

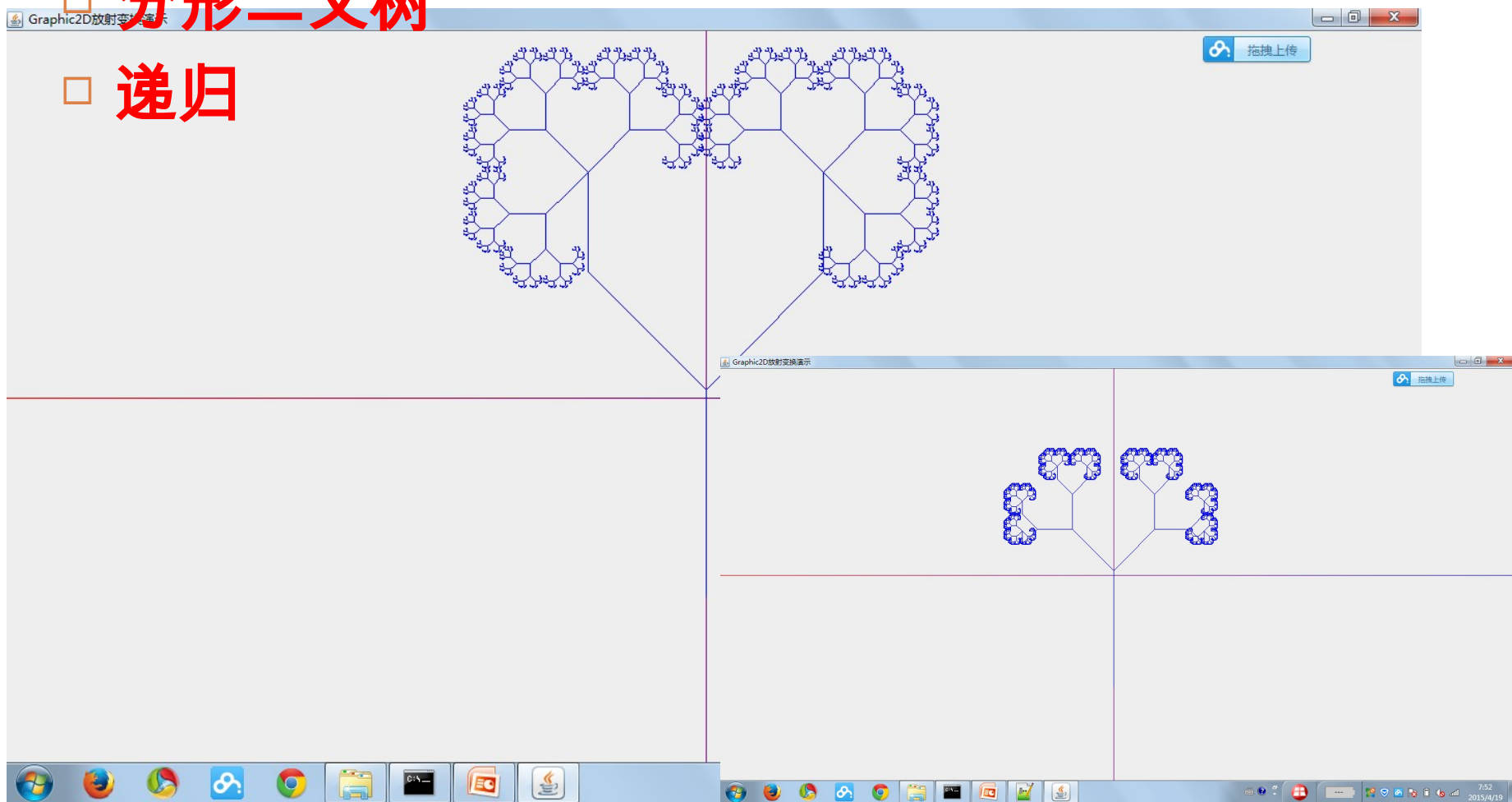
# 数据可视化

李春芳

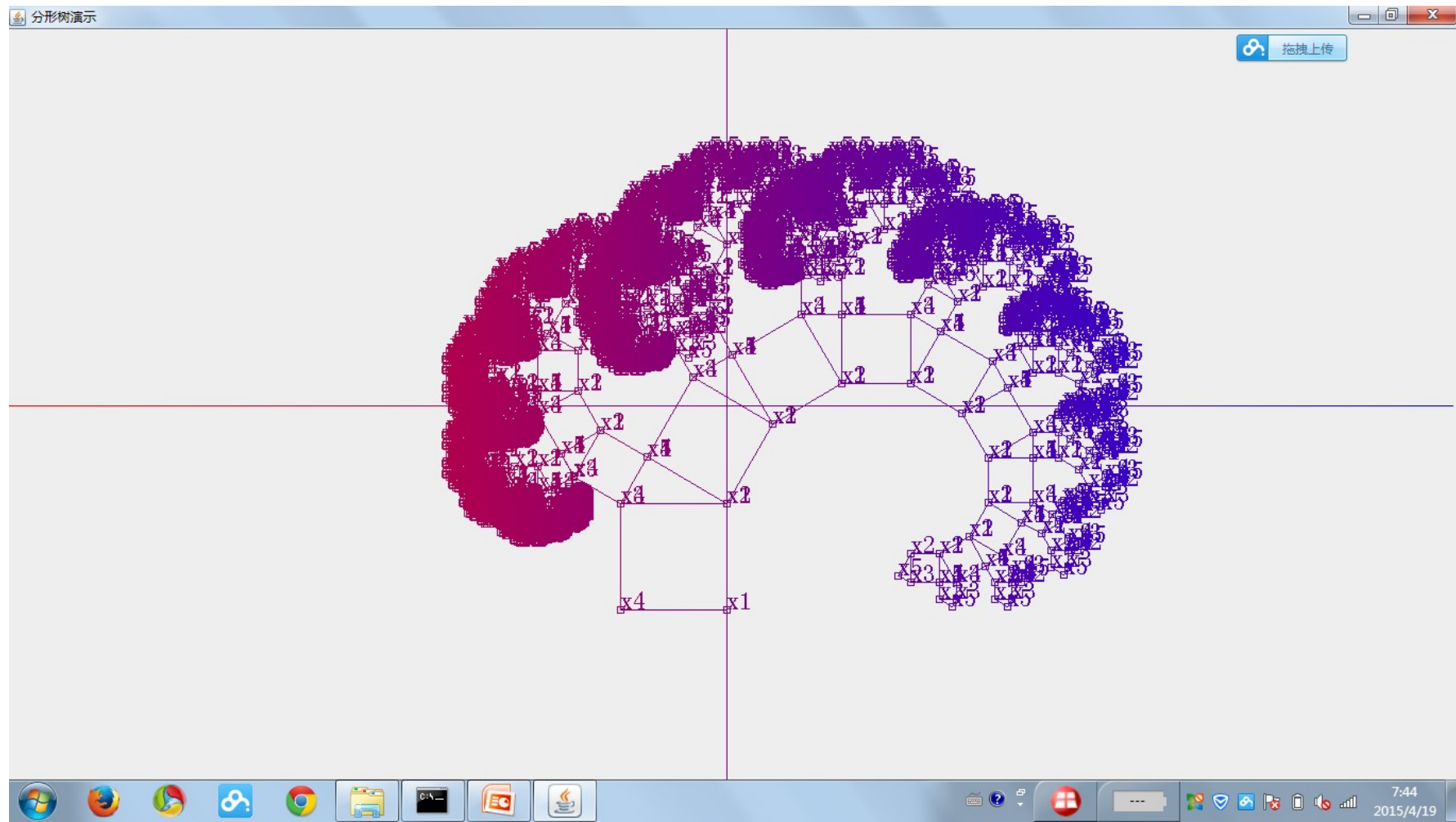
# 分形

□ 分形二叉树

□ 递归



# 分形

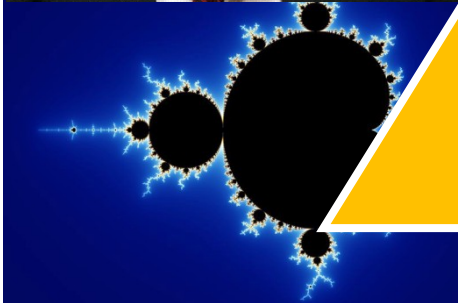


# 1. 分形学

**1975 年，本华·曼德博**

云不是球体，山不是圆锥体，海岸线不是圆，  
树皮不是光滑的，闪电传播的路径也不是直线

——曼德博，《大自然的分形几何学》绪论。



**性质**  
自相似  
(精确、半相似、统计)  
递归反馈系统

**造法：**

逃逸时间分形  
迭代函数系统  
随机分形  
奇异吸引子

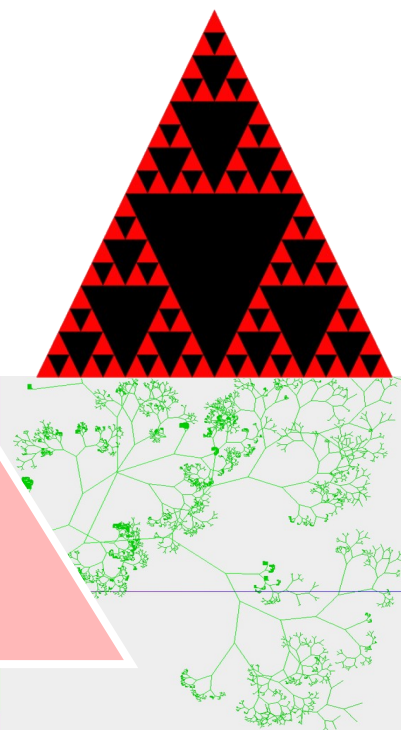
**分形学**  
**Fractal**

**自然存在**

云、山脉、闪电、海岸线  
、雪片、植物根、蔬菜  
(西兰花) 和动物毛皮等

**应用**

医学、土力学、地震学  
、技术分析、艺术.....



