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

中村，高橋，小林，橋本

宿題2-1：hw_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

- CircleClassを改良し、正方形が動き回るSquareClass、xが動き回るCrossClass、△が動き回る TriangleClassを作成せよ
- またこれを利用して10個の〇と，5個の正方形と，5個のxと、3個の△が画面内を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、〇と□は壁で跳ね返り，xと△は跳ね返らずに反対側から出てくるようにせよ

宿題2-2: hw_ClickHide

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

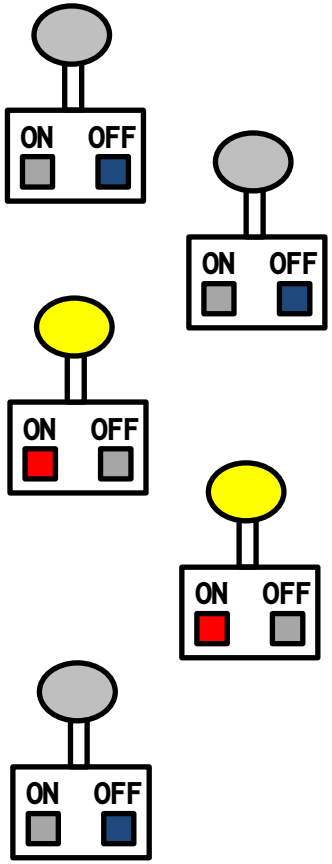
- CircleClassを改良し、20個の赤色の○（ただし、半径は10〜30のランダムとせよ）が600x400のウインドウ内を動き回るプログラムを作成せよ。
- また、その○の中がクリックされたら、そのクリックされた○が見えなくなるようにせよ（描画しないようにするだけで良い）

- ヒント
  - 描画するかどうかを判断する isShow みたいなインスタンス変数を追加して、その変数が true なら描画、 false なら描画しないみたいなことをすると良いよ！

宿題2-3: hw_SwitchLight

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

- 2つのボタンと1つのライトからなる照明を作成する．この照明をSwitchLightクラスとして作成せよ（形状については任せる）
- ボタンは左右に配置し，左側のボタンをクリックするとライトが光り，右側のボタンをクリックするとライトが消えるようにせよ
- また，SwitchLightクラスを利用して，5つの照明を画面に表示するようにせよ
- ヒント： RadioButtonクラス



本日の流れ

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

- 13:30-14:00 宿題の解説
- 14:00-15:40 座学＋演習（10分休憩）
- 15:40-15:45 課題提示
- 15:45-16:45 課題に取り組む
- 16:45が課題の提出期限
- 16:45-17:00 課題の解説

先週のまとめ

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

- オブジェクト指向のさわりを学んだ
  - インスタンス化
    - `CircleClass circle = new CircleClass();`
  - インスタンス
    - `circle`
  - インスタンス変数
    - `circle.speed`
  - インスタンスメソッド
    - `circle.move()`
  - コンストラクタ
    - `CircleClass(){ 初期化処理 }`
  - 先週間違ったファイル配布してごめんなさい（簡単にしたバージョンが同期されておらず間違って配布）

動きは図形に任せたい

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

- 丸、正方形、三角形を定義
  - それぞれの座標は意識したくない
    - circle.x, circle.y, square.x, square.y, triangle.x, triangle.y
    - 内部で適当に処理してもらう
  - それぞれの速度も意識したくない
    - circle.vx, circle.vy, square.vx, square.vy, triangle.vx, triangle.vy
  - 描画はシンプルにしたい
    - circle.draw(), square.draw(), triangle.draw()
  - 移動もシンプルにしたい
    - circle.move(), square.move(), triangle.move()

すべてを XXXX . 機能 という形に！

オブジェクト指向（クラス）

宿題2-1：hw_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

- CircleClassを改良し、正方形が動き回るSquareClass、xが動き回るCrossClass、△が動き回る TriangleClassを作成せよ
- またこれを利用して10個の〇と，5個の正方形と，5個のxと、3個の△が画面内を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、〇と□は壁で跳ね返り，xと△は跳ね返らずに反対側から出てくるようにせよ

Circle/Square/Cross

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
  - CircleClass
  - SquareClass
  - CrossClass

CircleClass 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 CircleClass
{
  float x;
  float y;
  float vx;
  float vy;

  CircleClass() {
    x = random(width);
    y = random(height);
    vx = random(-5, 5);
    vy = random(-5, 5);
  }

  void display() {
    fill(255, 0, 0);
    ellipse(x, y, 30, 30);
  }

  void move() {
    x += vx;
    y += vy;
    if(x > width) {
      x = width * 2 - x;
      vx = -vx;
    }
    if(x < 0) {
      x = -x;
      vx = -vx;
    }
    if(y > height) {
      y = height * 2 - y;
      vy = -vy;
    }
    if(y < 0) {
      y = -y;
      vy = -vy;
    }
  }
}
```

SquareClass

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 SquareClass
{
  float x;
  float y;
  float vx;
  float vy;

  SquareClass {
    x = random(width);
    y = random(height);
    vx = random(-5, 5);
    vy = random(-5, 5);
  }

  void display() {
    fill(0, 0, 255);
    rect(x-15, y-15, 30, 30);
  }
```

```
  void move() {
    x += vx;
    y += vy;
    if(x > width) {
      x = width * 2 - x;
      vx = -vx;
    }
    if(x < 0) {
      x = -x;
      vx = -vx;
    }
    if(y > height) {
      y = height * 2 - y;
      vy = -vy;
    }
    if(y < 0) {
      y = -y;
      vy = -vy;
    }
  }
}
```

CrossClass

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 CrossClass
{
  float x;
  float y;
  float vx;
  float vy;

  CrossClass() {
    x = random(width);
    y = random(height);
    vx = random(-5, 5);
    vy = random(-5, 5);
  }

  void display() {
    stroke(0, 0, 0);
    line(x-15, y-15, x+15, y+15);
    line(x-15, y+15, x+15, y-15);
  }
```

```
  void move() {
    x += vx;
    y += vy;
    x = (x + width) % width;
    y = (y + height) % height;
  }
}
```

Circle/Square/Cross

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 CircleClass
{
  float x;
  float y;
  float vx;
  float vy;

  CircleClass() {
    x = random(width);
    y = random(height);
    vx = random(-5, 5);
    vy = random(-5, 5);
  }

  void display() {
    fill(255, 0, 0);
    ellipse(x, y, 30, 30);

  }
```

```
class SquareClass
{
  float x;
  float y;
  float vx;
  float vy;

  SquareClass() {
    x = random(width);
    y = random(height);
    vx = random(-5, 5);
    vy = random(-5, 5);
  }

  void display() {
    fill(0, 0, 255);
    rect(x-15, y-15, 30,

  }
```

```
class CrossClass
{
  float x;
  float y;
  float vx;
  float vy;

  CrossClass() {
    x = random(width);
    y = random(height);
    vx = random(-5, 5);
    vy = random(-5, 5);
  }

  void display() {
    stroke(0, 0, 0);
    line(x-15, y-15, x+15,
    line(x-15, y+15, x+15,
  }
```

Circle/Square/Cross

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 CircleClass
{
  float x;
  float y;
  float vx;
  float vy;

  CircleClass() {
    x = random(width);
    y = random(height);
    vx = random(-5, 5);
    vy = random(-5, 5);
  }

  void display() {
    fill(255, 0, 0);
    ellipse(x, y, 30, 30);
  }
```

```
class SquareClass
{
  float x;
  float y;
  float vx;
  float vy;

  SquareClass() {
    x = random(width);
    y = random(height);
    vx = random(-5, 5);
    vy = random(-5, 5);
  }

  void display() {
    fill(0, 0, 255);
    rect(x-15, y-15, 30,
  }
```

```
class CrossClass
{
  float x;
  float y;
  float vx;
  float vy;

