明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

プログラミング演習2
クラスと継承とArrayList

中村，高橋，小林，橋本

Ball/Cross/Triangle

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

- 3つのクラスを作って、3つのオブジェクトをそれぞれ描画
  - Ball
  - Cross
  - Triangle

  - SampleBCT.zip をダウンロードして下さい

Ballクラス

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

```
class Ball
{
  float posX;
  float posY;
  float speedX;
  float speedY;

  Ball(){
    posX = random(width);
    posY = random(height);
    speedX = random(-5, 5);
    speedY = random(-5, 5);
  }

  void display(){
    fill(255, 0, 0);
    ellipse( posX, posY, 30, 30 );
  }
```

```
  void move(){
    posX += speedX;
    posY += speedY;
    if(posX > width){
      posX = width * 2 - posX;
      speedX = -speedX;
    }
    if(posX < 0){
      posX = -posX;
      speedX = -speedX;
    }
    if(posY > height){
      posY = height * 2 - posY;
      speedY = -speedY;
    }
    if(posY < 0){
      posY = -posY;
      speedY = -speedY;
    }
  }
}
```

Ballを改良しCrossを作る

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

```
class Cross
{
  float posX;
  float posY;
  float speedX;
  float speedY;

  Cross(){
    posX = random(width);
    posY = random(height);
    speedX = random(-5, 5);
    speedY = random(-5, 5);
  }

  void display(){
    line(posX-15, posY-15, posX+15, posY+15);
    line(posX+15, posY-15, posX-15, posY+15);
  }
```

```
  void move(){
    posX += speedX;
    posY += speedY;
    if(posX > width){
      posX = posX - width;
    }
    if(posX < 0){
      posX = posX + width;
    }
    if(posY > height){
      posY = posY - height;
    }
    if(posY < 0){
      posY = posX + height;
    }
  }
}
```

同じくTriangleを作る

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

```
class Triangle
{
  float posX;
  float posY;
  float speedX;
  float speedY;

  Triangle(){
    posX = random(width);
    posY = random(height);
    speedX = random(-5, 5);
    speedY = random(-5, 5);
  }

  void display(){
    triangle(posX, posY-15, posX-15,
      posY+15, posX+15, posY+15);
  }
```

```
  void move(){
    posX += speedX;
    posY += speedY;
    if(posX > width){
      posX = posX - width;
    }
    if(posX < 0){
      posX = posX + width;
    }
    if(posY > height){
      posY = posY - height;
    }
    if(posY < 0){
      posY = posX + height;
    }
  }
}
```

Cross/Triangle

```
Ball ball;
Cross cross;
Triangle triangle;

void setup() {
  size(600, 600);
  ball = new Ball();
  cross = new Cross();
  triangle = new Triangle();
}

void draw() {
  background(255);

  ball.move();
  cross.move();
  triangle.move();
  ball.display();
  cross.display();
  triangle.display();
}
```

Ball/Cross/Triangle

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室


```
class Ball
{
  float posX;
  float posY;
  float speedX;
  float speedY;

  Ball(){
    posX = random(width);
    posY = random(height);
    speedX = random(-5, 5)
    speedY = random(-5, 5)
  }

  void display(){
    fill(255, 0, 0);
    ellipse( posX, posY, 3
  }
```

```
class Cross
{
  float posX;
  float posY;
  float speedX;
  float speedY;

  Cross(){
    posX = random(width);
    posY = random(height)
    speedX = random(-5, 5
    speedY = random(-5, 5
  }

  void display(){
    line(posX-15, posY-15
    line(posX+15, posY-15
  }
```

```
class Triangle
{
  float posX;
  float posY;
  float speedX;
  float speedY;

  Triangle(){
    posX = random(width);
    posY = random(height);
    speedX = random(-5, 5);
    speedY = random(-5, 5);
  }

  void display(){
    triangle(posX, posY-15,
      posY+15, posX+15, pos
  }
```

Ball/Cross/Triangle

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室


```
class Ball
{
  float posX;
  float posY;
  float speedX;
  float speedY;

  Ball(){
    posX = random(width);
    posY = random(height);
    speedX = random(-5, 5)
    speedY = random(-5, 5)
  }

  void display(){
    fill(255, 0, 0);
    ellipse( posX, posY, 3
  }
```

```
class Cross
{
  float posX;
  float posY;
  float speedX;
  float speedY;

  Cross(){
    posX = random(width);
    posY = random(height)
    speedX = random(-5, 5
    speedY = random(-5, 5
  }

  void display(){
    line(posX-15, posY-15
    line(posX+15, posY-15
  }
```

```
class Triangle
{
  float posX;
  float posY;
  float speedX;
  float speedY;

  Triangle(){
    posX = random(width);
    posY = random(height);
    speedX = random(-5, 5);
    speedY = random(-5, 5);
  }

  void display(){
    triangle(posX, posY-15,
      posY+15, posX+15, pos
  }
```

Ball/Cross/Triangle

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

```
  void move(){
    posX += speedX;
    posY += speedY;
    if(posX > width){
      posX = width * 2 - p
      speedX = -speedX;
    }
    if(posX < 0){
      posX = -posX;
      speedX = -speedX;
    }
    if(posY > height){
      posY = height * 2 -
      speedY = -speedY;
    }
    if(posY < 0){
      posY = -posY;
      speedY = -speedY;
    }
  }
}
```

```
  void move(){
    posX += speedX;
    posY += speedY;
    if(posX > width){
      posX = posX - width;
    }
    if(posX < 0){
      posX = posX + width;
    }
    if(posY > height){
      posY = posY - height
    }
    if(posY < 0){
      posY = posX + height
    }
  }
}
```

```
  void move(){
    posX += speedX;
    posY += speedY;
    if(posX > width){
      posX = posX - width;
    }
    if(posX < 0){
      posX = posX + width;
    }
    if(posY > height){
      posY = posY - height;
    }
    if(posY < 0){
      posY = posX + height;
    }
  }
}
```

Ball/Cross/Triangle

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

```
  void move(){
    posX += speedX;
    posY += speedY;
    if(posX > width){
      posX = width * 2 - p
      speedX = -speedX;
    }
    if(posX < 0){
      posX = -posX;
      speedX = -speedX;
    }
    if(posY > height){
      posY = height * 2 -
      speedY = -speedY;
    }
    if(posY < 0){
      posY = -posY;
      speedY = -speedY;
    }
  }
}
```

```
  void move(){
    posX += speedX;
    posY += speedY;
    if(posX > width){
      posX = posX - width;
    }
    if(posX < 0){
      posX = posX + width;
    }
    if(posY > height){
      posY = posY - height
    }
    if(posY < 0){
      posY = posX + height
    }
  }
}
```

```
  void move(){
    posX += speedX;
    posY += speedY;
    if(posX > width){
      posX = posX - width;
    }
    if(posX < 0){
      posX = posX + width;
    }
    if(posY > height){
      posY = posY - height;
    }
    if(posY < 0){
      posY = posX + height;
    }
  }
}
```

Ball/Cross/Triangle

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

- Ballクラスと，Crossクラスと，Triangleクラスは似ている

クラス名	Ball	Cross	Triangle	
インスタンス変数	posX，posY speedX，speedY	posX，posY speedX，speedY	posX，posY speedX，speedY	一致
コンストラクタ	Ball() 座標速度初期化	Cross() 座標速度初期化	Triangle() 座標速度初期化	名は違うが 内部は一致
void move()	はねかえる	はねかえらない	はねかえらない	一部一致
void display()	円を描く	xを描く	△を描く	違う

無駄じゃね？
もっと手軽にできないの？

どう無駄をなくす？

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

- そもそもBallクラスというのがダメなのでは？
  - Objectクラスという名前にして、objectTypeなどの変数を用意し、displayの時に切り替えては？

```
class Object {
  float posX;
  float posY;
  float speedX;
  float speedY;
  int objectType;

  void display(){
    if(objectType == 0){
      ellipse(posX, posY, 30, 30);
    } else if(objectType == 1){
      rect(posX, posY, 30, 30);
    }
  }
```

これも一つの方法だが
跳ね返りでも条件が必要で
複雑なものだと厳しくなる

そこで継承！

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

- 大辞林 第三版
  1. 先の人の身分・権利・義務・財産などを受け継ぐこと。「王位を－する」
  2. インヘリタンス→（オブジェクト指向プログラミングにおいて，クラス間でデータの共有を行う機構。新しく定義するクラスを既存のクラスの下位クラスとして記述し，上位クラスより属性やメソッドを引き継ぐ仕組みをいう。上位クラスに対する差分のみを記述するだけで新しいクラスを定義することが可能となる。）

一緒の部分をまとめた
スーパークラス（親クラス）を作る

課題3-2: basic_boundAll3

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

- 講義中に扱ったObjectBaseクラスを継承し，赤色の○が動き回るBallクラスと，緑色の△が動き回るTriangleクラス，青色の□が動き回るSquareクラスを作成せよ
- ただし，○は上下左右の壁で跳ね返り，△は上下左右で逆から出てくるように，□は左右で跳ね返り，上下では逆から出てくるようにせよ
- また，この継承したクラスを利用して800x600のウインドウ内を100個の赤色○と，60個の緑色△と，80個の青色□を描画せよ

ここで・・・

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

- ObjectBaseというクラスを作る
  - その機能を，BallやCross，Triangleに提供する

クラス名	ObjectBase	Ball	Cross	Triangle
インスタンス変数	posX，posY speedX，speedY	posX，posY speedX，speedY	posX，posY speedX，speedY	posX，posY speedX，speedY
コンストラクタ	ObjectBase() 座標速度初期化	Ball() 座標速度初期化	Cross() 座標速度初期化	Triangle() 座標速度初期化
void move()	はねかえらない	はねかえる	はねかえらない	はねかえらない
void display()	何も描かない	円を描く	xを描く	△を描く

ObjectBaseクラス

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

```
class ObjectBase {
  float posX;
  float posY;
  float speedX;
  float speedY;

  ObjectBase(){
    posX = random(width);
    posY = random(height);
    speedX = random(-5, 5);
    speedY = random(-5, 5);
  }

  void display(){
  }
```

```
  void move(){
    posX = posX + speedX;
    posY = posY + speedY;
    if(posX > width){
      posX = posX - width;
    }
    if(posX < 0){
      posX = posX + width;
    }
    if(posY > height){
      posY = posY - height;
    }
    if(posY < 0){
      posY = posY + height;
    }
  }
}
```

display() は何もしないので空っぽにする

ObjectBaseとetc

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室


```
class ObjectBase
{
  float posX;
  float posY;
  float speedX;
  float speedY;

  ObjectBase(){
    posX = random(w
    posY = random(h
    speedX = random
    speedY = random
  }

  void display(){


  }
```

```
class Ball
{
  float posX;
  float posY;
  float speedX;
  float speedY;

  Ball(){
    posX = random(w
    posY = random(h
    speedX = random
    speedY = random
  }

  void display(){
    fill(255, 0, 0)
    ellipse( posX,
  }
```

```
class Cross
{
  float posX;
  float posY;
  float speedX;
  float speedY;

  Cross(){
    posX = random(w
    posY = random(h
    speedX = random
    speedY = random
  }

  void display(){
    line(posX-15, p
    line(posX+15, p
  }
```

```
class Triangle
{
  float posX;
  float posY;
  float speedX;
  float speedY;

  Triangle(){
    posX = random(w
    posY = random(h
    speedX = random
    speedY = random
  }

  void display(){
    triangle(posX,
      posY+15, posX
  }
```