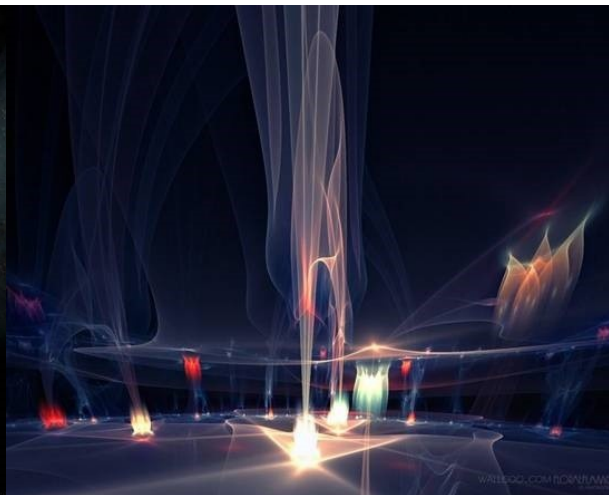
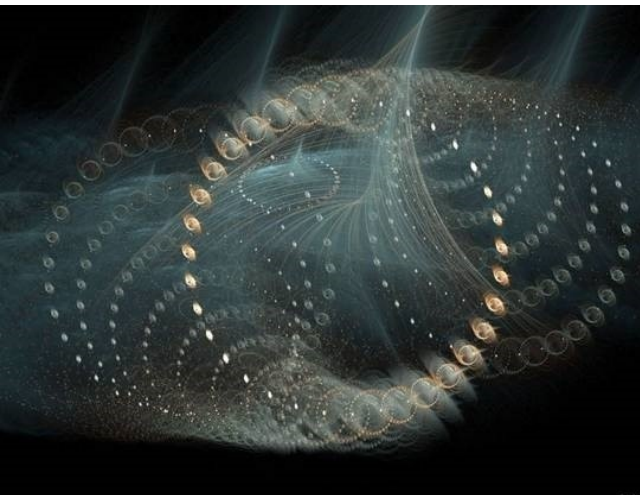
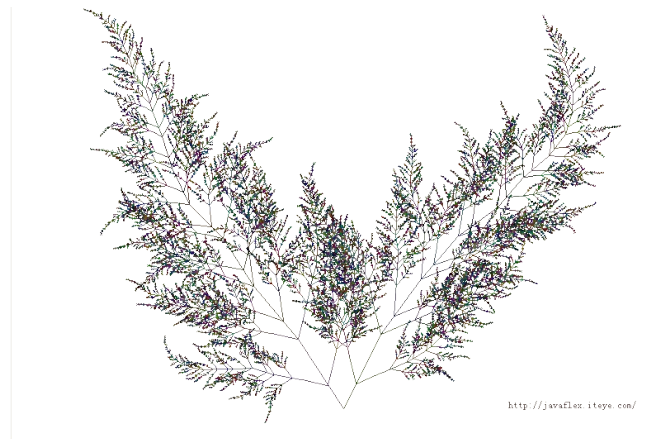


# 1. 自然界分形案例





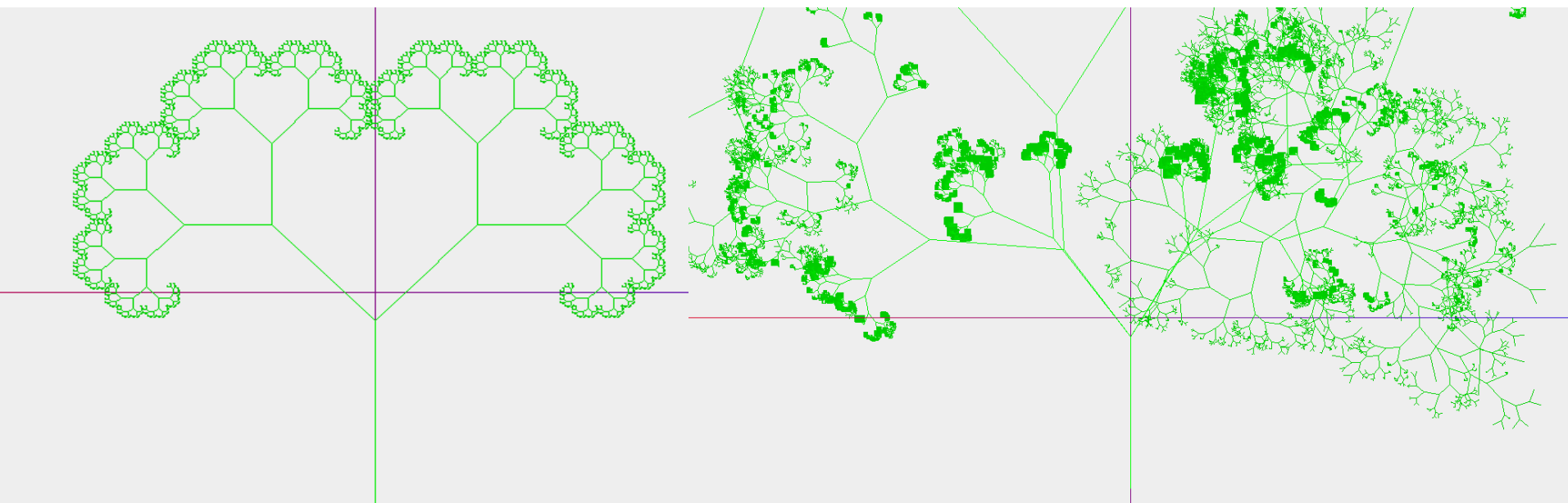
# 1. 人造分形图



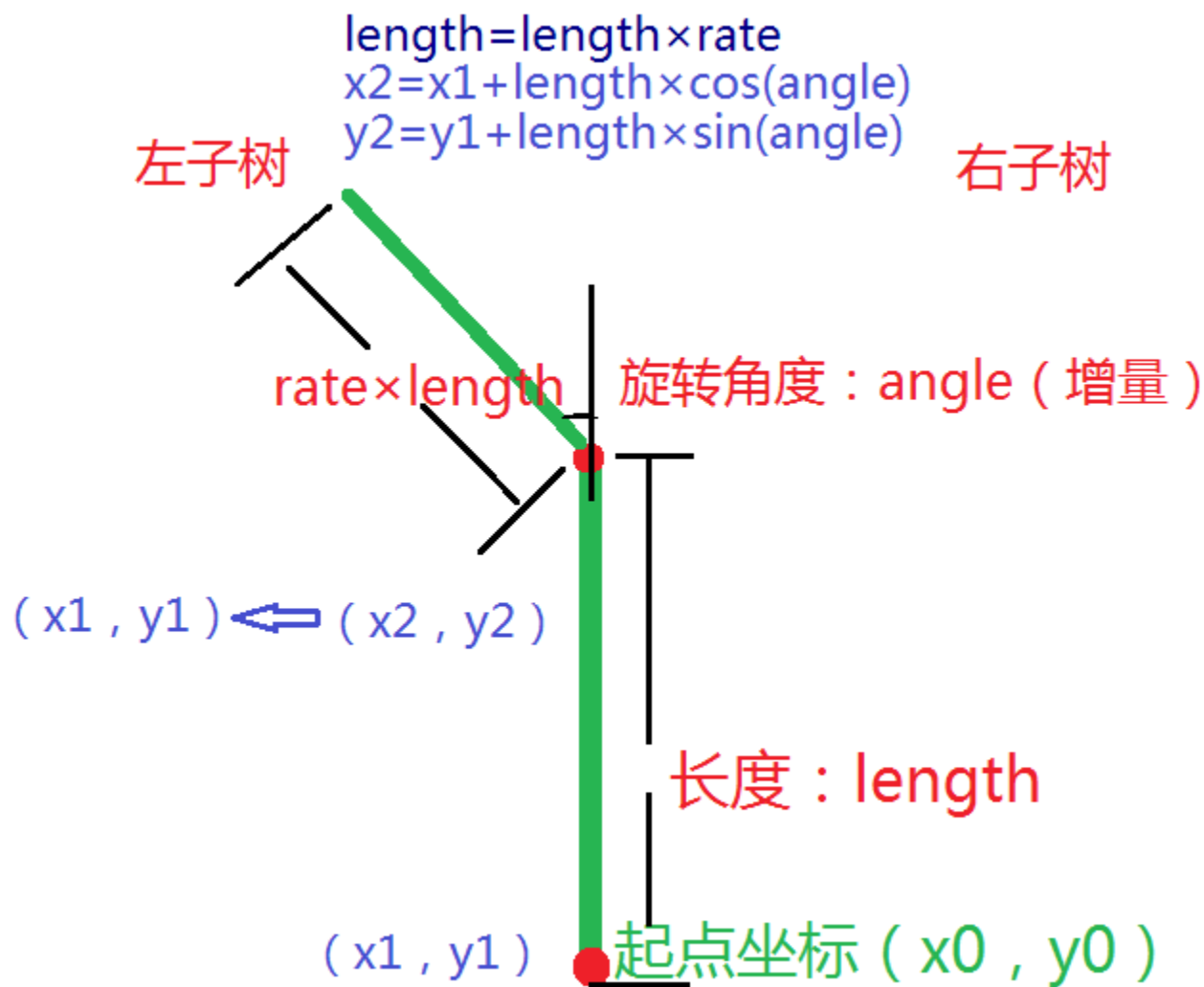
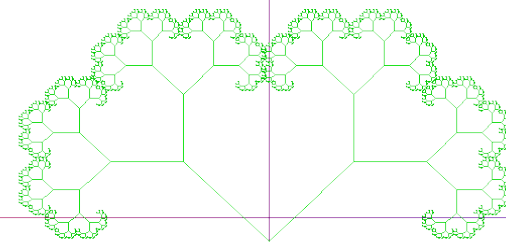
# 1. 分形二叉树——递归示意

□ 确定性：自相似

带随机性：统计自相似



# 1. 分形二叉树——递归示意





# 1. 分形二叉树——递归实现

## □ 递归函数：

`show(X 坐标, Y 坐标, 长度, 衰减系数, 角度, 迭代次数, 绘图区)`

## □ Java 实现：

`show(double x0,double y0,double length,double rate,double angle,int count,Graphics2D g2)`

