに置き換える。(省略可能)

```
1. String toStr(){  
2.   return(str(this.min) + ":" + str(this.sec));  
3. }
```

演習1

- Timely2Str.pde を元にして, 時刻オブジェクトを秒に変換する
Timely2Int.pde を書け
 - 秒 ← 分 * 60 + 秒 で変換する.
 - 秒に変換する静的メソッド `int toInt(timely)`
 - 秒に変換するインスタンスメソッド `int toInt()`
 - どちらのメソッドが適しているか?

```
■ Timely2Int.pde
1. void setup(){
2.   size(300, 300);
3. }

4. void draw(){
5.   Timely a = new Timely(8, 13);
6.   println(a.toStr() + " = "
7.   + a.toInt() + " [sec]");
8.   println(a.toStr() + " = "
9.   + toInt(a) + " [sec]");
10. noLoop();
11. }
```

実行結果

8:13 = 493 [sec]

8:13 = 493 [sec]

2. 時刻の和

■ 問題

□ a = 12:34

□ b = 8:42

□ a+b = 21:16

(34秒 + 42秒 = 76秒
= 1分16秒)

(12分 + 8分 + 1分 = 21分)

■ 静的メソッドの例

```
1. Timely plus(Timely x, Timely y){
2.   Timely z = new Timely();
3.   z.sec = x.sec + y.sec;
4.   z.min = x.min + y.min;
5.   if(z.sec >= 60){
6.     z.sec -= 60;
7.     z.min++;
8.   }
9.   return(z);
10. }
```

演習2

- 時刻オブジェクトの加算を行う

Timely4Plus.pde を書け

- インスタンスメソッド
Timely **plus**(Time x,
Timely y)

実行結果

```
a      = 12:34
b      = 8:42
a+b    = 21:16
a+b    = 21:16
```

- Timely4Plus.pde
1. void setup() {
 2. size(300, 300);
 3. }

 4. void draw() {
 5. Timely a = new Timely(12, 34);
 6. Timely b = new Timely(8, 42);
 7. println("a = " + a.toStr());
 8. println("b = " + b.toStr());
 9. println("a+b=" + plus(a, b).toStr());
 10. println("a+b=" + a.plus1(b).toStr());
 11. noLoop();
 12. }

3. 継承とサブクラス

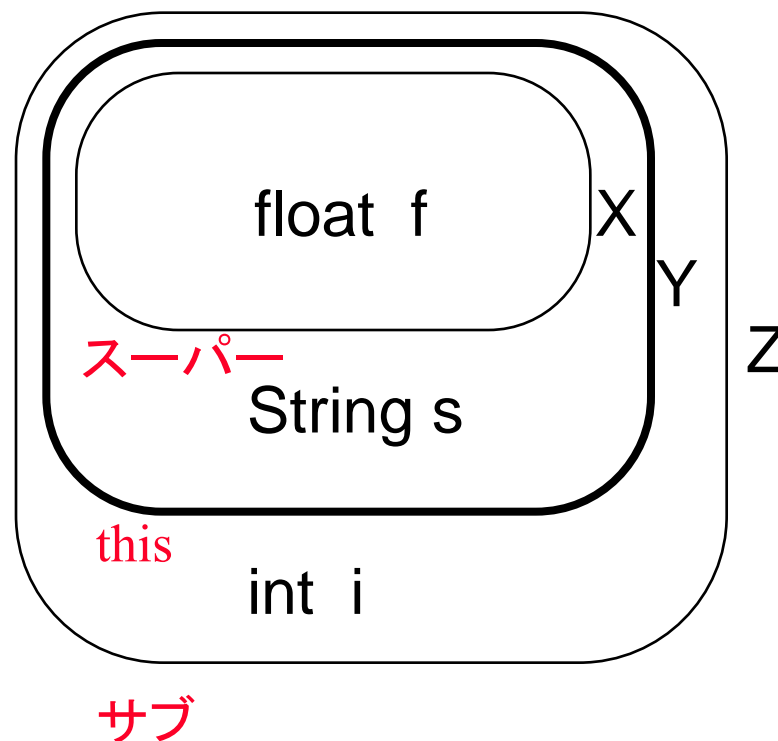
■ XYZ.pde

(p. 181)

```
1. class X{  
2.   float f;  
3. }  
4. class Y extends X{  
5.   String s = "Y";  
6. }  
7. class Z extends Y{  
8.   int i;  
9. }
```

クラスの継承

- サブクラス
 - ZはYの **クラス**
 - YはZの **クラス**
- (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



継承関係の形成

■ クラス継承

(p. 177)

□ class サブクラス名

スーパークラス{

拡張するクラス変数,
拡張するメソッドの定義

}

□ 同じ名前の変数: サブクラスが優先参照

» 例) class E { int x; }

class F extends E {int x}

■ 親クラスの参照

□ **super**.親クラスのメソッド()

