

Aplety

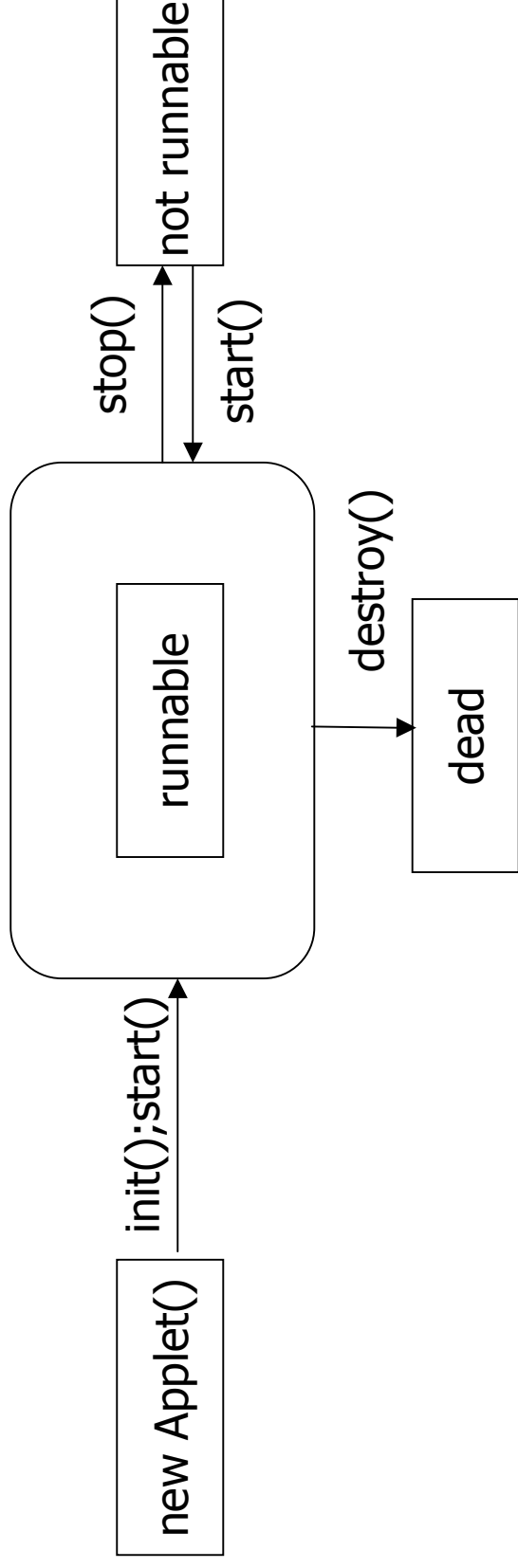
Na podstawie:

<http://java.sun.com/docs/books/tutorial/uiswing>

Aplety

- Aplikacje graficzne pobierane ze serwera www i uruchamiane w przeglądarkach internetowych
- Ze względów bezpieczeństwa działania apletów są kontrolowane przez obiekt klasy `SecurityManager`, implementacja tego obiektu w przeglądarkach zazwyczaj uniemożliwia:
 - utworzenie połączenia sieciowego do innego komputera niż ten, z którego dany aplet został *ściągnięty*
 - korzystanie z lokalnych zasobów komputera np.: dyski, drukarki
 - uruchamianie kodu lokalnych bibliotek systemowych (dll)

Cykl życia apletu



- Operator `new` i metoda `init()` oraz `start()` jest wywoływane przez przeglądarkę po zakodowaniu kodu apletu
- Metoda `stop()` jest wywoływana przez przeglądarkę przed przejściu na następną stronę lub zminiaturyzowanie przeglądarki do ikonki
- Metoda `start()` jest wywoływana przez przeglądarkę przed powrocie na stronę apletu lub powiększenie przeglądarki
- Metoda `destroy()` jest wywoływana przez przeglądarkę przed zamknięciem okna

Uruchamianie apletu

- Przez przeglądarkę - zagnieżdżenie w HTML znacznika

```
<applet>:  
<APPLET  
    CODE = "RectAnimApplet.class"  
    CODEBASE = "myHome/"  
    ARCHIVE = "myPackage.jar"  
    WIDTH = 300  
    HEIGHT = 50 >  
< PARAM NAME = fps VALUE = 10 >  
Nie masz zainstalowanej Javy  
</APPLET>
```

- Za pomocą appletviewer (np.: do testów) jak wyżej +:
C:>appletviewer myhtml.html

Przykładowy aplet 1/2

5

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

public class RectAnimApplet extends JApplet {
    RectangleAreaAnim animator;

    public void init() {
        String str;
        int fps = 0;
        str = getParameter("fps");
        try { fps = Integer.parseInt(str);
            } catch (Exception e) {}
        int delay = (fps > 0) ? (1000 / fps) : 100;
        animator = new RectangleAreaAnim(delay);
        setContentPane(animator);
    }
    public void start() {
        animator.startAnimation();
    }
    public void stop() {
        animator.stopAnimation();
    }
    public void destroy() {
        animator.stopAnimation();
    }
}
```

Przykładowy aplet 2/2

6

```
class RectangleAreaAnim extends JPanel implements
    ActionListener {
    ...
    public RectangleAreaAnim(int delay) {
        addMouseListener(new MouseAdapter() {
            ...
            public void mousePressed(MouseEvent e) {
                ...
                startAnimation();
                repaint();
            }
        });
    }

    public synchronized void startAnimation() {
        if (point!=null & !timer.isRunning()) timer.start();
    }

    public synchronized void stopAnimation() {
        if (timer.isRunning()) timer.stop();
    }
    ...
}
```

Wątki (apletu) i komponenty Swing

- Metody `init()`, `start()`, `stop()`, `destroy()` nie są wykonywane w wątku *event-dispatching*
- Operowanie na komponentach jest bezpieczne tylko w wątku *event-dispatching* z wyjątkiem:
 - wątku *main*, ale tylko przed *realizacją* komponentu, czyli wywołaniem metod `setVisible(true)`, `pack()`, `show()` lub dodaniem do innego zrealizowanego komponentu
 - metody `init()` (w większości nowoczesnych przeglądark)
 - metod klasy `JComponent`: `repaint()`, `revalidate()`
 - metod klasy `JComponent`: `addListenerXXX()`, `removeListenerXXX()`
 - i niektórych innych zaznaczonych w dokumentacji
- W innych przypadkach należy skorzystać z metod klasy `SwingUtilities`: `invokeLater`, `invokeAndWait`