Multimedia

Obrazy i dźwięki w apletach Javy, praca z tekstem, zdarzenia od myszki, obsługa wyjątków w Javie, aplety z wątkami, animacja poklatkowa.

FUW

Wczytywanie grafiki

- **Standardowo Java obsługuje pliki w formacie GIF oraz JPG.**
- **Do wczytywania grafiki wykorzystuje się metodę getImage() klasy Applet. Metoda ta pobiera spod wskazanego adresu URL (Uniform Resorce Locator) plik z obrazkiem i przekształca go w obiekt Image:**

  Image img = getImage(URL, FileName);

  - URL - jest bazowym adresem katalogu zawierającego plik graficzny,
  - FileName - określa nazwę tego pliku.
- **Adres URL uzyskuje się za pomocą metody getCodeBase(), która zwraca ścieżkę do katalogu z apletem.**
- **Można też użyć metody getDocumentBase(), która zwraca adres dokumentu HTML z zagnieżdżonym apletem.**
- **Najczęściej plik z grafiką znajduje się w tym samym katalogu co kod klasy apletu i można go pobrać za pomocą wywołania:**

  Image img = getImage(getCodeBase(), "plik.gif");

Wyświetlanie obrazków

- **Do wyświetlania obrazków służy metoda drawImage() klasy Graphics.**
- **Jej najprostsza postać to:**

  g.drawImage(img, 20, 20, this);

  - img - referencja na obiekt Image,
  - 20, 20 - współrzędne lewego górnego rogu wyświetlanego obrazka
  - this - parametr odnoszący się do bieżącego komponentu, w którym obrazek jest wyświetlany.
- **Często stosowana jest druga forma tej metody, która pozwala zmienić rozmiar obrazka. Na przykład jeśli chcemy wyświetlić obrazek w obszarze o szerokości 100 i wysokości 80 pikseli, wprowadzamy dodatkowe dwa parametry:**

  g.drawImage(img, 20, 20, 100, 80, this);

Dźwięki w apletach Javy

- **Obsługę plików dźwiękowych zapewnia klasa AudioClip z pakietu java.applet.**
- **Wcześniejsze wersje Javy mogły obsługiwać tylko pliki dźwiękowe w formatach AU lub CLIP. Obecnie Java potrafi obsługiwać również formaty: AIFF, WAV i MIDI**
- **Aby załadować plik dźwiękowy, należy użyć metody getAudioClip() z parametrami w postaci adresu URL i nazwy pliku. Na przykład:**

  AudioClip ac = getAudioClip(getCodeBase(), "beep.au");
- **Po załadowaniu pliku z muzyką można go odtworzyć za pomocą metod klasy AudioClip:**

  play() - jednorazowe odtwarzanie klipu

  loop() - odtwarzanie klipu w pętli
- **Jeśli odgrywamy klip w pętli, musimy pamiętać o przerwaniu odtwarzania dźwięku za pomocą metody stop() wywoływanej na rzecz obiektu AudioClip.**

Praca z tekstem

Klasa FontMetrics zawiera metody przeznaczone do pracy z tekstem. Metody te pozwalają ustalić własności używanej czcionki, takie jak wysokość czy szerokość wyświetlanych znaków.

Metoda	Opis
`FontMetrics(Font)`	Konstruktor - tworzy nowy obiekt FontMetrics na podstawie określonej czcionki.
`getFont()`	Zwraca bieżącą czcionkę.
`getHeight()`	Zwraca całkowitą wysokość czcionki.
`charWidth(char)`	Zwraca szerokość (w pikselach) danego znaku tej czcionki.
`stringWidth(String)`	Zwraca szerokość (w pikselach) danego łańcucha.