  CrossClass() {
    x = random(width);
    y = random(height);
    vx = random(-5, 5);
    vy = random(-5, 5);
  }

  void display() {
    stroke(0, 0, 0);
    line(x-15, y-15, x+15,
    line(x-15, y+15, x+15,
  }
```

Circle/Square/Cross

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() {
   x += vx;
   y += vy;
   if(x > width) {
     x = width * 2 - x;
     vx = -vx;
   }
   if(x < 0) {
     x = -x;
     vx = -vx;
   }
   if(y > height) {
     y = height * 2 - y;
     vy = -vy;
   }
   if(y < 0) {
     y = -y;
     vy = -vy;
   }
  }
}
```

```
void move() {
   x += vx;
   y += vy;
   if(x > width) {
     x = width * 2 - x;
     vx = -vx;
   }
   if(x < 0) {
     x = -x;
     vx = -vx;
   }
   if(y > height) {
     y = height * 2 - y;
     vy = -vy;
   }
   if(y < 0) {
     y = -y;
     vy = -vy;
   }
  }
}
```

```
void move() {
   x += vx;
   y += vy;
   x = (x + width) % width
   y = (y + height) % heig


  }
}
```

Circle/Square/Cross

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() {
    x += vx;
    y += vy;
    if(x > width) {
      x = width * 2 - x;
      vx = -vx;
    }
    if(x < 0) {
      x = -x;
      vx = -vx;
    }
    if(y > height) {
      y = height * 2 - y;
      vy = -vy;
    }
    if(y < 0) {
      y = -y;
      vy = -vy;
    }
  }
}
```

```
void move() {
    x += vx;
    y += vy;
    if(x > width) {
      x = width * 2 - x;
      vx = -vx;
    }
    if(x < 0) {
      x = -x;
      vx = -vx;
    }
    if(y > height) {
      y = height * 2 - y;
      vy = -vy;
    }
    if(y < 0) {
      y = -y;
      vy = -vy;
    }
  }
}
```

```
void move() {
    x += vx;
    y += vy;
    x = (x + width) % width
    y = (y + height) % heig
  }
}
```

Circle/Square/Cross

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

- CircleClassと，SquareClassと，CrossClassは似ている

クラス名	CircleClass	SquareClass	CrossClass	
インスタンス変数	x，y，vx，vy	x，y，vx，vy	x，y，vx，vy	一致
コンストラクタ	CircleClass() 座標速度初期化	SquareClass() 座標速度初期化	CrossClass() 座標速度初期化	名は違うが 内部は一致
void move()	はねかえる	はねかえる	はねかえらない	一部一致
void display()	円を描く	正方形を描く	x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CircleClassというのがダメなのでは？
  - ObjectClassというものを作って、objectTypeなどの変数を用意して、displayの時に表示するものを切り替えては？

```
class ObjectClass {
  float x;
  float y;
  float vx;
  float vy;
  int objectType;

  void display(){
    if(objectType == 0){
      ellipse(x, y, 30, 30);
    } else if(objectType == 1){
      rect(x-15, y-15, 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。）

スーパークラス（親クラス）を作って
そこから機能や変数を引き継ぐ！

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，CircleClassやSquareClass，CrossClassに提供する
  - そのためには共通項を探る

クラス名	ObjectBase	CircleClass	SquareClass	CrossClass
インスタンス変数	x，y，vx，vy	x，y，vx，vy	x，y，vx，vy	x，y，vx，vy
コンストラクタ	ObjectBase() 座標速度初期化	CircleClass() 座標速度初期化	SquareClass() 座標速度初期化	CrossClass() 座標速度初期化
void move()	はねかえる	はねかえる	はねかえる	はねかえらない
void display()	点を描く	円を描く	正方形を描く	x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 x;
  float y;
  float vx;
  float vy;

  ObjectBase() {
    x = random(width);
    y = random(height);
    vx = random(-5, 5);
    vy = random(-5, 5);
  }

  void display(){
    point(x, y);
  }
```

display() は点を描くに
move() は跳ね返るに

```
  void move() {
    x += vx;
    y += vy;
    if(x > width) {
      x = width * 2 - x;
      vx = -vx;
    }
    if(x < 0) {
      x = -x;
      vx = -vx;
    }
    if(y > height) {
      y = height * 2 - y;
      vy = -vy;
    }
    if(y < 0) {
      y = -y;
      vy = -vy;
    }
  }
}
```

ObjectBaseと○□X

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 x;
  float y;
  float vx;
  float vy;

  ObjectBase() {
    x = random(widt
    y = random(heig
    vx = random(-5,
    vy = random(-5,
  }

  void display(){
    point(x, y);


  }
```

```
class CircleClass
{
  float x;
  float y;
  float vx;
  float vy;

  CircleClass() {
    x = random(widt
    y = random(heig
    vx = random(-5,
    vy = random(-5,
  }

  void display() {
    fill(255, 0, 0)
    ellipse(x, y, 3

  }
```

```
class SquareClass
{
  float x;
  float y;
  float vx;
  float vy;

  SquareClass() {
    x = random(widt
    y = random(heig
    vx = random(-5,
    vy = random(-5,
  }

  void display() {
    fill(0, 0, 255)
    rect(x-15, y-15


  }
```

```
class CrossClass
{
  float x;
  float y;
  float vx;
  float vy;

  CrossClass() {
    x = random(widt
    y = random(heig
    vx = random(-5,
    vy = random(-5,
  }

  void display() {
    stroke(0, 0, 0)
    line(x-15, y-15
    line(x-15, y+15
  }
```

ObjectBaseと○□X

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() {
    x += vx;
    y += vy;
    if(x > width) {
      x = width * 2
      vx = -vx;
    }
    if(x < 0) {
      x = -x;
      vx = -vx;
    }
    if(y > height) {
      y = height * 2
      vy = -vy;
    }
    if(y < 0) {
      y = -y;
      vy = -vy;
    }
  }
}
```

```
void move() {
    x += vx;
    y += vy;
    if(x > width) {
      x = width * 2
      vx = -vx;
    }
    if(x < 0) {
      x = -x;
      vx = -vx;
    }
    if(y > height) {
      y = height * 2
      vy = -vy;
    }
    if(y < 0) {
      y = -y;
      vy = -vy;
    }
  }
}
```

```
void move() {
    x += vx;
    y += vy;
    if(x > width) {
      x = width * 2
      vx = -vx;
    }
    if(x < 0) {
      x = -x;
      vx = -vx;
    }
    if(y > height)
      y = height *
      vy = -vy;
    }
    if(y < 0) {
      y = -y;
      vy = -vy;
    }
  }
}
```

```
void move() {
    x += vx;
    y += vy;
    x = (x + width)
    y = (y + height
  }
}
```

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

- CircleClassはObjectBaseの変数（x，y，vx，vy）や機能（移動や初期化）をもち，独自の丸の描画の機能をもつ
- SquareClassはObjectBaseの変数（x，y，vx，vy）や機能（移動や初期化）をもち，独自の□の描画の機能をもつ
- CrossClassはObjectBaseの変数（x，y，vx，vy）や機能（初期化）をもち，独自の移動とx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

- CircleClassはObjectBaseの変数（x，y，vx，vy）や機能（移動や初期化）をもち，独自の丸の描画の機能をもつ
- SquareClassはObjectBaseの変数（x，y，vx，vy）や機能（移動や初期化）をもち，独自の□の描画の機能をもつ
- CrossClassはObjectBaseの変数（x，y，vx，vy）や機能（初期化）をもち，独自の移動とx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 x;
  float y;
  float vx;
  float vy;

  ObjectBase() {
    x = random(widt
    y = random(heig
    vx = random(-5,
    vy = random(-5,
  }

  void display(){
    point(x, y);


  }
```

```
class CircleClass
{
  float x;
  float y;
  float vx;
  float vy;

  CircleClass() {
    x = random(widt
    y = random(heig
    vx = random(-5,
    vy = random(-5,
  }

  void display() {
    fill(255, 0, 0)
    ellipse(x, y, 3

  }
```

```
class SquareClass
{
  float x;
  float y;
  float vx;
  float vy;

  SquareClass() {
    x = random(widt
    y = random(heig
    vx = random(-5,
    vy = random(-5,
  }

  void display() {
    fill(0, 0, 255)
    rect(x-15, y-15


  }
```

```
class CrossClass
{
  float x;
  float y;
  float vx;
  float vy;

  CrossClass() {
    x = random(widt
    y = random(heig
    vx = random(-5,
    vy = random(-5,
  }

  void display() {
    stroke(0, 0, 0)
    line(x-15, y-15
    line(x-15, y+15
  }
```