ObjectBaseとetc

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室


```
  void move(){
    posX += speedX;
    posY += speedY;
    if(posX > width)
      posX = posX -
    }
    if(posX < 0){
      posX = posX +
    }
    if(posY > heigh
      posY = posY -
    }
    if(posY < 0){
      posY = posX +
    }



  }
}
```

```
  void move(){
    posX += speedX;
    posY += speedY;
    if(posX > width)
      posX = width -
      speedX = -spee
    }
    if(posX < 0){
      posX = -posX;
      speedX = -spee
    }
    if(posY > heigh
      posY = height
      speedY = -spee
    }
    if(posY < 0){
      posY = -posY;
      speedY = -spee
    }
  }
}
```

```
  void move(){
    posX += speedX;
    posY += speedY;
    if(posX > width
      posX = posX -
    }
    if(posX < 0){
      posX = posX +
    }
    if(posY > heigh
      posY = posY -
    }
    if(posY < 0){
      posY = posX +
    }



  }
}
```

```
  void move(){
    posX += speedX;
    posY += speedY;
    if(posX > width
      posX = posX -
    }
    if(posX < 0){
      posX = posX +
    }
    if(posY > heigh
      posY = posY -
    }
    if(posY < 0){
      posY = posX +
    }



  }
}
```

一緒の部分をまとめる

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

- CrossクラスはObjectBaseクラスの変数（posX，posY，speedX，speedY）や機能（移動や初期化）をもち，独自のバツの描画の機能をもつ
- TriangleクラスはObjectBaseクラスの変数（posX，posY，speedX，speedY）や機能（移動や初期化）をもち，独自の三角形描画の機能をもつ
- BallクラスはObjectBaseクラスの変数（posX，posY，speedX，speedY）や機能（初期化）をもち，独自のはねかえる移動と独自の円の描画の機能をもつ

- インスタンス変数や，インスタンスメソッドを引き継ぐことを継承と呼ぶ！

一緒の部分をまとめる

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

- Crossクラスは ObjectBaseクラスの変数（posX，posY，speedX，speedY）や機能（移動や初期化）をもち，独自のバツの描画の機能をもつ
- Triangleクラスは ObjectBaseクラスの変数（posX，posY，speedX，speedY）や機能（移動や初期化）をもち，独自の三角形描画の機能をもつ
- Ballクラスは ObjectBaseクラスの変数（posX，posY，speedX，speedY）や機能（初期化）をもち，独自のはねかえる移動と独自の円の描画の機能をもつ

- インスタンス変数や，インスタンスメソッドを引き継ぐことを継承と呼ぶ！

ObjectBaseとetc

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室


```
class ObjectBase
{
  float posX;
  float posY;
  float speedX;
  float speedY;

  ObjectBase(){
    posX = random(w
    posY = random(h
    speedX = random
    speedY = random
  }

  void display(){


  }
```

```
class Ball
{
  float posX;
  float posY;
  float speedX;
  float speedY;

  Ball(){
    posX = random(w
    posY = random(h
    speedX = random
    speedY = random
  }

  void display(){
    fill(255, 0, 0)
    ellipse( posX, p
  }
```

```
class Cross
{
  float posX;
  float posY;
  float speedX;
  float speedY;

  Cross(){
    posX = random(w
    posY = random(h
    speedX = random
    speedY = random
  }

  void display(){
    line(posX-15, p
    line(posX+15, p
  }
```

```
class Triangle
{
  float posX;
  float posY;
  float speedX;
  float speedY;

  Triangle(){
    posX = random(w
    posY = random(h
    speedX = random
    speedY = random
  }

  void display(){
    triangle(posX,
      posY+15, posX
  }
```

ObjectBaseとetc

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

```
  void move(){
    posX += speedX;
    posY += speedY;
    if(posX > width){
      posX = posX -
    }
    if(posX < 0){
      posX = posX +
    }
    if(posY > height
      posY = posY -
    }
    if(posY < 0){
      posY = posX +
    }
  }
}
```

```
  void move(){
    posX += speedX;
    posY += speedY;
    if(posX > width){
      posX = width +
      speedX = -spee
    }
    if(posX < 0){
      posX = -posX;
      speedX = -spee
    }
    if(posY > height
      posY = height
      speedY = -spee
    }
    if(posY < 0){
      posY = -posY;
      speedY = -spee
    }
  }
}
```

```
  void move(){
    posX += speedX;
    posY += speedY;
    if(posX > width
      posX = posX -
    }
    if(posX < 0){
      posX = posX +
    }
    if(posY > heigh
      posY = posY -
    }
    if(posY < 0){
      posY = posX +
    }
  }
}
```

```
  void move(){
    posX += speedX;
    posY += speedY;
    if(posX > width
      posX = posX -
    }
    if(posX < 0){
      posX = posX +
    }
    if(posY > heigh
      posY = posY -
    }
    if(posY < 0){
      posY = posX +
    }
  }
}
```

継承

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

- 継承すると，親の力をすべて引き継ぐ！
- 継承の方法は extends とやるだけ！

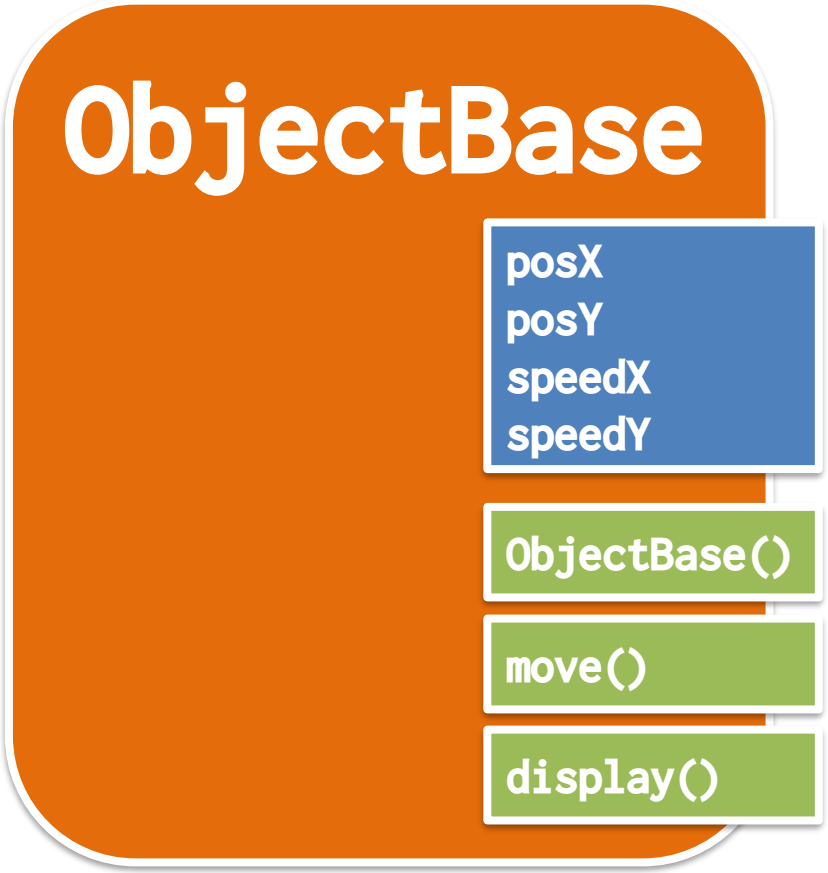
```
class クラス名 extends 親クラス名 {

}
```

- SampleObj.zip をダウンロードして下さい

Crossをどう作る？

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

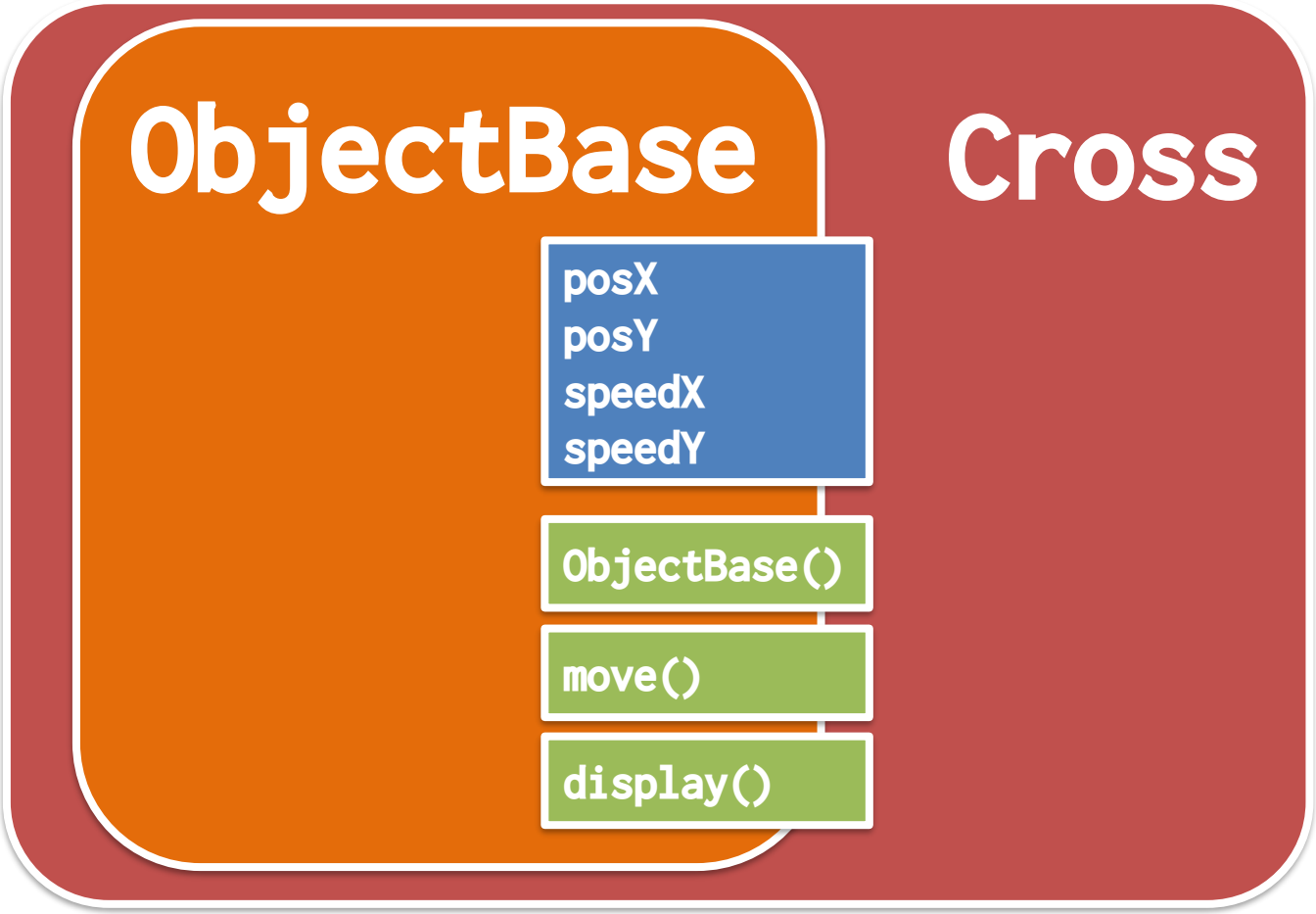
- ObjectBaseクラスを継承してCrossクラスを作る！



Crossをどう作る？

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室


```
class Cross extends ObjectBase
{
}
```



