第九章 文字与图形GUI

主要内容

1. GUI设计概述
2. font类
3. Color类
4. Graphic类

1 GUI设计概述

- 图形用户界面(Graphics User Interface,简称GUI)就是为应用程序提供一个图形化的界面。
- GUI使用图形的方式，借助菜单、按钮等标准界面元素和鼠标操作，向计算机系统发出命令操作，并将系统运行的结果以图形的方式显示给用户，使一个应用程序具有画面生动、操作简便的效果。

- 为了方便编程人员开发图形用户界面，Java提供了两个工具包：

(1) 抽象窗口工具包(Abstract Windowing ToolKit，缩写为AWT)

(2) Swing图形用户界面工具包。

- 工具包提供丰富的类库用于创建与平台无关的用户界面。编程人员可方便地使用这些类库来生成各种标准图形界面元素并处理图形界面的各种事件。

1.1 图形用户界面元素分类

(1) 容器

用来组织或容纳其他界面成分和元素的组件。一个容器可以包含许多组件，同时它本身也可以作为一个组件，放进另一容器中。

容器是Java中的类，例如框架(JFrame)、面版(JPanel)及滚动面板(JScrollPanel)等。

(2) 控制组件

与容器不同，控制组件是图形用户界面的最小单位之一，它里面不再包含其他的成分。

作用：完成与用户的一次交互，包括接收用户的一个命令(如按钮命令)，接收用户输入的一个文本或选择，向用户显示一段文本或一个图形等等。

使用控制组件，需要如下步骤：

i. 创建某控制组件类的对象，指定其大小等属性。

ii. 使用某种布局策略，将控制组件对象加入到容器指定位置。

iii. 注册对象，重载事件处理方法，实现交互功能。

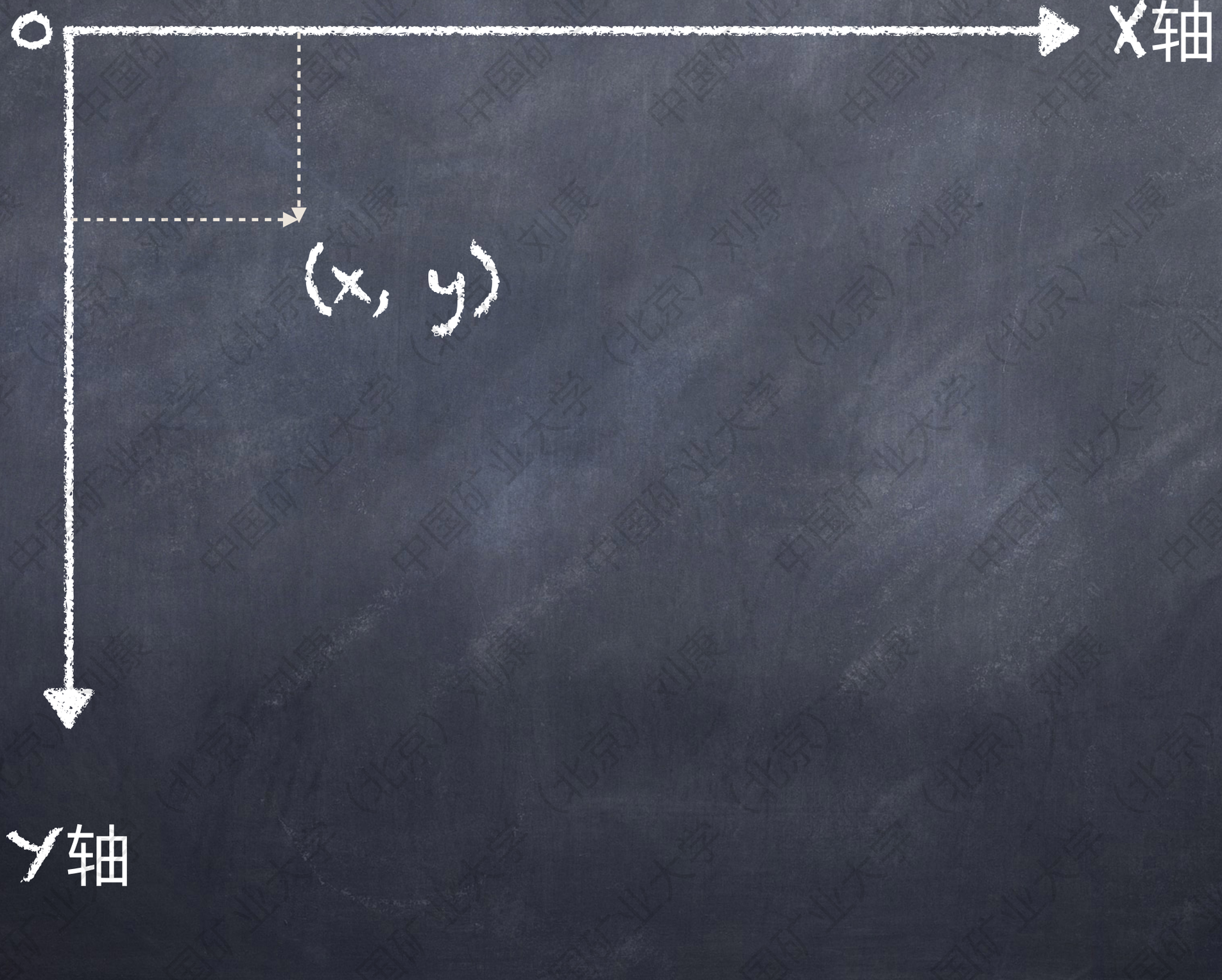
(3) 用户自定义成分

根据用户的需要，使用各种字型、字体和色彩，设计几何图形、标志图案等，被称为用户自定义成分。

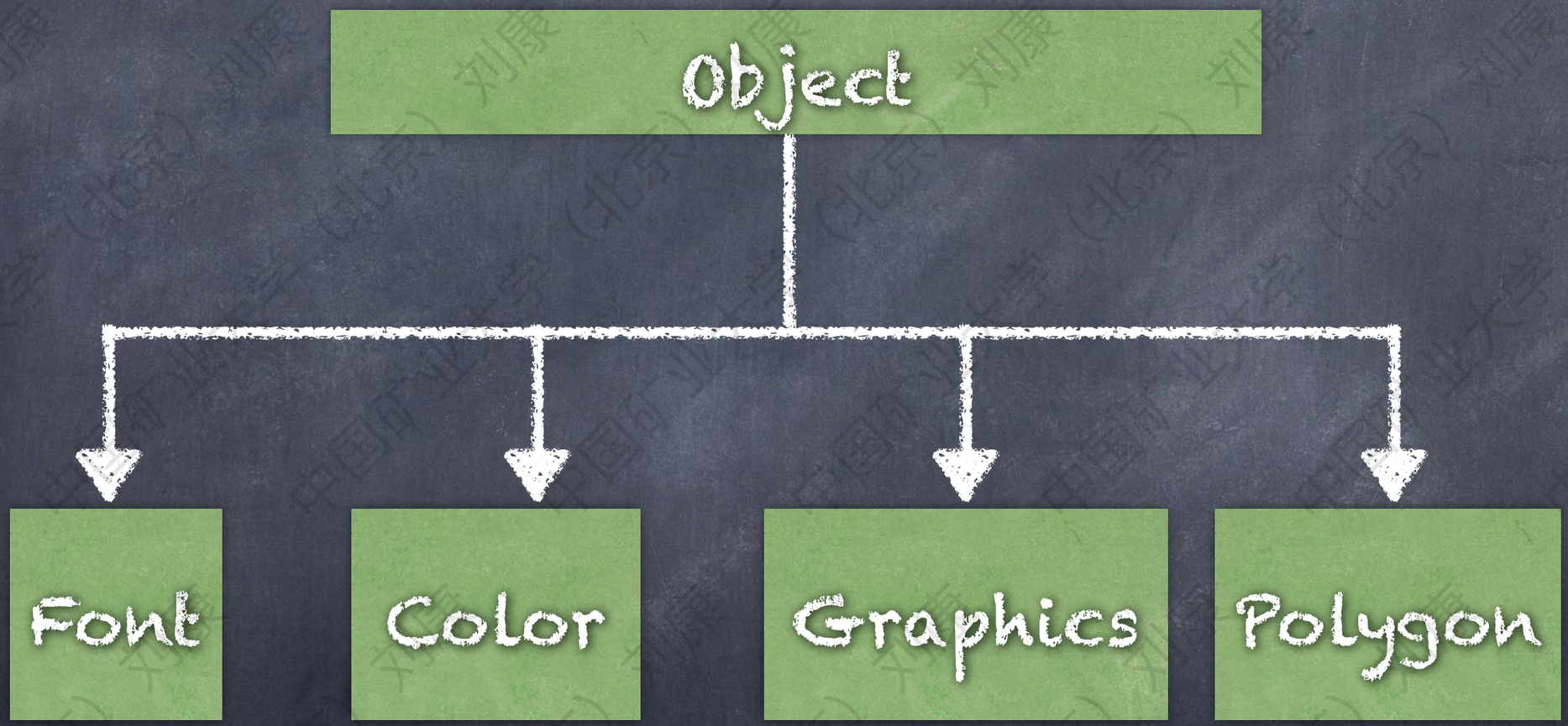
用户自定义成分通常只能起到装饰、美化的作用，而不能响应用户的动作，也不具有交互功能。

