АПЛЕТИ

гл.ас. д-р Мария Евтимова

https://github.com/marias83837/JavaPresentations

Аплет

- малки програми работещи в рамките на уеб браузър
- интегрират се в HTML
- не изискват инсталиране освен на java plug-ins

Пример

```
import javax.swing.*;
import java.awt.*;
public class AppletExample extends JApplet{
public void init(){
getContentPane().add(new JLabel(“Hello Students”));
}
}
```

Пример за десктоп приложение

```
public class Applet1c extends JApplet {
  public void init() {
    getContentPane().add(new JLabel("Applet!"));
  }
  // main() за приложението:
  public static void main(String[] args) {
    JApplet applet = new Applet1c();
    JFrame frame = new JFrame("Applet1c");
    // За затваряне на приложението:
    Console.setupClosing(frame);
    frame.getContentPane().add(applet);
    frame.setSize(100,50);
    applet.init();
    applet.start();
    frame.setVisible(true);
  }
} ///:~
```

Аплета може да се реализира, като наследник на класа JApplet

Жизнен цикъл на аплета

- init()- инициализиране на аплета
- start()- започва изпълнението на аплета
- paint()- за създаване на GUI
- stop()- аплета спира да работи
- destroy()- унищожава аплета

Клас- javax.swing.Japplet

- **public Japplet()-** конструира нов екземпляр на аплета
- **public Container getContentPane()-** връща съдържанието на аплета
- **public void setContentPane(Container contentPane)–** установява съдържанието на аплета
- **public Component add(Component comp)-** добавя компонентата comp към аплета
- **public void paint(Graphics gwin)-** изчертава аплета с графично съдържание gwin

- **public void repaint()**

пречертава текущия аплет

- **public Image getImage(URL url, String name)**

възстановява изображение от определения URL

- **public URL getCodeBase()**

връща URL, на който се намира байткодът на аплета

- **public void setBackground(Color color)**

установява фоновия цвят

- **public void setSize(int width, int height)**

установява размера на аплета

Клас java.awt.Graphics
съдържа методи за изчертаване на графични примитиви

- **public void setColor(Color color)**
- **public void getColor(Color color)**

установява/връща цвета color на графичното съдържание

- **public void drawString(String str, int x,int y)**

изчертава низа **str** от точката **(x,y)** надясно

- **public void drawLine(int x1, int y1, int x2, int y2)**

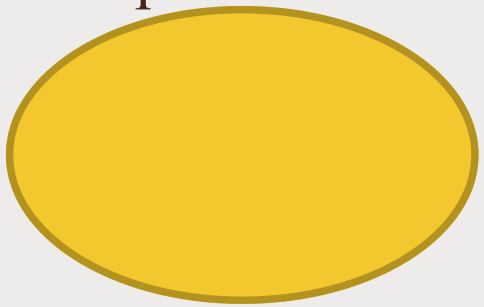
Изчертава линия между точките **(x1,y1)** и **(x2,y2)**

- **public void drawRect(int x, int y, int width, int height)**
- **public void fillRect(int x, int y, int width, int height)**

изчертава/запълва правоъгълник с горен ляв ъгъл **(x,y)** и размери **width** и **height**

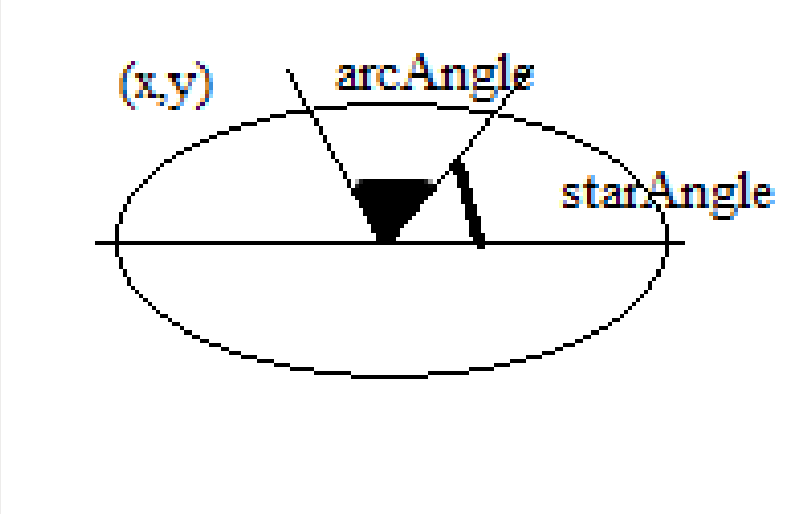
- **public void drawOval(int x,int y, int width, int height)**
- **public void fillOval(int x, int y, int width, int height)**

изчертава / елипса, вписана в правоъгълник с горен ляв ъгъл (x,y) и размери width и height



- **void drawArc(int x, int y, int width, int height, int startAngle, int arcAngle)**
- **void fillArc(int x, int y, int width, int height, int startAngle, int ArcAngle)**

Изчертава/запълва със съответния цвят дъга/сектор от елипса, ограничена от правоъгълника с горен ляв ъгъл (x,y) и размери width и height. Дъгата/секторът започва от ъгъл startAngle и завършва на разстояние, дефинирано от arcAngle.



public void clearRect(int x, int y, int width, int height)

Изчиства зададения правоъгълник, като го запълва с фоновия цвят на текущата повърхност за изчертаване.

