class化

1 もっとも簡単なversion

もとは関数型のrobotとしましょう.

Listing 1: robot5_functionのhead部.

```
void setup() {
    size(720, 480);
    strokeWeight(2);
    ellipseMode(RADIUS);
}

void draw() {
    background(0, 0, 255);
    drawRobot(120,420,110,140);
    drawRobot(270,460,260,95);
}

void drawRobot(int x, int y, int bodyHeight, int neckHeight){
  int radius = 45;
  int ny = y - bodyHeight - neckHeight -radius;

  ... 中略 ...
}
```

これをclass化するとつぎのとおりになります.

Listing 2: robotx_classのhead部-I.

```
Robot myRobot1;                          // step 4

void setup() {
    size(720, 480);
    strokeWeight(2);
    ellipseMode(RADIUS);
    myRobot1 = new Robot(120,420,110,140); // step 5
}

void draw() {
    background(0, 0, 255);
    myRobot1.display();                  // step 6
}
```



```
class Robot {                            // step 1
                                         // step 3
  int x,y;
  int bodyHeight, neckHeight;
  int radius = 45;

                                         // step 2
  Robot(int a_x, int a_y, int a_bodyHeight, int a_neckHeight){
    x = a_x;
    y = a_y;
    bodyHeight = a_bodyHeight;
    neckHeight = a_neckHeight;
  }

  void display(){
    int ny = y - bodyHeight - neckHeight -radius;
  ... 中略 ...
  }
}
```

つまりclass化する手順としては

1. class名を決めてclass宣言する
2. 同じ名前の初期化関数をつくる
3. class変数をつける
4. 変数(Robot)を宣言(Robot myRobot)する
5. newする.
6. displayする.

です.

2 classにする意義=よく似ているけど異なる振る舞い

このような簡単な例をみると, その機能を実現するには functionで十分のような気がします. しかし, たくさんのrobotをつくって, それぞれに違う「振る舞い」をさせたいとします. 例えば, myRobot1はそのまま止まっていて, myRobot2はmouseXでeasingさせたいとします. そうするとcodeは次のようになります.

Listing 3: robotx_classのhead部-II.

```
Robot myRobot1;
Robot myRobot2;   // myRobot2

void setup() {
```

```
void setup() {
    size(720, 480);
    strokeWeight(2);
    ellipseMode(RADIUS);
    myRobot1 = new Robot(120,420,110,140);
    myRobot2 = new Robot(240,470,220,70);    // myRobot2
}

void draw() {
    background(0, 0, 255);
    myRobot1.display();
    myRobot2.easing();   // myRobot2
    myRobot2.display();    // myRobot2
}

class Robot {
  ... 中略 ...

  float easing = 0.02;
  void easing(){
    int targetX = mouseX;
    x += (targetX-x)*easing;
  }

  ... 以下略 ...
```

myRobot1もeasing機能をもっているのですが，それをcodeで書いてないのでeasingしません．あるいはランダムに動くやつとか，clickに反応するやつとか，ほとんどの振る舞いやcodeは共通なのに，それぞれ異なった「振る舞い」をさせることが可能になるんです．便利でしょ?

3 自分のものにするコツ

いろんなexamplesをみて，自分の想像を膨らませてください．自分のコードにするときのコツは，まず

> どんなものにどんなふるまいをさせたいか

を自分で決めておくことです．

「こういう振る舞いをさせたいので，ある関数を組み込んで パラメータをいじりました」だといいんですが，「面白いサンプルがあったので，それをコピペして

parametersをいじりました」では，単なる劣化版にしかならないですよ．

あと，何を参考にしたかの引用を忘れないように．自分のcredit(license)とともにheadに書き込んでおいてください．