1.2 屏幕坐标系

- Java坐标系是一个二维网络，可以标识屏幕上每个点的坐标位置。
- 坐标单位使用像素度量。一个像素是显示的最小分辨单位。
- 在缺省的状态下，原点为屏幕左上角坐标(0，0)


O
X轴
(x, y)
Y轴

1.3 与文字、图形有关类



字体　　颜色　　几何图形　　多边形

1.4 Applet执行程序

- 每个Applet应用程序都是Applet类的子类
- 在实际运行中，浏览器在下载字节码的同时，会自动创建一个用户Applet子类的对象，并在适当事件发生时自动调用该对象的主要方法。

I. init()方法

当Applet程序启动时，自动调用init()方法。该方法仅用作初始化操作。用户可以重载父类的init方法，通过init方法初始化图像文件、声音文件、字体等等。

II. start()方法

运行init()后，自动调用start()方法。该方法体现小程序要完成的功能。可覆盖父类的start()方法。

III. paint()方法

主要用于Applet界面中显示文字、图形和其他界面元素。

浏览器调用paint()主要有三种方式：

1. 首次显示Applet时，自动调用paint()
2. 用户调整窗口大小或移动窗口时，调用paint()
3. 当repaint()方法被调用时，系统首先调用update()，将Applet对象所占用对象清空，再调用paint()方法重画。

IV. stop()方法

当用户浏览Applet程序所在Web页面切换到到其他页面时，浏览器自动调用stop()。终止运行Applet。用户又回到Applet所在页面，重启Applet的start()。

V. destroy()方法

用户关闭Applet程序窗口时，浏览器自动执行此方法，结束程序，释放所占资源。

1.5 Swing概述

1.5.1 JApplet

所有的Swing GUI组件都包含在JApplet小程序中，且javax.Swing.JApplet是Applet的子类，所以本章主要介绍javax.Swing的“J组件”，应用程序均以JApplet小程序为例。

JApplet使用与Applet小程序相似，它们与HTML文件的配合没有什么差别，差别仅在默认布局策略与个别方法的使用上。

1.5 Swing概述

1.5.2 Swing组件

重量级组件，也称顶级容器。包含：JFrame、jDialog、JWindow、JApplet。任何Swing应用程序中都必须至少有一个重量级组件。

轻量级组件，继承自JComponent抽象类的组件，实现人机交互的基本组件，必须放到重量级组件中才能显示。

1.5 Swing概述

1.5.3 Swing容器结构

1. 透明面板(glassPane)

位于JRootPane中所有组件之上，完全透明的。主要功能用于捕获鼠标事件和为所有组件上绘图提供方便，默认不可见。

2. 分层面板(JLayeredPane)

是一个容器，提供添加组件的能力。如同多层置物架，每一层可放置物品。但上面的物品会遮住下面的物品。

3. 内容面板(contentPane)

分层面板内的一层，顶级容器的内容面板，可将组件加入其中。

加入Swing组件时，可利用getContentPane()得到contentPane容器，利用add()添加组件到容器中。

4. 菜单条(JMenuBar)

属于可选项，与分层面板在同一层。

1.5 Swing概述

1.5.4 Swing轻量级组件的分类

(1)中间容器

常用于Swing应用程序中，JPanel、JScrollPane、JSplitPane、JTabbedPane。

(2)专用容器

JInternalFrame(内窗体)、JLayeredPane、JRootPane。

1.5 Swing概述

1.5.4 Swing轻量级组件的分类

(3)基本组件

JButton(按钮)

JcheckBox(复选框)、JRadioButton(多选按钮)

JcomboBox(下拉式列表)、JList(列表)

JTextField(文本框)、JTextArea(文本域)

JMenu(菜单)