Zdarzenia od myszki

- **Wyróżnia się dwa rodzaje zdarzeń - zdarzenia typu Mouse:**
  - kliknięcie, a także tylko naciśnięcie lub zwolnienie dowolnego klawisza myszy,
  - najechanie kursorem myszy na obszar zajmowany przez dany komponent,
  - opuszczenie obszaru komponentu.
- **Zdarzenia typu MouseMotion:**
  - przesuwanie kursora myszy,
  - przeciąganie kursora myszy.
- **Aby wykrywać określone zdarzenia pochodzące od myszy, należy zaimplementować interfejsy:**
  - MouseListener
  - MouseMotionListener
- **MouseMotionListener posiada tylko dwie metody:**
  - public void mouseDragged(MouseEvent e){}
  - public void mouseMoved(MouseEvent e){}

Zdarzenia od myszki

- **MouseListener wymaga zdefiniowania pięciu metod:**
  - public void mouseClicked(MouseEvent e){}
  - public void mousePressed(MouseEvent e){}
  - public void mouseReleased(MouseEvent e){}
  - public void mouseEntered(MouseEvent e){}
  - public void mouseExited(MouseEvent e){}
- **Parametrem każdej z tych metod jest referencja do obiektu MouseEvent - zawierającego m.in. informację o aktualnych współrzędnych kursora myszy.**
- **Na rzecz obiektu MouseEvent można wywoływać następujące metody:**
  - getClickCount() - zwraca całkowitą liczbę kliknięć przyciskiem myszy,
  - getPoint() - zwraca obiekt Point - współrzędne (x,y) klikniętego punktu,
  - getX() - zwraca współrzędną x pozycji kursora,
  - getY() - zwraca współrzędną y pozycji kursora.

Obsługa wyjątków w Javie

- **Czasami zdarza się tak, że program kompiluje się poprawnie, natomiast po jego uruchomieniu pojawiają się problemy.**
- **Wszystkie zdarzenia, które mogą spowodować awarię w pracy programu nazywane są wyjątkami (*exceptions*).**
- **Do obsługi wyjątków służy blok try... catch, który jest wykonywany automatycznie w przypadku pojawienia się błędu.**
  - Słowem kluczowym try rozpoczynamy blok kodu, który może zgłaszać wyjątki. Jeśli wewnątrz tego bloku pojawi się wyjątek jest on wyrzucany (throw).
  - Słowo kluczowe catch używane jest do przechwytywania wyjątków.
- **Najczęściej używana postać bloku try...catch jest następująca:**

```
try {// blok kodu, który może spowodować wyjątek
}
catch (Typ_wyjątku identyfikator_wyjątku) {
     // obsługa wyjątku
}
```

Aplety z wątkami

- **Wątki to sekwencje kodu, które mogą wykonywać się niezależnie od siebie (asynchronicznie).**
- **Umieszczenie czasochłonnych zadań w ich własnych wątkach zwiększa wydajność programu - pozwala wykorzystać czas bezczynności, który występuje w większości programów.**
- **Wątki są reprezentowane przez klasę Thread, która znajduje się w pakiecie java.lang.**
- **Istnieją dwa sposoby tworzenia wątków:**
  1. utworzenie klasy implementującej interfejs Runnable,
  2. wyprowadzenie nowej klasy z klasy Thread, która sama implementuje interfejs Runnable
- **Klasy działające jako wątki, tzw. klasy wątkowe muszą zawierać metodę public void run() zawierającą cały kod wątku.**
- **Uruchomienie wątku następuje poprzez wywołanie metody Thread.start();**

Konstruktory i metody klasy Thread

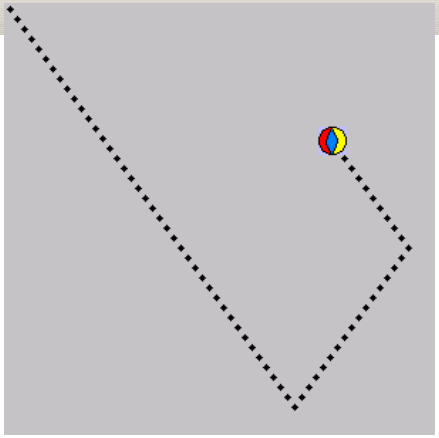
Metoda	Opis
Thread()	Tworzy nowy wątek (obiekt klasy Thread)
Thread(String nazwa)	Tworzy nowy wątek o określonej nazwie
Thread(Runnable o)	Tworzy nowy wątek dla obiektu Runnable
Thread(Runnable o, String nazwa)	Tworzy nowy wątek dla obiektu Runnable i nadaje mu określoną nazwę
getName()	Pobiera i zwraca nazwę wątku
getPriority()	Pobiera i zwraca priorytet wątku
isAlive()	Sprawdza, czy wątek jest aktywny
setName(String nazwa)	Ustawia nazwę wątku
setPriority()	Ustawia priorytet wątku

