

# 独習Javaゼミ

---

13. 5 カラーの使用

13. 6 テキストの表示

07/07/13

鈴木 慧

# 13.5 カラーの使用

---

java.awt.Color クラスを使って色を扱う事ができる

Color コンストラクタ

Color (int *red*, int *green*, int *blue*)

Color (int *rgb*)

Color (float *r*, float *g*, float *b*)

*red*, *green*, *blue* はそれぞれ 0~255 の整数が入る

*rgb* は 24ビットで表され、23~16ビットが赤、15~8ビットが緑、7~0ビットが青を表す。

*r*, *g*, *b* はそれぞれ 0.0~1.0f の float 型の値が入る

# Color クラスのメソッド

---

Colorクラスに定義されている主な  
静的メソッド/インスタンスメソッド

Color brighter() Color darker() int getBlue()	色を明るくしたオブジェクトを返す 色を暗くしたオブジェクトを返す 青の値を返す
static int HSBtoRGB (float <i>h</i> , float <i>s</i> , float <i>b</i> )	<i>h,s,b</i> で指定された色相,彩度,明度をint型の値にコード化して返す

(p.395 表13-2参照)

# カラーの使用例

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

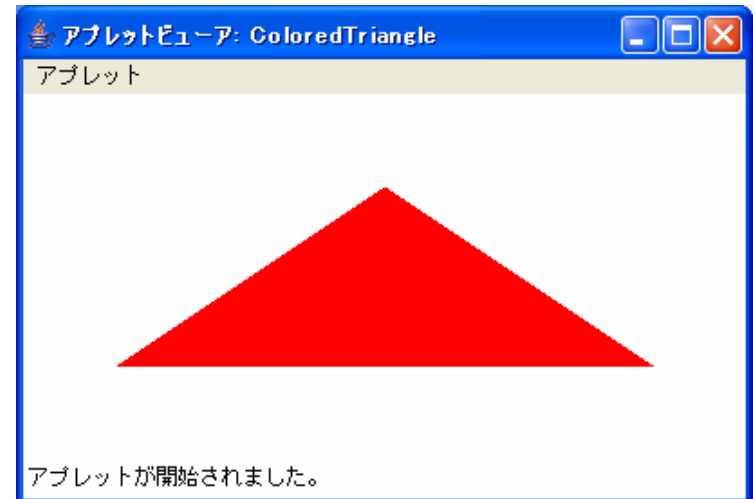
/*
  <applet code="ColoredTriangle"
    width=400 height=200>
  </applet>
*/

public class ColoredTriangle extends
    Applet {

    public void paint(Graphics g) {
        int n = 3;
        int xdata[] = new int[n];
        int ydata[] = new int[n];
        xdata[0] = 50;
        ydata[0] = 150;
        xdata[1] = 200;
```

```
        ydata[1] = 50;
        xdata[2] = 350;
        ydata[2] = 150;
        int rgb = Color.HSBtoRGB (1.0f,
            1.0f, 1.0f);
        g.setColor(new Color(rgb));
        g.fillPolygon(xdata, ydata, n);
    }
}
```

## 実行結果



## 13.6 テキストの表示

---

アプレットに表示する文字列の外観と位置を、  
java.awt.Font クラスやjava.awt.FontMetrics  
クラスを使う事で制御できる

Font コンストラクタ

Font (String *フォント名*, int *書体*, int *文字の大きさ*)

setFont() メソッド

Abstract void setFont (Font *フォント名*)

# FontMetrics クラス

---

FontMetricsコンストラクタ

FontMetrics (Font **フォント名**)

FontMetrics クラスに定義されている主なインスタンスメソッド

int charWidth(char **c**)

**c** の幅を返す

int getAscent()

アセントを返す

int getDescent()

ディセントを返す

int getHeight()

高さを返す

# テキストの表示例(1/4)

---

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

/*
  <applet
    code="FontMetricsDemo"
    width=300 height=200>
  </applet>
*/

public class FontMetricsDemo
  extends Applet {

  public void paint (Graphics
    g) {
```

```
// ベースラインの描画 (1)
int baseline = 100;
g.setColor (Color.lightGray);
g.drawLine (0, baseline, 300,
  baseline);

// 文字列の描画
Font font = new Font
("Serif", Font.BOLD, 36);
g.setFont(font);
g.setColor(Color.black);
g.drawString("Wxyz", 5,
  baseline);
g.setColor(Color.lightGray);
```