使ってみよう！

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

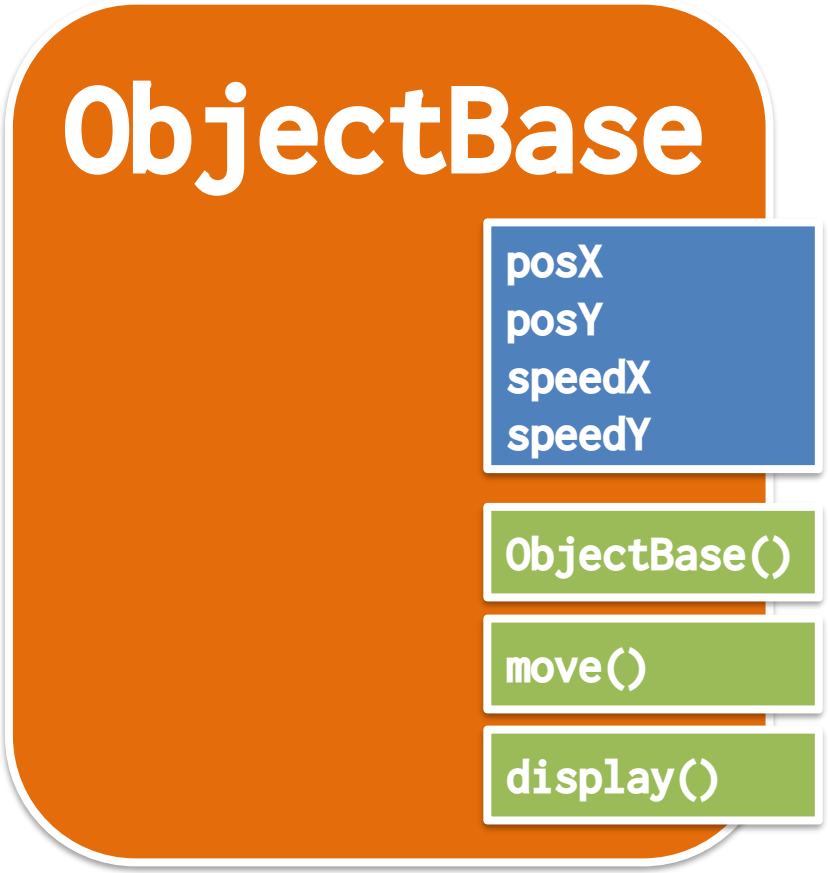
```
Cross cross1;
Cross cross2;

void setup() {
  size(600, 600);

  cross1 = new Cross();
  cross2 = new Cross();
}

void draw() {
  background(255);

  cross1.move();
  cross2.move();
  cross1.display();
  cross2.display();
}
```

- なにも表示されない
  - なんで？
- ObjectBaseのdisplay()は何も描画しないから！
  - バツを描画するにはどうしたら良い？
  - ObjectBaseのdisplayを書き換える？
    - → だめ！！

Crossをどう作る？

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

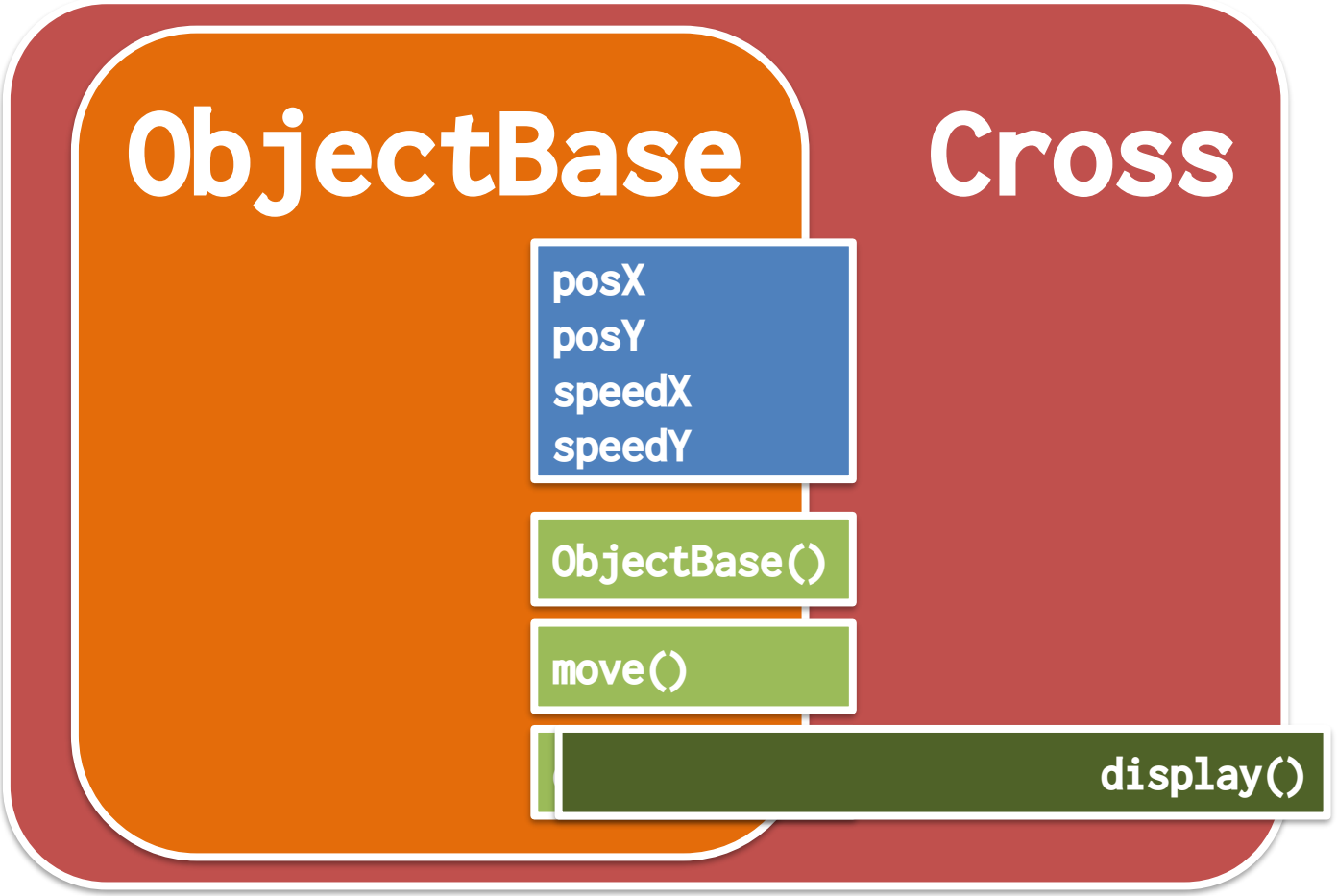
- ObjectBase の変数やメソッドを引き継ぎつつ，必要なところを上書きする！



Crossをどう作る？

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

- display() を上書きしてしまう！
  - displayメソッドをオーバーライドする



ObjectBaseクラスを使う

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

```
class Cross extends ObjectBase
{
  void display(){
    line(posX-15, posY-15, posX+15, posY+15);
    line(posX+15, posY-15, posX-15, posY+15);
  }
}
```

displayをオーバーライドして
親のdisplayメソッドが
呼ばれないようにする

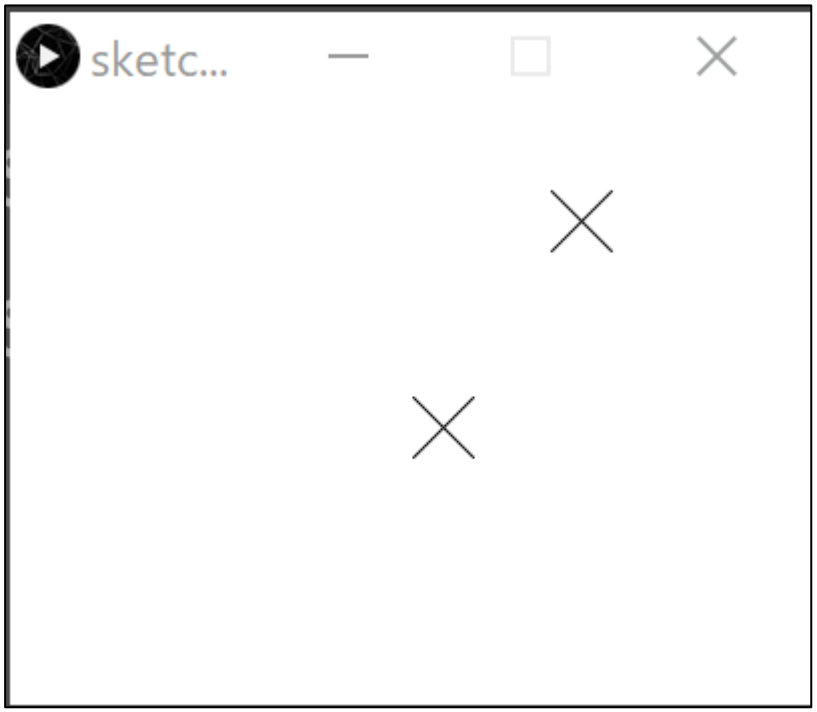
Crossクラスが劇的に短く！

使ってみよう！

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

```
Cross cross1;
Cross cross2;

void setup() {
  size(600, 600);

  cross1 = new Cross();
  cross2 = new Cross();
}

void draw() {
  background(255);

  cross1.move();
  cross2.move();
  cross1.display();
  cross2.display();
}
```

