

メディアプログラミング演習—第4回（第2テーマ1日目）—

演習4-2：再帰図形

再帰関数を用いて、繰り返しのある平面図形を描く。

最初の簡単な例を示す。図1は、円の中に、半径が半分の円を繰り返し描いた図形である。これを実現する再帰プログラムは、図2となる (maru-pro)。

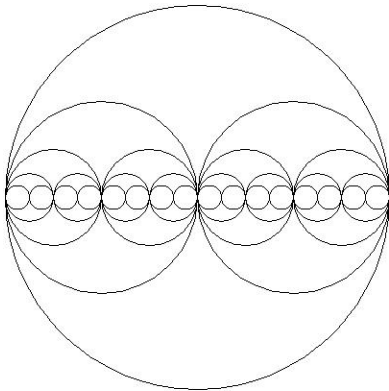


図1 円の繰り返し

```
void draw() {
    Maru(5, 250, 250, 200);
}

void Maru(int n, float x, float y, float r) {
    ellipse(x, y, r*2, r*2);
    if (n>1) {
        Maru(n-1, [redacted], y, [redacted]); //右側の円
        Maru(n-1, [redacted], y, [redacted]); //左側の円
    }
}
```

図2. 図1を描くプログラム

図形例2：次いで、図3の図形を「再帰的に」描くことを考える。

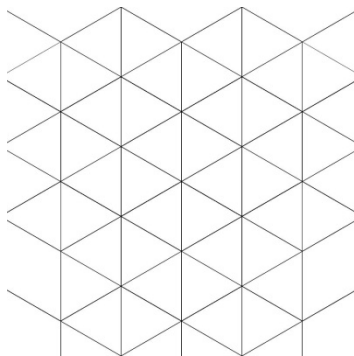


図3 最初の図形

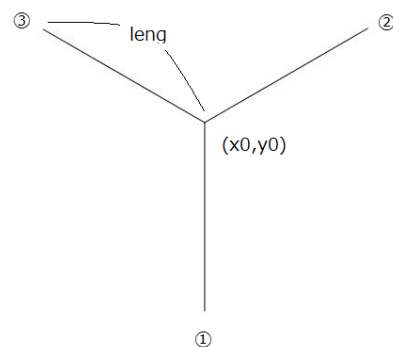


図4. 基本図形

この図形は、図4のようなY字図形を描く手続きを再帰的に①～③を(x0,y0)として適用することで描ける。プログラムは、以下の通りである。プログラム中(A)は、図4の周囲3点の座標を求めるとともに、中心から各点への線分を描いている。繰り返し数nが0でなければ、①～③の各頂点にて同じことを繰り返す(Bの部分) (lattice-pro)。

```

void draw(){
  drawfig1(5, 200.0, 500.0,500.0);  }
void drawfig1(int n, float leng,float XO,float YO) {
  float x, y, ang;  float[] rx,ry;
  rx=new float[3]; ry=new float[3];
  for(int k=0;k<3;k++)
    { ang=3.141592/180*k*120;
      rx[k] = leng*sin(ang) + XO;
      ry[k] = leng*cos(ang) + YO;
      line(XO,YO,rx[k],ry[k]);
    }
};

```

(A) }

```

if (n == 0) {return;};
  drawfig1(n - 1, leng, rx[0], ry[0]);
  drawfig1(n - 1, leng, rx[1], ry[1]);
  drawfig1(n - 1, leng, rx[2], ry[2]);

```

(B)

図5. 図3を描くプログラム

図形例3：例1を発展させると、同類の図6が描ける。この基本図形は図7に示す5分木線分であり、各頂点で、長さを変えて再帰的に描いた図形である (*penta-pro*)。

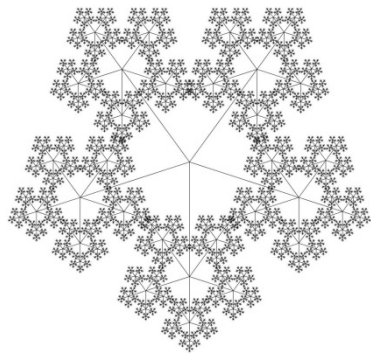


図6 繰り返し図形2

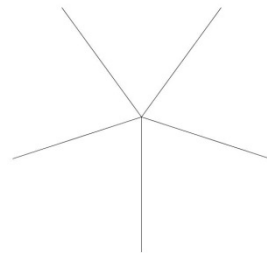


図7 図6の基本図形

図形例4：図8および図9を描きなさい

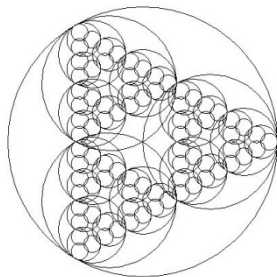


図8 図1の応用

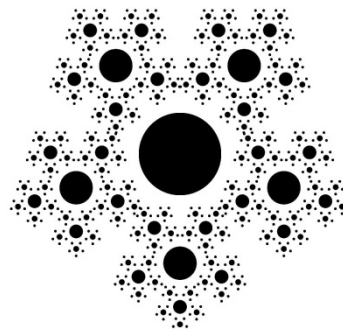


図9 図6の応用

球の塗りつぶし：setup 中の noFill() を削除し、

fill(r, g, b);ellipse(<適切なパラメータ>); とすればよい。