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() {
    x += vx;
    y += vy;
    if(x > width) {
      x = width * 2
      vx = -vx;
    }
    if(x < 0) {
      x = -x;
      vx = -vx;
    }
    if(y > height) {
      y = height * 2
      vy = -vy;
    }
    if(y < 0) {
      y = -y;
      vy = -vy;
    }
  }
}
```

```
void move() {
    x += vx;
    y += vy;
    if(x > width) {
      x = width * 2
      vx = -vx;
    }
    if(x < 0) {
      x = -x;
      vx = -vx;
    }
    if(y > height) {
      y = height * 2
      vy = -vy;
    }
    if(y < 0) {
      y = -y;
      vy = -vy;
    }
  }
}
```

```
void move() {
    x += vx;
    y += vy;
    if(x > width) {
      x = width * 2
      vx = -vx;
    }
    if(x < 0) {
      x = -x;
      vx = -vx;
    }
    if(y > height)
      y = height *
      vy = -vy;
    }
    if(y < 0) {
      y = -y;
      vy = -vy;
    }
  }
}
```

```
void move() {
    x += vx;
    y += vy;
    x = (x + width)
    y = (y + height
  }
}
```

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

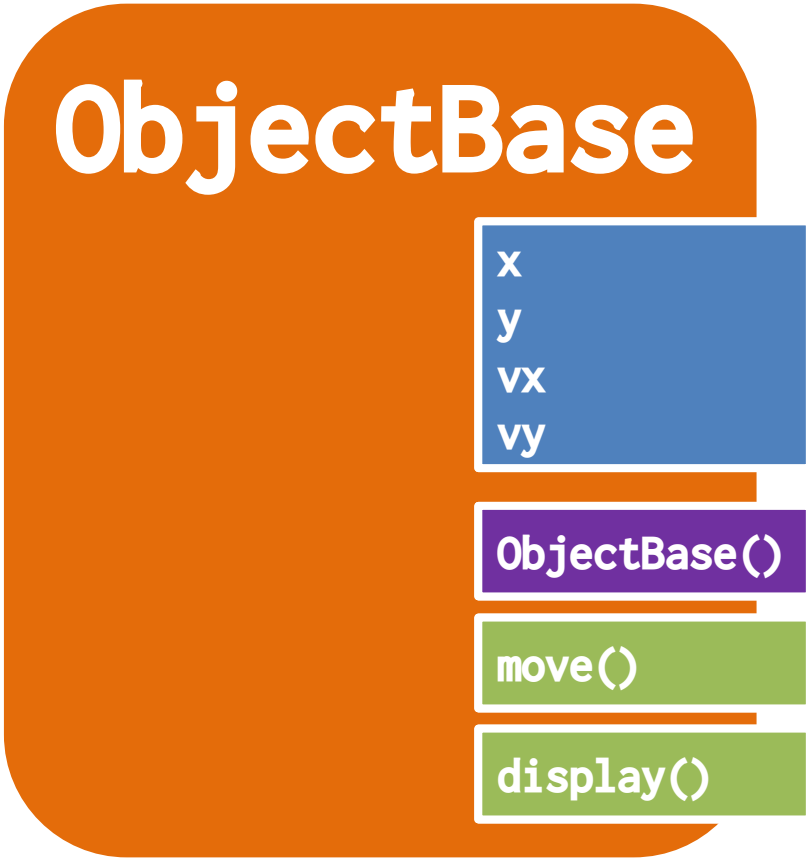
- 継承すると、親の力をすべて引き継ぐ！
  - 親の資産、能力、機能をもらいうける
- 継承の方法は extends とやるだけ！

```
class クラス名 extends 親クラス名
{
}
```

CircleClass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

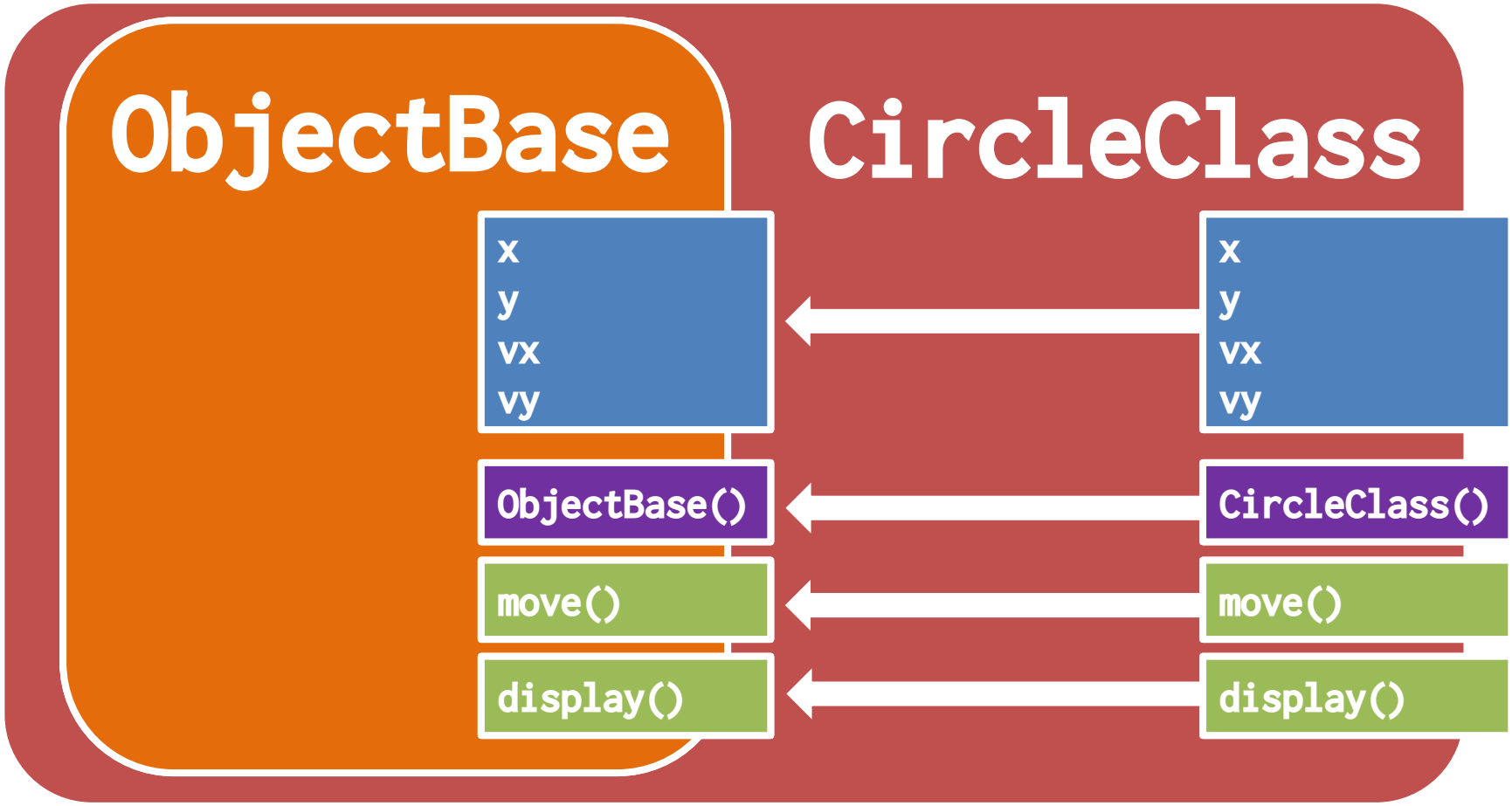
- ObjectBaseクラスを継承してCircleClassを作る！



CircleClass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 CircleClass 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

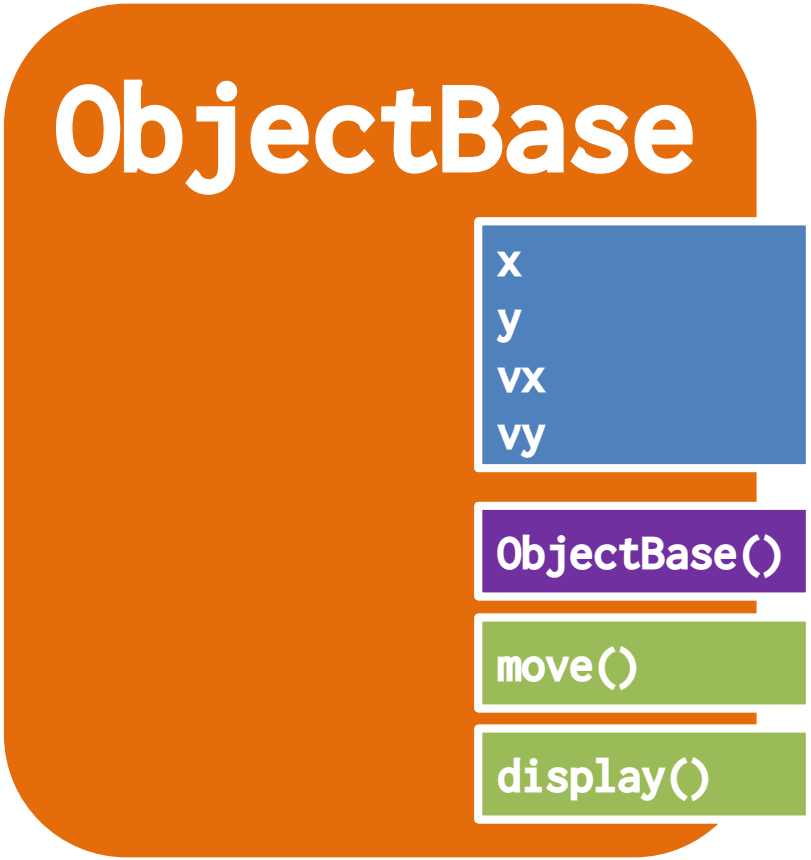
```
CircleClass circle1;
CircleClass circle2;