1.5 Swing概述

1.5.5 添加组件到顶级容器

Swing组件放入Swing顶层容器的内容面板上。两种方案：

(1) 利用getContentPane()，获得当前容器的内容面板对象，再引用add()添加组件。

(2) 建立JPanel之类的中间容器，添加组件到中间容器，再使用顶级容器的setContentPane()将该中间容器置为顶级容器的内容面板。

2 绘制文字

2.1 绘制文字的成员方法

由于字符串可以用字符串对象、字符数组、字节数组三种不同形式表示，Java在Graphics类中提供三个成员方法绘制不同形式的字符串。

成员方法	参数说明	功能
drawString {String string, int x, int y}	string字符串对象；x，y起始坐标	以坐标x，y为起始位置，用当前的字体和颜色绘制String代表的字符串
drawChars {char[] ch, int offset, int number, int x, int y}	ch字符组数名；offset要绘制的第一个字符在数组中的下标；number绘制的个数；x，y起始坐标	从ch数组下标为offset的位置开始截取number个字符，从坐标x，y处开始用当前的字体和颜色绘制number个字符
drawBytes {byte[] by, int offset, int number, int x, int y}	by字节数组数名；offset要绘制的第一个字符在数组中的下标；number绘制的个数；x，y起始坐标	从by数组下标为offset的位置开始截取number个字符，从坐标x，y处开始用当前的字体和颜色绘制number个字符

2.2 Font类

1. Font类的构造方法

Font类用来描述字体的名称(Name)、大小(Size)和样式(Style)。使用Font类的构造方法创建一个Font类的对象，其格式如下：

```
Font(String fontname, int style, int size)
```

其中：fontname为字型名

style为字体样式

size为用像素点表示的字体大小

2. Font类成员方法

成员方法	功能说明
public static Font decode (String str)	使用参数指定的字体
public string getFamily ()	获得指定平台的字体名
public string getName ()	获得字体的名称
public int getStyle ()	获得字体的样式
public int getSize ()	获得字体的大小
public string toString ()	将此对象转换为一个字符串表示

3、设置字体

可以用java.awt.Graphics类的成员方法来设置自己希望使用的字体，其格式如下：

```
setFont(Font myFont)
```

3 Color类

1. Color类的构造方法

构造方法	功能说明
public Color (int r, int g, int b)	使用0~255范围内整数指定红、绿、蓝三种颜色比例创建Color对象
public Color (float r, float g, float b)	使用0.0~1.0范围内浮点数指定的红、绿、蓝比例创建Color对象
public Color (int rgb)	使用指定的rgb组合创建Color对象

2. Color类数据成员常量

用户可使用构造方法指定RGB值外，Java还提供常用的颜色常量。

颜色数据成员常量	颜色	RGB值
public final static Color red	红	255, 0, 0
public final static Color green	绿	0, 255, 0
public final static Color blue	蓝	0, 0, 255
public final static Color black	黑	0, 0, 0
public final static Color white	白	255,255,255
public final static Color yellow	黄	255, 255, 0
public final static Color orange	橙	255, 200, 0
public final static Color cyan	青蓝	0, 255, 255
public final static Color magenta	洋红	255, 0, 0
public final static Color pink	淡红色	255,175,175
public final static Color gray	灰	128,128,128
public final static Color lightGray	浅灰	192,192,192
public final static Color darkGray	深灰	64, 64, 64

3. Color类的成员方法

成员方法	功能说明
public int getRed()	获得对象的红色值
public int getGreen()	获得对象的绿色值
public int getBlue()	获得对象的蓝色值
public int getRGB()	获得对象的RGB值
public Color brighter()	获取此颜色的一种更亮版本
public Color darker()	获取此颜色的一种更暗版本

- 此外，还可以用java.awt.Graphics类的方法设定颜色或获取颜色。这些方法及其功能如下：

```
setColor(Color c);
```

设定前景颜色，c代表颜色。

```
setColor(new Color(int r,int g,int b));
```

```
setPaint(new Color(int r,int g,int b));
```

