



Lập trình Java cơ bản

Cao Đức Thông - Trần Minh Tuấn

cdthong@ifi.edu.vn, tmtuan@ifi.edu.vn

Bài 3. Lập trình GUI (Applet)

- Giới thiệu các loại ứng dụng
- Thư viện AWT
- Tạo các applet
- Đối tượng đồ họa Graphics
- Kỹ thuật khung hình phụ
- Bài tập

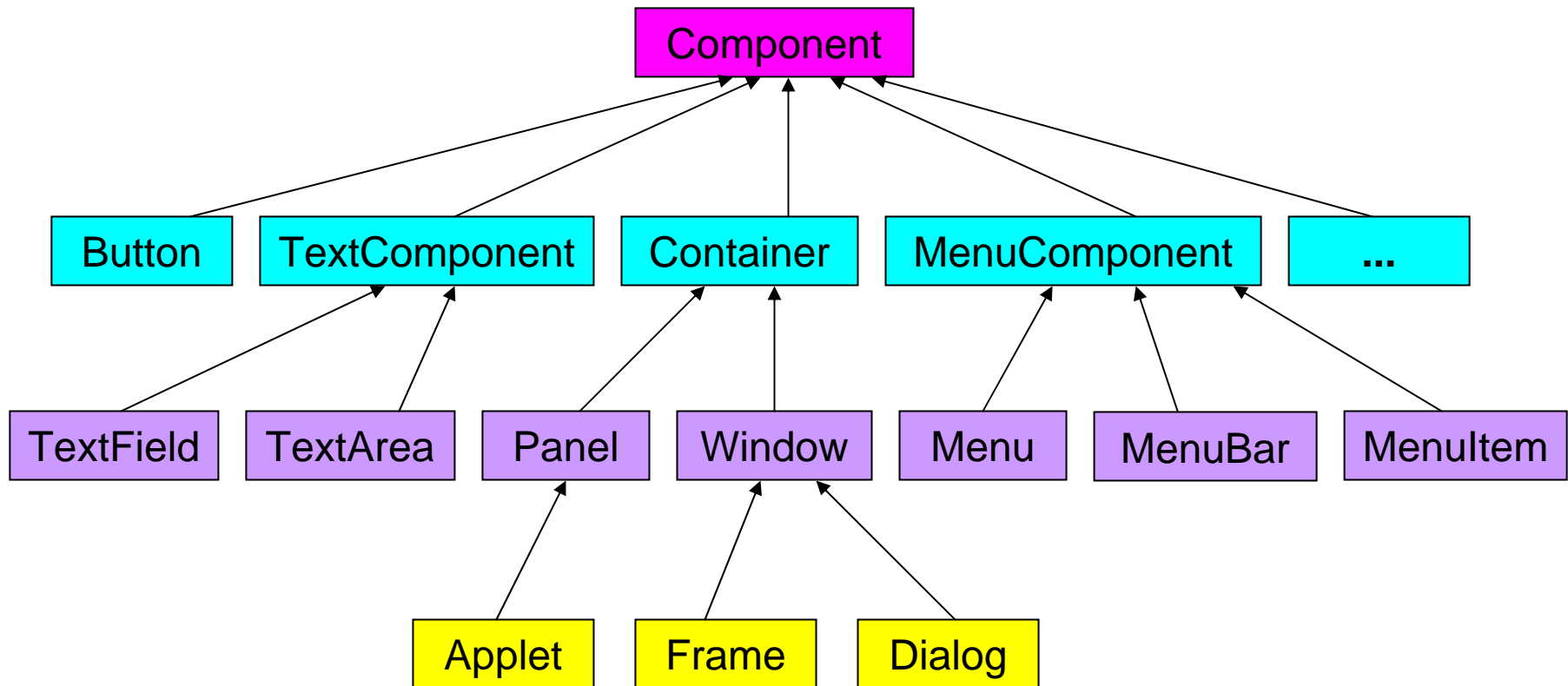
Hai loại ứng dụng Java

- Applet
 - Chương trình Java chạy trong một trang web nhờ vào trình duyệt hỗ trợ Java.
- Stand-alone Application
 - Giao diện dòng lệnh (console): Tương tác với người dùng thông qua các dòng ký tự.
 - Giao diện đồ họa (GUI): Tương tác với người dùng bằng nhiều cách khác nhau như hình ảnh, nút nhấn, biểu tượng... Việc xử lý ứng dụng dựa trên các sự kiện.

