

Swing 用于表示“被绘制的”用户界面类，AWT 表示事件处理等窗口工具箱的底层机制。所有代码无条件加入

```
import java.awt.*;
import javax.swing.*;
```

顶层窗口（未被包含在其他窗口中的窗口）称为**框架（frame）**。在 Swing 中框架类为 JFrame，它不绘制在画布上，而是由窗口系统绘制。

类似下图的代码可创建窗口

```
public static void main(String[] args) {
    EventQueue.invokeLater(() -> {
        MyFrame frame = new MyFrame();
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setVisible(true);
    });
}
```

注意让框架显示需要调用 setVisible 方法。可以用 setLocation, setBounds, setIconImage, setTitle, setResizable 方法设置其他属性。

Toolkit 类包含许多于本地窗口系统打交道的方法，可以借助

```
Toolkit kit = Toolkit.getDefaultToolkit();
Dimension screenSize = kit.getScreenSize();
```

获得屏幕大小。

为了向 frame 中添加组件，需要向 frame 的**内容窗格（content pane）**中添加之，然而实际上可以直接添加为

```
frame.add(c);
```

为了定义 component 类，下列方法是必须的：

```
public void paintComponent(Graphics g) {
    g.drawString("Hello World!", x, y);
}
```

paintComponent 方法由系统自动调用，无需自行调用。Graphics 类型的对象 g 保留绘制图像和文本的设置，所有绘制都依赖于它。

同时还应当重写 `getPreferredSize` 方法，返回首选宽度和高度。

```
public Dimension getPreferredSize() {
    return new Dimension(w, h);
}
```

最后，不同于上述空框架的 `setSize`，可以直接用 `pack` 方法将 `frame` 设置为满意的大小。

`Graphics2D` 提供了比 `Graphics` 更强大的绘图功能，可以通过下列方法调用

```
Graphics2D g2 = (Graphics2D) g;
Rectangle2D r = new Rectangle2D.Double(...);
g2.draw(r);
```

使用内部类 `Double` 是为了使用 `Double` 指定坐标。

`Rectangle2D` 和 `Ellipse2D` 在很大程度上类似，并且可以通过

```
Rectangle2D r = new Rectangle2D.Double();
r.setFrameFromDiagonal(px, py, qx, qy);
```

或者

```
Point2D p = ...; Point2D q = ...;
r.setFrameFromDiagonal(p, q);
```

从对角点设定大小。此外还有 `Line2D` 可供绘图使用。

```
g2.setPaint(Color.RED);
g2.draw(r);
g2.fill(r);
```

可以以给定的颜色绘图或填充。`SystemColor.*` 可获得预设颜色。