## □ 主要代码：

初值：`show(myWidth/2,(int)(myHeight*3/4),200,0.5,-Math.PI/2,14,g2);`

绘直线： 起点：`x1=x0; y1=y0;`

终点：`x2=(int)(x1+length*Math.cos(angle));`

`y2=(int)(y1+length*Math.sin(angle));`

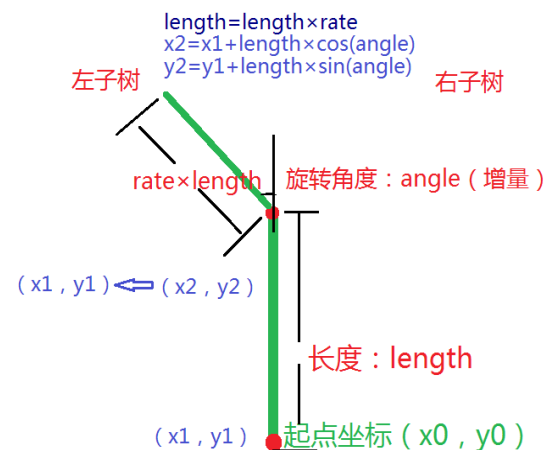
`lineL=new Line2D.Double(x1,y1,x2,y2);`

`g2.draw(lineL);`

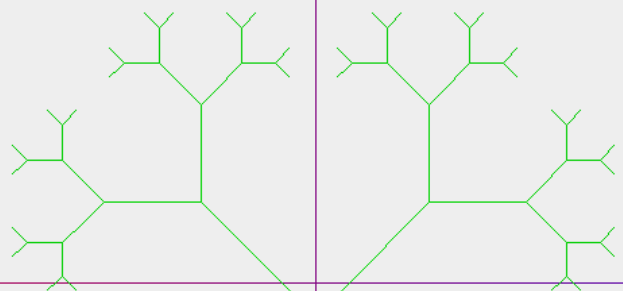
左子树：`show(x2,y2,length*rate,0.6,angle-Math.PI/4,count-1,g2)`

右子树：`show(x2,y2,length*rate,0.6,angle+Math.PI/4,count-1,g2);`

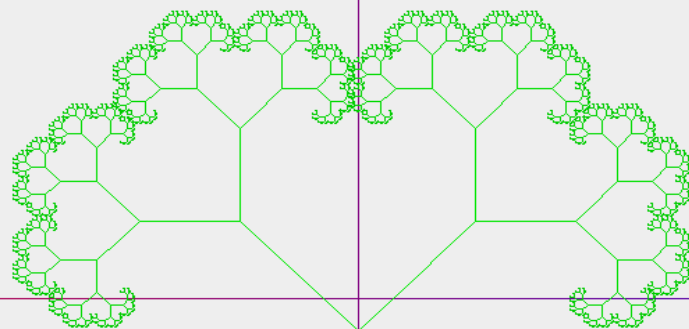
程序：`MyFractalTree.java`



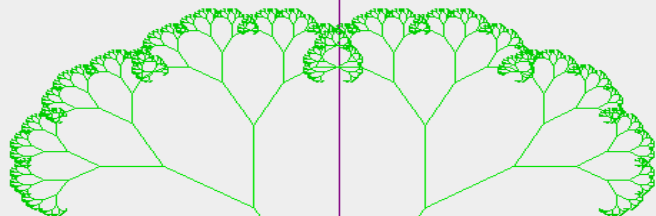
# 1. 分形二叉树——运行结果



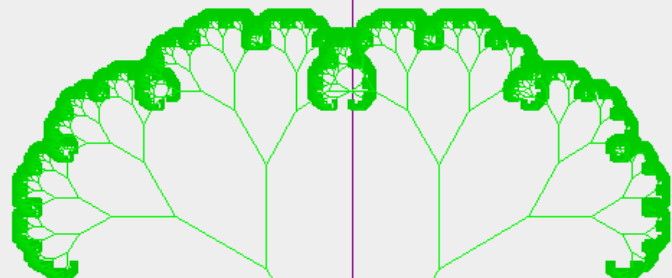
迭代 6 次，角度  $45^\circ$



迭代 12 次，角度  $45^\circ$



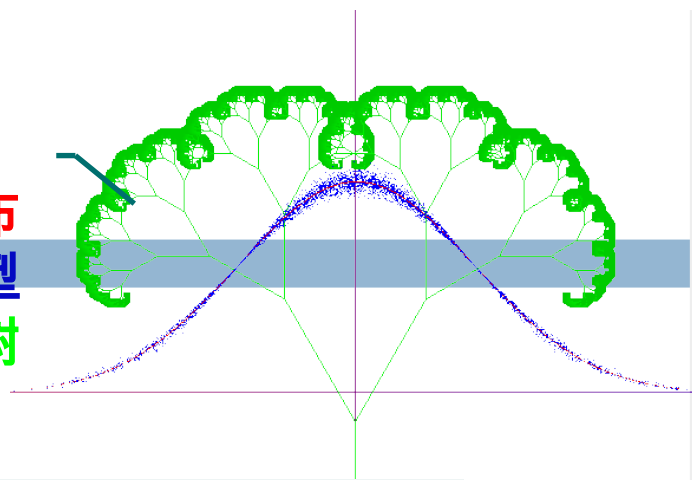
迭代 12 次，角度  $30^\circ$



迭代 18 次，角度  $30^\circ$

## 2. 正态分布 & 云模型

红色——正态分布  
蓝色——云模型  
绿色——分形二叉树



真实  
世界

本例：  
随机驱动分形二叉树的  $l_e$   
和  $\text{angle}$

李德毅院士提出  
随机的  
模糊的  
保持规则性、蕴含随机性



云模型  
二阶正态分布

正态分布

世界不是完全规则的  
也不是完全随机的  
规则中包含随机



词人：蔡世平  
现代旧体词，婉约派  
生活是毛绒绒的……

## 2. 正态分布——分形二叉树

### □ java.util.Random 类

方法 `public double nextGaussian()` : 返回下一个伪随机数，服从高斯分布，均值 0，标准差 1

### □ 正态分布密度函数

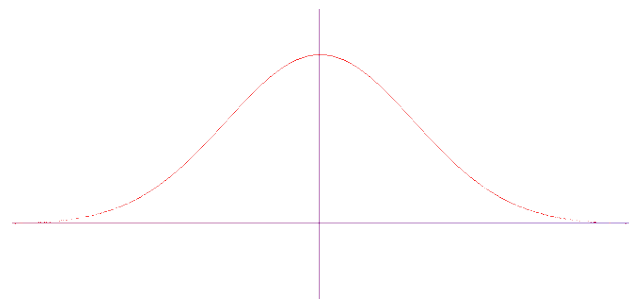
$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

### □ 主要代码：u : 均值    q : 方差

```
java.util.Random r1 = new java.util.Random();
```

```
x=(int)(u+q*r1.nextGaussian());
```

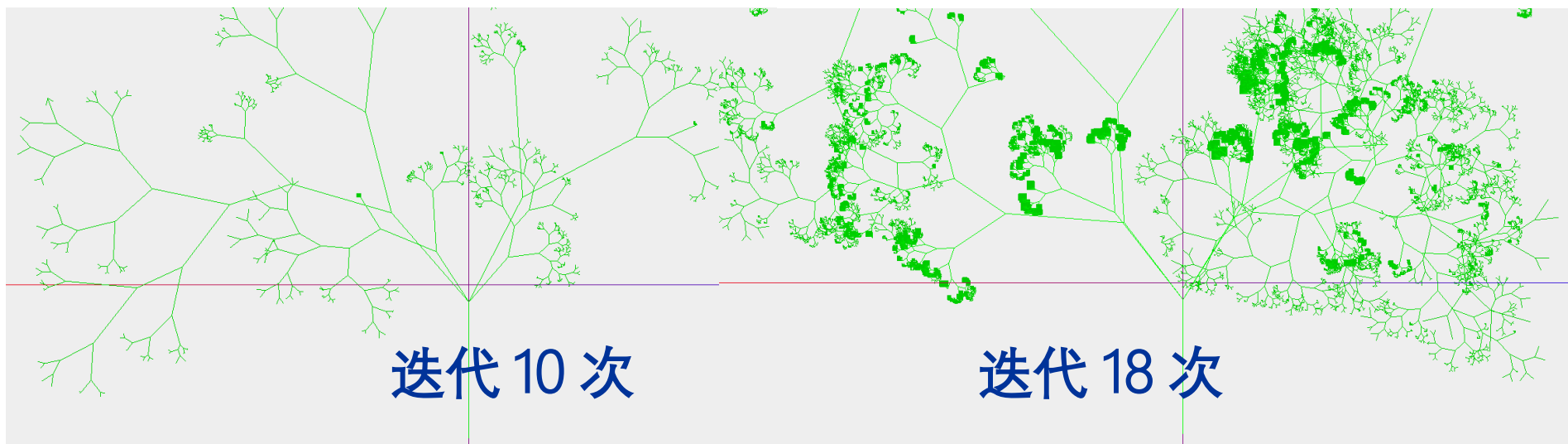
```
y=(int)(1/(q*Math.sqrt(2*Math.PI))*Math.exp(-(x-u)*(x-u)/(2*q*q))*y0*100);
```





## 2. 正态分形二叉树——运行结果

- `double ru=0.6,rp=0.2; // 比例均值, 方差`
- `double au=0.8,ap=0.2; // 角度均值, 方差`
- `rate=ru+rp*rt.nextGaussian(); // 正态分布比例`
- `double angleAddL=au+ap*rt.nextGaussian();`
- `angle=angle-Math.PI/4*angleAddL; // 左边角度`
- **自然、仿真**



## 2. 云模型

### □ 参数

- ⊙ 期望： $Ex$ ----- 均值
- ⊙ 熵： $En$ ----- 方差
- ⊙ 超熵： $He$ ----- 方差的变化

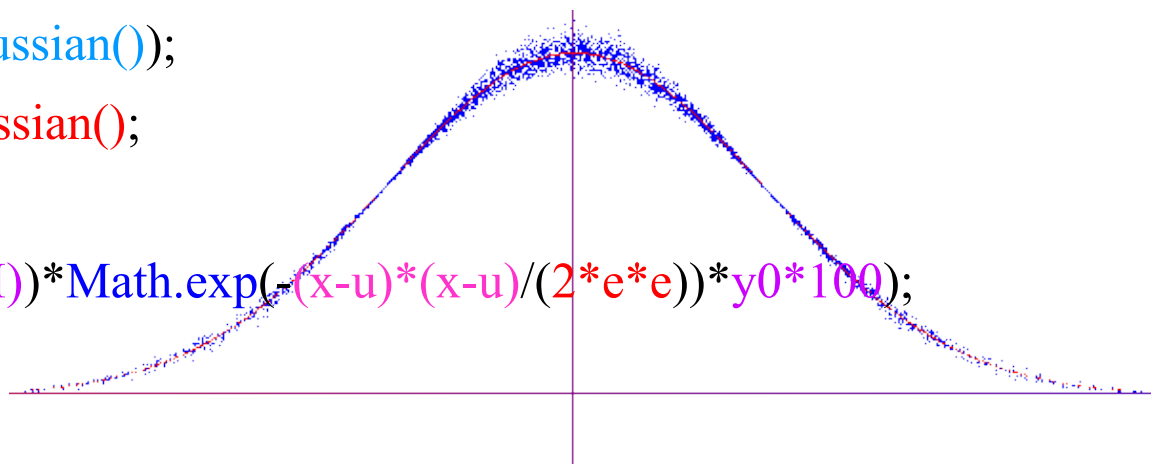
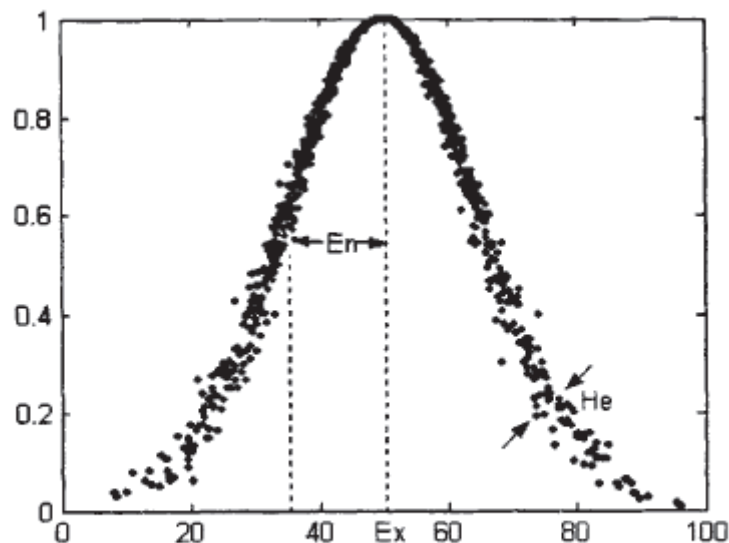
$u$  : 均值     $q$  : 方差     $h$  : 超熵