Thư viện AWT

- AWT (Abstract Windowing Toolkit)
 - Là bộ thư viện dùng để xây dựng các ứng dụng giao diện đồ hoạ (GUI)
 - Các thành phần giao diện đồ hoạ nằm trong gói **java.awt**

Các thành phần của AWT



Các thành phần của AWT

- Các thành phần container: dùng để “chứa” và quản lý các thành phần khác
 - Applet
 - Frame
 - Dialog
- Các thành phần khác: nằm trong các thành phần container

Nút nhấn (Button)	Thực đơn (Menu)
Ô văn bản (TextField)	Nút chọn (Radio button)
Nhãn (Label)	Danh sách (List)
Vùng văn bản (TextArea)	Lựa chọn (Choice)
....	

Một số phương thức của Component

- `void setVisible(boolean)`: hiển thị hoặc ẩn component
- `Dimension getSize()`: trả về kích thước của component
- `void setSize(Dimension)`: thay đổi kích thước
- `void setEnabled()`: “bật” hoặc “tắt” component
- `void repaint()`: cập nhật lại component
- `void update(Graphics g)`: được gọi qua `repaint()`
- `void paint(Graphics g)`: được gọi qua `update()`
- `void setBackground(Color)`: đặt màu nền
- ...

Xây dựng các applet

- Lớp Applet
 - Java có lớp **java.applet.Applet** kế thừa từ lớp `java.awt.Component` cho phép tạo ra các applet trong Web.
 - Mọi lớp applet do người dùng tạo ra đều phải kế thừa từ lớp Applet.

Xây dựng các applet

- Ví dụ 1: Tạo file TestApplet.java

```
import java.applet.Applet;  
import java.awt.Graphics;  
  
public class TestApplet extends Applet  
{  
    public void paint( Graphics g)  
    {  
        g.drawString("Hello world!", 50, 25);  
    }  
}
```

- Dịch: javac TestApplet.java

Xây dựng các applet

- Thực thi applet
 - Cách 1: Tạo file TestApplet.html có nội dung như sau:

```
<APPLET CODE="TestApplet.class" WIDTH=500 HEIGHT=500  
</APPLET>
```

- Mở file này bằng trình duyệt WEB
- Cách 2: Dùng công cụ appletviewer.
- Gõ lệnh:
appletviewer TestApplet.html

Khung của một applet cơ bản

```
import java.applet.Applet;  
import java.awt.Graphics;  
  
public class TestApplet extends Applet  
{  
    public void init() {...}  
    public void start() {...}  
    public void stop() {...}  
    public void destroy {...}  
    public void paint( Graphics g) {...}  
}
```

Hoạt động của Applet

- `init()`: khởi tạo applet
- `start()`: applet bắt đầu hoạt động
- `stop()`: applet chấm dứt hoạt động
- `destroy()`: giải phóng applet
- Chú ý:
 - `paint()` không phải là phương thức của Applet mà là của Component.
 - `paint()` được gọi mỗi khi cửa sổ được vẽ lại.