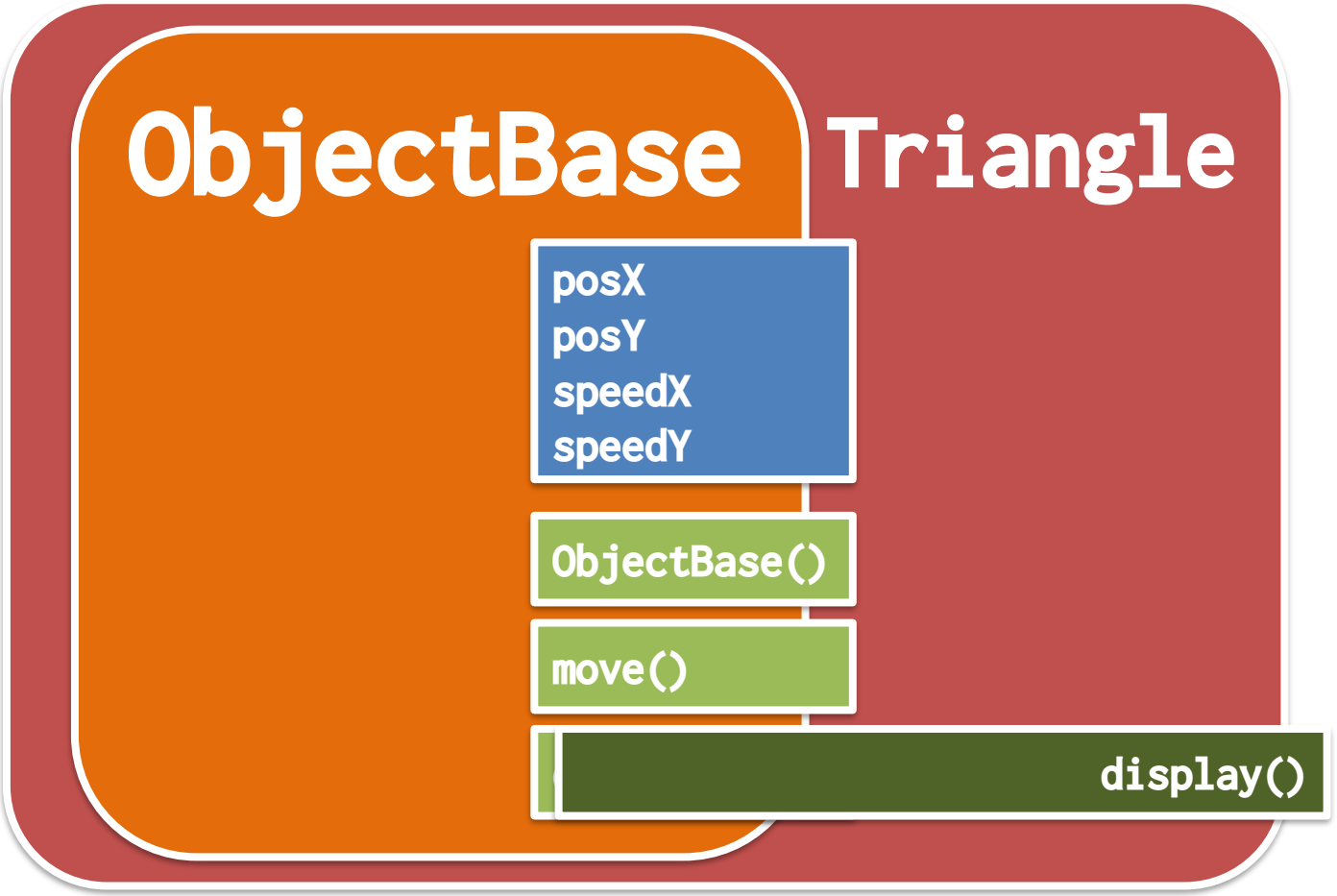
- 表示された！
  - display()が上書きされている！



Triangleをどう作る？

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

- display() を上書きしてしまう！
  - displayメソッドをオーバーライドする



ObjectBaseクラスを使う

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

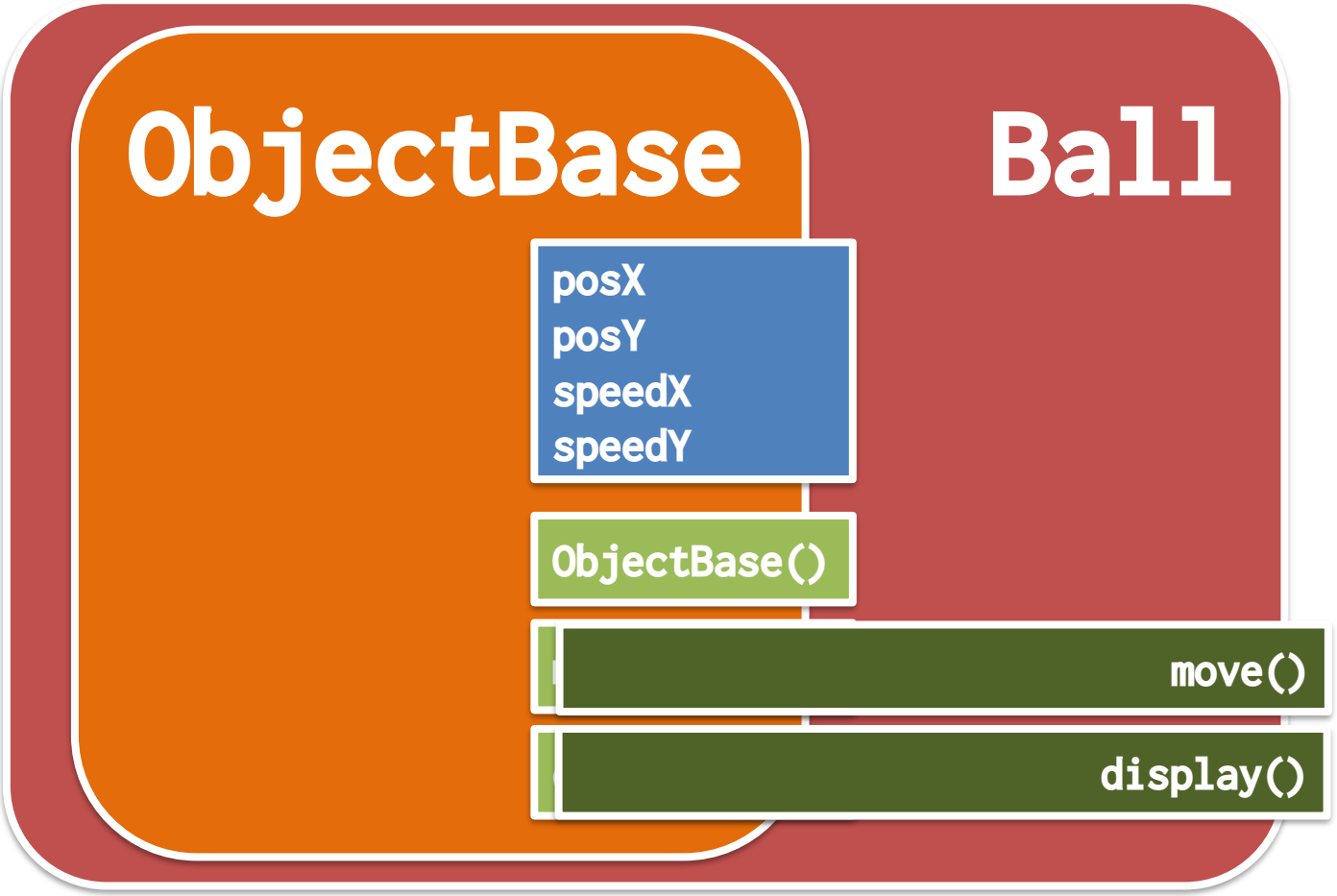
```
class Triangle extends ObjectBase
{
  void display(){
    fill(0, 255, 0);
    triangle(posX, posY-15, posX+15, posY+15, posX-15, posY+15);
  }
}
```

displayをオーバーライドして
親のdisplayメソッドが
呼ばれないようにする

Triangleクラスが劇的に短く！

Ball をどう作る？

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室



- move() と display() をオーバーライド！

Ballクラスはどう作る？

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

```
class Ball extends ObjectBase
{
  void display(){
    fill(255, 0, 0);
    ellipse(posX, posY, 30, 30);
  }
```

```
  void move(){
    posX += speedX;
    posY += speedY;
    if(posX > width){
      posX = width * 2 - posX;
      speedX = -speedX;
    }
    if(posX < 0){
      posX = -posX;
      speedX = -speedX;
    }
    if(posY > height){
      posY = height * 2 - posY;
      speedY = -speedY;
    }
    if(posY < 0){
      posY = -posY;
      speedY = -speedY;
    }
  }
}
```

**display と move を
オーバーライドして
親の move と display が
呼ばれないようにする**

ObjectBaseクラスを使う

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

- 使うときは，継承していることは気にしないで，対象とするクラスを使うだけでよい！
  - move()
  - display()

  - すばらしい！！

```
Ball ball;
Cross cross;
Triangle triangle;

void setup() {
  size(600, 600);
  ball = new Ball();
  cross = new Cross();
  triangle = new Triangle();
}

void draw() {
  background(255);
  ball.move();
  cross.move();
  triangle.move();
  ball.display();
  cross.display();
  triangle.display();
}
```

ObjectBaseとetc

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室


```
class ObjectBase
{
  float posX;
  float posY;
  float speedX;
  float speedY;

  ObjectBase(){
    posX = random(w
    posY = random(h
    speedX = random
    speedY = random
  }

  void display(){



  }
```

```
class Ball extends
{
  float posX;
  float posY;
  float speedX;
  float speedY;

  Ball(){
    posX = random(w
    posY = random(h
    speedX = random
    speedY = random
  }

  void display(){
    fill(255, 0, 0)
    ellipse( posX,
  }
```

```
class Cross extends
{
  float posX;
  float posY;
  float speedX;
  float speedY;

  Cross(){
    posX = random(w
    posY = random(h
    speedX = random
    speedY = random
  }

  void display(){
    line(posX-15, p
    line(posX+15, p
  }
```

```
class Triangle exte
{
  float posX;
  float posY;
  float speedX;
  float speedY;

  Triangle(){
    posX = random(w
    posY = random(h
    speedX = random
    speedY = random
  }

  void display(){
    triangle(posX,
      posY+15, posX
  }
```

ObjectBaseとetc

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室


```
  void move(){
    posX += speedX;
    posY += speedY;
    if(posX > width)
      posX = posX -
    }
    if(posX < 0){
      posX = posX +
    }
    if(posY > height
      posY = posY -
    }
    if(posY < 0){
      posY = posX +
    }



  }
}
```

```
  void move(){
    posX += speedX;
    posY += speedY;
    if(posX > width)
      posX = width +
      speedX = -spee
    }
    if(posX < 0){
      posX = -posX;
      speedX = -spee
    }
    if(posY > height
      posY = height
      speedY = -spee
    }
    if(posY < 0){
      posY = -posY;
      speedY = -spee
    }
  }
}
```