设定前景颜色的另一个方法。

```
setBackground(Color c);
```

设定背景颜色，c代表颜色。

```
getColor();
```

获取当前使用颜色。

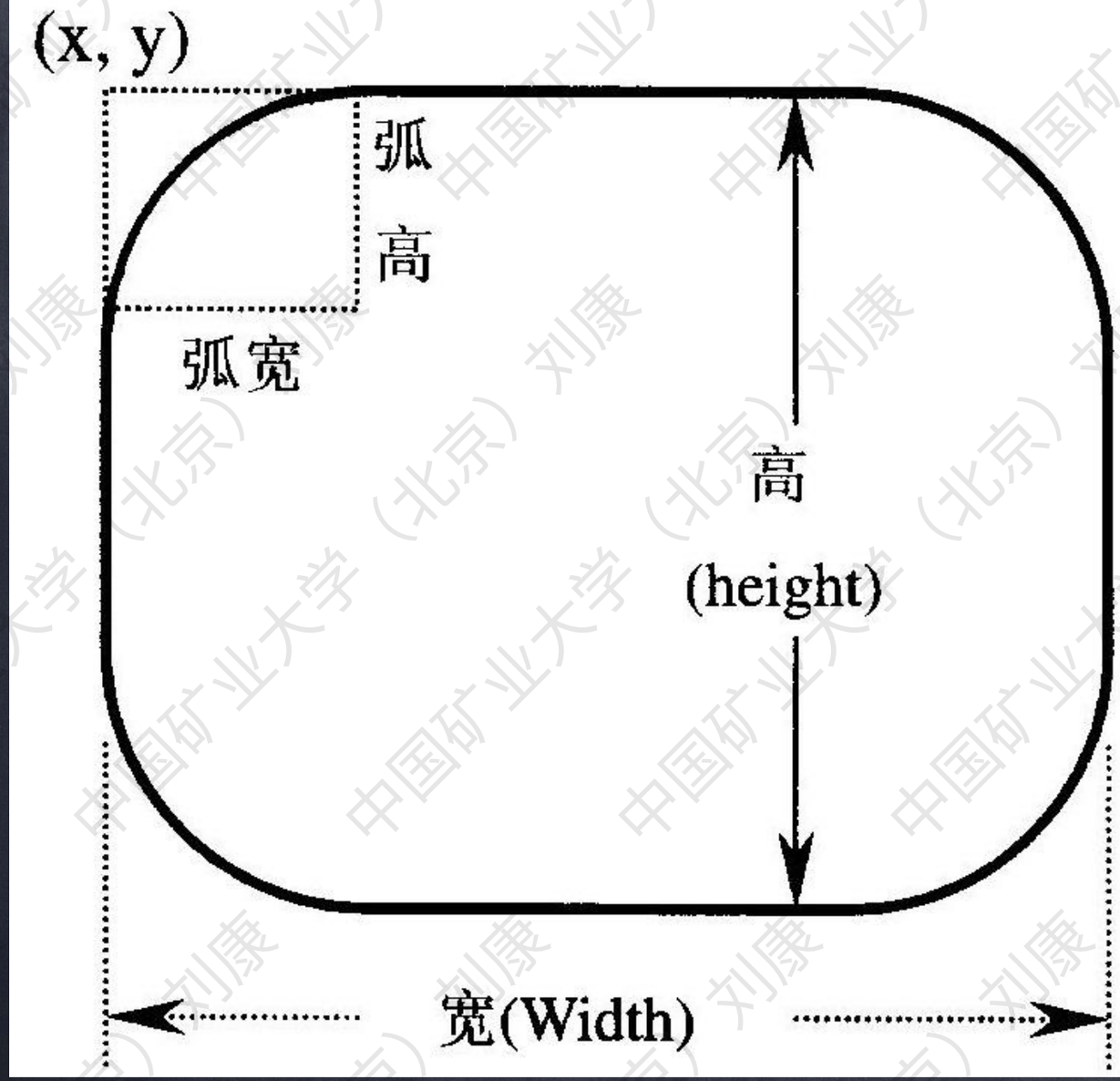
4 Graphics类

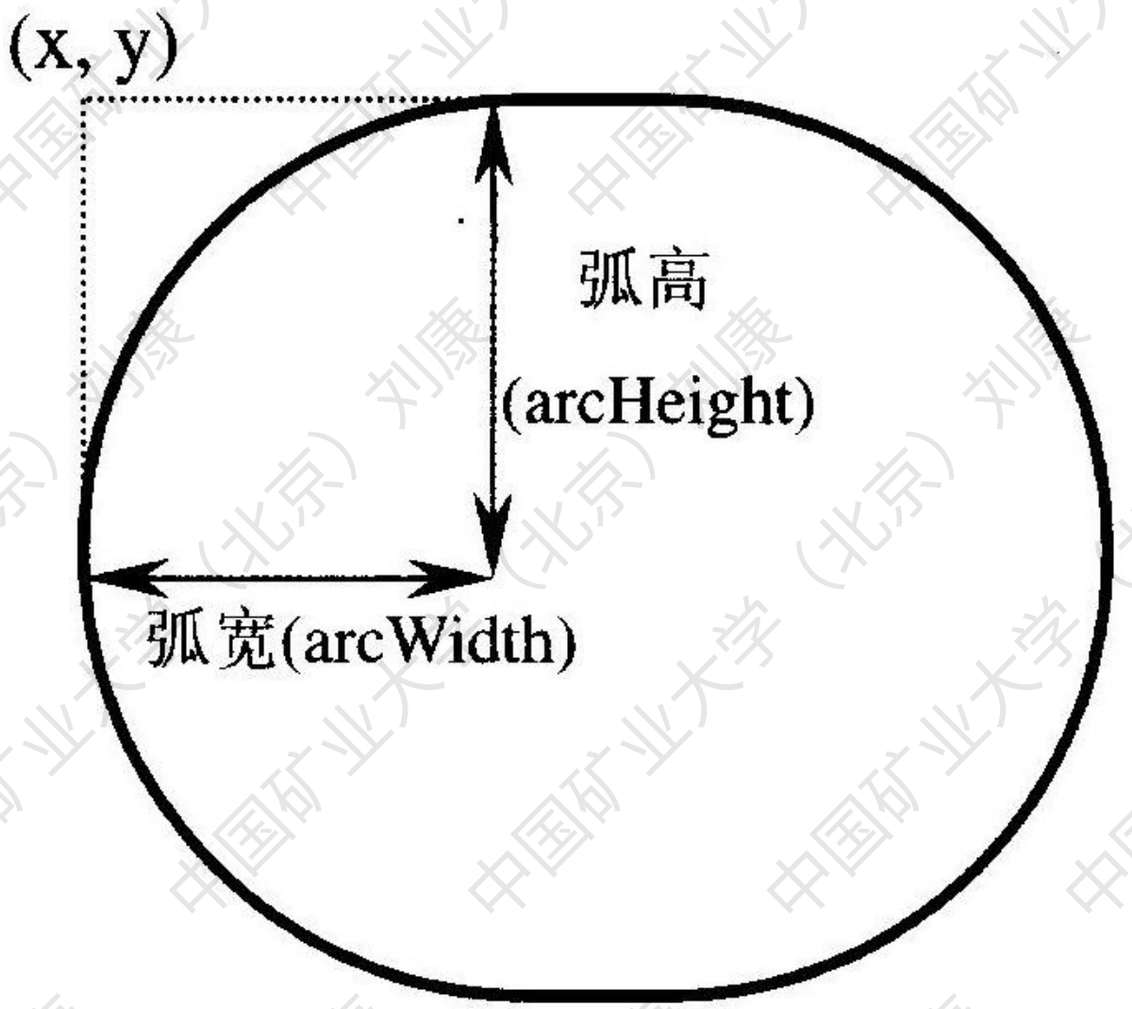
1. 绘制线段与矩形

Graphics类中提供了绘制直线、平面矩形、圆角矩形、三维立体矩形等图形的成员方法。

成员方法	功能	参数说明
drawLine (int x1, int y1, int x2, int y2)	绘制一条直线	x1,y1为起点坐标，x2,y2为终点坐标
drawRect (int x, int y, int width, int height)	绘制平面矩形	x,y为矩形左上角顶点坐标； width为矩形的宽度； height为矩形的高度
fillRect (int x, int y, int width, int height)	绘制当前颜色填充的矩形	
clearRect (int x, int y, int width, int height)	清除矩形区域	

drawRoundRect (int x, int y, int widht, int height, int arcWidth, int arcHeight)	绘制圆角矩形	x, y, width, height四个参数意义同上； arcWidth为圆角的弧宽； arcHeight为圆角的弧高
fillRoundRect (int x, int y, int width, int height, int arcWidth, int arcHeight)	绘制用当前颜色填充的圆角矩形	
Draw3Drect (int x, int y, int width, int height, boolean b)	绘制三维矩形	x, y, width, height四个参数意义同上； b用于确定矩形的凸凹： 当b为true时矩形凸出， 当b为false时矩形凹下。


(x, y)
弧高
弧宽
高
(height)
宽(Width)


(x, y)
弧高
(arcHeight)
弧宽(arcWidth)