void setup() {
  size( 400, 300 );

  circle1 = new CircleClass();
  circle2 = new CircleClass();
}

void draw() {
  background(255);

  circle1.move();
  circle2.move();
  circle1.display();
  circle2.display();
}
```

- 点が描画されるだけ
  - なんで？
- ObjectBaseのdisplay()は店の描画だから！
  - 丸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！！

CircleClass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

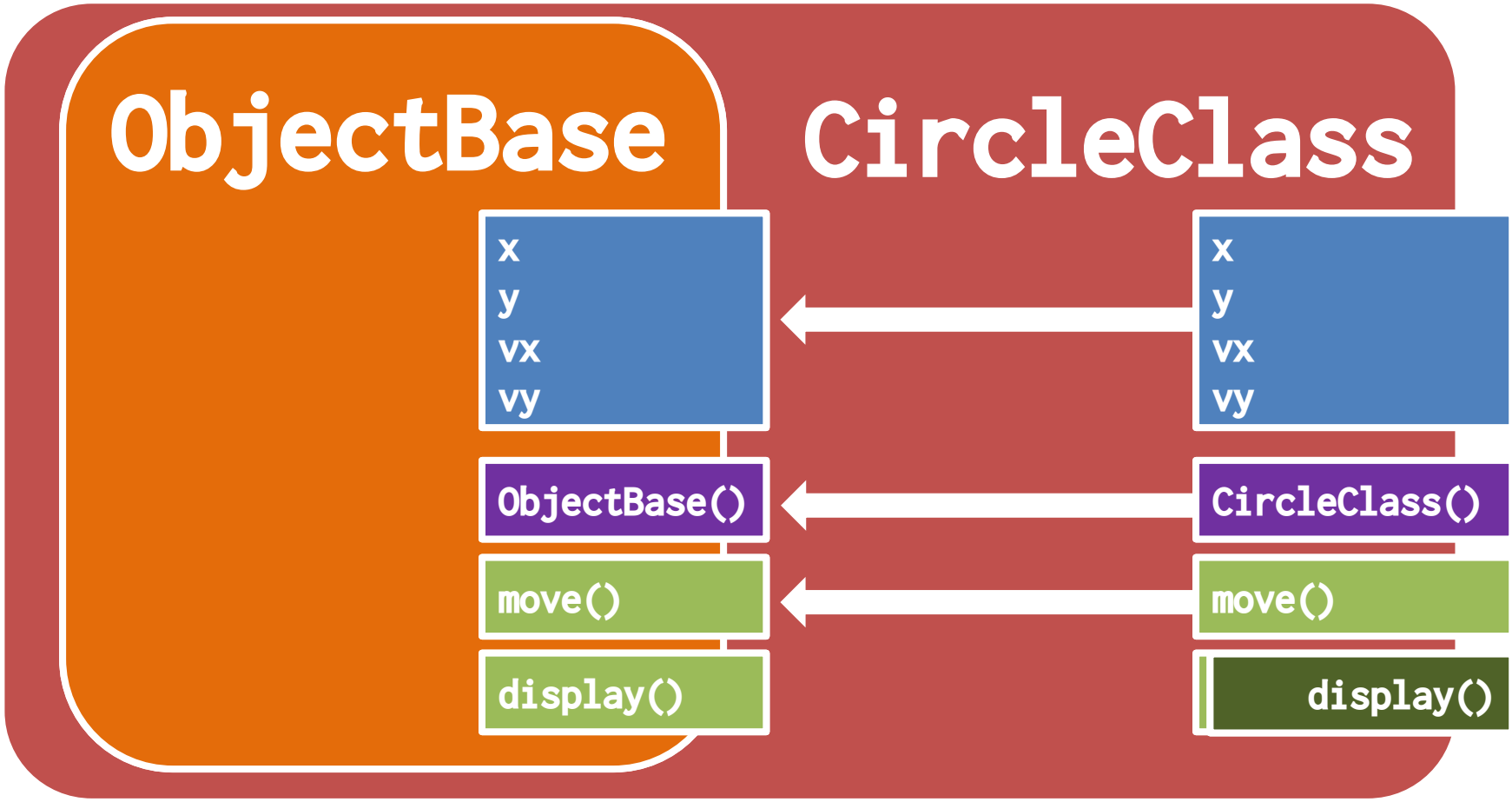
- ObjectBase の変数やメソッドを引き継ぎつつ，必要なところを上書きする！



CircleClass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 CircleClass extends ObjectBase
{
  void display(){
    fill(255, 0, 0);
    ellipse(x, y, 30, 30);
  }
}
```