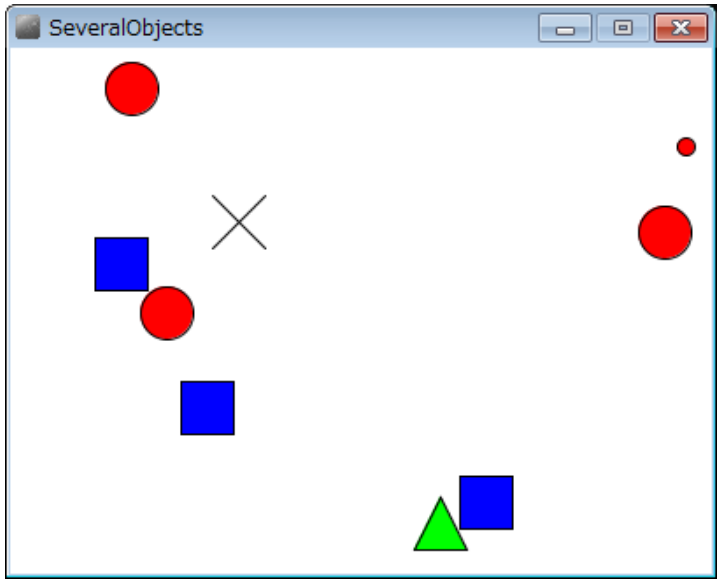
```
  void move(){
    posX += speedX;
    posY += speedY;
    if(posX > width
      posX = posX -
    }
    if(posX < 0){
      posX = posX +
    }
    if(posY > heigh
      posY = posY -
    }
    if(posY < 0){
      posY = posX +
    }



  }
}
```

```
  void move(){
    posX += speedX;
    posY += speedY;
    if(posX > width
      posX = posX -
    }
    if(posX < 0){
      posX = posX +
    }
    if(posY > heigh
      posY = posY -
    }
    if(posY < 0){
      posY = posX +
    }



  }
}
```

じゃあ

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

- ObjectBase クラスを継承して青色の四角形を描画するSquareクラスを作るには？
  - 四角形は左右の壁は跳ね返り，上下の壁はすり抜けて反対側から出てくる



四角形のクラス

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

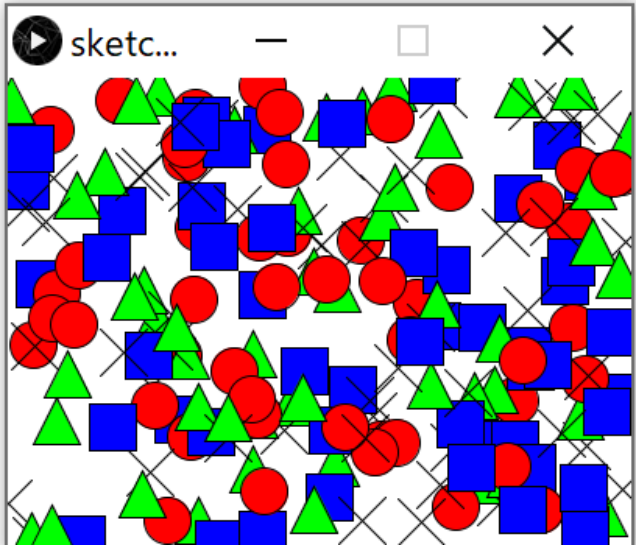
- 動く青色の四角形のクラス

```
class Square extends ObjectBase
{
  void display(){
    fill(0, 0, 255);
    rect(posX-15, posY-15, 30, 30);
  }
  void move(){
    posX += speedX;
    posY += speedY;
    if(posX > width-15){
      posX = width-15;
      speedX = -speedX;
    }
    if(posX < 15){
      posX = 15;
      speedX = -speedX;
    }
    if(posY > height) posY = posY - height;
    if(posY < 0)      posY = posY + height;
  }
}
```

50個の○△□xを描画！

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

- 50個の○と，50個のxと，50個の△と，50個の□が動き回るプログラムを作成せよ
  - ただし，その速度はx、y方向それぞれ-5～5の実数値とせよ
  - また，○は壁で跳ね返り，xと△と跳ね返らずに反対側から出てくるように，□は上下では跳ね返らず反対側から出てきて左右では跳ね返るようにせよ



配列＋クラス

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室


```
Ball[] balls = new Ball[50];
Cross[] crosses = new Ball[50];
Triangle[] triangles = new Triangle[50];
Square[] squares = new Square[50];

void setup() {
  size(400, 300);
  for (int i=0; i<50; i++) {
    balls[i] = new Ball();
    crosses[i] = new Cross();
    triangles[i] = new Triangle(
    squares[i] = new Square();
  }
}
```

```
void draw() {
  background(255);
  for(int i=0; i<50; i++){
    balls[i].move();
    balls[i].display();
    crosses[i].move();
    crosses[i].display();
    triangles[i].move();
    triangles[i].display();
    squares[i].move();
    squares[i].display();
  }
}
```

ArrayList＋クラス

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室


```
ArrayList<Ball> balls = new ArrayList<Ball>();
ArrayList<Cross> crosses = new ArrayList<Cross>();
ArrayList<Triangle> triangles = new ArrayList<Triangle>();
ArrayList<Square> squares = new ArrayList<Square>();

void setup() {
  size(400, 300);
  for (int i=0; i<50; i++) {
    balls.add(new Ball());
    crosses.add(new Cross());
    triangles.add(new Triangle());
    squares.add(new Square());
  }
}
```

```
void draw() {
  background(255);
  for(int i=0; i<50; i++){
    balls.get(i).move();
    balls.get(i).display();
    crosses.get(i).move();
    crosses.get(i).display();
    triangles.get(i).move();
    triangles.get(i).display();
    squares.get(i).move();
    squares.get(i).display();
  }
}
```

ArrayListでもあまり改善されてない？

ポリモーフィズム

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

- Ball, Cross, Triangle, Squareの親クラス（継承元）はObjectBaseで，同じメソッドを持っている
- ポリモーフィズム（多態性，多様性）
  - 複数の型に属することを許すこと
  - 親クラスから継承されたクラスのインスタンスは，それぞれ親クラスの型に代入できる．呼び出されるのは，継承されたクラスのメソッド
  - つまり，ObjectBaseとして定義しておき，BallもCrossもTriangleもSquareも放り込んで，ただのメソッドで呼び出しが可能！

配列＋ObjectBase

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

- ObjectBaseにはBallもCrossもTriangleもSquareも入れることができるよ！

- ポリモーフィズムバンザイ！

```
ObjectBase[] list = new ObjectBase[200];

void setup() {
  size(400, 300);
  for (int i=0; i<list.length; i+=4) {
    list[i+0] = new Ball();
    list[i+1] = new Cross();
    list[i+2] = new Triangle();
    list[i+3] = new Square();
  }
}

void draw() {
  background(255);
  for(int i=0; i<list.length; i++){
    list[i].move();
    list[i].display();
  }
}
```

ArrayList＋ObjectBase

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室


```
ArrayList<ObjectBase> list = new ArrayList<ObjectBase>();

void setup() {
  size(400, 300);
  for(int i=0; i<50; i++){
    list.add( new Ball() );
    list.add( new Cross() );
    list.add( new Triangle() );
    list.add( new Square() );
  }
}

void draw() {
  background(255);
  for( int i=0; i<list.size(); i++ ){
    list.get(i).move();
    list.get(i).display();
  }
}
```

<ObjectBase> で型を定義

BallもCrossもTriangleもSquareも
入れることができるよ！

list.size()で数を取得

list.get(i)でi番目を取得

ArrayList＋ObjectBase

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室


```
ArrayList list = new ArrayList();

void setup() {
  size(400, 300);
  for(int i=0; i<50; i++){
    list.add( new Ball() );
    list.add( new Cross() );
    list.add( new Triangle() );
    list.add( new Square() );
  }
}

void draw() {
  background(255);
  for(int i=0; i<list.size(); i++){
    ObjectBase obj = (ObjectBase)list.get(i);
    obj.move();
    obj.display();
  }
}
```

型を定義しなくても使えます
その場合は使うときにキャストを！

(ObjectBase) でキャスト

ArrayList＋拡張for文

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室


```
ArrayList<ObjectBase> list = new ArrayList<ObjectBase>();

void setup() {
  size(400, 300);
  for(int i=0; i<50; i++){
    list.add( new Ball() );
    list.add( new Cross() );
    list.add( new Triangle() );
    list.add( new Square() );
  }
}

void draw() {
  background(255);
  for( ObjectBase obj: list ){
    obj.move();
    obj.display();
  }
}
```

めっちゃシンプル！

何に使えるの？

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

- 継承とかポリモーフィズムとかなんか色々あるけど何に使うの？
  - シューティングで色んな動きをする敵を作る
  - RPGゲームで色んな戦い方をする敵を作る
  - 色々な光り方・広がり方をする花火を作る
  - 色々な結晶の雪を各々の特性で降らせる
  - 草原に色々な草木花を生えさせる
  - ひとの混雑に関するシミュレーションをする