Metody klasy Thread

Metoda	Opis
`start()`	Uruchamia wątek poprzez wywołanie jego metody run()
`sleep(long)`	Usypia wątek na określoną liczbę milisekund
`wait()`	Powoduje zablokowanie wątku do chwili wywołania metody notify()
`notify()`	Odblokowanie wątku
`notifyAll()`	Odblokowanie wszystkich wstrzymanych wątków
`join()`	Czeka na zakończenie wątku
`destroy()`	Zniszczenie wątku
`yield()`	Sprawia, że wątek oddaje sterowanie

Ćwiczenie 1

Narysuj w edytorze grafiki kulkę o wymiarach 20x20 pix. i zapisz rysunek w pliku "kulka.gif". Wykonaj prostą animację ruchu kulki odbijającej się od brzegów apletu. Przesunięcie kulki uzyskuje się poprzez wyczyszczenie apletu metoda clearRect i narysowanie jej w nowym miejscu.



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

public class Grafika extends Applet
{
   Image image;
   int x, y, vx, vy;

   public void init()
   {
      x=0; y=0; vx=7; vy=5;
      image = getImage(getCodeBase(),"kulka.gif");
   }
```

FUW

Animacja ruchu kulki

```
public void paint(Graphics g)
{
    int xm = getSize().width;
    int ym = getSize().height;
    while (true)
    {
       g.drawImage(image, x, y, this);
       x=x+vx;
       y=y+vy;
       if (x<=0) vx=-1*vx;
       if (x>=xm-20) vx=-1*vx;
       if (y<=0) vy=-1*vy;
       if (y>=ym-20) vy=-1*vy;
       for (int k=0; k<3e6; k++);
       g.clearRect(0, 0, xm, ym);
    }
}
}
```

Wersja wielowątkowa

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

public class Grafika extends Applet
implements Runnable
{
   Thread thread = null;
   Image image;
   int x,y,vx,vy;

   public void init()
   {
     x=0; y=0; vx=7; vy=5;
     image = getImage(getDocumentBase(),"kulka.gif");
     thread = new Thread(this);
   }
   publÏc void start()
   {
     thread.start();
   }
```

Wersja wielowątkowa - c.d.

```
    public void run()
    {
      while (thread!=null)
      {
        try { thread.sleep(50);
        } catch (InterruptedException e) {}
        repaint();
      }
    }

    public void paint(Graphics g)
    {
      int xm = getSize().width;
      int ym = getSize().height;
      g.drawImage(image, x, y, this);
      x=x+vx;
      y=y+vy;
      if (x<=0) vx=-1*vx;
      if (x>=xm-20) vx=-1*vx;
      if (y<=0) vy=-1*vy;
      if (y>=ym-20) vy=-1*vy;
    }
}
```

Ćwiczenie 2

Utwórz aplet z grafiką i efektami dźwiękowymi. Na aplecie umieść dwa przyciski: Start - rozpoczynający odtwarzanie pliku dźwiękowego i Stop - zatrzymujący odtwarzanie tego klipu.