Hoạt động của Applet

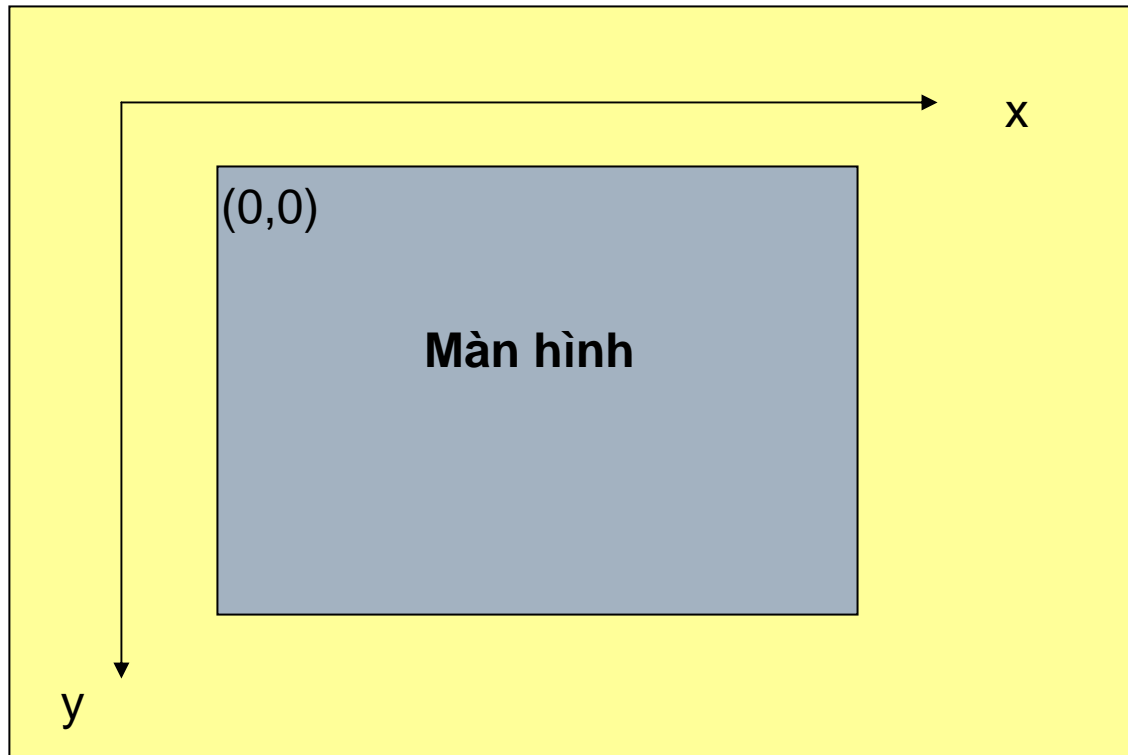
- Vòng đời của một Applet
 - Nạp một applet: applet được khởi tạo và thực thi
 - Chuyển hoặc trở về trang Web: Các phương thức stop và start sẽ được gọi
 - Nạp lại applet: như quá trình nạp applet
 - Thoát khỏi trình duyệt: phương thức stop và destroy sẽ được gọi

Lớp Graphics

- `java.awt.Graphics` là lớp cung cấp các phương thức vẽ đồ họa cơ bản:
 - Đường thẳng (Line)
 - Đường oval (Oval)
 - Hình chữ nhật (Rectangle)
 - Đa giác (Polygon)
 - Văn bản (Text)
 - Hình ảnh (Image)
 - ...

Lớp Graphics

- Hệ toạ độ



Lớp Graphics

- Vẽ đường thẳng
 - `public void drawLine(int x1, int y1, int x2, int y2);`
- Vẽ hình chữ nhật
 - `public void drawRect(int x, int y, int width, int height);`
- Tô một hình chữ nhật