親クラスを定義して継承してクラスを作り
ArrayListで意識せずまとめて動かす！

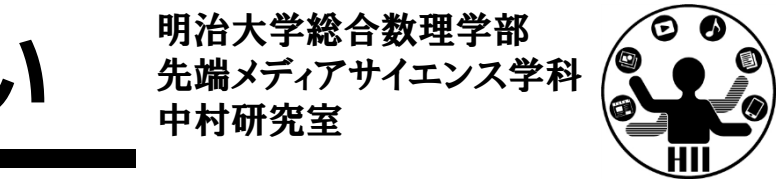
覚える必要はないけれど

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

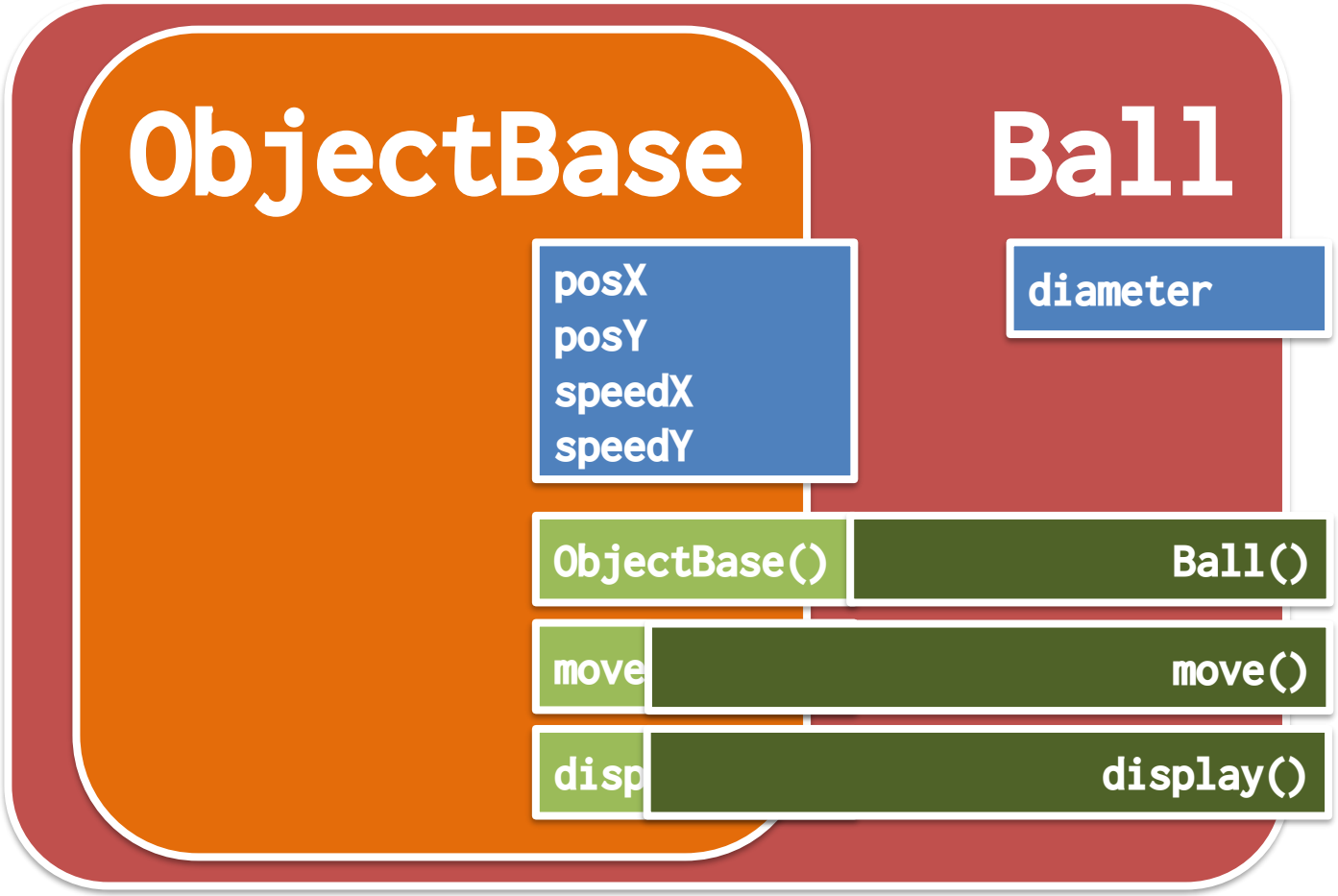
- アップキャスト
  - 親クラスで定義されたものに，継承されたクラスのインスタンスが代入されると，自動的に親の型に変換されること
- ちなみに，ダウンキャストもあるよ
  - ダウンキャストは安全じゃないよ！

継承先で変数追加したい

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室



- ObjectBaseを継承するけど，○だけはその直径（diameter）もランダムに決定したい



継承先で変数追加したい

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

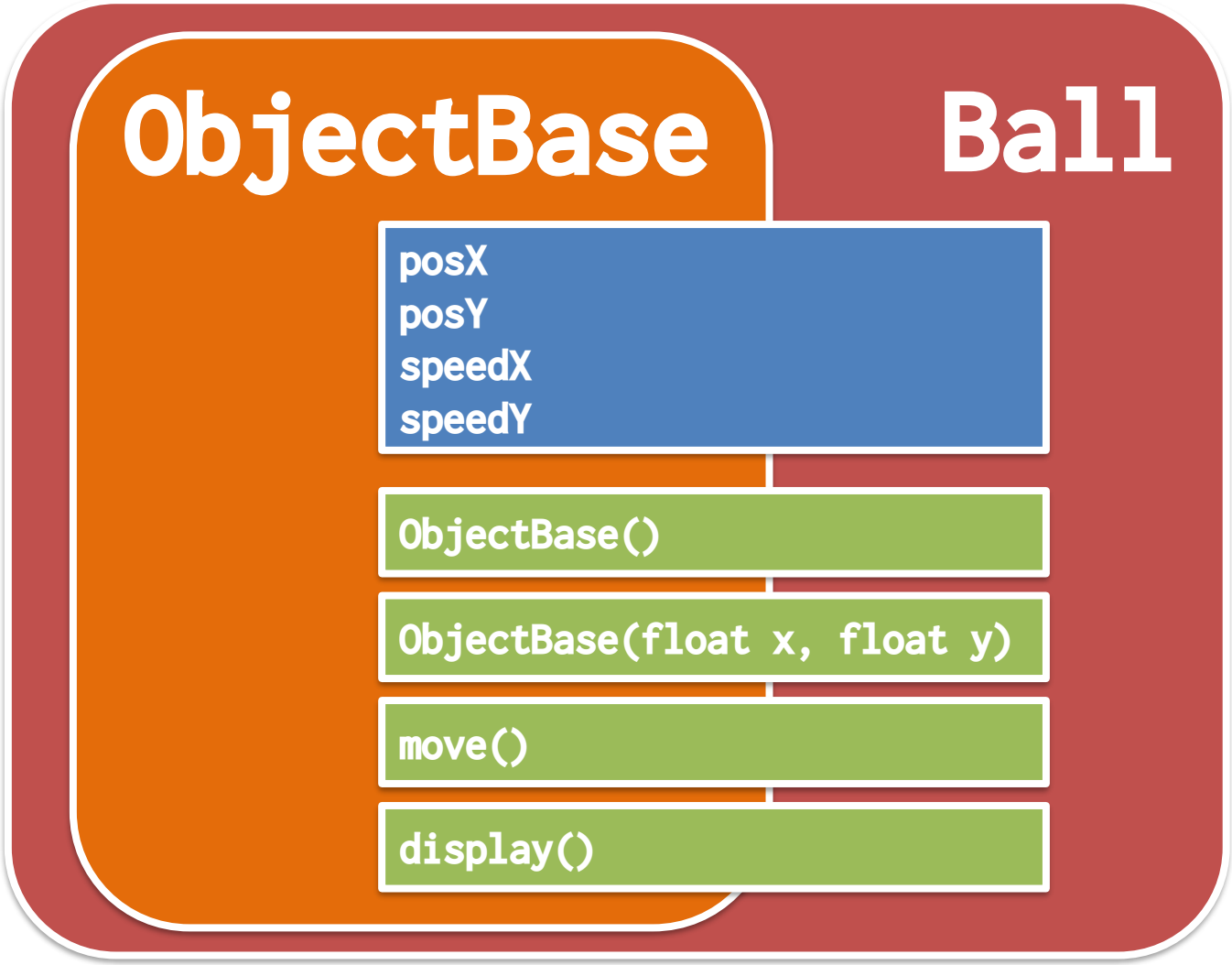
- Ballにインスタンス変数diameterを追加して，10-50の間でランダムに決定！
  - ObjectBaseのコンストラクタを呼び出してないのに動く！（コンストラクタはオーバーライドされない）

```
class Ball extends ObjectBase {
  int diameter;

  Ball(){
    diameter = (int)random(10, 50);
  }

  void display(){
    fill(255, 0, 0);
    ellipse(posX, posY, diameter, diameter);
  }
```

注意

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

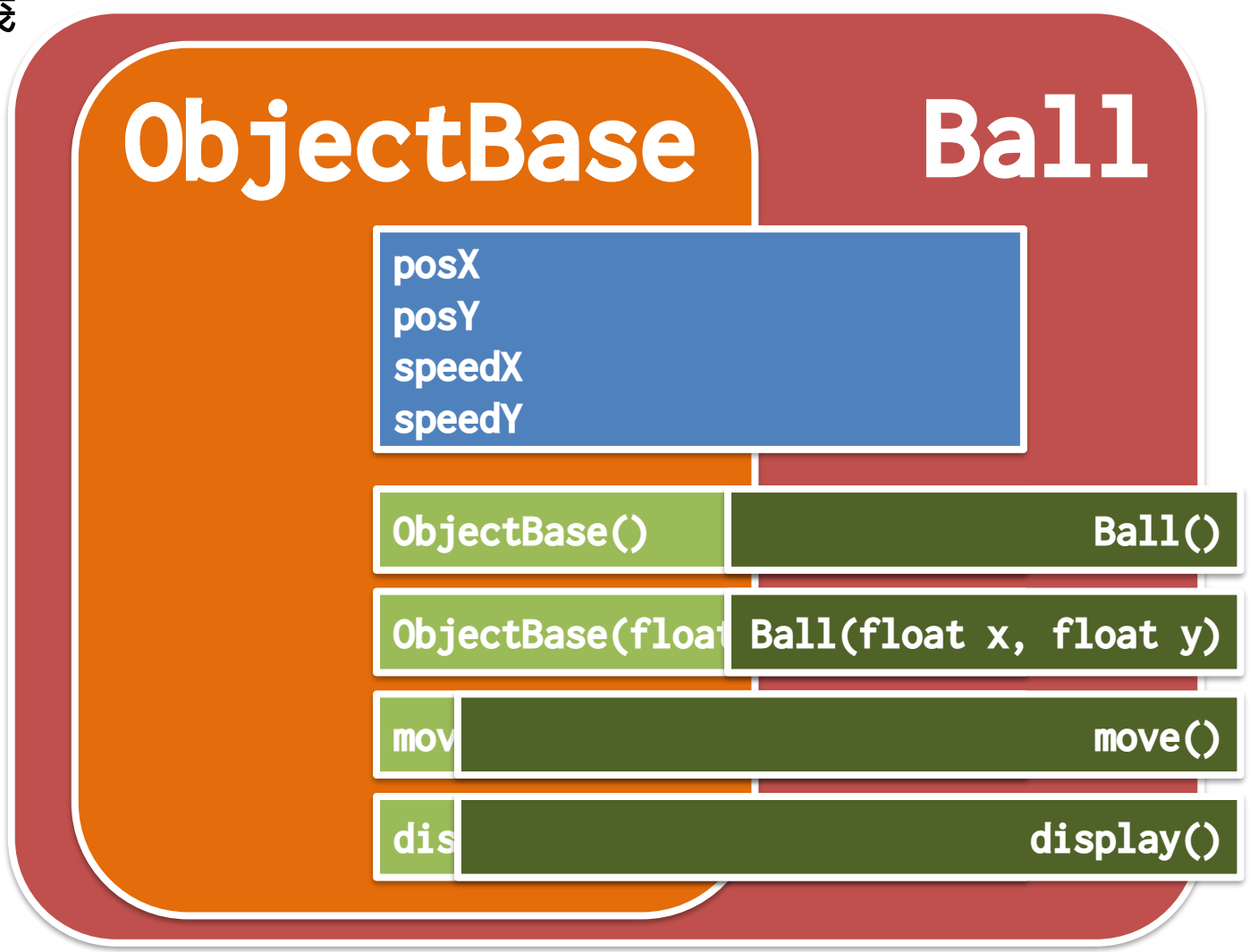
- コンストラクタに引数がある場合は要コンストラクタの定義



注意

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

- コンストラクタに引数がある場合は要コンストラクタの定義



親コンストラクタの呼び出し

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

- 空っぽで同じ引数のコンストラクタを作る

```
class ObjectBase {
  float posX;
  float posY;
  float speedX;
  float speedY;
  ObjectBase(){
    posX = random(width);
    posY = random(height);
    speedX = random(-5, 5);
    speedY = random(-5, 5);
  }
  ObjectBase(float x, float y){
    posX = x;
    posY = y;
    speedX = random(-5, 5);
    speedY = random(-5, 5);
  }
```

```
class Ball extends ObjectBase {

  Ball(){
  }

  Ball(float x, float y){
    super(x, y);
  }
```