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

public class Muzyka extends Applet
implements ActionListener
{
 AudioClip audioclip;
 Image image;
 Button start, stop;

 public void init()
 {
  audioclip = getAudioClip(getCodeBase(), "song.mid");
  image = getImage(getCodeBase(),"choinka.jpg");
  start = new Button("Start");
  start.addActionListener(this);
  add(start);
```

Dźwięk i grafika

```
    stop = new Button("Stop");
    stop.addActionListener(this);
    add(stop);
  }

  public void paint(Graphics g)
  {
    g.drawImage(image, 10, 40, 280, 250, this);
  }

  public void actionPerformed(ActionEvent e)
  {
    Object o = e.getSource();

    if (o==start) audioclip.loop();
    if (o==stop)  audioclip.stop();
  }
}
```

FUW

Ćwiczenie 3

Napisz klasę obsługującą zdarzenia od myszki. Pojedyncze kliknięcie myszką w obszarze apletu ma spowodować narysowanie punktu oraz wyświetlenie jego współrzędnych.

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

public class Myszka extends Applet
implements MouseListener
{
   Point p;
   String string="";

   public void init()
   {
      addMouseListener(this);
   }

   public void paint(Graphics g)
   {
      g.drawString(string, 20, 20);
      g.drawLine(p.x, p.y, p.x, p.y);
   }
```

Metody interfejsu MouseListener

```
    public void mouseClicked(MouseEvent e) {
    }
    public void mousePressed(MouseEvent e) {
    }
    public void mouseReleased(MouseEvent e){
       p = e.getPoint();
       string = "x = " + p.x + " y = " + p.y;
       repaint();
    }
    public void mouseEntered(MouseEvent e) {
    }
    public void mouseExited(MouseEvent e)  {
    }
}
```

Przykład - falujący tekst

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

public class Napis extends Applet
implements Runnable {
  String napis = "Java";
  Font font = new Font("Serif", Font.BOLD, 24);
  FontMetrics fm = getFontMetrics(font);
  int dl = napis.length();
  char[] znak = new char[dl];
  Thread thread = null;

  public void init() {

  }

  public void start() {
    if (thread == null) {
      thread = new Thread(this);
      thread.start();
    }
  }
  public void stop() {
    thread = null;
  }
```

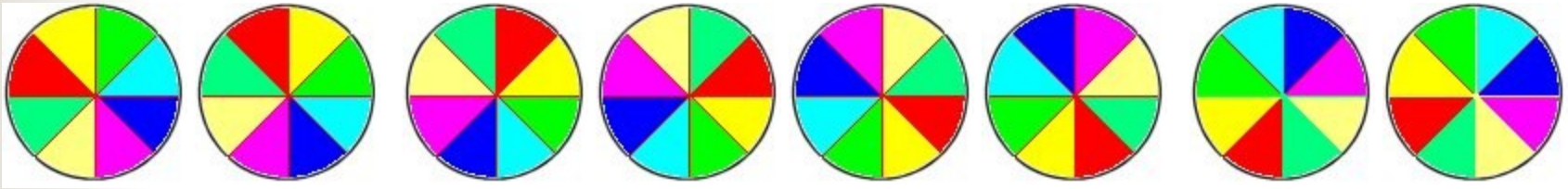
Falujący tekst - c.d.

```
  public void run() {
    while (thread!=null) {
      try { Thread.sleep(300);
      } catch (InterruptedException e) {}
      repaint();
    }
  }

  public void losuj_kolor(Graphics g) {
     int red = (int)(Math.random()*255);
     int green = (int)(Math.random()*255);
     int blue = (int)(Math.random()*255);
     g.setColor(new Color(red, green, blue));
  }
  public void paint(Graphics g) {
       g.setFont(font);
       int x=40, y=40;
       for (int i=0; i<dl; i++) {
         losuj_kolor(g);
         g.drawChars(znak, i, 1, x, y);
         x += fm.charWidth(znak[i]);
         y += (int)(Math.random()*20 - 10);
       }
}
```

Animacja poklatkowa

Jest to prosta technika realizująca animację, polegająca na cyklicznym wyświetlaniu obrazków zapisanych na dysku, np. a1.jpg, a2.jpg, ..., a8.jpg



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

public clas football extends Applet
implements Runnable{

 Image A[] = new Image[8];
 Image image;
 Thread thread;

 public void init()
 {
   for (int i=0; i<8; i++)
    A[i]=getImage(getDocumentBase(),"a"+(i+1)+".jpg");
 }
```

Animacja poklatkowa

```
public void start() {
  thread = new Thread(this);
  thread.start();
}

public void run() {
  int i=0;
```