 - `public void fillRect(int x, int y, int width, int height);`
- Xoá một vùng chữ nhật
 - `public void clearRect(int x, int y, int width, int height);`
- Vẽ đa giác
 - `public void drawPolygon(int[] x, int[] y, int numPoint);`
 - `public void drawPolygon(Polygon p);`

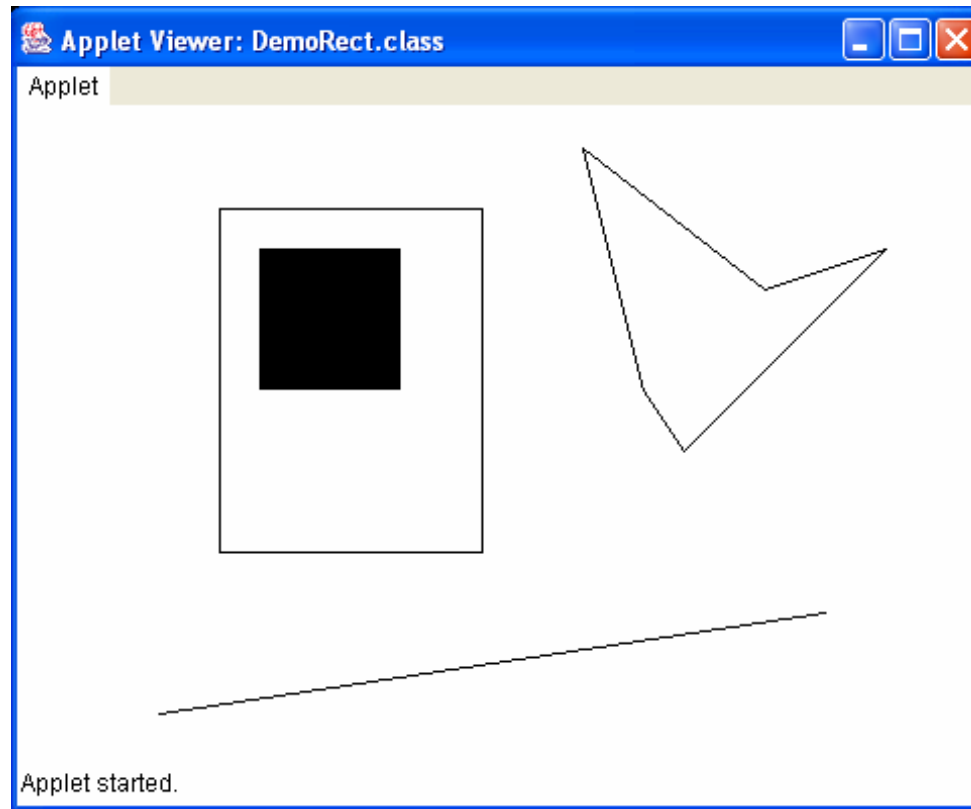
Lớp Graphics

```
import java.applet.Applet;
import java.awt.Graphics;

public class DemoRect extends Applet
{
    public void init()
    {
        System.out.println("Demonstration of some simple figures");
    }
    public void paint(Graphics g)
    {
        g.drawLine(70, 300, 400, 250);
        g.drawRect(100, 50, 130, 170);
        g.fillRect(120, 70, 70, 70);

        int[] x = { 280, 310, 330, 430, 370 };
        int[] y = { 200, 140, 170, 70, 90 };
        g.drawPolygon(x, y, x.length);
    }
}
```