惑星と衛星の様なオブジェクト

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

ObjectBaseを継承し，800x600のウインドウ内で，任意の場所x，yから任意の速度で移動する3つの赤色の円（直径30ピクセルの惑星）を描画し，右端・左端・上端・下端に来ると跳ね返るようにする．また，赤色の円（惑星）には円の中心から50の距離があるところに1つの衛星（直径10ピクセル）があり，5度ずつ円の周りを回転するようにせよ

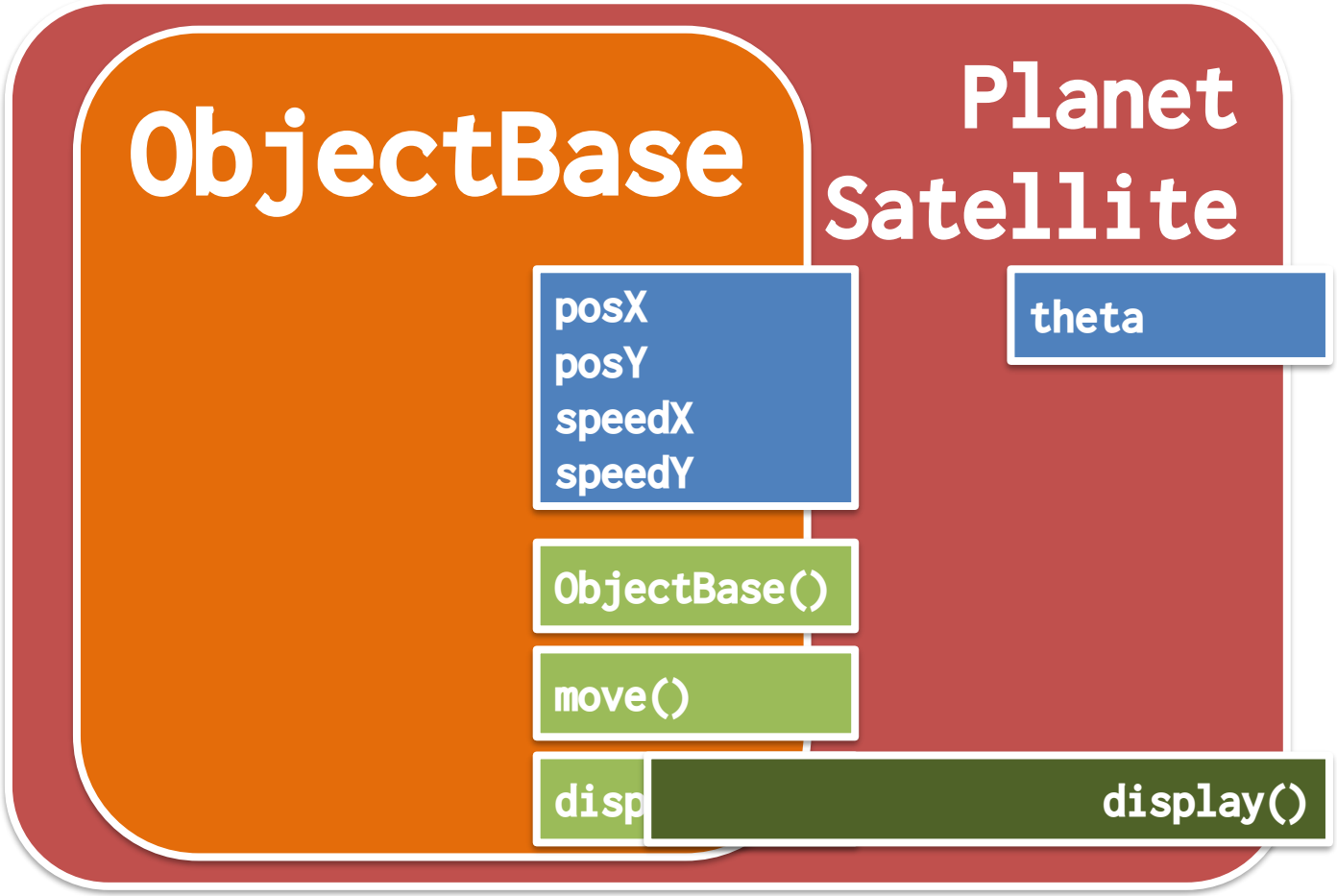
- 考え方
  - 円の中心（x，y）から衛星の方向の角度（0〜360度）をthetaとすると，衛星の座標は

```
(x+50*cos(radians(theta)), y+50*sin(radians(theta)))
```

継承すると・・・

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

- ObjectBase から継承して PlanetSatellite を作成し，theta という変数を追加し，display() というメソッドを上書き（オーバーライド）



惑星と衛星の様なオブジェクト

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

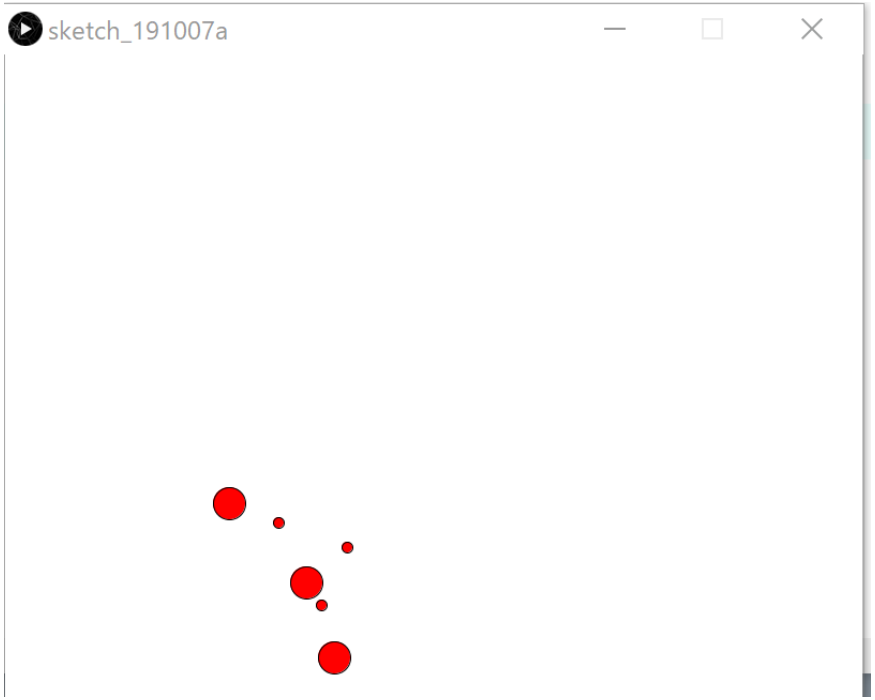
- ObjectBaseを継承して，変数を追加する

```
class PlanetSatellite extends ObjectBase
{
  int theta;

  void display()
  {
    fill( 255, 0, 0 );
    ellipse(posX, posY, 30, 30);
    theta = theta + 5;
    float sx = posX + 50*cos(radians(theta));
    float sy = posY + 50*sin(radians(theta));
    ellipse(sx, sy, 10, 10);
  }
}
```

惑星と衛星の様なオブジェクト

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室


```
PlanetSatellite ps1;
PlanetSatellite ps2;
PlanetSatellite ps3;
void setup()
{
  size( 800, 600 );
  ps1 = new PlanetSatellite();
  ps2 = new PlanetSatellite();
  ps3 = new PlanetSatellite();
}

void draw()
{
  background(255);
  ps1.move();
  ps2.move();
  ps3.move();
  ps1.display();
  ps2.display();
  ps3.display();
}
```



惑星と衛星の様なオブジェクト

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

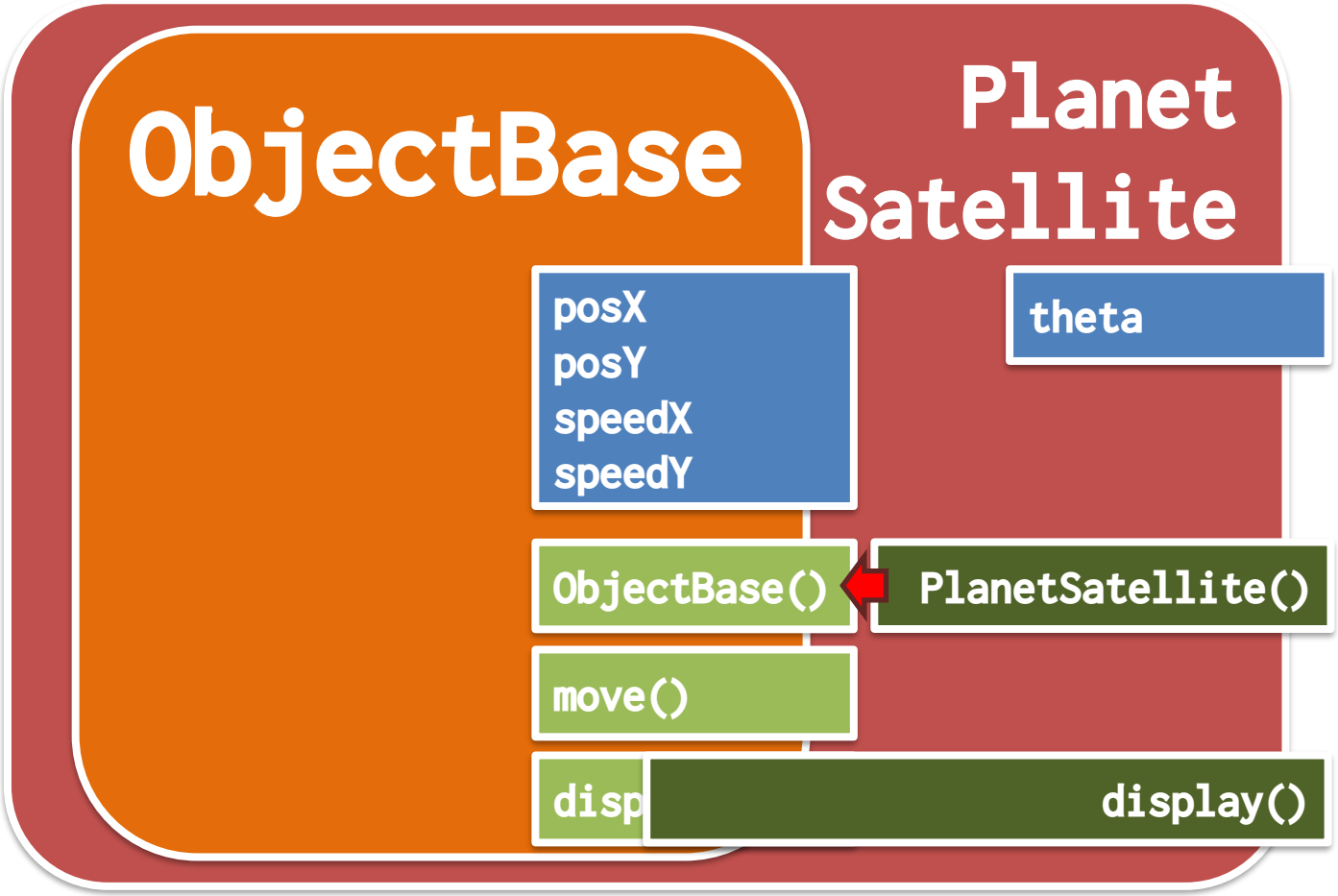
衛星の開始角を0～360度の任意の場所にしたい．どうするか？

- 考え方
  - コンストラクタで指定したらOK？

継承すると・・・

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

- コンストラクタで定義する
  - 勝手に親が呼び出されるので便利！（明示的に呼び出す場合は super() と書く）



やってみる

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

```
class PlanetSatellite extends ObjectBase {
  int theta;
  PlanetSatellite(){
    theta = (int)random(360);
  }
  void display(){
    fill(255, 0, 0);
    ellipse(posX, posY, 30, 30);
    // 回転させる
    theta = theta + 5;
    int sx = (int)(posX+50*sin(radians(theta)));
    int sy = (int)(posY+50*cos(radians(theta)));
    ellipse(sx, sy, 10, 10);
  }
}
```

