

## Workshop#3 “Object”



Workshop#3は、「クラスを使ったオブジェクトを利用して作品を作ってみよう」という内容のワークショップでした。エージェントやアトラクターなどに続く仕組みを理解するため、サンプルプログラムでもclassを使って実装しています。

```
class Square {
  int x, y;
  Square(int x, int y) {
    this.x = x;
    this.y = y;
  }
  boolean isHover() {
    boolean isHoverX = x < mouseX && mouseX < x+width/4;
    boolean isHoverY = y < mouseY && mouseY < y+height/4;
    return isHoverX && isHoverY;
  }
  void draw() {
    if (isHover()) {
      fill(0, 0, 100);
    } else {
      fill(0, 0, 50);
    }
    rect(x, y, width/4, height/4);
  }
}
```

カーソルを載せると白くなる  
四角形を表す、Squareクラス  
を作りました。

関数isHoverによって、カー  
ソルが四角形の上に乗ってい  
るのかを判定します。

Squareクラスを利用するに  
は、クラスの外でSquareクラ  
スのインスタンスを作り、関  
数drawを呼び出します。

## Workshop#4 “Image”



Workshop#4は、「PGraphicsを使って作品を作ってみよう」という内容のワークショップでした。サンプルプログラムでは、画像を表示するレイヤーで表示された画像に図形が描画されていき、コマが進んでも維持されていることが分かります。

```
PGraphics backLayer, artLayer;
void setup() { // Draw only once
  size(960, 540);
  rectMode(CENTER); imageMode(CENTER);
  textAlign(CENTER, CENTER);
  textFont(createFont("Avenir-Black", 40));
  artLayer = createGraphics(400, 400);
  // 省略: artLayerで表示する画像を読み込み、描画
  backLayer = createGraphics(width, height);
  // 省略: artLayerの影として、ぼかした黒四角を描画
}
void draw() { // Draw 60 frames per second
  // 省略: カーソルの位置でartLayerに図形を描画
  push(); // Save previous axis
  translate(width/2, height/2);
  image(backLayer, 0, 0);
  image(artLayer, 0, 0);
  text(frameCount, 0, 0);
  pop(); // Revert to the previous axis
}
```

読み込んだ画像を表示する  
artLayerと、artLayerで表示  
した画像が落とした影とし  
て、黒い四角をずらして描画  
するbackLayerを作ります。

関数draw内で、artLayer上に  
カーソルが乗るときに図形  
(円など)を描画します。  
artLayerの状態が維持され、  
frameCountの数字が更新さ  
れていきます。