随机数： $x = (\text{int})(u + q * \text{r1.nextGaussian}());$

变化的方差： $e = q + h * \text{r2.nextGaussian}();$

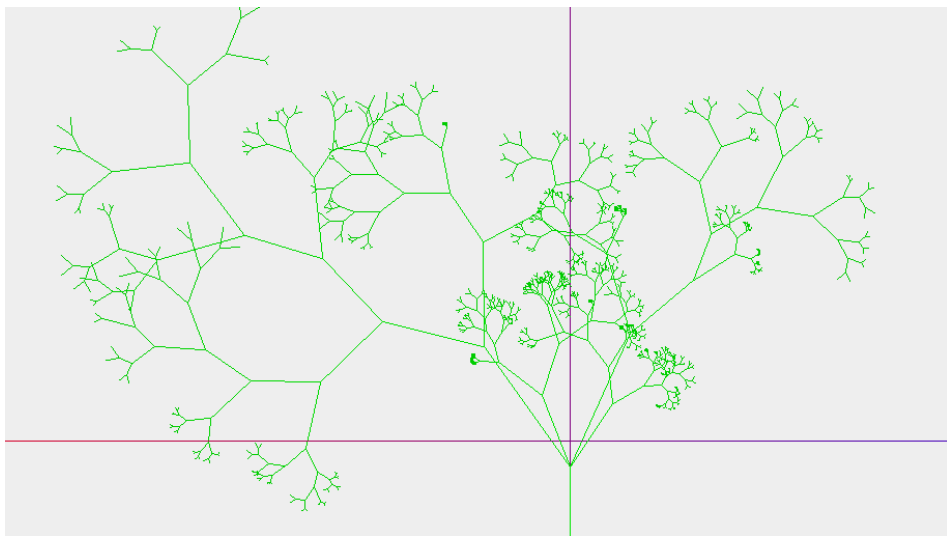
密度：

$yc = (\text{int})(1 / (e * \text{Math.sqrt}(2 * \text{Math.PI})) * \text{Math.exp}(-(x - u) * (x - u) / (2 * e * e)) * y0 * 100);$

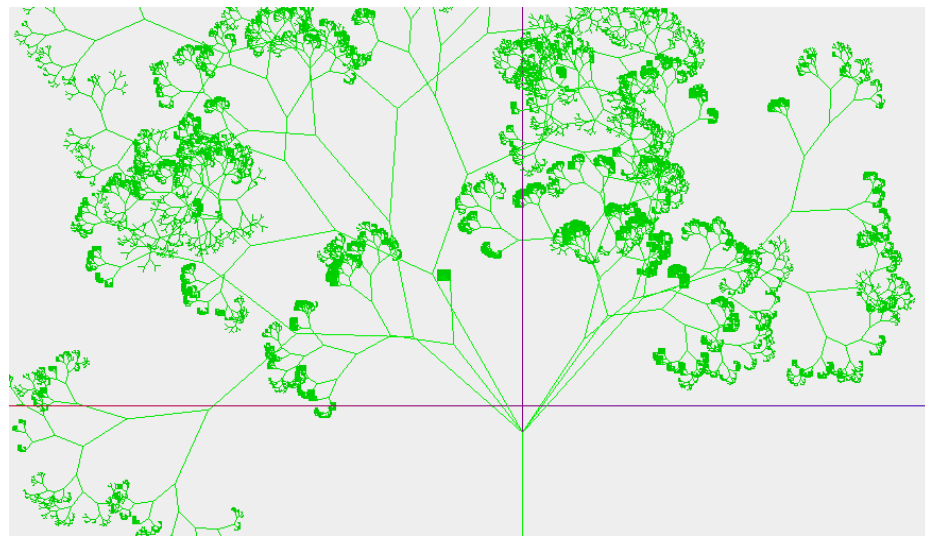


### 3. 基于云模型的分形二叉树

- `double ru=0.6,rp=0.2,rh=0.05; // 比例均值，方差，超熵`
- `double au=0.8,ap=0.2,ah=0.1; // 角度均值，方差，超熵`
- `rate=ru+(rp+rh*rt.nextGaussian())*rt.nextGaussian(); // 云模型比例`
- `double angleAddL=au+(ap+ah*rt.nextGaussian())*rt.nextGaussian();`
- `angle=angle-Math.PI/4*angleAddL; // 左边角度`
- **更加自然、仿真**



迭代 10 次



迭代 15 次

```

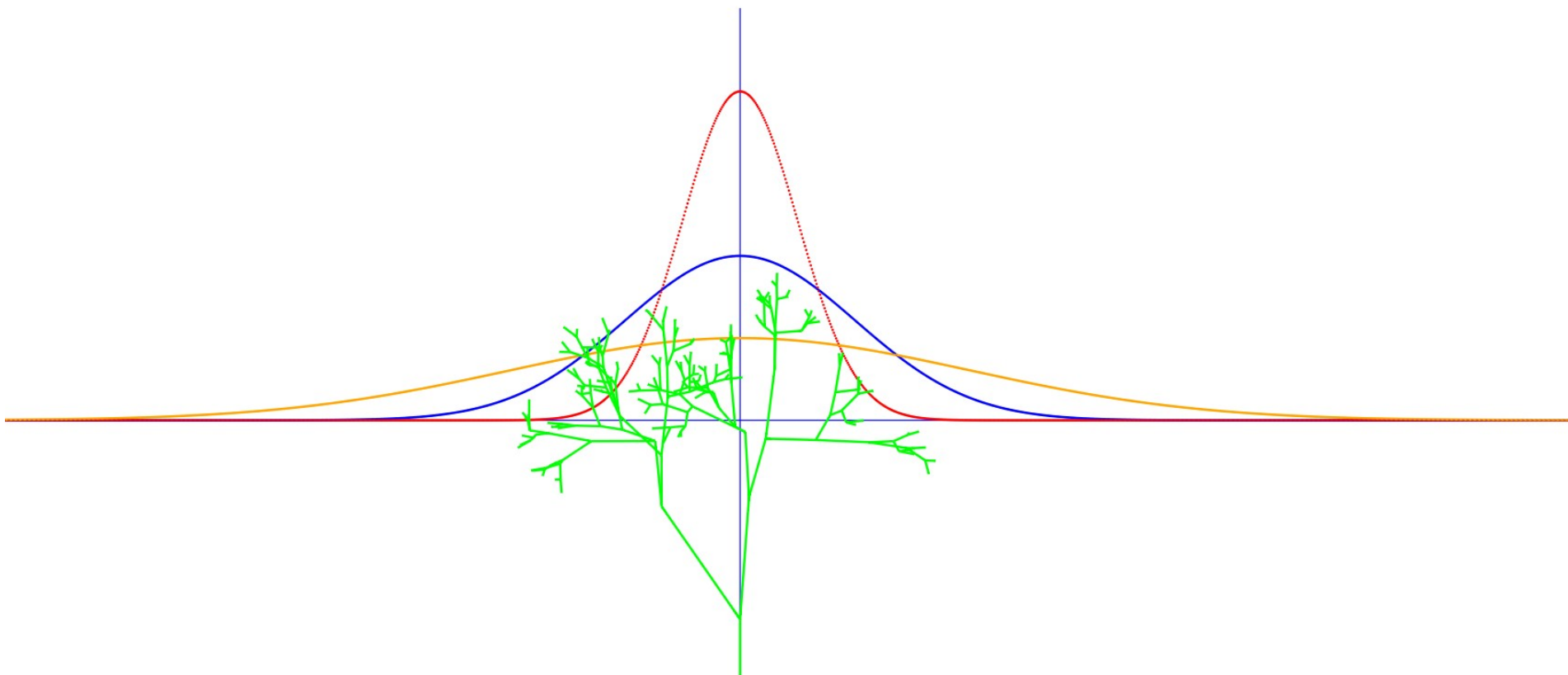
for(var i=0;i<w;i++){
    var point= document.createElement("circle");
    mysvg.appendChild(point);

    var u=w/2; // 均值
    var q=100; // 方差
    var x=i;
    var y=h/2-(1/(q*Math.sqrt(2*Math.PI))*Math.exp(-(x-u)*(x-u)/(2*q*q))*100*h/2);
    point.outerHTML="<circle cx="+x+" cy="+y+" r=1 fill='blue' />";
}

