

## 今回のテーマ

オブジェクト指向プログラミング  
衝突判定、跳ね返り  
ソーティング（並び替え）アルゴリズム

前回のやり残し（キーボードイベント、テキスト表示）

## オブジェクト指向プログラミングの続き クラスの拡張

### 練習問題 1 :

Areaクラスを拡張して、どのAreaオブジェクト内にマウスポインタがあるか判定できるようにする。  
Area クラスにinsideメソッドを追加し、マウスポインタのある場所のAreaオブジェクトの色が変わるようにする。  
マウスポインタのある座標値をすべてのAreaオブジェクトに対してinsideメソッドで確認すれば、どのオブジェクトのエリア内にマウスポインタがあるか判定出来る。

```
class Area {  
  
    boolean inside(float x, float y){  
        // 描画範囲内にあったら、true  
        // そうじゃなかったら、falseを返す  
    }  
}
```

### 衝突判定と反射

ウインドウの枠に到達したら、跳ね返るようにする  
既に前回、repeatH, repeatVメソッドの作成において、  
ウインドウの枠に達した事の判定を行っているが、さらに  
斜め方向にぶつかった場合も考慮した跳ね返りを実現する。  
ウインドウのサイズは、width, height 変数内に自動的に入っている。

### 練習問題 2 :

円を描いて、ランダムな方向に一定速度で動くようなアニメーションを作成する。  
方向を決定するのは、進行方向の角度をランダムで決定し、その値から  
sin, cosを使って、x方向、y方向の変化量を決定する。

計算

移動速度は、適当な変数、speedなど、を使って調整できるようにしておくが良い。

---

```
float cx, cy;  
float dx, dy;  
float angle;  
float speed;  
color c;
```

```
void setup(){  
  background(0);  
  size(400,400);  
  c = color(random(255), random(255),random(255));  
  stroke(c);  
  fill(c);  
  speed = 10;  
}
```

```
void draw(){  
  必要な処理を記述  
}
```

```
void mousePressed(){  
  無くてもかまわない。  
}
```

---

### 練習問題3：

練習問題2を改良し、ウインドウの枠に達したら、跳ね返るようにする。  
跳ね返るときにどのような変化が起きるか？  
普通は、進行方向を示す変数も必要となる。  
図に描いて確認してみるとわかりやすい。

## 授業内課題 1 :

練習問題 1 のマウスポインタの位置の代わりに練習問題 3 で作った反射する円を利用して、円の場所にあるエリアオブジェクトの色が変化するようにする。

## ソーティング

頭の体操として、並べ替えの方法を考えてみる。

10 個の数値があるときに小さい順に並べ替える。

色々な方法がありますが、今回は以下のような方法を用いるとする。

1. 10 個のランダムな数値が適当な順番に並んでいるとする。  
配列で考えれば、配列の 0 番目から 9 番目にランダムな数値が入っていると考えることができる。
2. 数値の 0 番目の値と 1 番目の値を比較して、0 番目の値の方が大きければ、0 番目の値と 1 番目の値を入れ替える。
3. 同様にして、1 番目と 2 番目の比較、2 番目と 3 番目の比較と続け、8 番目と 9 番目まで実行する。(比較して大きければ入れ替え)
4. ここまでで、9 番目に一番大きい値が入っている。
5. 同様に 0 番目と 1 番目の比較から始め、7 番目と 8 番目まで比較していく。2 回目なので、比較の回数が一回減ることに注意。
6. これで、8 番目に次に大きい値が入る。
7. これを、繰り返すと毎回比較の回数が減りながら、最後まで行くと並べ替えが終了する。

**0 1 2 3 4 5 6 7 8 9**

□□□□□□□□□□

まずは、紙にでも書いて仕組みを理解しましょう。

## 練習問題 4 :

10 個の大きさを持つ int 型の配列を用意し、ランダムな値を入れる。初期状態を出力したあと、上のアルゴリズムを使って並べ替えを行い、結果を出力する。

----

```
int[] ia = new int[10];  
PFont font;
```

```

void setup(){
  size(400,400);
  font = loadFont("Monospaced-48.vlw");
  textFont(font,24);

  for(int i = 0; i < 10 ; i++){
    ia[i] = int(random(1000));
    String str = nf(ia[i], 4);
    text(str, 10, i * 25 + 25);
  }

  // 並べ替えを実施
  // 二重ループを使う

  for(int i = 0; i < 10 ; i++){
    String str = nf(ia[i], 4);
    text(str, 210, i * 25 + 25);
  }
}
---
```

## 授業内課題 2 :

Areaオブジェクトを10個使って、ソーティングの様子をアニメーションにする。  
Areaオブジェクトの色をソートするための数値として利用して、  
練習問題4と同様なソーティングをdraw()内に記述する事で、  
色が並べ替える様子が見えるようにする。  
色のうち、特定の要素（RGBならRだけ、HSBならBだけ）に注目して  
並べ替えれば、変化がわかりやすい。  
frameRate(10);  
程度にゆっくりと見せれば、徐々に並べ替えられていく様子が見えるでしょう。  
並べ終わったら、noLoop()でループを終了する。

## 課題 :

次回は、ネットワークを使って、プログラム同士でやり取りができるようにします。  
これまでに習ったことをベースに、ネットワーク越しに対戦が出来るゲーム、  
あるいは、ネットワークを使う事でインタラクティブに変化していく表現を考えて  
次回までにメールで送る事。

あまりに内容がかぶるようなら再考してもらう場合もあります。

条件としては、

- オブジェクト指向プログラミングによって実現する事。
  - Areaクラス、またはそれを拡張したものを利用する事。
  - Tenクラスのサブクラスを作成して、円にしたり、必要に応じて機能を加えていくのは自由です。
- ネットワーク越しに2つ以上のプログラムが無いと実現出来ない内容であること。
  - 2つ以上のプログラムは同一のものが複数であってもよいし、サーバ的なものとクライアントの違うプログラムが連携するものでもよい。
- キーボードやマウスなどによって変化するインタラクティブなものであること。