Lớp Graphics



Lớp Graphics

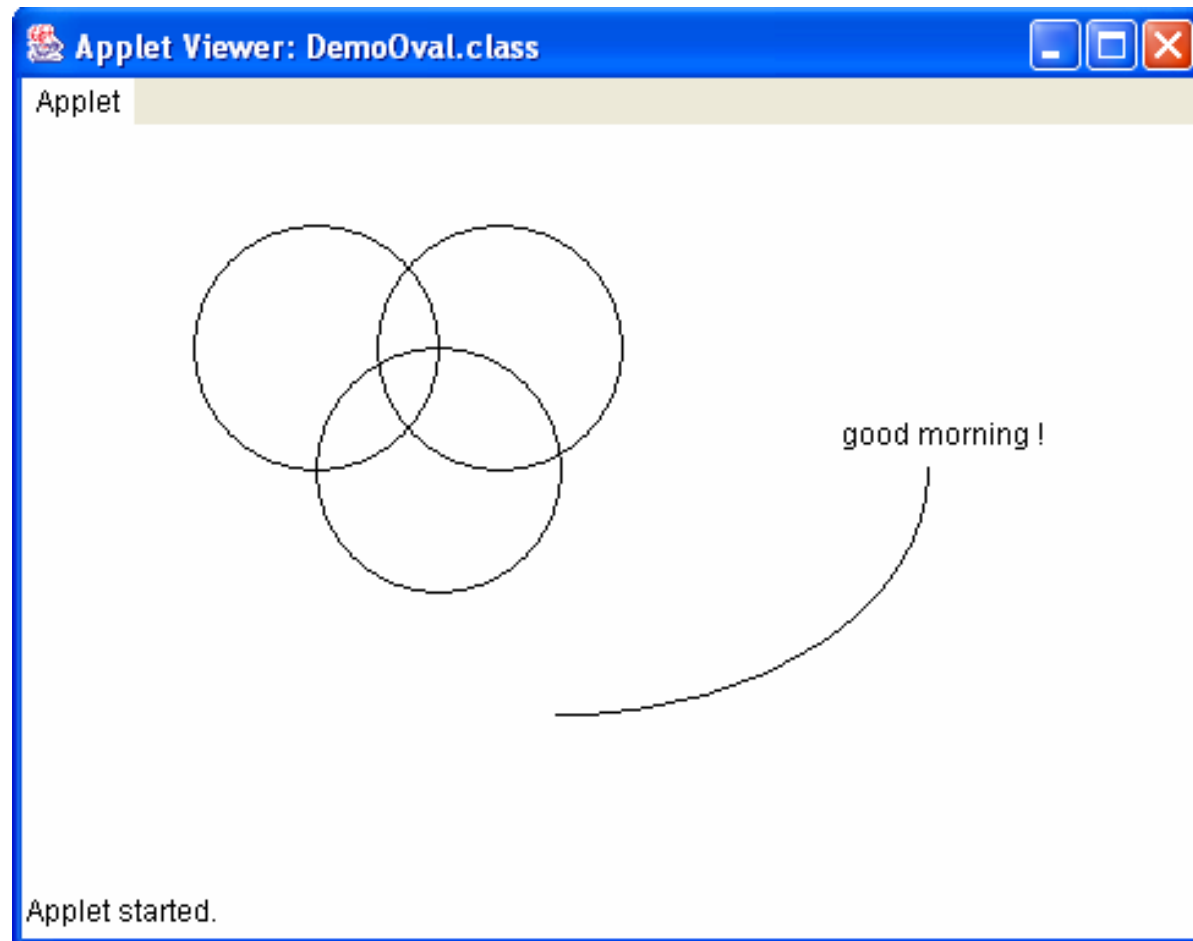
- Vẽ đường tròn/elip
 - `public void drawOval(int x, int y, int width, int height);`
- Tô đường tròn/elip
 - `public void fillOval(int x, int y, int width, int height);`
- Vẽ cung tròn
 - `public void drawArc(int x, int y, int width, int height, int startAngle, int arcAngle);`
- Vẽ xâu kí tự
 - `public void drawString(String str, int x, int y);`
- Vẽ ảnh
 - `public void drawImage(Image img, int x, int y,...);`