```

127.0.0.1:8080/dviz2019/binTreeRR.htm

拖拽上传





# 小结

## 程序实现

分形二叉树  
正态分布图、云模型  
随机分形二叉树

-4-

作业

-3-

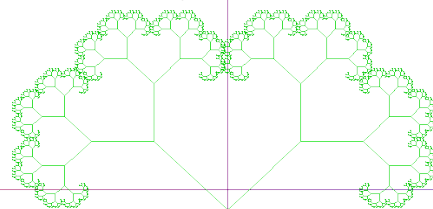
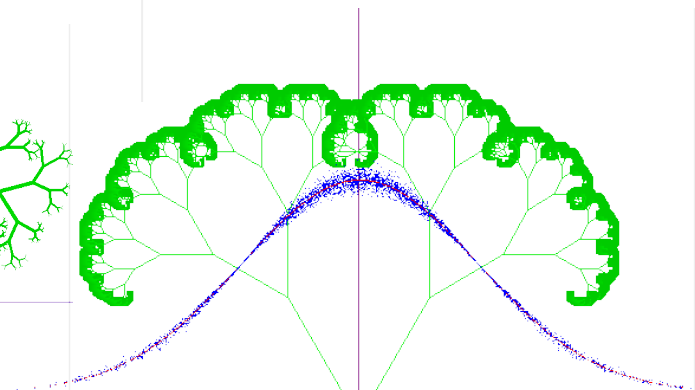
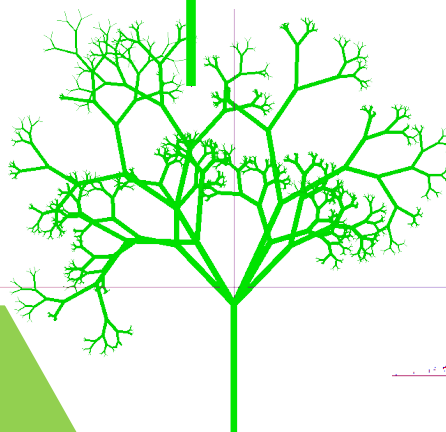
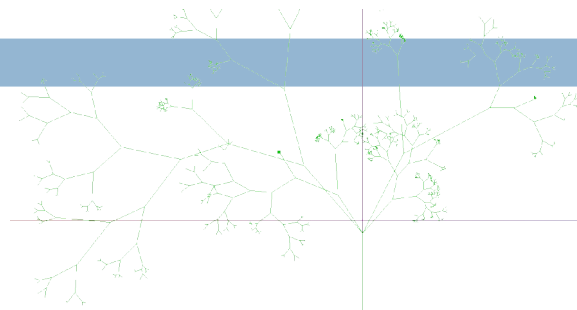
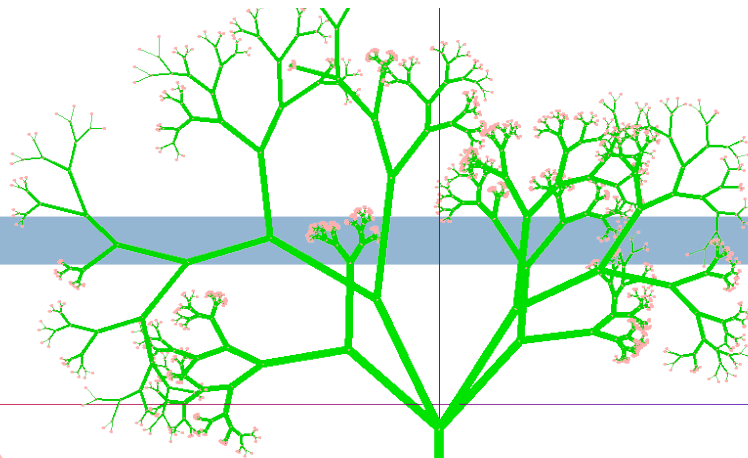
随机性分形二叉树

-2-

正态分布、云模型

-1-

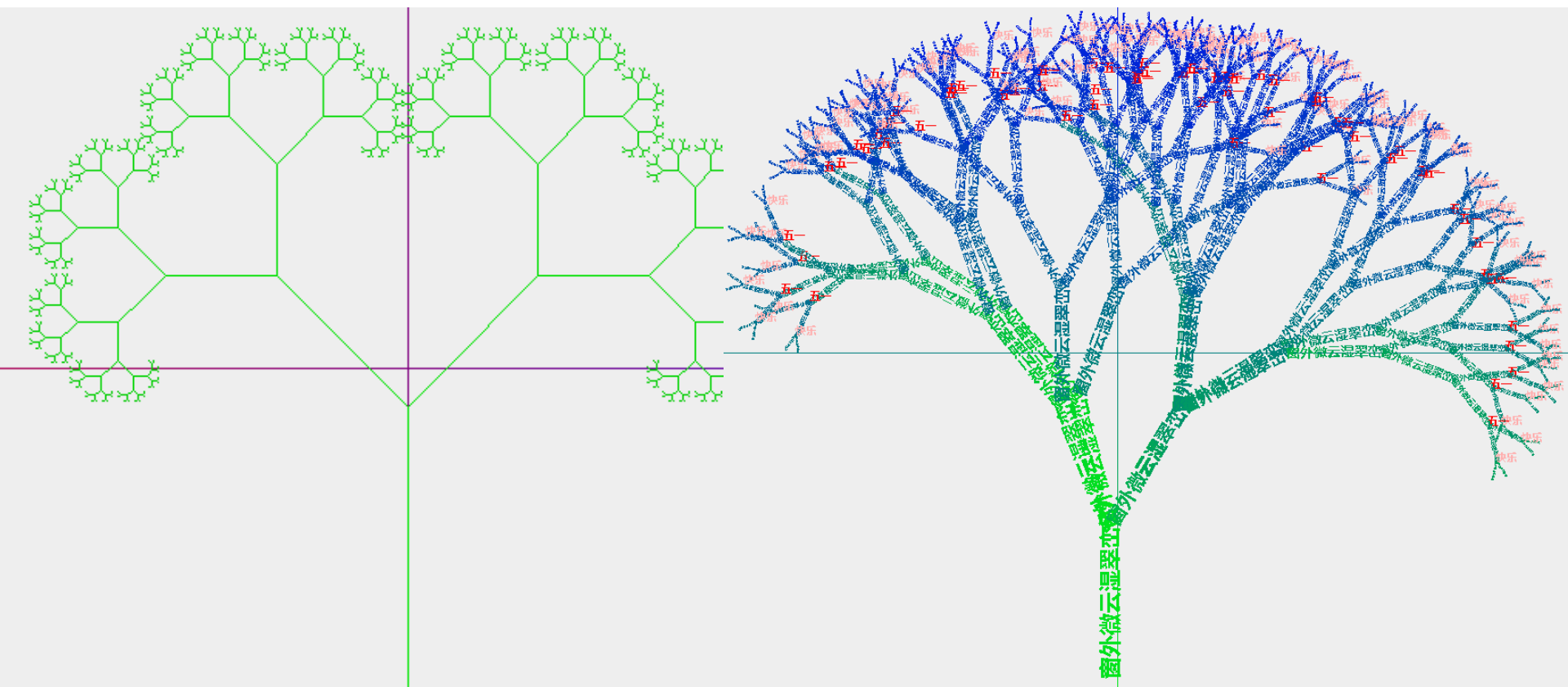
分形学之递归二叉树





# 二叉树变形之 WordTree

# 二叉树变形之 WordTree



# 二叉树变形之 WordTree

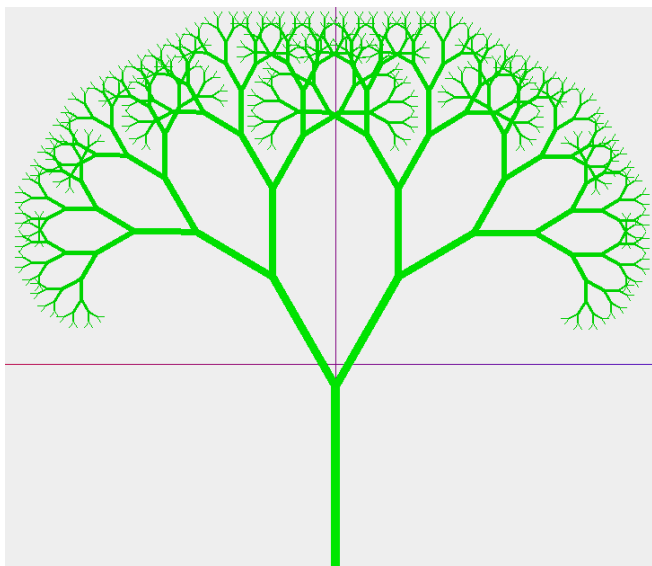
## □ 二叉树

### □ 线宽

```
Stroke stroke = new BasicStroke(count,  
    BasicStroke.CAP_BUTT, BasicStroke.JOIN_BEVEL);  
g2.setStroke(stroke);
```