メソッドのオーバーライド

■ Hello.pde

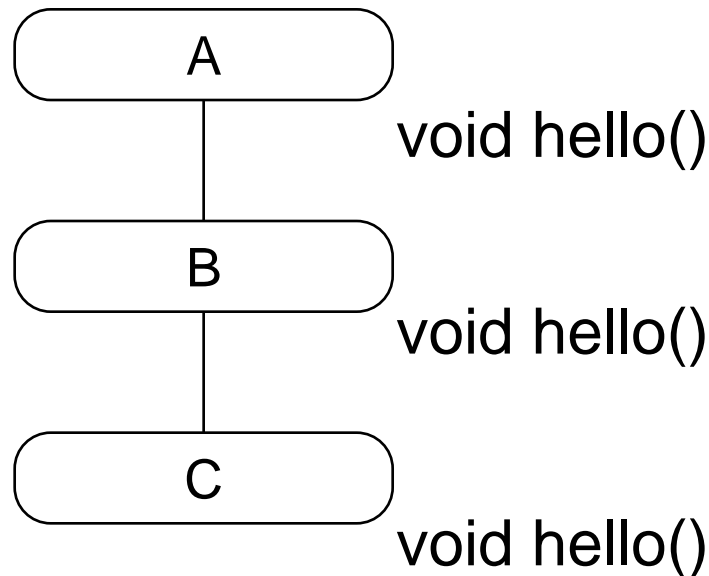
```
1. class A1{
2.   void hello(){
3.     text("Hello from A1", 20, 100);
4.   }
5. }
6. class B1 extends A1{
7.   void hello(){
8.     text("Hello from B1", 20, 100);
9.   }
10. }
11. class C1 extends B1{
12.   void hello(){
13.     text("Hello from C1", 20, 100);
14.   }
15. }
```

■ Override.pde

```
1. void setup(){
2.   size(300, 300);
3.   fill(255, 0, 0);
4.   textSize(36);
5. }
6. void draw(){
7.   C1 a = new C1();
8.   a.hello();
9.   noLoop();
10. }
```

どのhello()が呼ばれるか?

-
- **override** (優越する, 上に重ねる)
 - どのhello()が実行されるだろう?
 - サブクラスでスーパークラスのメソッドを修正できる.



Superの例

■ Zz.pde

```
1. class X{
2.   float f;
3. }
4. class Y extends X{
5.   String s = "Y";
6. }
7. class Z extends Y{
8.   int i;
9. }
10. class Z3 extends Y{
11.   String s = "Z3";
12.   String disp(){
13.     return(super.s);
14.   }
15. }
```

```
16. void setup(){
17.   size(300, 300);
18. }
19. void draw(){
20.   Z z = new Z();
21.   z.f = 4.5;
22.   z.s = "Object Z";
23.   z.i = 10;
24.   println(z.f + "¥t" + z.s + "¥t" + z.i);
25.   Z3 z3 = new Z3();
26.   println("z3.s = " + z3.s);
27.   println("z3.super.s = " + z3.disp());
28. }
```

実行結果

4.5 Object Z 10

z3.s = Z3

z3.super.s = Y

演習3

- Timelyクラスを拡張して、時間 hour を表現できる様にした Hourly クラス を書け
 - toStr(), plus(Hourly y)をオーバーライドせよ.

実行結果

a = 1:12:34

b = 2:48:42

a+b = 4:1:16

- Hourly1.pde
 1. void setup(){
 2. size(300, 300);
 3. }

 4. void draw(){
 5. Hourly a = new Hourly(1, 12, 34);
 6. Hourly b = new Hourly(2, 48, 42);
 7. println("a = " + a.toStr());
 8. println("b = " + b.toStr());
 9. println("a+b = " + a.plus(b).toStr());
 10. noLoop();
 11. }

ヒント

```
■ Hourly.pde 14.
1. class Hourly extends _____ { 15.
2.     int hour; 16.
3.     Hourly(int h, int m, int s){ 17.
4.         hour = h; min = m; sec = s; 18.
5.     } 19.
6.     String toStr(){ 20.
7.         return(str(hour) + ":" + 21.
8.             super.toStr()); 22.         return(z);
9.     } 23.     }
10.     Hourly plus(Hourly y){ 24. }
11.         Hourly z = new Hourly(0,0,0);
12.
13.
```

まとめ

- class A extends Bはクラスの拡張を表す。
AをBの _____
BをAの _____ という。
- クラスAのメソッドをBで書換えることを
_____ とよぶ。
- スーパークラスAのメソッドは、Bにも
_____ (inheritance)されている。BからAのメ
ソッドを呼ぶには、_____ メソッド()を用いる。

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

スタティックメソッド zero

■ Timely1.pde

```
1. void setup() {
2.   size(200, 200);
3. }
4. void draw() {
5.   String s = "";
6.   Timely a = new Timely(2, 13);
7.   s = zero(a.min) + ":"
      + zero(a.sec);
8.   println(s);
9.   noLoop();
10. }
```

■ Timely.pde

```
1. String zero(int x){
2.   if(x < 10){
3.     return("0" + x);
4.   }else{
5.     return("" + x);
6.   }
7. }
```

02:13