

プログラミング基礎

アニメーション課題 補足！

良いプログラムの原則

- できるだけシンプルに書く！
- 同じことを2回以上書かない！
 - コピペでもダメ！
- アニメーション課題を通じて学びます

Q. 猫と犬を登場させたい(1)

画像が違うだけの場合

× Cat クラスをコピーして Dog クラスを作る

○ 猫も犬も同じクラスで表す

(画像はメンバ変数として覚えておくようにする)

```
class Animal{
    int x, y;
    int vx, vy;
    PImage img;

    void setImage(PImage img){
        this.img = img;
    }

    void draw(){
        image(img, x - img.width / 2, y - img.height / 2);
    }
}
```

以下省略

(つづき)

```
Animal cats[];
Animal dog;

void setup(){
    PImage catImage = loadImage("...");
    PImage dogImage = loadImage("...");

    for(int i = 0; i < cats.length; i++){
        cats[i] = new Animal();
        cats[i].setImage(catImage);
        // 場所や速度も設定する
    }

    dog = new Animal();
    dog.setImage(dogImage);
    // 場所や速度も設定する
}
```

```
void draw(){
    for(int i = 0; i < cats.length; i++){
        // 猫たちを動かしたり、描画したりする
    }
    // 犬も動かしたり、描画したりする
}
```

Q. 猫と犬を登場させたい(2)

画像が違い、動きも違う

- 例) 猫は飛び跳ねるが、犬は地を這うだけ
 - △ 今度こそ Cat クラスと Dog クラスを作ろう
 - Animal クラスに両方の機能を盛り込む
 - Animal クラスを継承して Cat, Dog クラスを作る

クラスを継承する (やや難)

```
class Cat extends Animal {  
    int g = 2;  
  
    void move() {  
        super.move();  
  
        if (y == height) {  
            vy = -30; // jump  
        }  
        else {  
            vy = min(vy + g, 10);  
        }  
  
        if (!isInsideX()) {  
            bounceX();  
        }  
    }  
}
```

```
class Dog extends Animal {  
    void move() {  
        super.move();  
  
        if (!isInsideX()) {  
            bounceX();  
        }  
    }  
}
```

継承したクラスを使う（やや難）

- Dog型のオブジェクトはAnimal型の変数に入れられる（Catも同様）

```
Animal dog;  
dog = new Dog();
```

- それぞれのクラスの move 関数が使われる

```
for(int i = 0; i < cats.length; i++){  
    cats[i].move();  
    cats[i].draw();  
}  
  
dog.move();  
dog.draw();
```