### □ 画线

```
lineL=new Line2D.Double(x1,y1,x2,y2);  
g2.draw(lineL);
```



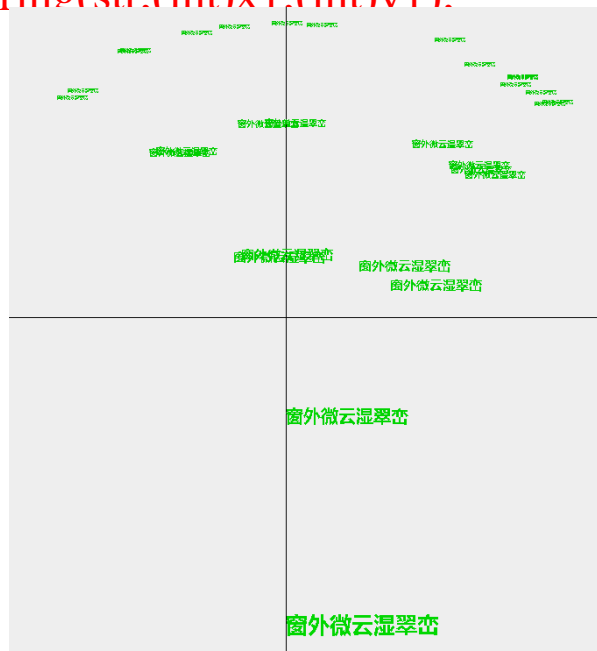
## □ WordTree

### □ 字号

```
Font font1 = new Font(" 微软雅黑 ", Font.BOLD,  
    count*3);  
g2.setFont(font1);
```

### □ 画字符

```
String str=" 窗外微云湿翠峦 ";  
g2.drawString(str,(int)x1,(int)y1);
```

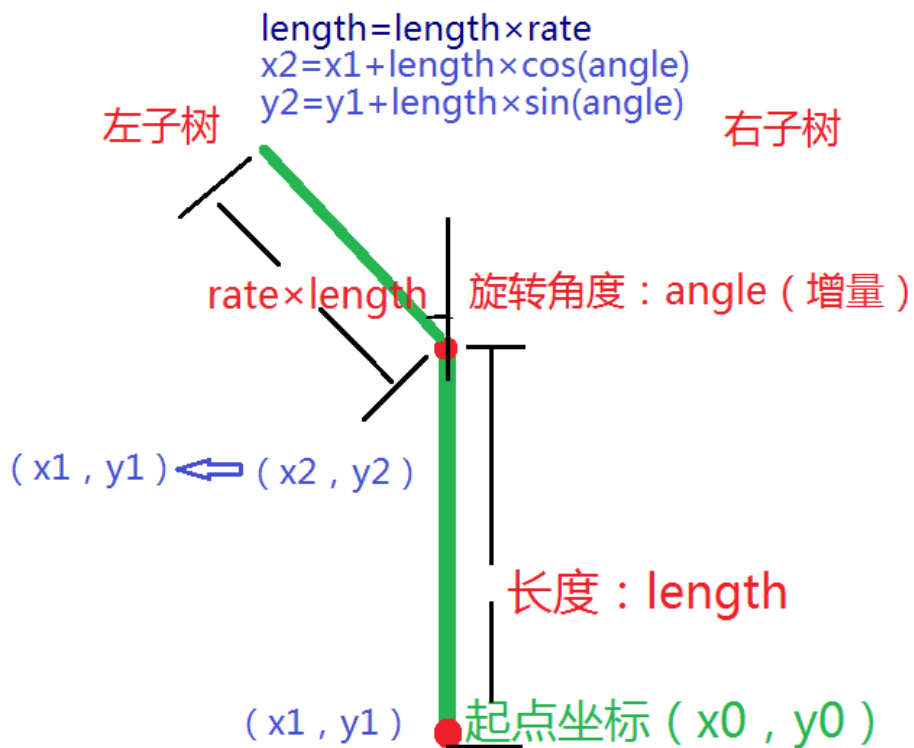




# 二叉树变形之 WordTree

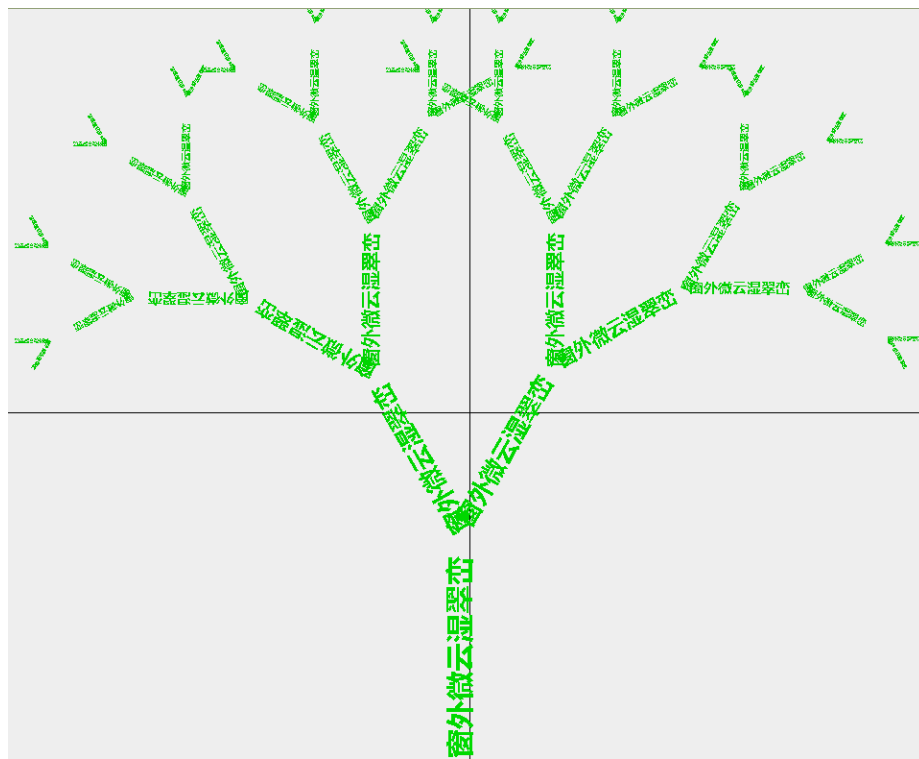
## 画线

- 起点和终点画线段
- 旋转角度决定终点位置



## 画字符

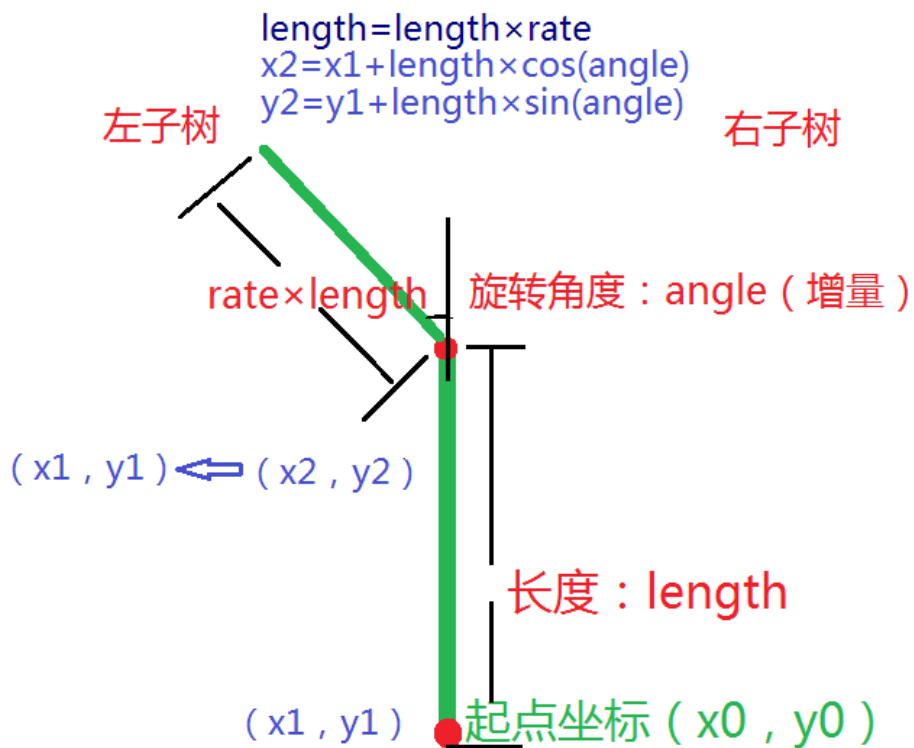
- 以字符基线左下角旋转  $angle$
- 打印字符
- 以字符基线左下角旋转  $-angle$



# 二叉树变形之 WordTree

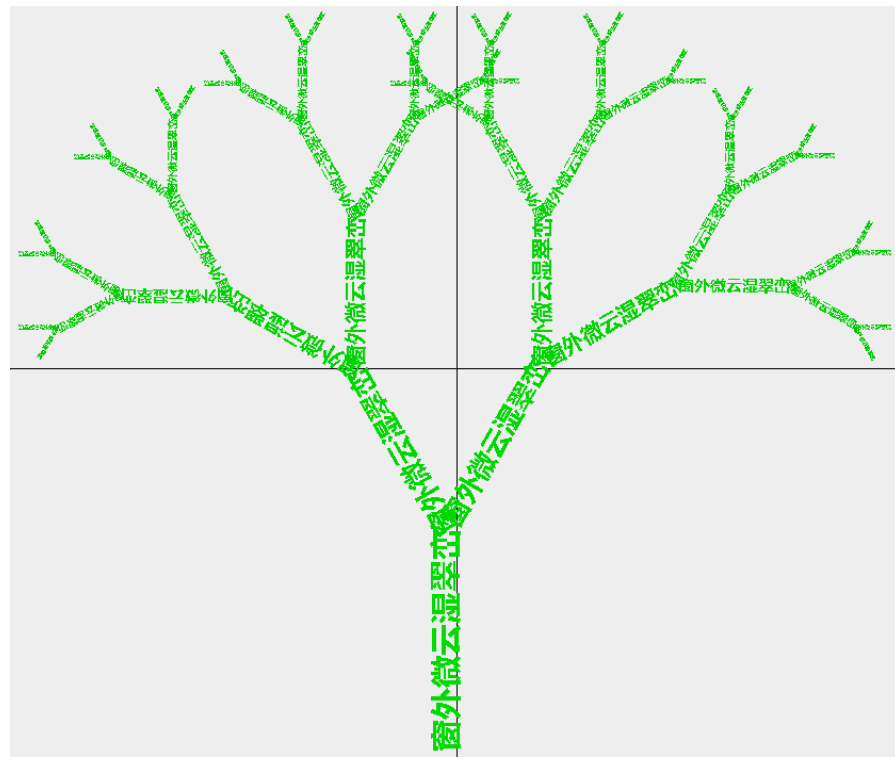
## 画线

### 线段长度衰减



## 画字符

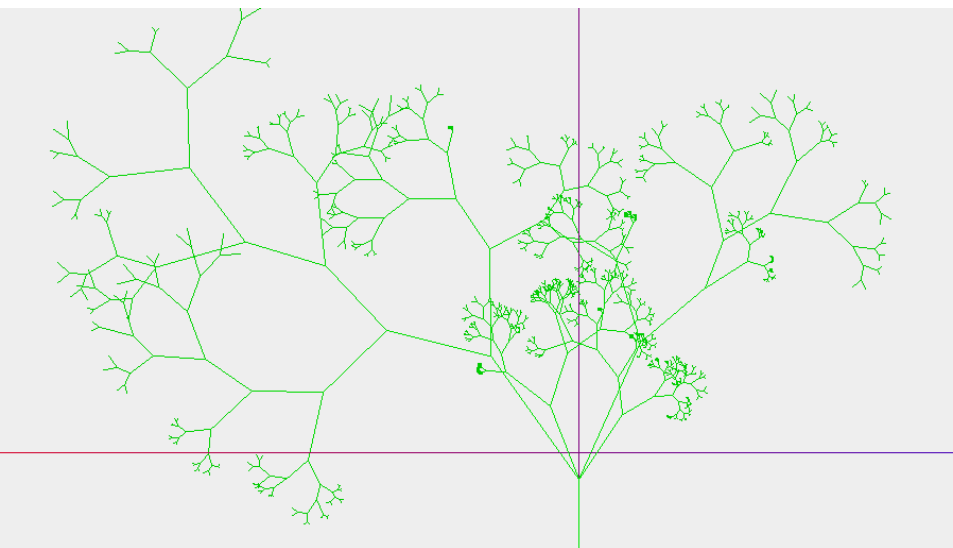
- 长度 = 字符数  $\times$  字号
- String str=" 窗外微云湿翠峦 ";
- length=count\*5\*str.length();



