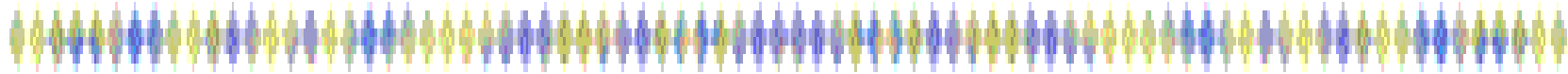


第14回 Javaゼミ



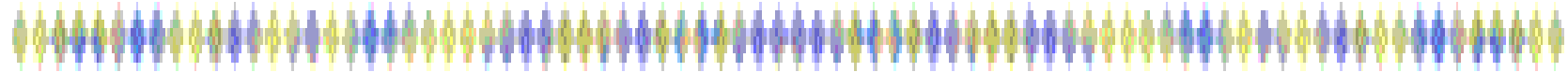
13.12 スレッドの使用

13.13 ダブルバッファリング

発表日 : 2007/7/20

発表者 : 阿部 竜之介

スレッドの使用



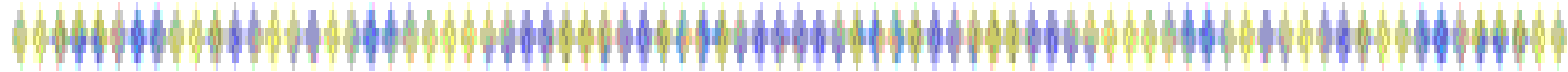
スレッドを使うアプレット

- スレッドを使うアプレットには、一般にウィンドウの表示の更新が必要
- repaint()メソッドを呼び出してウィンドウの更新を要求する

repaint()メソッド

- repaint()メソッドを呼び出すと、JVMでウィンドウを更新する適切なタイミングが判断され、update()メソッドが呼び出される

update()メソッド



update()メソッド

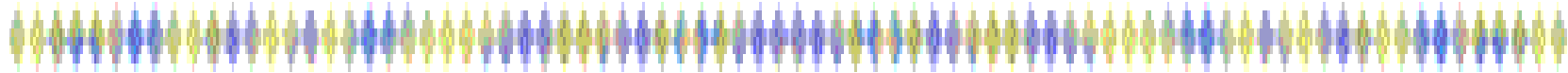
- 既定の実装では、アプレットウィンドウを背景色で塗りつぶしてからpaint()メソッドを呼び出す
- update()メソッドはオーバーライドすることができる

例

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

//背景色で塗りつぶさずにpaint()メソッドを呼び出している

ダブルバッファリング



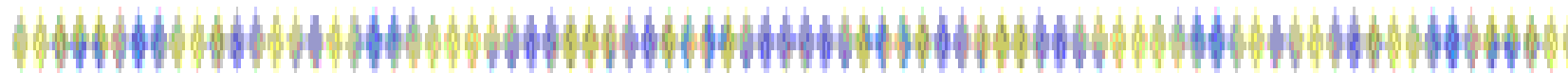
ダブルバッファリングとは

- アプレットウィンドウのちらつきを防ぐ手法のこと
- ちらつきはupdate()メソッドの既定の実装での、全面塗りつぶしと、別の色による描画が頻繁に行われることが原因

ダブルバッファリングのやり方

- update()メソッドをオーバーライドし、全面塗りつぶしをしないように、ただpaint()メソッドだけを呼び出すことにする
- paint()メソッド内では、全ての描画操作をバッファに対して行う
- 最後にdrawImage()メソッドを使ってバッファをウィンドウにコピーする