Lớp Graphics

```
import java.applet.Applet;
import java.awt.Graphics;

public class DemoOval extends Applet
{
    public void init()
    {
        System.out.println("Demonstration of some simple figures");
    }
    public void paint(Graphics g)
    {
        int xstart = 70, ystart = 40, size = 100;
        g.drawOval(xstart, ystart, size, size);
        g.drawOval(xstart + (size*3)/4, ystart, size, size);
        g.drawOval(xstart + size/2, ystart + size/2, size, size);
        g.drawArc(xstart, ystart, 300, 200, 0, -90);
        g.drawString("good morning !", xstart + 265, ystart + 90);
    }
}
```

Lớp Graphics



Lớp Graphics

```
import java.applet.Applet;
import java.awt.Graphics;
import java.awt.Image;

public class DemoImage extends Applet
{
    public void init()
    {
        System.out.println("Demonstration of imaging");
    }
    public void paint(Graphics g)
    {
        Image image = getToolkit().getImage("summer.jpg");
        g.drawImage(image, 0, 0, this);
    }
}
```

Lớp Graphics



Các lớp tiện ích khác

- Lớp Point: biểu diễn điểm trên màn hình
- Lớp Dimension: biểu diễn kích thước về chiều rộng và chiều cao của một đối tượng
- Lớp Rectangle: biểu diễn hình chữ nhật
- Lớp Polygon: biểu diễn đa giác
- Lớp Color: biểu diễn màu sắc

Các lớp tiện ích khác

```
import java.applet.Applet;
import java.awt.*;

public class DemoColor extends Applet
{
    public void paint(Graphics g)
    {
        Dimension size = getSize();
        g.setColor(Color.orange);
        g.fillRect(0, 0, size.width, size.height);

        Color color = new Color(10, 150, 20);
        g.setColor(color);
        g.drawString("I am a colorful string",
                     size.width/2 - 50, size.height/2);
    }
}
```

Các lớp tiện ích khác



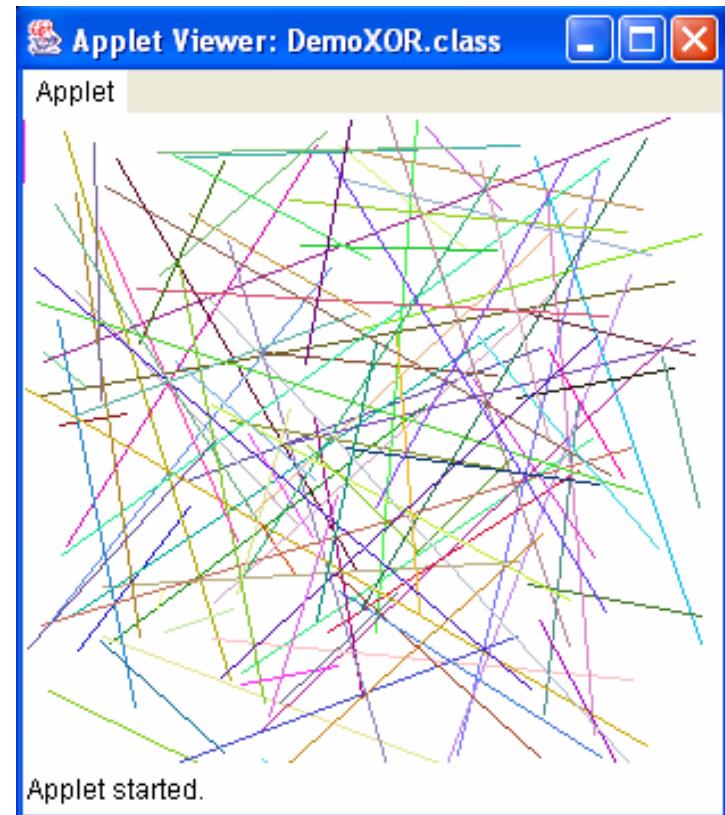
Xử lý font vẽ

```
import java.applet.Applet;  
import java.awt.*;  
  
public class DemoFont extends Applet  
{  
    public void paint(Graphics g)  
    {  
        Font font = new Font("Arial", Font.BOLD, 30);  
        g.setFont(font);  
        g.drawString("I am font Arial, bold, size 30", 50, 50);  
    }  
}
```