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

CircleClass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

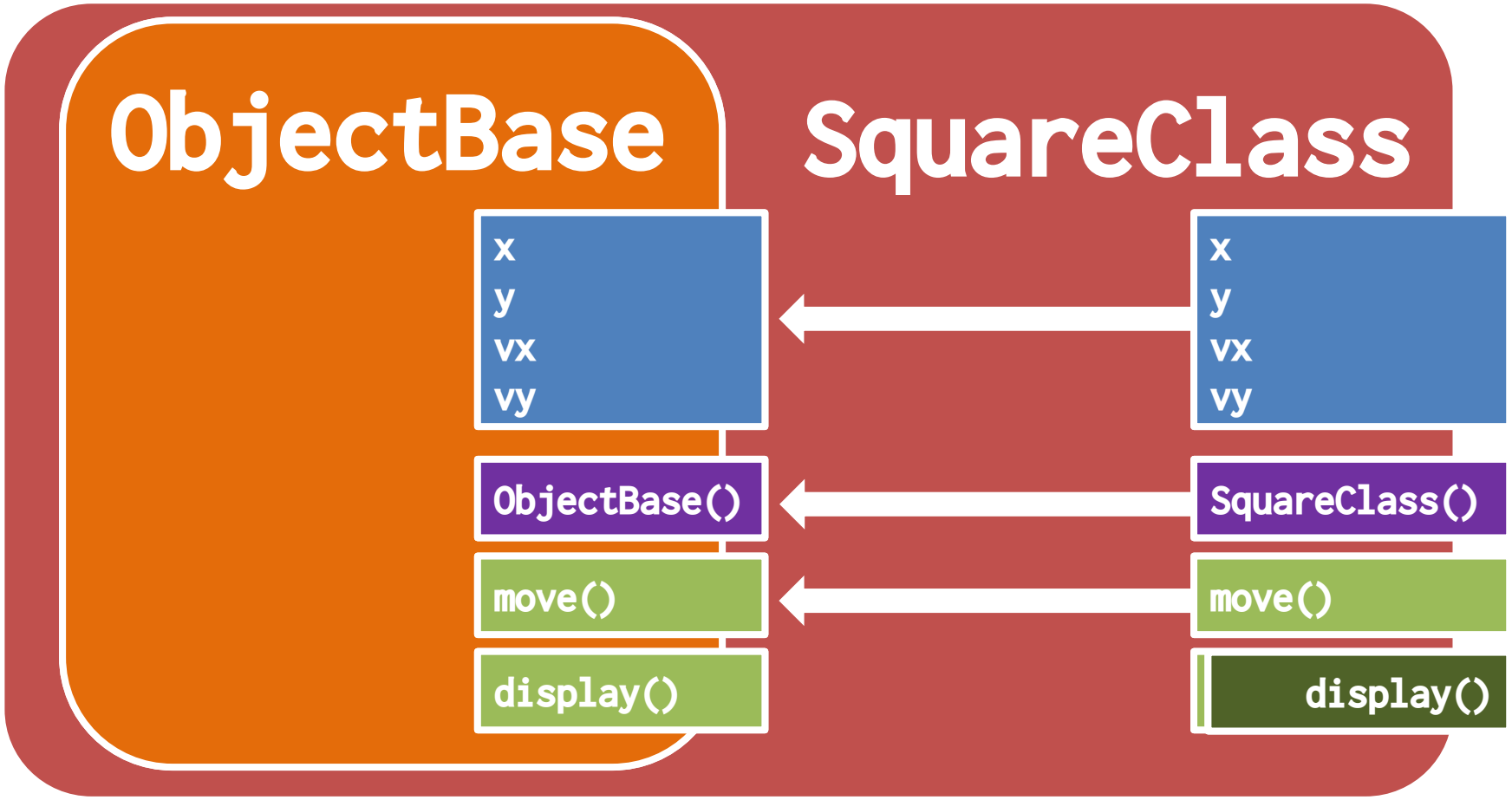
```
CircleClass circle1;
CircleClass circle2;

void setup() {
  size( 400, 300 );

  circle1 = new CircleClass();
  circle2 = new CircleClass();
}

void draw() {
  background(255);

  circle1.move();
  circle2.move();
  circle1.display();
  circle2.display();
}
```

- 表示された！
  - display()が上書きされている！

SquareClass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 SquareClass extends ObjectBase
{
  void display(){
    fill(0, 0, 255);
    rect(x-15, y-15, 30, 30);
  }
}
```

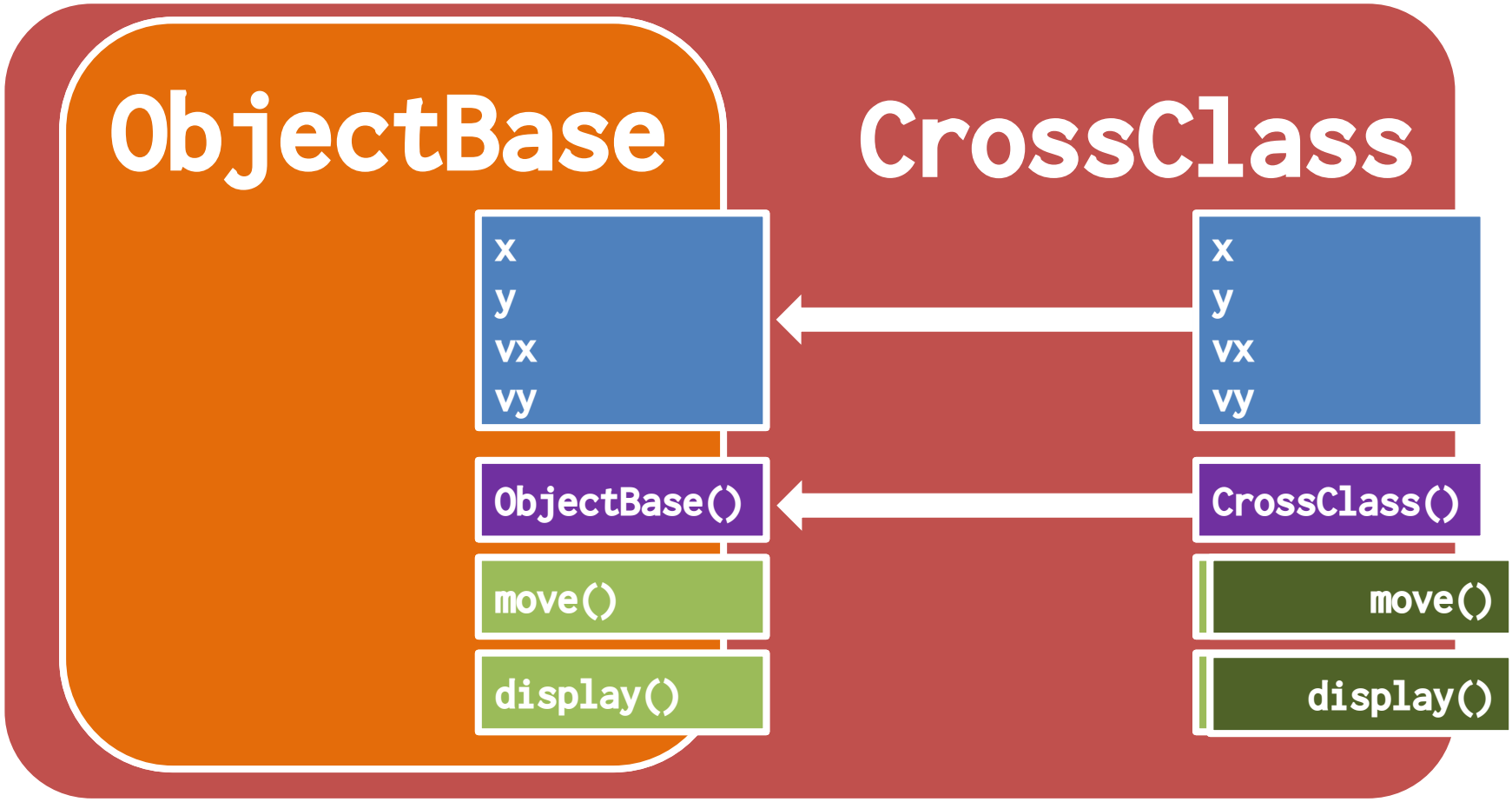
displayをオーバーライドして
親のdisplayメソッドが
呼ばれないようにする

SquareClassが劇的に短く！

CrossClass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() を上書きしてしまう
  - 跳ね返らないなので move をオーバーライド！



CrossClass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 CrossClass extends ObjectBase
{
  void display(){
    stroke(0, 0, 0);
    line(x-15, y-15, x+15, y+15);
    line(x-15, y+15, x+15, y-15);
  }

  void move() {
    x += vx;
    y += vy;
    x = (x + width) % width;
    y = (y + height) % height;
  }
}
```