createImage()メソッド

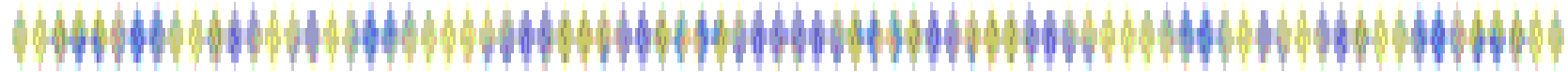


- バックグラウンドのバッファを作成するには、ComponentクラスのcreateImage()メソッドを使う

createImage()メソッド

```
Image createImage( int width, int height )
```

getGraphics()メソッド

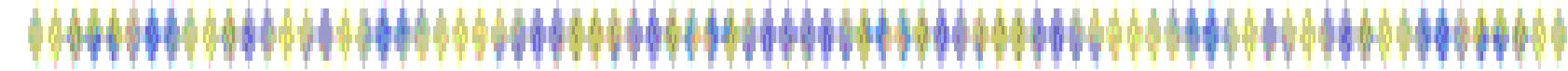


■ createImage()メソッドで作成したバッファのグラフィックコンテキストは、ImageのgetGraphics()メソッドを使って取得する

```
abstract Graphics getGraphics() getGraphics()メソッド
```

■ このグラフィックコンテキストを使って描画操作を実行する

プログラム例



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

/*
  <applet code="DoubleBuffer"
  width=300 height=100>
  </applet>
*/

public class DoubleBuffer extends
Applet implements Runnable {
  int x = 0;
  Thread t;
  Image buffer;
  Graphics bufferg;
```

```
public void init() {
  //スレッドを開始する
  t = new Thread( this );
  t.start();
  //バッファを作成する
  Dimension d = getSize();
  buffer = createImage( d.width,
d.height );
}

public void run() {
  try {
    while( true ) {
      //再描画を要求する
      repaint();
```

プログラム例

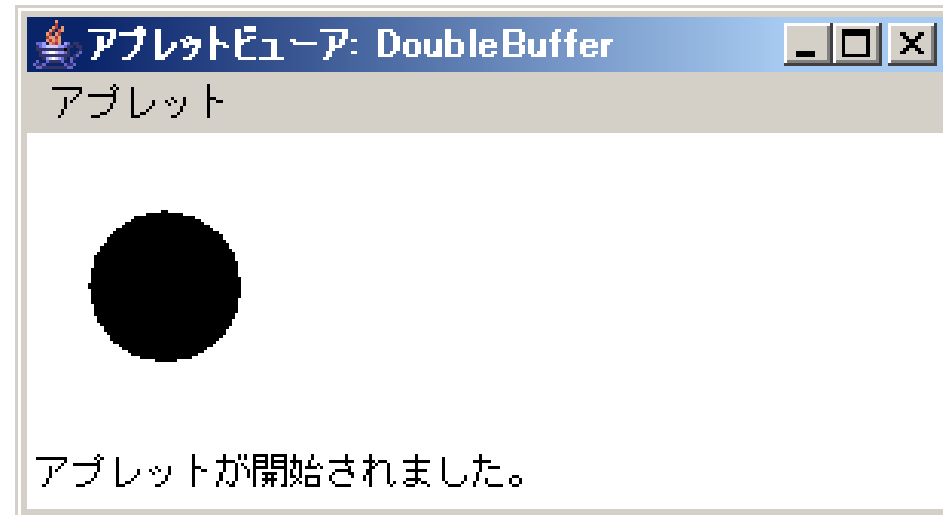
```
        //ウィンドウを更新する前に休止
        する
        Thread.sleep( 100 );
    }
}
catch( Exception e ) {
}
}

public void update( Graphics g ) {
    paint( g );
}

public void paint( Graphics g ) {
    //バッファのグラフィックコンテキスト
    トを取得する
    if( bufferg == null ) bufferg =
```

```
buffer.getGraphics();
    //バッファを描画する
    Dimension d = getSize();
    bufferg.setColor( Color.white );
    bufferg.fillRect( 0, 0, d.width,
d.height );
    bufferg.setColor( Color.black );
    bufferg.fillOval( x, d.height / 4, 50,
50 );
    //ウィンドウを更新する
    g.drawImage( buffer, 0, 0, this );
    //x座標を増やす
    x += 5;
    if( x + 50 > d.width ) x = 0;
}
}
```

実行結果



ウィンドウを左から右へ移動する黒く塗りつぶした円を表示する。

練習問題

秒針のみのアナログ時計を描画するアプレットを作成しなさい。ただし、表示のちらつきを防ぐため、ダブルバッファリングを用いること。

