クラスとメソッド

プログラミング演習Ⅱ (2)

中村，小松，菊池

参考図書

- 参考書
  - 独習Java 第4版
  - ジョゼフ･オニール (著), 武藤 健志 (監修)
  - 翔泳社(3,200円)
- 本文中のページ数はこの資料のもの

1. 文字列クラス

- 例）

```
1. String s = "NAKANO";
2. println(s);
3. println(s.substring(2));
4. println(s.substring(2,4));
5. println(s.length());
6. int i = s.indexOf("K");
7. println(i);
8. println(s.substring(i));
```

2. Stringクラスの操作

- インスタンス （例）
  - String s = "NETWORK" 文字列の「例」
- 主なインスタンスメソッド （pp. 54-55)
  - メソッド (Method) = （処理）方法
  - String substring(int st, int en) stからenの まで
    - String s = "NETWORK";
    - s.substring(2,5) → "TWO"
  - int length() オブジェクトの 数
    - s.length() → 7
  - String toLowerCase() 大文字へ変換
    - s.toLowerCase() → "network"

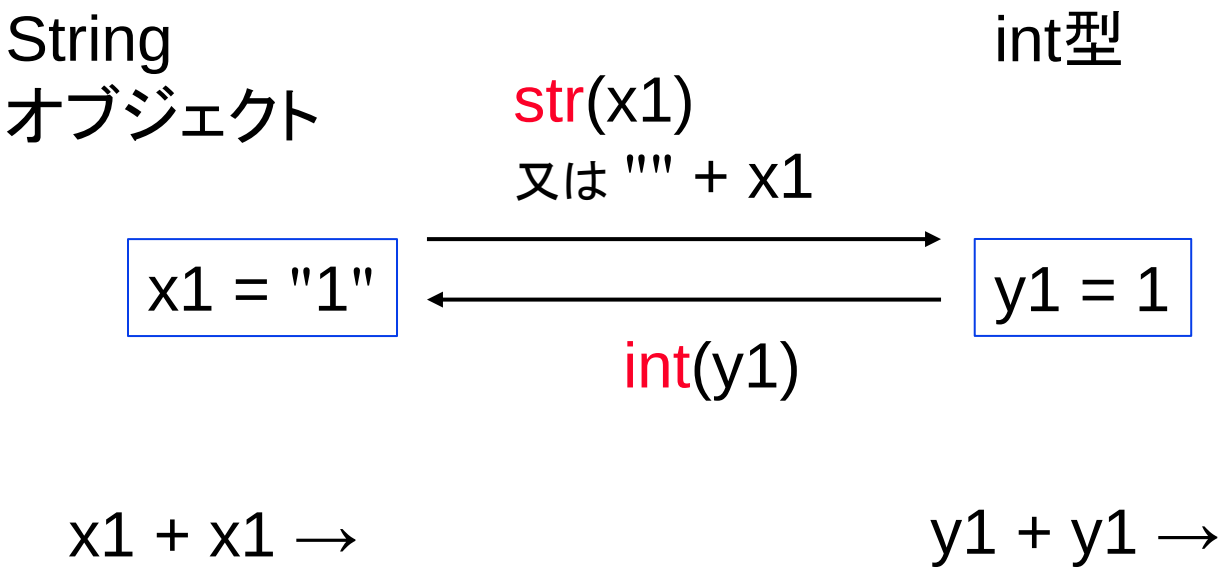
文字と数値

- 文字列の連結
  - ❑ + 演算子

  例1) "NET" + "WORK" = "NETWORK"
  例2) "Oct" + 10 = "Oct 10"
  例3) int x = 3; "Three = " + x　　= "Three = 3"
- 次の出力を予測せよ

```
String x1 = "1";
String x2 = "2";
int    y1 = 1;
int    y2 = 2;
System.out.println(x1 + x2);
System.out.println(y1 + y2);
System.out.println(y1 + y2 + y2);
System.out.println("y1 + y2 = " + y1 + y2);
System.out.println("y1 + y2 + y2= " + y1 + y2 + y2);
```