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

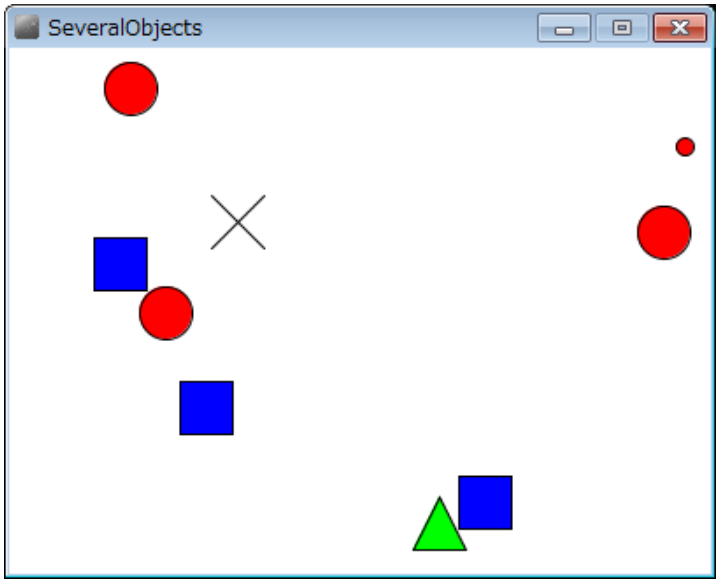
- 使うときは，継承していることは気にしないで，対象とするクラスを使うだけでよい！
  - move()
  - display()
  - すばらしい！！

```
CircleClass circle;
SquareClass square;
CrossClass cross;

void setup() {
  size( 400, 300 );
  circle = new CircleClass();
  square = new SquareClass();
  cross = new CrossClass();
}

void draw() {
  background(255);
  circle.move();
  square.move();
  cross.move();
  circle.display();
  square.display();
  cross.display();
}
```

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 クラスを継承して緑色の三角形を描画するTriangleClassを作るには？
  - 三角形は壁をすり抜けて反対側から出てくる



三角形のクラス

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

- 動く緑色の三角形のクラス

```
class TriangleClass extends ObjectBase
{
  void display(){
    stroke(0, 255, 0);
    triangle(x, y-15, x+10, y+15, x-10, y+15);
  }

  void move() {
    x += vx;
    y += vy;
    x = (x + width) % width;
    y = (y + height) % height;
  }
}
```

宿題2-1：hw_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

- CircleClassを改良し、正方形が動き回るSquareClass、xが動き回るCrossClass、△が動き回る TriangleClassを作成せよ
- またこれを利用して10個の〇と，5個の正方形と，5個のxと、3個の△が画面内を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、〇と□は壁で跳ね返り，xと△は跳ね返らずに反対側から出てくるようにせよ

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


```
CircleClass[] circle = new CircleClass[10];
SquareClass[] square = new SquareClass[5];
CrossClass[] cross = new CrossClass[5];
TriangleClass[] triangle = new TriangleClass[3];

void setup() {
  size(400, 300);
  for (int i=0; i<10; i++)
    circle[i] = new CircleClass();
  for (int i=0; i<5; i++)
    square[i] = new SquareClass();
  for (int i=0; i<5; i++)
    cross[i] = new CrossClass();
  for (int i=0; i<3; i++)
    triangle[i] = new TriangleClass();
}
```

```
void draw() {
  background(255);
  for(int i=0; i<10; i++){
    circle[i].move();
    circle[i].display();
  }
  for(int i=0; i<5; i++){
    square[i].move();
    square[i].display();
  }
  for(int i=0; i<5; i++){
    cross[i].move();
    cross[i].display();
  }
  for(int i=0; i<3; i++){
    triangle[i].move();
    triangle[i].display();
  }
}
```

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

- CircleClassとSquareClassとCrossClassとTriangleClassはObjectBaseを継承

- ObjectBase型はCircleClassもSquareClassもCrossClassもTriangleClassも含んでいるので、入れることができるよ！

- ポリモーフィズム！

```
ObjectBase[] list = new ObjectBase[10+5+5+3];

void setup() {
  size(400, 300);
  for (int i=0; i<10; i++)
    list[i] = new CircleClass();
  for (int i=0; i<5; i++)
    list[i+10] = new SquareClass();
  for (int i=0; i<5; i++)
    list[i+10+5] = new CrossClass();
  for (int i=0; i<3; i++)
    list[i+10+5+5] = new TriangleClass();
}

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

ArrayList型

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型は，いくつでも追加可能で色々なものを格納できるクラス

  （例） ArrayList list = new ArrayList(); で定義

  - ArrayListの要素数を取得

```
list.size();
```

  - ArrayListに対するaddで要素を追加

```
list.add( value ); // valueを追加
```

  - ArrayListに対するremoveで要素を削除

```
list.remove( index ); // index番目を削除
```

  - ArrayListに対するgetで要素を取得

```
list.get( index ); // index番目のオブジェクトを取得
```

ArrayList

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

- add()でリストに追加
- size()でリストのアイテム数を取得
- get(n)でリストからn番目のアイテム（オブジェクト）を取得
  - get(0)なら0番目
  - get(1)なら1番目
- remove(n)でリストからn番目を削除

```
ArrayList member = new ArrayList();

member.add("Nakamura");
member.add("Kobayashi");
member.add("Hashimoto");
member.add("Takahashi");

for(int i=0; i<member.size(); i++){
  println(member.get(i));
}

member.remove(0);
member.add("Miyashita");
member.add("Fukuchi");

for(int i=0; i<member.size(); i++){
  println(member.get(i));
}
```

ArrayListの定義

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

- 一応こちらでOK（getのとき要キャスト）

```
ArrayList list = new ArrayList();
```

- でも，こんな感じで型（クラス）もセットにしたほうが後々わかりやすいので推奨
  - <> をGenericsと言います
  - intやfloatはなく，Integer，Floatと書きます

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

```
ArrayList<Integer> list = new ArrayList<Integer>();
ArrayList<Float> list = new ArrayList<Float>();
ArrayList<String> member = new ArrayList<String>();
```

ArrayList<型>

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

- サイコロを1000回ふって、最後の10回分を表示するには？

```
ArrayList<Integer> history = new ArrayList<Integer>();

for(int i=0; i<1000; i++)
{
  int dice = (int)random(1, 7);
  history.add( dice );
}

for(int i=history.size()-10; i<history.size(); i++){
  println(history.get(i));
}
```

Generics で Integer を指定

ArrayList<型>

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

- サイコロを1000回ふって、最後の10回分を表示するには？