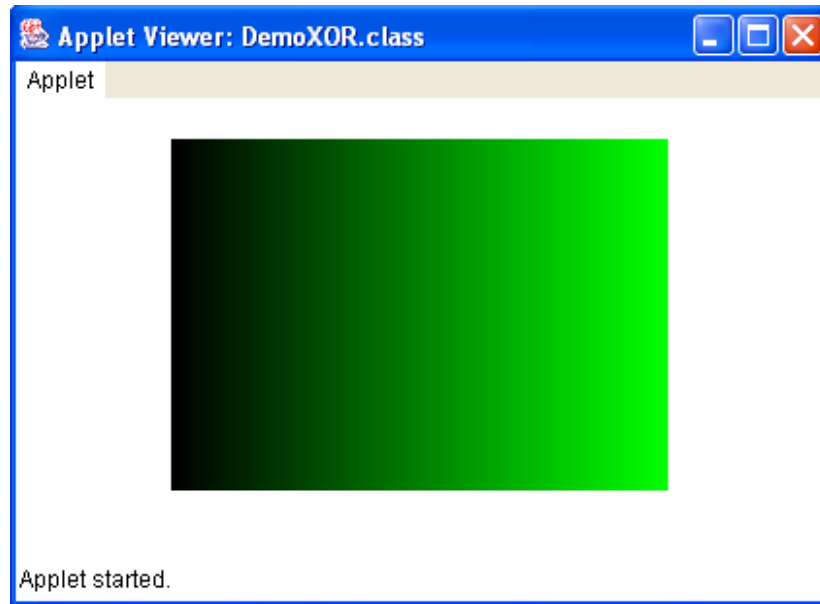
Bài tập tại lớp

- Bài 1: Sinh 100 đường thẳng có tọa độ ngẫu nhiên và màu ngẫu nhiên



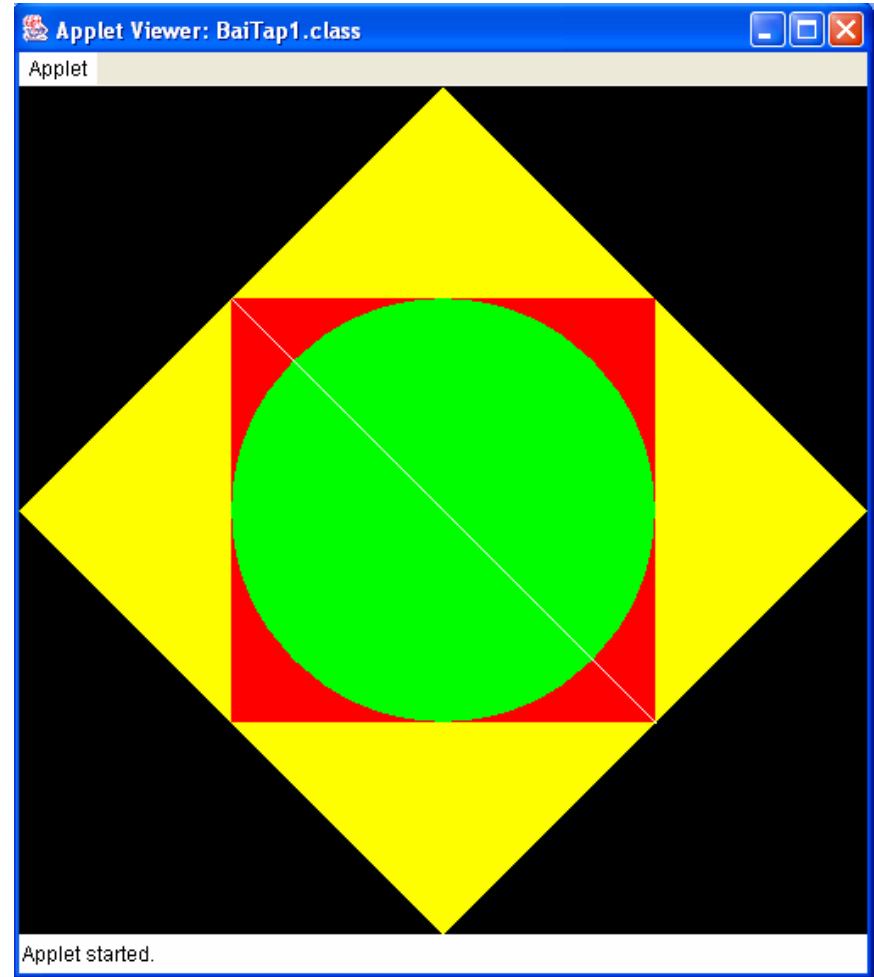
Bài tập tại lớp

- Bài 2: Vẽ sự thay đổi của màu xanh lá cây



Bài tập tại lớp

- Bài 3: Tính toán tọa độ và vẽ ra màn hình các hình sau.



Kĩ thuật khung hình phụ

- Ví dụ về sự di chuyển “bị nháy”

```
import java.applet.Applet;
import java.awt.*;

public class DemoMove1 extends Applet
{
    private int x = 50;
    private int y = 50;

    public void paint(Graphics g)
    {
        if (x > 300) x = 50;
        g.fillOval(x, y, 100, 100);
        delay(100);
        move();
    }
}
```

Kĩ thuật khung hình phụ

- Ví dụ về sự di chuyển “bị nháy”

```
public void delay(int milisecond)
{
    try
    {
        Thread.sleep(milisecond);
    } catch (Exception e) {}
}
```

```
public void move()
{
    x += 5;
    repaint(); // vẽ lại cửa sổ
}
}
```


Kỹ thuật khung hình phụ

- Giải thích lý do nháy:
 - Mỗi lần gọi `repaint()` thì `update()` được gọi
 - *`update()` xoá nội dung cửa sổ và gọi `paint()`*
- Khắc phục:
 - Vẽ hình vào một khung hình phụ trong bộ nhớ thay vì vẽ trực tiếp ra màn hình.
 - Khi vẽ xong, khung hình phụ được hiển thị ra màn hình trong phương thức `paint()`.
 - Nạp chồng `update()` để tránh việc xoá nội dung cửa sổ.

Kỹ thuật khung hình phụ

- Giải pháp

```
//...
public class DemoMove2 extends Applet
{
    //...
    private Image offImage;
    private Graphics offGraphics;

    public void init()
    {
        // tao khung hinh phu
        offImage = createImage(500, 500);
        // lay doi tuong do hoa de ve vao khung hinh phu
        offGraphics = offImage.getGraphics();
    }
}
```

Kỹ thuật khung hình phụ

- Giải pháp

```
public void paint(Graphics g)
{
    //...
    offGraphics.clearRect(0, 0, 500, 500);
    offGraphics.fillOval(x, y, 100, 100);
    g.drawImage(offImage, 0, 0, this);
    //...
}

public void update(Graphics g)
{
    paint(g);
}
//...
}
```