String型とint型の変換



演習1 文字列の操作

- 次の空欄を埋めて，指定されたように出力するように完成させよ．提出ファイル名 FMS.pde

```
String a = "明治大学2013年中野にFMS学科登場!";

println(a);
println("Len = " +                  ); // 文字数
println(                            ); // 明治大学
println(                            ); // 2013
int x =                ;      // FMSの位置
println("FMS is " + x + "th char of a."); // 12
println(        + "先端メディアサイエンス" +     );  // FMSの部分を正式名称にする.
```

3. オブジェクト

- new演算子 (p.60)
  - クラス名 変数 = クラス名(初期化引数）
  - 例）
    - » char data[] = {'a', 'b', 'c'};
    - » String s = new String(data);
    - » String s = "abc";
  - 例）ラップクラス
    - » Integer ix = new Intger(5);
    - » int型のオブジェクト.「ラップ（ ）」

コンストラクタ

- オブジェクトの生成
  - String s = new **String**({‘a’, ‘b’, ‘c’});

  - インスタンス化:「　　」＝メモリ※へ実割当
  - インスタンス(instance):「具体化したオブジェクト」= s
  - 　　(constructor)「構成器」
    クラス名と同じ名前 = String(初期化引数)

※メモリ: データを格納する領域

オブジェクトの比較

- 例）
  - String s が “ABE”かどうかテストしたい.
- インスタンスメソッド
  - boolean equals(Object s)
    - » (現在の）オブジェクトとsが同じなら真，違えば偽を返す
  - http://java.sun.com/javase/ja/6/docs/ja/api/java/lang/String.html#equals(java.lang.Object)
  - 例）

    String a = "ABE";

    a.equals("ABC")　　"ABE" == "ABC"

    a.equals("abe")

演習2 文字列の比較

- Strcmp.pde の実行を予測し，コンパイルして確かめ，全てtrueを表示するように書換えよ.
  提出ファイル名 Strcmp.pde

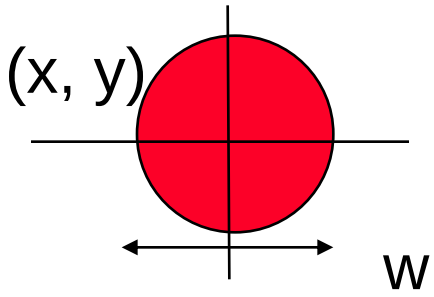
```
1.  String s1 = "Akito";
2.  String s2 = "Takuma";
3.  String s3 = new String("Akito");
4.  String s4 = s1.substring(0);
5.  String s5 = s1.substring(0, 1) + s1.substring(1);

6.  println("s1 == s2: " + (s1 == s2));
7.  println("s1 == s3: " + (s1 == s3));
8.  println("s1 == s4: " + (s1 == s4));
9.  println("s1 == s5: " + (s1 == s5));
```

- true (真)，false (偽)を表す論理値　(boolean)

4. クラスの作成

- Ballクラス　赤い円のオブジェクト

```
1.  class Ball {
2.    int  x;
3.    int  y;
4.    int  w;
5.  }

6.  void setup() {
7.    size(200, 200);
8.    fill(255, 0, 0);
9.  }
10. void draw() {
11.   background(255);
12.   Ball  p = new Ball();
13.   p.x = 100;
14.   p.y = 30;
15.   p.w = 20;
16.   ellipse(p.x,  p.y,  p.w,  p.w);
17. }
```