# 二叉树变形之 WordTree

## □ 画线

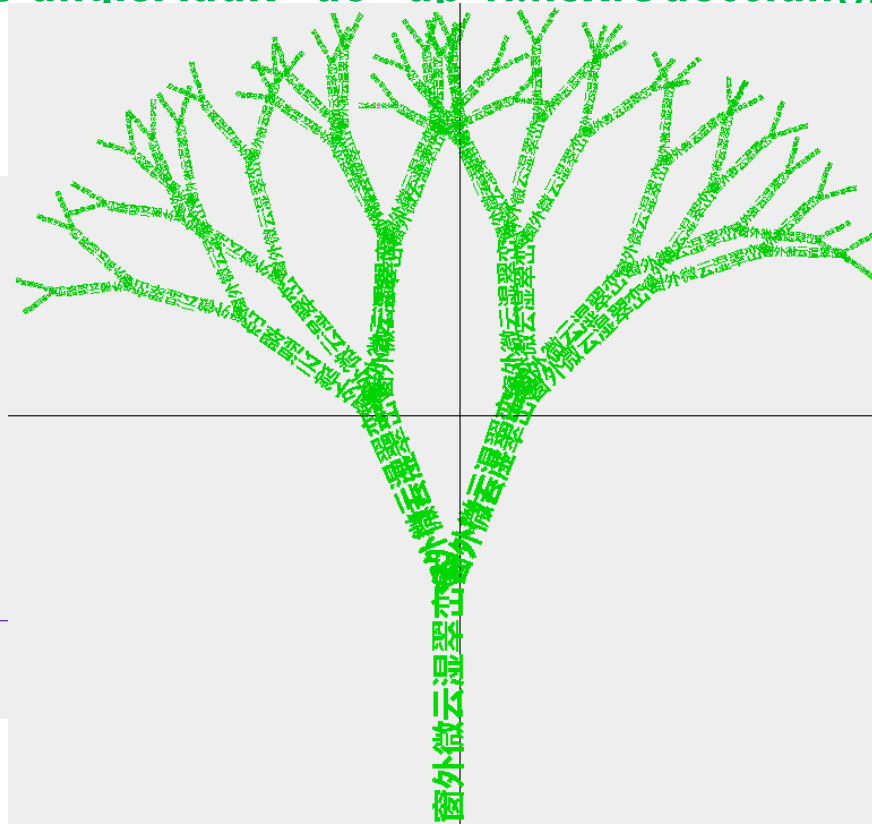
- ⊙ 正态分布调整角度
- ⊙ 正态分布调整线段长度



## □ 画字符

用正态分布调整角度

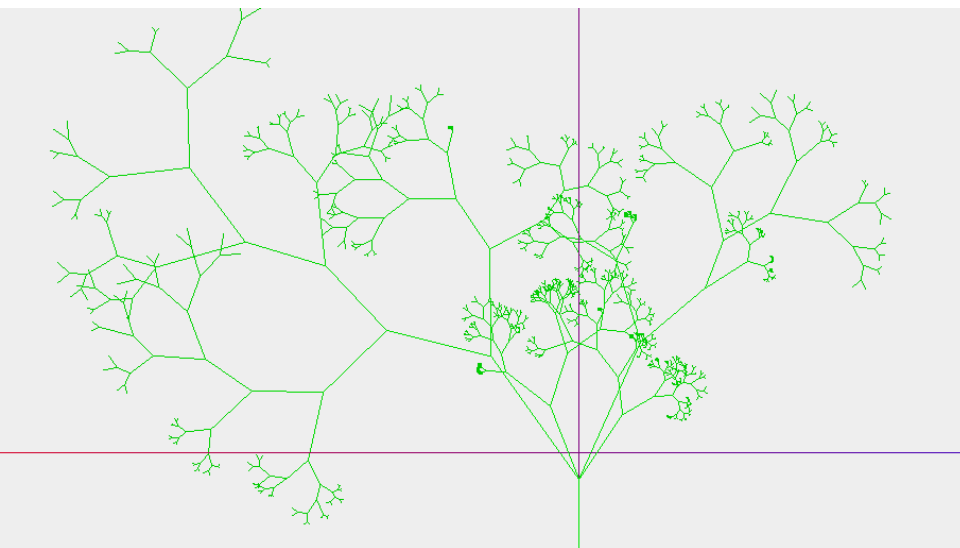
```
java.util.Random rt = new java.util.Random();  
double au=0.8,ap=0.2;  
double angleAddL=au+ap*rt.nextGaussian();  
double angleAddR=au+ap*rt.nextGaussian();
```