2. 绘制椭圆

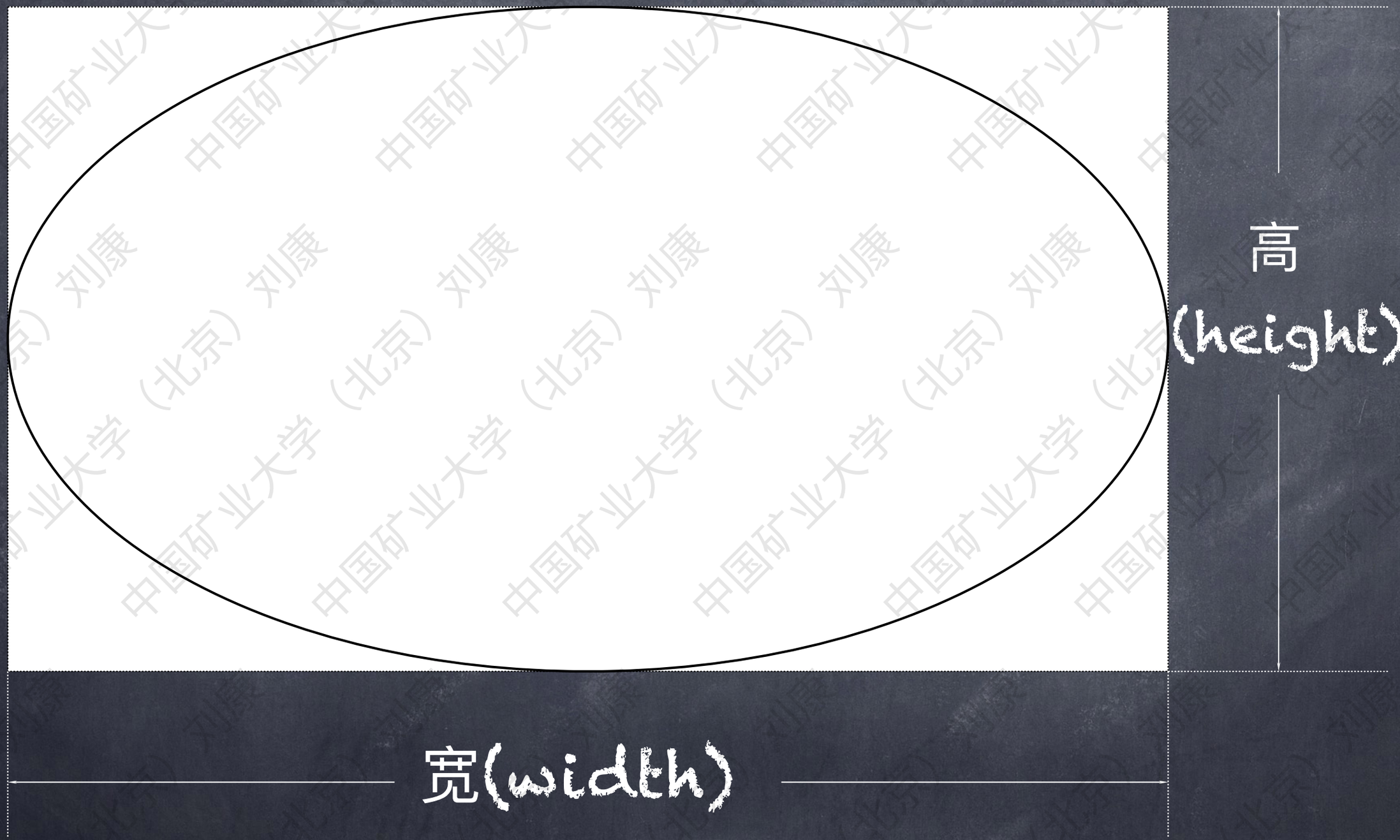
在Java中，绘制一个椭圆需要指定4个参数，分别是外切矩形左上角顶点坐标x，y和椭圆的宽度与高度。Graphics类中提供了两种成员方法：

用于绘制椭圆

```
drawOval(int x, int y, int width, int height);
```

用于绘制一个用当前颜色填充的椭圆

```
fillOval(int x, int y, int width, int height);
```


高
(height)
宽(width)