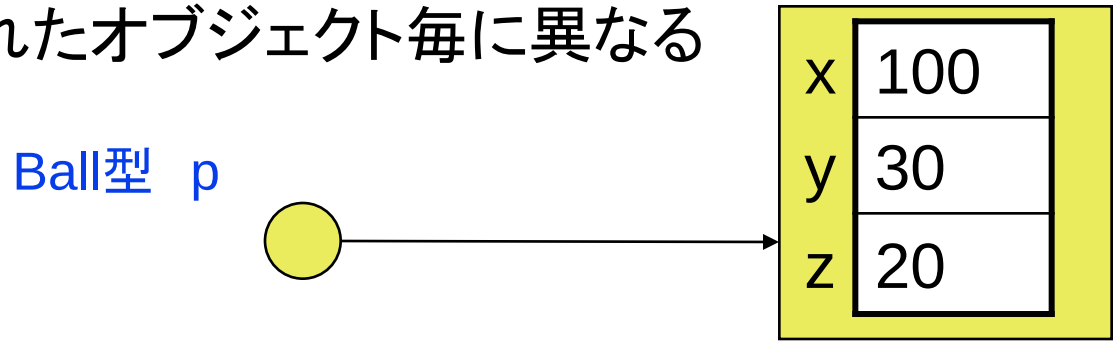


クラス (p.136)

- クラスの構成要素
  - クラス変数
  - メソッド
  - コンストラクタ
- 基本文法
  - class クラス名{
    型　　変数名 ;
    クラス名 {　初期化処理 }
    型　メソッド名(引数){ メソッド処理 }
    }

変数　(p. 150)

- オブジェクトに結びついた変数
  - class Ball{
    int x; int y; int w;
    }
  - 生成されたオブジェクト毎に異なる

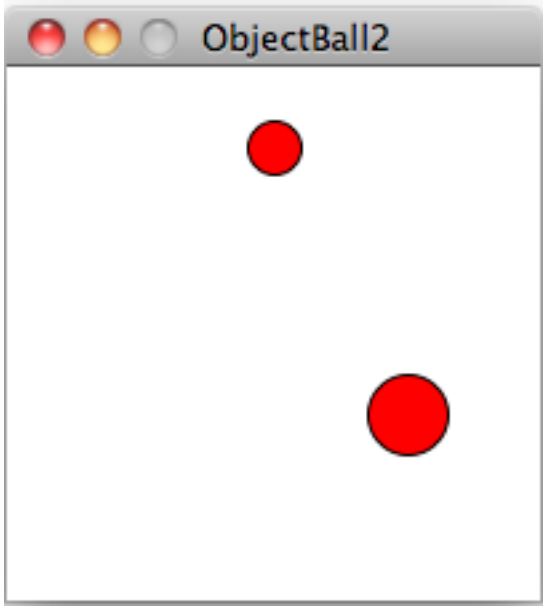


  - 参考）　局所（ローカル）変数，静的変数

インスタンス化

- ObjectBall2.pde

```
1.  void draw(){
2.    background(255);
3.    Ball  p = new Ball();
4.    Ball  q = new Ball();
5.    p.x = 100;  p.y =   30; p.w = 20;
6.    q.x = 150;  q.y = 130; q.w = 30;
7.    ellipse(p.x,  p.y,  p.w,  p.w);
8.    ellipse(q.x,  q.y,  q.w,  q.w);
9.  }
```



の追加

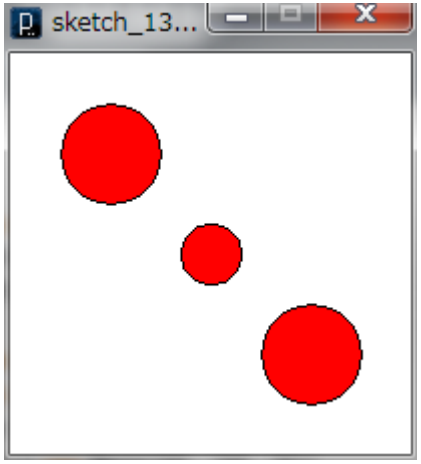
- ObjectBall3.pde

```
1.  class Ball{
2.    int  x;
3.    int  y;
4.    int  w;

5.    Ball(int ax, int ay, int aw){
6.      x = ax;
7.      y = ay;
8.      w = aw;
9.    }
10. }
```

```
11. void setup(){
12.   size(200, 200);
13.   fill(255, 0, 0);
14. }
15. void draw(){
16.   background(255);
17.   Ball  p = new Ball(100, 30, 20);
18.   Ball  q = new Ball(150, 130, 30);
19.   ellipse(p.x,  p.y,  p.w,  p.w);
20.   ellipse(q.x,  q.y,  q.w,  q.w);
21.   noLoop();
22. }
```