Applet

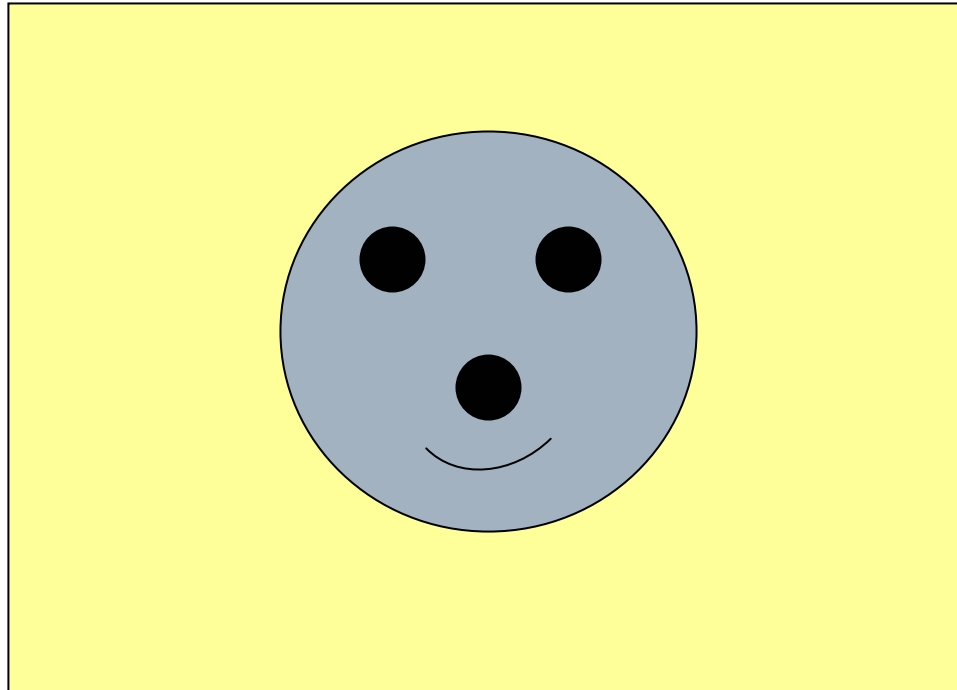
- Khả năng của Applet
 - Applet được đặt tại một Server trên mạng
 - Applet được chuyển tới máy Client theo một trang HTML nào đó
 - Khi một trình duyệt (tương thích với Java) nhận được trang web này, nó sẽ tải mã của Applet và thực thi trên máy client

Applet

- Giới hạn của Applet
 - Không được nạp các thư viện hay các phương thức sử dụng mã gốc (native code).
 - Không được đọc và ghi lên các tập tin của máy đang chạy chúng.
 - Không được khởi động bất kỳ chương trình nào trên máy đang chạy.
 - Không được đọc bất kỳ tính chất nào của hệ thống.

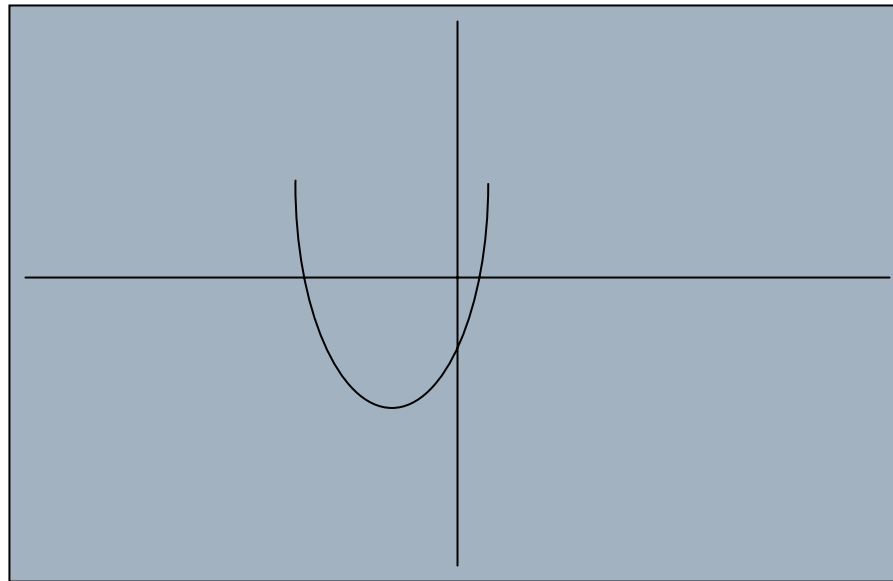
Bài tập

1. Xây dựng một applet để vẽ ra một khuôn mặt như hình vẽ sau:



Bài tập

2. Vẽ đồ thị hàm số $y = 3x^2 - 6x + 1$
3. Vẽ đồ thị hàm số $y = \sin(x)$



Bài tập

4. Dùng kĩ thuật khung hình phụ để vẽ một quả bóng chuyển động tự do trong một applet.
5. Mở rộng bài toán với 2 quả bóng.
6. Mở rộng bài toán với nền applet là một hình ảnh.
7. Mở rộng bài toán với quả bóng là một hình ảnh và có nhiều quả bóng chuyển động đồng thời.