# テキストの表示例(2/4)

---

```
// FontMetricsの取得
```

```
fm = g.getFontMetrics (font);
```

```
// ベースラインからアセントを引いた位置に線を描画する (2)
```

```
int ascent = fm.getAscent();
```

```
int y = baseline - ascent;
```

```
g.drawLine(0, y, 300, y);
```

```
// ベースラインにディセントを足した位置に線を描画する (3)
```

```
int descent = fm.getDescent();
```

```
y = baseline + descent;
```

```
g.drawLine(0, y, 300, y);
```

```
// ベースラインとディセントとレディングを足した位置に線を描画する (4)
```

```
int leading = fm.getLeading();
```

```
y = baseline + descent + leading;
```

```
g.drawLine(0, y, 300, y);
```

# テキストの表示例(3/4)

---

```
// ベースラインと高さを足した位置に線を描画する (5)
```

```
int height = fm.getHeight();  
y = baseline + height;  
g.drawLine(0, y, 300, y);
```

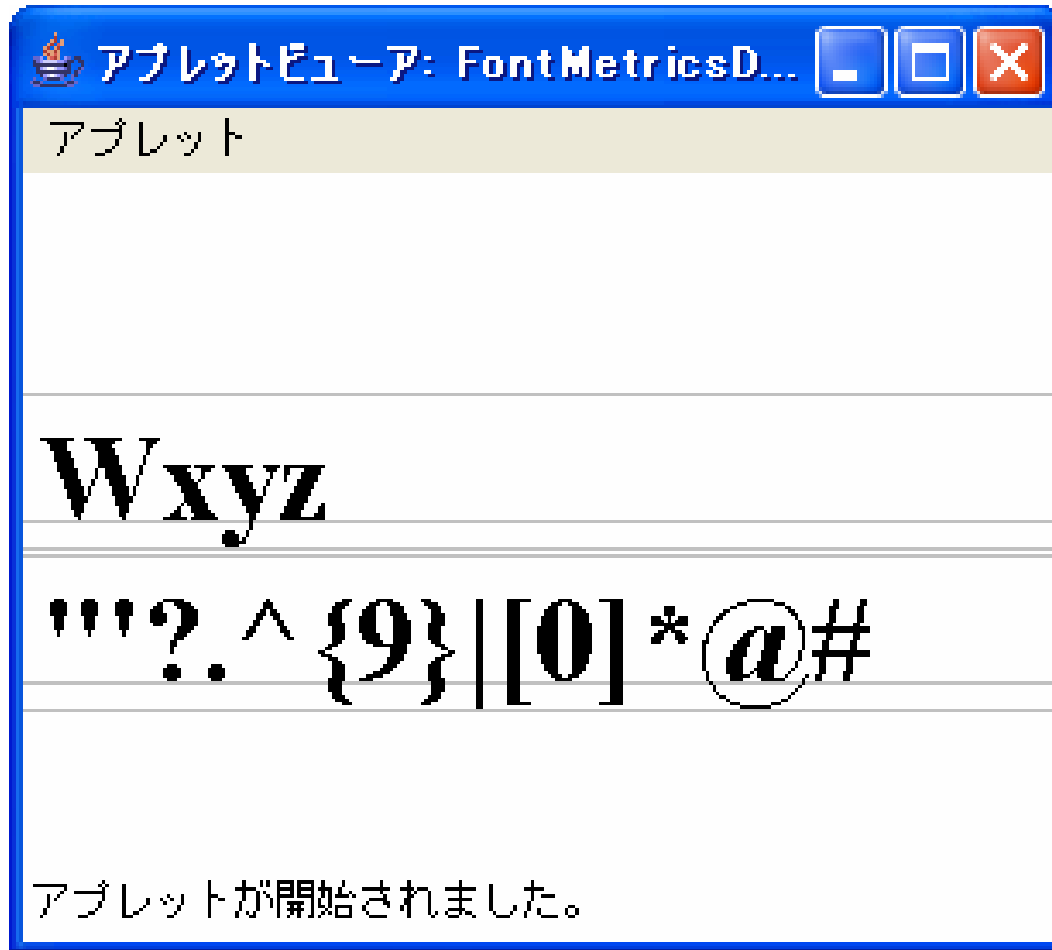
```
// 文字列の描画
```

```
g.setColor(Color.black);  
g.drawString("¥"?".^{9}|[0]*@#", 5, y);  
g.setColor(Color.lightGray);
```

```
// ベースラインとディセントと高さを足した位置に線を描画する (6)
```

```
y = baseline + height + descent;  
g.drawLine(0, y, 300, y);  
}  
}
```

# テキストの表示例(4/4)



実行結果

(2)

(1)

(3)

(4)

(5)

(6)

# 宿題

---

中を塗りつぶした星(★)の絵を書き、  
その中に Star という文字を入れる  
プログラムを作成せよ

色やフォントは何でもよいものとする。