```
ArrayList history = new ArrayList();

for(int i=0; i<1000; i++)
{
  int dice = (int)random(1, 7);
  history.add( dice );
}

for(int i=history.size()-10; i<history.size(); i++){
  int num = (int)history.get(i);
  println( num );
}
```

Genericsを指定してないので
(int)でキャスト

追加/削除が自由！

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

- どうやって実現しているんだろう？ということを気にせずに，配列として使うことができる！
  - add(末尾に追加したいもの)
  - remove(削除したいアイテム番号)
  - get(取得したいアイテムの番号)
  - size()
  - を使えば，大きさを気にせず配列みたいなことを実現することができる！

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

<ObjectBase> で型を定義

```
ArrayList<ObjectBase> list = new ArrayList<ObjectBase>();
void setup() {
  size(400, 300);
  for (int i=0; i<10; i++){
    CircleClass circle = new CircleClass();
    list.add( circle );
  }
  for (int i=0; i<5; i++){
    SquareClass square = new SquareClass();
    list.add( square );
  }
  for (int i=0; i<5; i++){
    CrossClass cross = new CrossClass();
    list.add( cross );
  }
  for (int i=0; i<3; i++){
    TriangleClass triangle = new TriangleClass();
    list.add( triangle );
  }
}
```

CircleClassも
SquareClassも
CrossClassも
TriangleClassも
入れることができるよ！

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

<ObjectBase> で型を定義

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

void setup() {
  size(400, 300);
  for (int i=0; i<10; i++){
    list.add( new CircleClass() );
  }
  for (int i=0; i<5; i++){
    list.add( new SquareClass() );
  }
  for (int i=0; i<5; i++){
    list.add( new CrossClass() );
  }
  for (int i=0; i<3; i++){
    list.add( new TriangleClass() );
  }
}
```

new ClassName()
を直接 add できるよ！

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<10; i++)
    list.add( new CircleClass() );
  for (int i=0; i<5; i++)
    list.add( new SquareClass() );
  for (int i=0; i<5; i++)
    list.add( new CrossClass() );
  for (int i=0; i<3; i++)
    list.add( new TriangleClass() );
}

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

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<10; i++)
    list.add( new CircleClass() );
  for (int i=0; i<5; i++)
    list.add( new SquareClass() );
  for (int i=0; i<5; i++)
    list.add( new CrossClass() );
  for (int i=0; i<3; i++)
    list.add( new TriangleClass() );
}

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<10; i++)
    list.add( new CircleClass() );
  for (int i=0; i<5; i++)
    list.add( new SquareClass() );
  for (int i=0; i<5; i++)
    list.add( new CrossClass() );
  for (int i=0; i<3; i++)
    list.add( new TriangleClass() );
}

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

めっちゃシンプル！

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

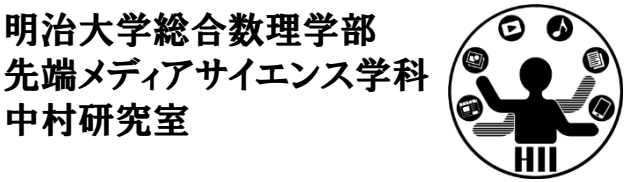
ObjectBaseを継承し，800x600のウインドウ内で，任意の場所x，yから任意の速度で移動する3つの赤色の円（直径30ピクセルの惑星）を描画し，右端・左端・上端・下端に来ると跳ね返るようにする．また，赤色の円（惑星）には円の中心から50の距離があるところに1つの衛星（直径10ピクセル）があり，5度ずつ円の周りを回転するようにせよ

- 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

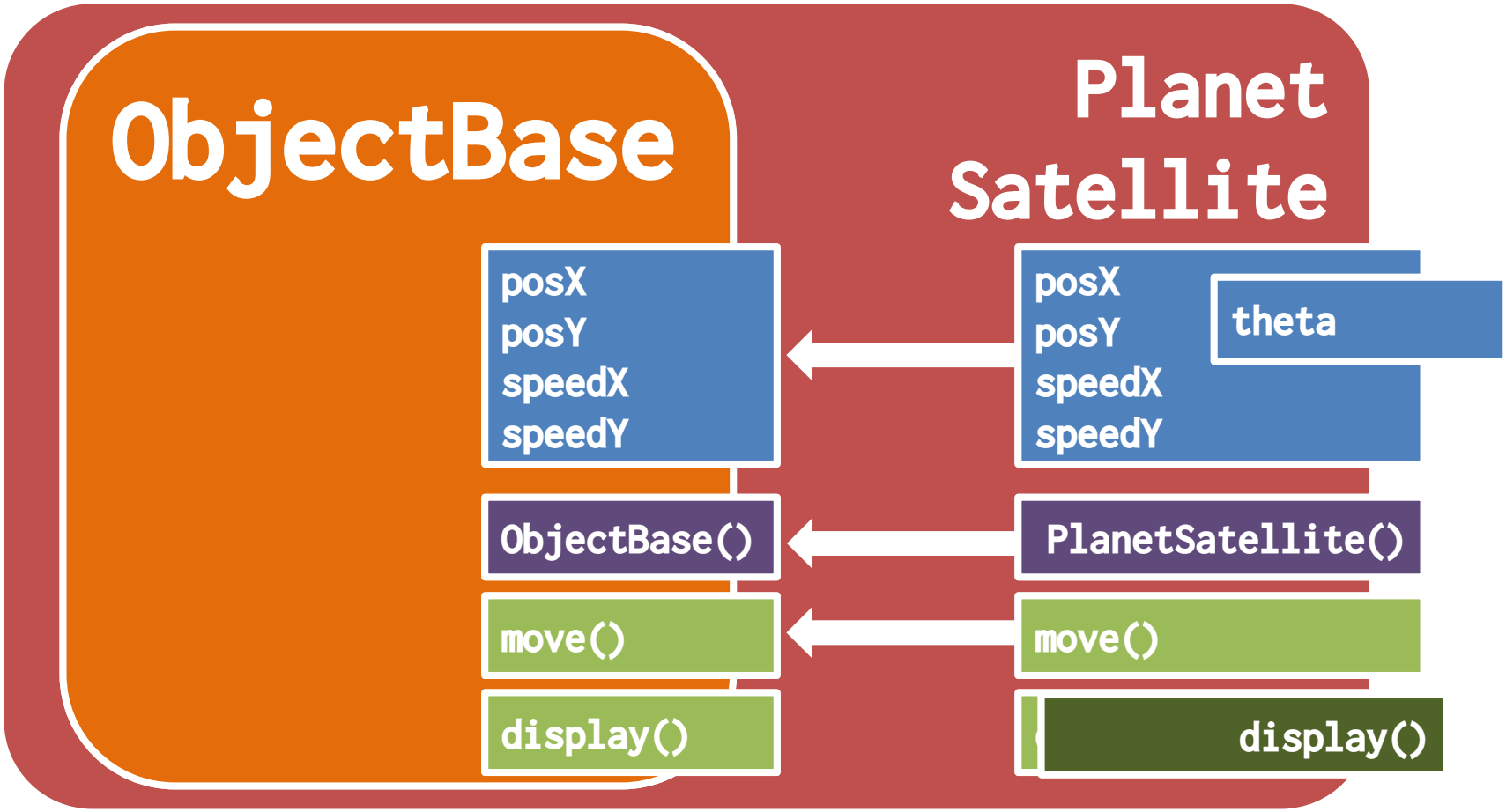
```
(x+50*cos(radians(theta)), y+50*sin(radians(theta)))
```