演習3 インスタンス化

- ObjectBall3.pde
  - ObjectBall3.pdeを，次のように表示する様に書き換えよ.
  - 実行例）



5. 静的メソッドの追加

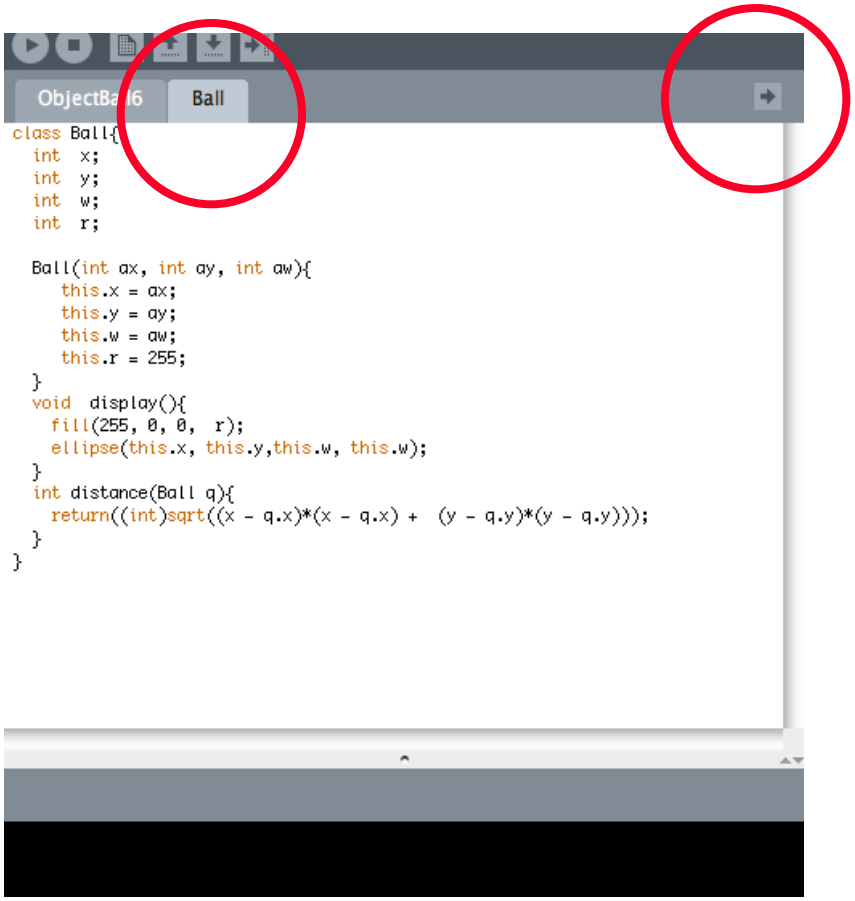
- ObjectBall4.pde

```
1. void display(Ball p){
2.   ellipse(p.x, p.y, p.w, p.w);
3. }

4. void setup(){
5.    size(200, 200);
6.   fill(255, 0, 0);
7. }

8. void draw(){
9.   background(255);
10.  Ball  p = new Ball(100, 30, 20);
11.  Ball  q = new Ball(150, 130, 30);
12.  display(p);
13.  display(q);
14.  noLoop();
15. }
```

```
1.  class Ball{
2.    int  x;
3.    int  y;
4.    int  w;
5.
6.    Ball(int ax, int ay, int aw){
7.       x = ax;
8.       y = ay;
9.       w = aw;
10.   }
11. }
```

クラスをサブタブへ.