課題4-1: basic_boundAll3

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

- 講義中に扱ったObjectBaseクラスを継承し，赤色の○が動き回るBallクラスと，黒色のxが動き回るCrossクラス，緑色の△が動き回るTriangleクラス，青色の□が動き回るSquareクラスを作成せよ
- ただし，○は上下左右の壁で跳ね返り，xと△は上下左右で逆から出てくるように，□は左右で跳ね返り，上下では逆から出てくるようにせよ
- また，この継承したクラスを利用して800x600のウインドウ内を100個の赤色○と，40個のxと，60個の緑色△と，80個の青色□を描画せよ

課題4-2: basic_ManyChara

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

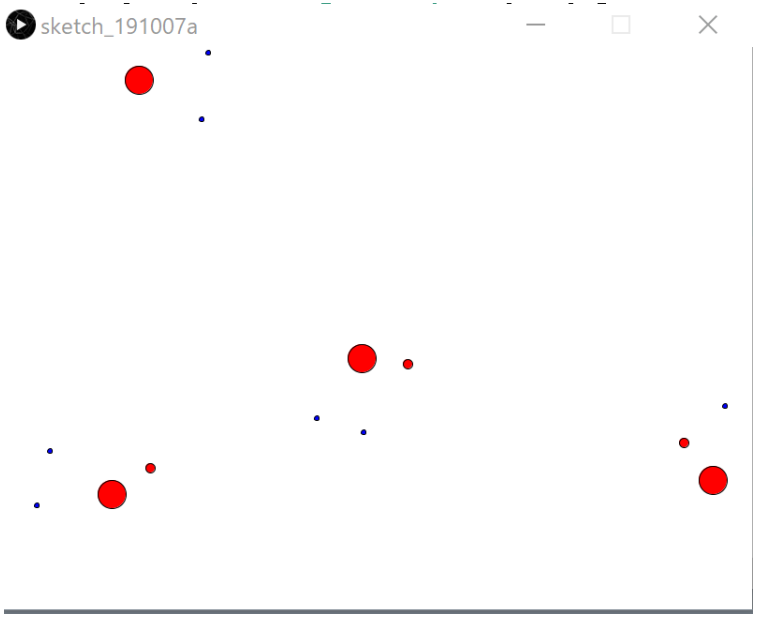
- まずはじめに、CharacterBaseクラスを継承し、自身のキャラクタを描画する ExXXXXX クラスを作成せよ
  - 先週の課題を使い回すと良い（基本的には、move(), displayHappy(), displaySad()の中身をコピペするだけ）
- この作成した ExXXXXX クラスを研究室のSlackで共有せよ
  - メディア基礎実験のチャンネルを使っても良いよ
- 自分と他人のキャラクタを使って、5つ以上のキャラクタが動き回るプログラムを作成せよ
  - 同じ研究室のものを5つ以上利用して（同じ研究室で足りなければ他の研究室のを使ってもよい）、キャラクタが画面内を動き回るプログラムを作成せよ
  - 他の研究室のものを使用しつつかなりたくさんのキャラクタを入れてもよい

（ヒント） ArrayListに放り込むだけ！！

課題4-3: basic_PlanetSatellite

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

- PlanetSatellite クラスを継承し，惑星の中心から80ピクセル離れた位置に青色の小さな衛星が2つある（直径5ピクセルで毎フレーム3度ずつ動く）PlanetSatelliteEx クラスを作成せよ．また，そのクラスを用いて4個の惑星（3つの衛星がある）がウインドウ内を動きまわるようにせよ
  - 考え方
    - インスタンス変数を追加する
    - display メソッドを
      オーバーライドする



宿題4-1: hw_JankenGuriko

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

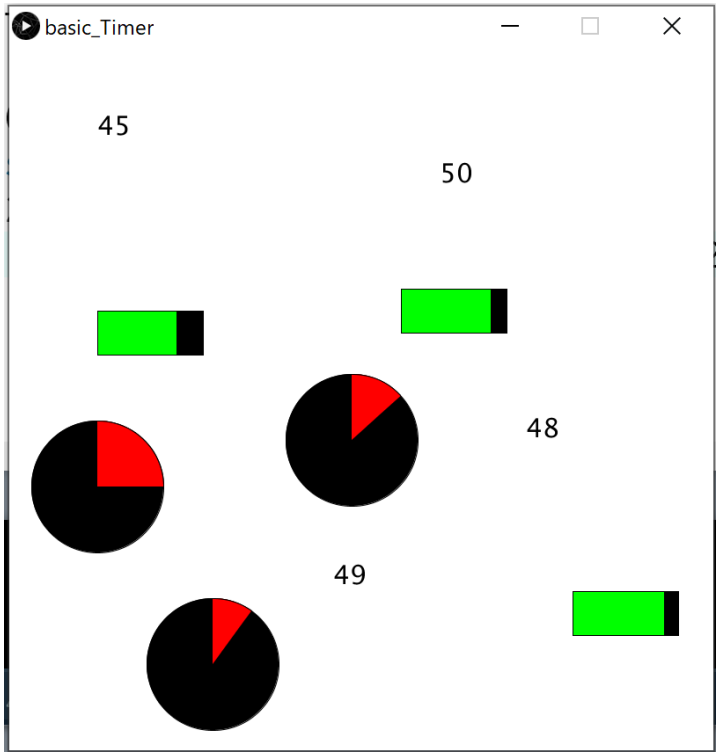
- グリコというゲーム（グーで勝つと+3，チョキで勝つと+5，パーで勝つと+6の点が入る）がある
  - チョキで+6が一般的だがここでは+5とする
- 毎回じゃんけんを繰り返し，10000点を先に獲得したほうが勝ちになる
- 配布する hw_JankenGuriko 内のJankenAgentクラスを継承し，自身のエージェントを作成せよ
  - ただしコンストラクタで strAgentName に自身のエージェントの名前をセットせよ
- battleメソッドに自身のエージェントと敵のエージェントを引数として与え，10000点になるまで戦わせて勝とう！
  - 少なくとも JankenAgent自身，AgentNakamura334には勝ちましょう
  - できれば，知り合いのAgentももらって，勝ちましょう！！

https://mathtrain.jp/grk

宿題4-2: hw_Timer

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

- 配布する hw_Timer.zip を利用し，生成されてから60秒経過すると自動的に消滅する（何も描画されなくなる）タイマーを，TimerCoreクラスを継承して，TextTimerやProgressTimerを参考にしつつ2つ以上実現せよ
- また，マウスクリックしたときにそれぞれのタイマーがランダムに生成されるようにせよ



宿題4-3: hw_VendingMachine

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

- 自動販売機（VendingMachineクラス）では日本円の10円・50円・100円の3種類の硬貨を扱い、販売される商品は全て200円以下で、10円～200円で10円刻みとなっています。
- 自動販売機には投入金額と購入金額からおつりを計算し、おつりを返却するか、商品を購入できないことを知らせる機能が必要です。これらの機能は、以下のルールに従って動作します。
  - お釣りがない場合は「ありがとうございました。おつりはありません」と表示する。
  - お釣りがある場合は「ありがとうございました。おつりは10円3枚と50円1枚です」と表示し、おつりを返却する。
  - 投入金額が足りないまたはおつりを返却できない場合は「商品を購入できません。10円0枚、50円1枚、100円1枚を返却します」と商品を購入できないことを知らせるとともに投入金額を返却する。
  - なお投入されたお金は、おつりとしても利用することが可能です。

宿題4-3: hw_VendingMachine

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

- ヒント
  - VendingMachineクラスには，内部に10円玉，50円玉，100円玉が何枚あるかを管理する変数を用意
  - 初期の枚数をセットするメソッドを作成

    initialize(10円の数, 50円の数, 100円の数) メソッド
  - お金を投入するメソッドを作成

    insert(10円の数, 50円の数, 100円の数) メソッド
  - 購入する商品を指定するメソッドを作成

    buy(値段) メソッド
    - buy メソッドは，標準出力で結果を返すようにせよ．ただし，
    - おつりは「投入金額 - 購入金額」で計算されます。
    - おつりは自動販売機の内部にある硬貨から枚数が最も少なくなるように選んだ硬貨の組合せで返却せよ

宿題4-3: 動作チェック

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

- 色々なパターンを用意して問題ないかを確認しよう

```
vMachine.initialize( 5, 5, 5 );
vMachine.insert( 0, 1, 1 );
vMachine.buy( 130 );
vMachine.insert( 0, 0, 2 );
vMachine.buy( 110 );
vMachine.insert( 0, 0, 2 );
vMachine.buy( 140 );

vMachine.initilize( 9, 8, 7 );
vMachine.insert( 0, 0, 2 );
vMachine.buy( 110 );
vMachine.insert( 0, 0, 2 );
vMachine.buy( 120 );
vMachine.insert( 0, 0, 2 );
vMachine.buy( 130 );
vMachine.insert( 2, 0, 2 );
vMachine.buy( 180 );
```

```
ありがとうございました。おつりは10円2枚と50円0枚と100円0枚です
商品を購入できません。10円0枚、50円0枚、100円2枚を返却します
ありがとうございました。おつりは10円1枚と50円1枚と100円0枚です
ありがとうございました。おつりは10円4枚と50円1枚と100円0枚です
ありがとうございました。おつりは10円3枚と50円1枚と100円0枚です
ありがとうございました。おつりは10円2枚と50円1枚と100円0枚です
商品を購入できません。10円2枚、50円0枚、100円2枚を返却します
```

クラス学習の参考資料

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

THE **NATURE** OF CODE

DANIEL SHIFFMAN

- The Nature of Code
  - Processingでクラスを使いつつ，物理世界の再現について学ぶことができるうえ，ニューラルネットワークとかも学べるよ！
  - 休みの期間に一通り入力してみると色々学びがあっていいと思います！
  - 英語も難しくないから学習に良いよ！
  - https://natureofcode.com/book/