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

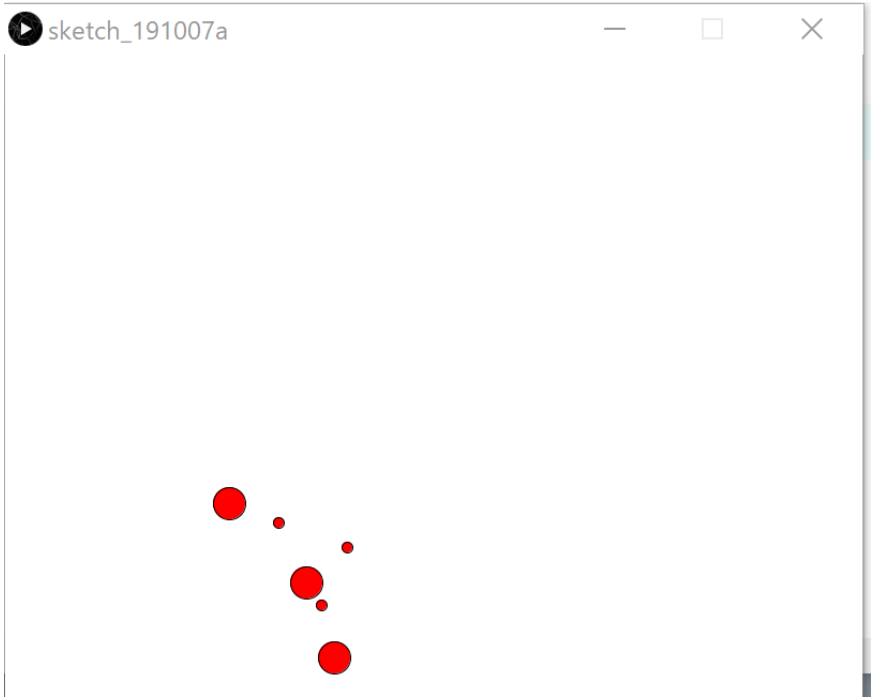
- ObjectBaseを継承して，変数を追加する

```
class PlanetSatellite extends ObjectBase
{
  int theta;

  void display()
  {
    fill( 255, 0, 0 );
    ellipse(x, y, 30, 30);
    theta = theta + 5;
    float sx = x + 50*cos(radians(theta));
    float sy = y + 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

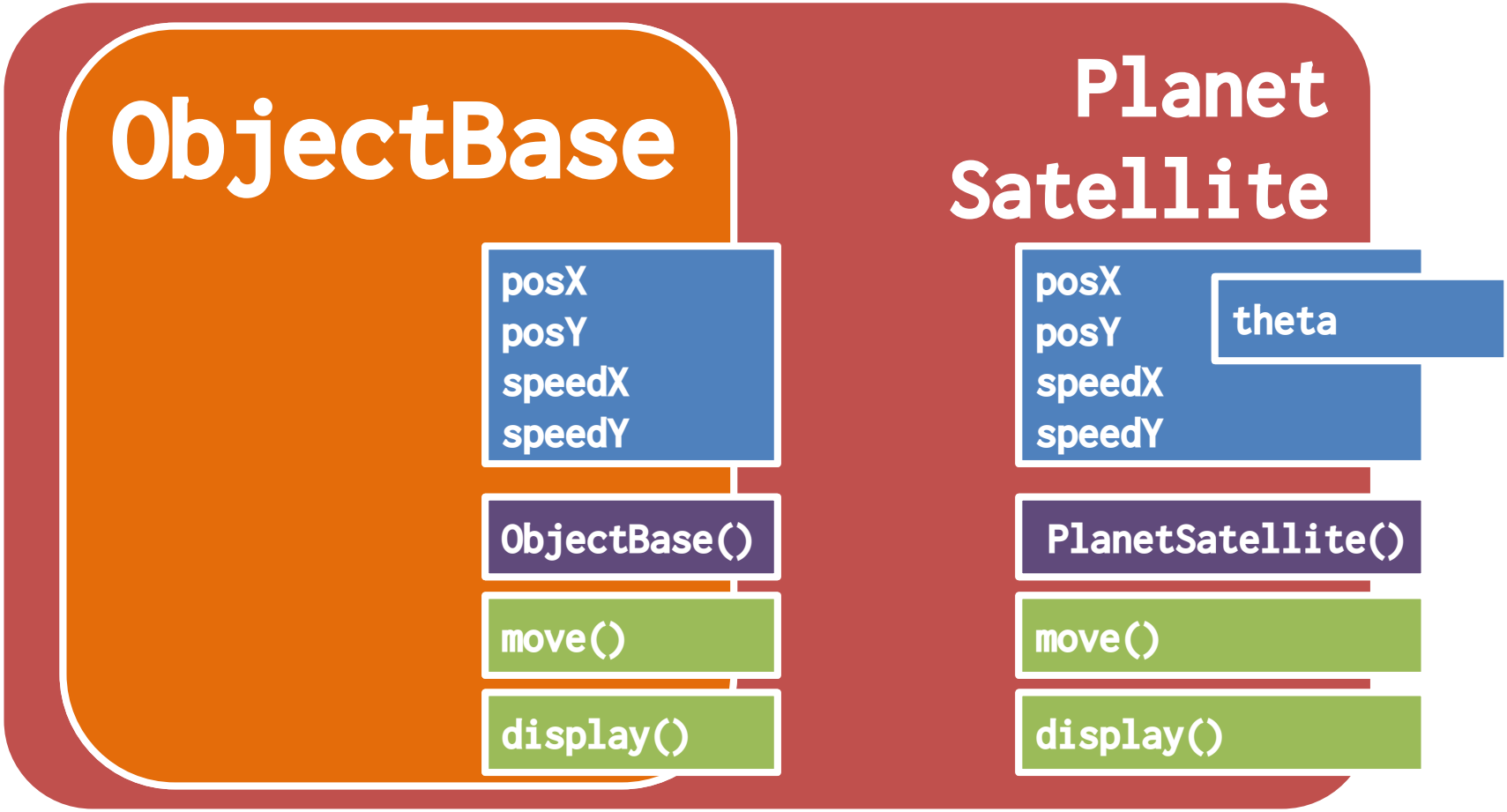
衛星の開始角を0～360度の任意の場所にしたい．どうするか？

- 考え方
  - コンストラクタで指定したらOK？

継承すると・・・

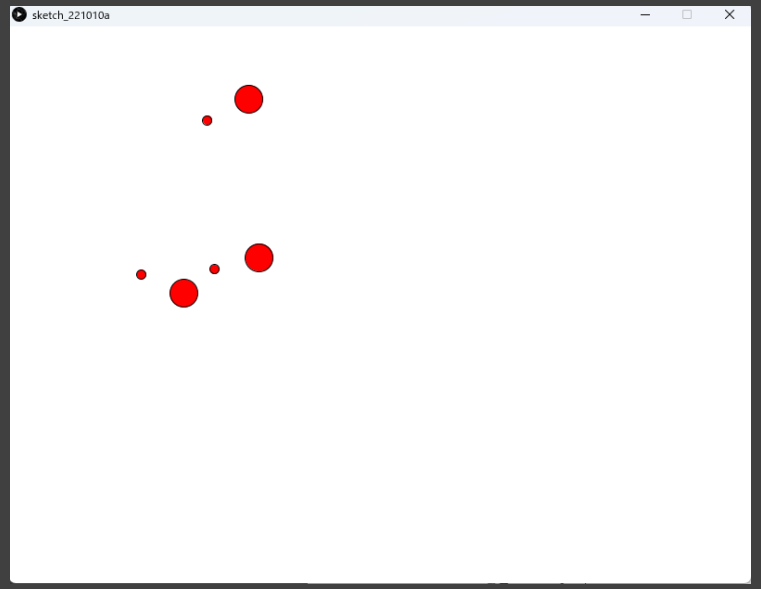
明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

- 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！



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);
  }
}
```

今週のまとめ

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ArrayListを紹介
  - 継承は変数や関数を引き継ぐ
  - 継承されたものはまとめて扱うことが可能
  - ArrayListは柔軟な配列みたいなもので便利なのでマスターしよう！