- ウィンドウメニューをクリック
  - →New Tab
  - タブBall：Ballクラスのソース
  - タブObjectBall: それ以外の，setup(), draw()メソッド，静的メソッドの定義
  - 分けなくても動作する．分かりやすさの為.



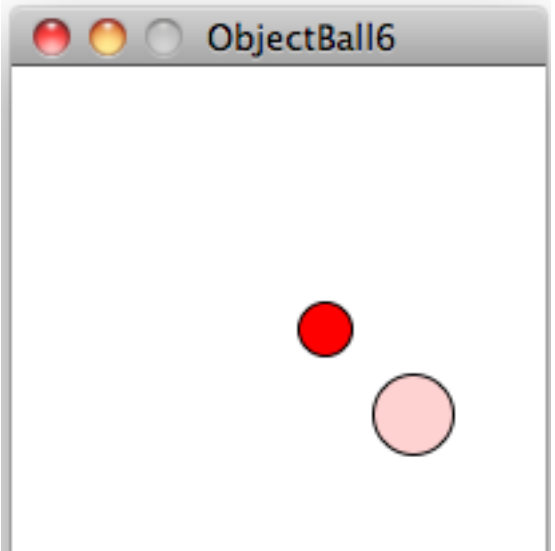
インスタンスメソッドの追加

- ObjectBall5.pde

```
1.  void setup(){
2.     size(200, 200);
3.     smooth();
4.    fill(255, 0, 0);
5.  }

6.  void draw(){
7.    background(255);
8.    Ball  p = new Ball(100, 30, 20);
9.    Ball  q = new Ball(150, 130, 30);
10.   p.display();
11.   q.display();
12. }

13. class Ball{
14.   int  x;
15.   int  y;
16.   int  w;
17.
18.   Ball(int ax, int ay, int aw){
19.     this.x = ax;
20.     this.y = ay;
21.     this.w = aw;
22.   }
23.   void  display(){
24.     ellipse(x, y, w, w);
25.   }
26. }
```

演習4. 静的メソッドの追加

- ObjectBall6.pde
  - 小さなボールはマウスカーソルに追従して移動する.
  - 大きなボールに近づいたら大きなボールの色が薄くなる. (fill(255,0,0,透明度))
  - ボール間の距離を求める静的メソッド
    int distance(Ball p, Ball q);
    を作れ. $\sqrt{x^2+y^2}$



ヒント　（プログラムの一部のみ）

- Ball.pde

```
1.  class Ball {
2.    int  x;
3.    int  y;
4.    int  w;
5.    int  r;

6.    void  display() {
7.      fill(255, 0, 0,  r);
8.      ellipse(x, y, w, w);
9.    }
10. }
```

- ObjectBall6.pde

```
1.  void draw() {
2.    background(255);
3.    p.x = mouseX;
4.    p.y = mouseY;
5.    p.display();
6.    q.display();
7.    q.r =
8.    println("dist = " +  q.r);
9.  }
10. int distance(Ball a, Ball b){
11.   return(                    );
12. }
```

まとめ

- 次の概念を学んだ
  - オブジェクトを作成すること＝
  - 具体化されたオブジェクト＝
  - 作成するための構成器＝
  - オブジェクトの変数＝
  - 2種類のメソッド＝

宿題

- 2.1 ObjectBall7.pde の静的メソッド distance() をインスタンスメソッドに書き換えた ObjectBall8.pde を書け.
- 2.2 カンマ付きの通貨（例1,200円）の文字列を2倍する Double.pde を書け.　(ヒント: indexOf() や split())
  - 例） 1,234,000 * 2 = 2468000
- 2.3 2つのボールの間の角度を求める静的メソッド degree(Ball a, Ball b) = atan(Δy/Δx) を書け.（角度に応じて小さなボールの色の濃さが増える様にせよ.
  fill(255, 0, 0, r);

$$r = 255.0 \cdot \frac{2}{\pi}|\Delta(p,q)|$$