# 二叉树变形之 WordTree

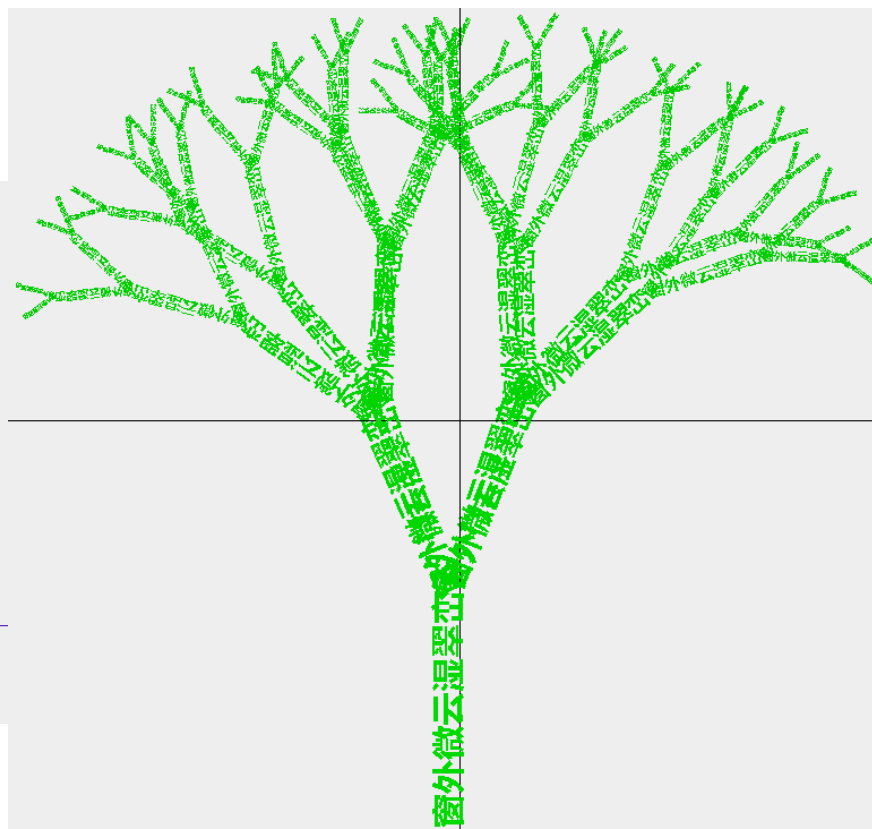
## □ 画线

- ◎ 渐变色
- ◎ 果实



## □ 画字符

- ◎ 渐变色
- ◎ 果实

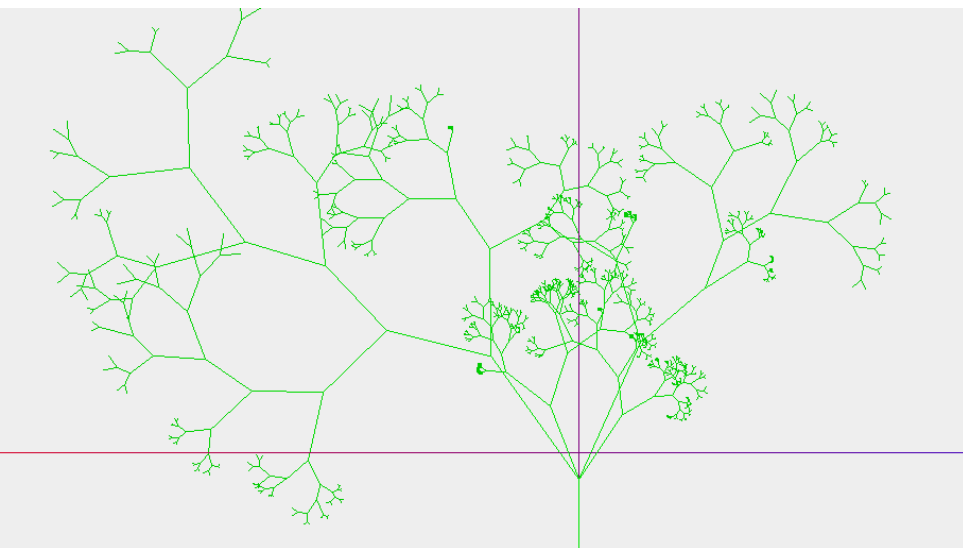




# 二叉树变形之 WordTree

## □ 画线

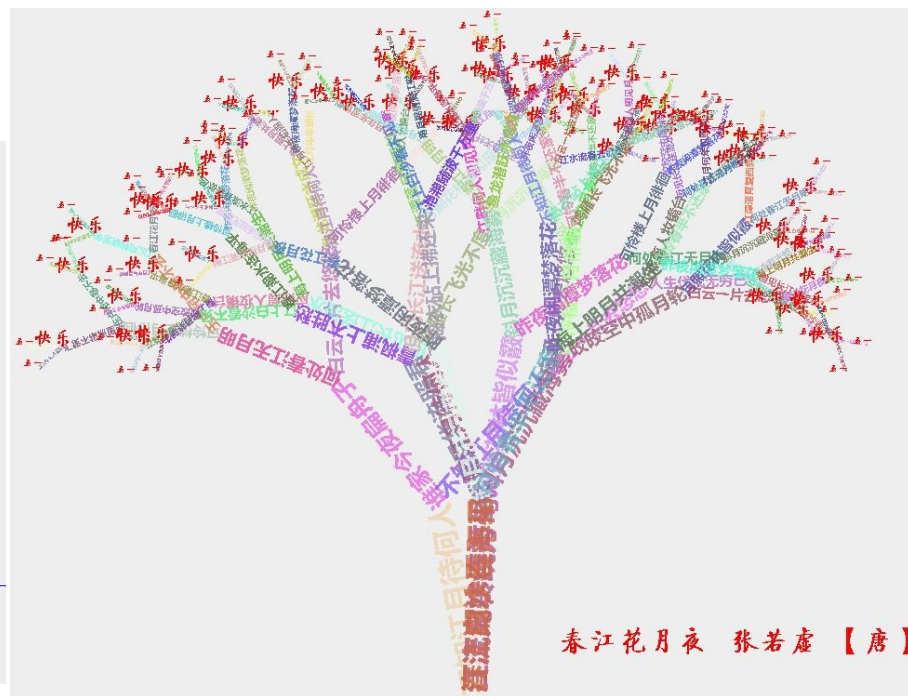
### ⊙ 随机色



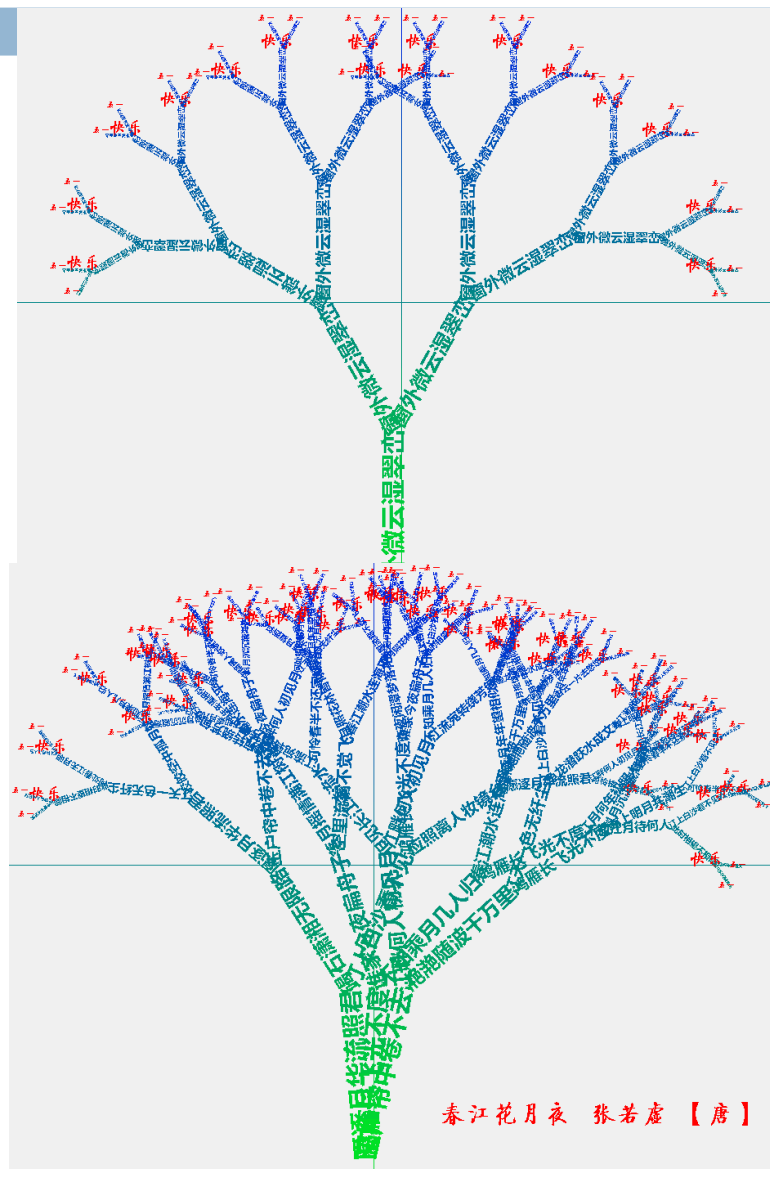
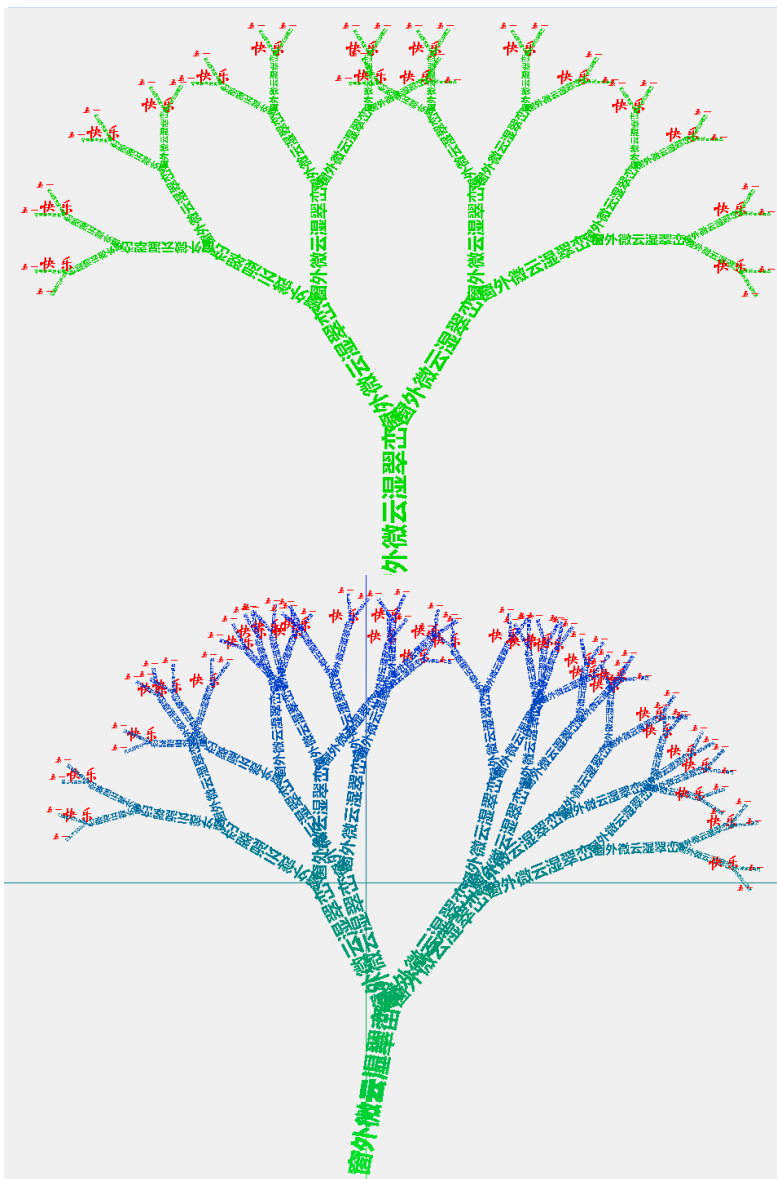
## □ 画字符

### ⊙ 随机色

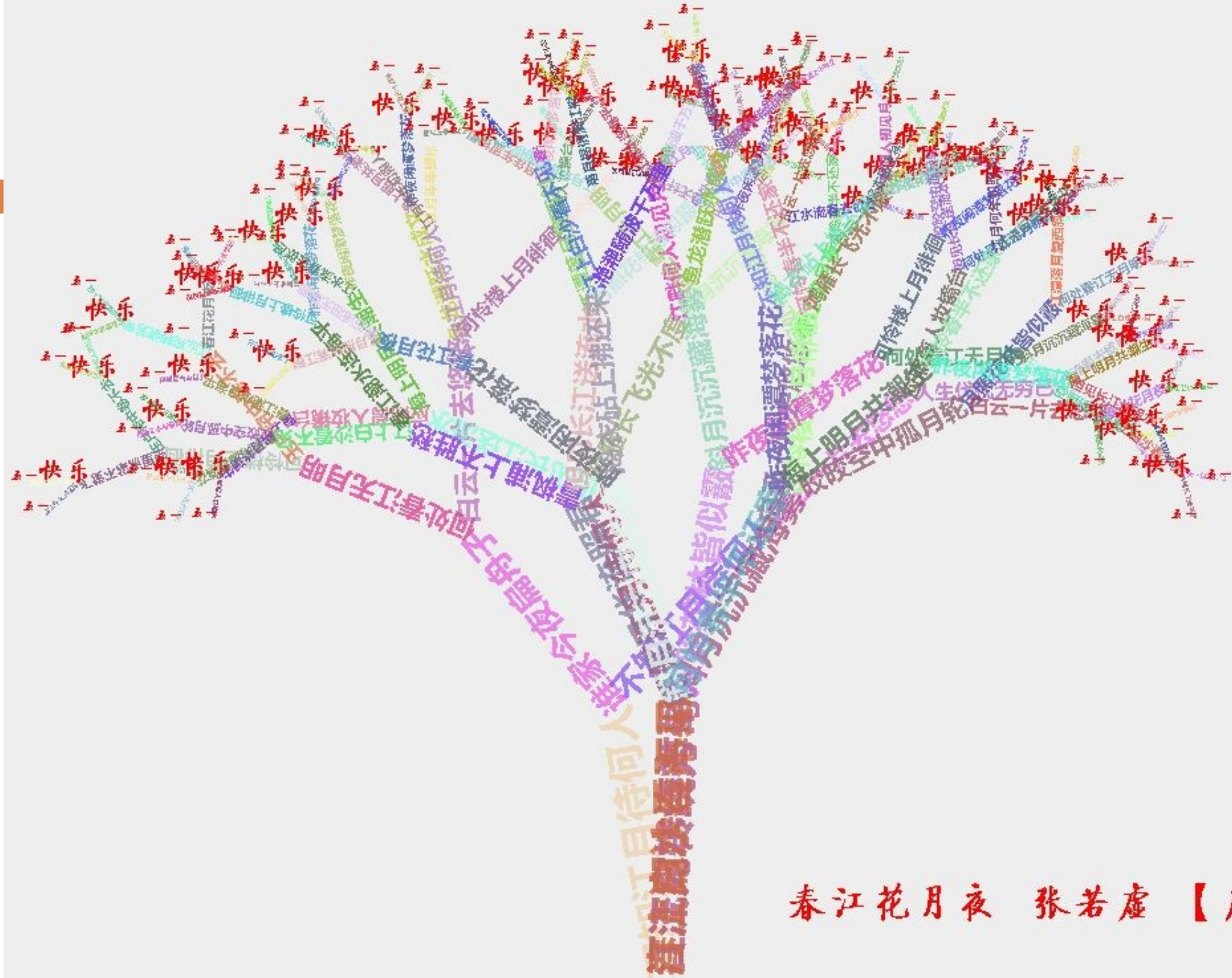
### ⊙ 字符



# 二叉树变形之 WordTree



春江花月夜 张若虚【唐】



春江花月夜 张若虚 【唐】

# 数据 + 可视化 = 数据可视化

- 只教了可视化技术，数据是学生兴趣去找
- 两人一组，最多 1 组有 3 人一组，3 人一组会  
有人偷懒滴
- **数据要求有三个原则：**
  - 1- 最新。近期数据。或者实时数据。**
  - 2- 完整。整年的数据。连续 N 年的数据。**
  - 3- 权威。国家权威官方数据，国家统计局数据、重要学术机构数据，部委数据、国内外知名咨询公司数据，重要文献数据等。**