3. 绘制弧

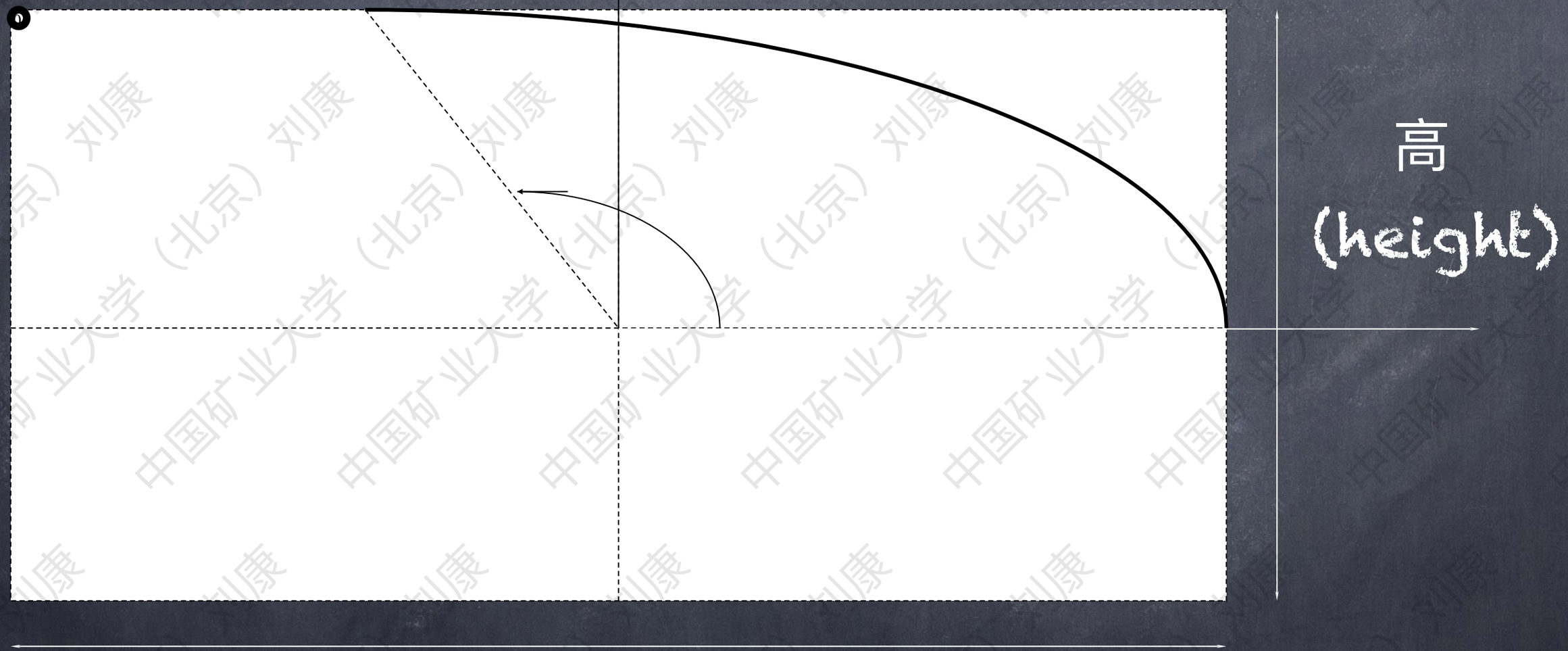
画弧实际上是画部分椭圆。绘制一个弧需要指定6个参数，这6个参数中的前4个的含义与椭圆中的相同，另外的2个参数一个为起始角度，一个为弧角度。

弧的角度用度数来衡量，表明一段弧在两个角度之间绘制。

Graphics提供两种成员方法：

drawArc(int x,int y,int width,int height,int startAngle,int arcAngle)

fillArc(int x,int y,int width,int height,int startAngle,int arcAngle)


高
(height)
宽(width)

4. 绘制多边形

多边形是由一系列首尾相接的直线组成的，用户只需按照某种次序连续绘制每一条直线，Java自动将最后一条直线的终点与第一条直线的起点相连，就可形成多边形。

首先需要创建Polygon类的对象，再向该对象中添加点的坐标，最后调用drawPolygon()成员方法或fillPolygon()成员方法画多边形。

① 创建Polygon类对象

创建Polygon类对象需要使用Polygon类的下述两个构造方法之一：

```
public Polygon();
public Polygon(int xPoints[],int yPoints[],int nPoints);
```

② 绘制多边形的Graphics方法

成员方法	功能说明
public abstract void drawPolygon (int xPoints[], int yPoints[], int nPoints)	绘制一个由x，y数组定义的多边形
public abstract void drawPolygon (Polygon p)	绘制一个由指定多边形定义的多边形
public abstract void fillPolygon (int xPoints[], int yPoints[], int nPoints)	使用当前颜色填充一个多边形
public abstract void fillPolygon (Polygon p)	使用当前颜色填充一个指定多边形

5、复制图形

在制作动画时，常常需要将一个图形稍微移动位置后重新显示出来，即将图中的一块区域复制到另一块区域中。Graphics类提供的copyArea(拷贝)方法就是实现这一操作的。

```
copyArea(int x,int y,int width,int height,int dx,int dy)
```

6. 绘图模式

AWT提供了两种绘图模式：

覆盖模式：在一个图形上画另一个图形时，直接将后一图形画在前一个图形的上面，使前一图形中与后一图形的重叠部分不可见。缺省的绘图模式为覆盖模式。

异或模式：后一图形并不直接覆盖在前一个图形之上，而是将两个图形的重叠区域的像素进行异或运算，以运算结果作为图形的设置来显示。这是制作多彩图形、创造动画效果的一种高效方式。

setXORMode(Color c)