ИЗПЪЛНЕНИЕ НА АПЛЕТ

1.Редактиране- име_на_клас.java

2.Компилация javac име_на_клас.java

3.Вграждане на аплета в HTML документ

Името на файла с разширение .html не е необходимо да съвпада с името на класа

```
<HTML>
<HEAD>
<TITLE>Аплет</TITLE>
</HEAD>
<BODY>
<APPLET CODE="име_на_клас.class" WIDTH= 100 HEIGHT=50>
</APPLET>
</BODY>
</HTML>
```

4.Стартиране на аплет

- в браузър

File/OpenFile/**име_на_клас.html**

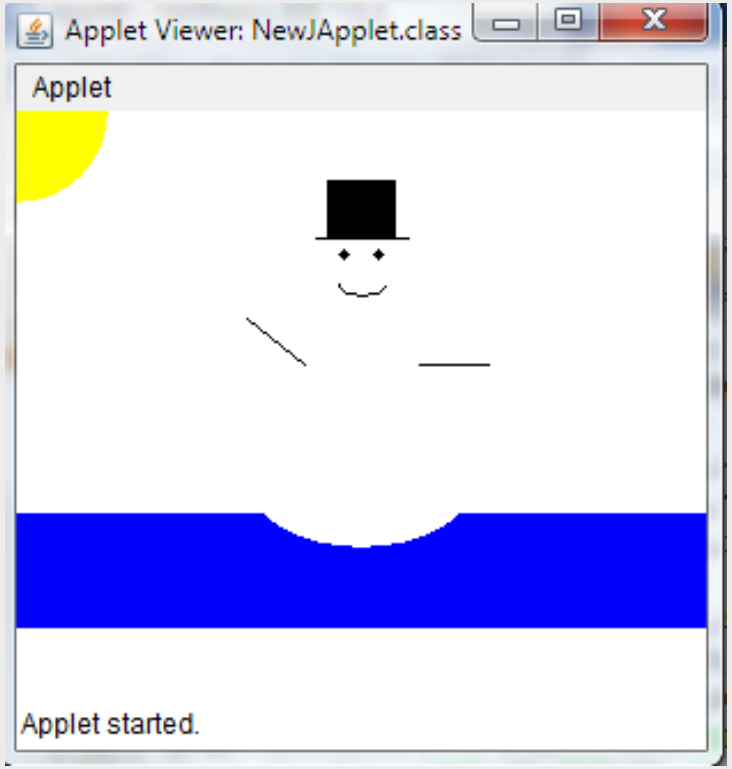
- с AppletViewer

Appletviewer име_на_клас.html

Клас java.awt.Color
за представяне на цветовете.
цвят- червен, зелен, син 0÷255

Цвят	Обект	RGB стойност
черен	Color.BLACK	0,0,0
син	Color.BLUE	0,0,255
синьозелен	Color.CYAN	0,255,255
сив	Color.GRAY	128,128,128
тъмно сив	Color.DARK_GRAY	64,64,64
светло сив	Color.LIGHT_GRAY	192,192,192
зелен	Color.GREEN	0,255,0
пурпурен	Color.MAGENTA	255,0,255
оранжев	Color.ORANGE	255,200,0
розов	Color.PINK	255,175,175
червен	Color.RED	255,0,0
бял	Color.WHITE	255,255,255
жълт	Color.YELLOW	255,255,0

Пример за аплет за снежен човек



```
import javax.swing.JApplet;
import java.awt.*;
import javax.swing.*;
public class NewJApplet extends JApplet {
private final int MID =150;
private final int TOP=50;
public void init(){
    setSize(300,255);//размер
    setBackground(Color.CYAN);//фон}
public void paint(Graphics page){
  page.setColor(Color.BLUE);
  page.fillRect(0, 175,300, 50);//земя
  page.setColor(Color.YELLOW);
  page.fillOval(-40, -40, 80,80);//слънце
  page.setColor(Color.WHITE);
   page.fillOval(MID-20,TOP,40,40);//глава
   page.fillOval(MID-35,TOP+35,70,50);//горен торс
   page.fillOval(MID-50,TOP+80,100,60);//долен торс
   page.setColor(Color.BLACK);
   page.fillOval(MID-10,TOP+10,5,5);//ляво око
   page.fillOval(MID+5,TOP+10,5,5);//дясно око
page.drawArc(MID-10, TOP+20, 20, 10, 190,160);//уста
page.drawLine(MID-25,TOP+60,MID-50,TOP+40);
//лява ръка
page.drawLine(MID+25,TOP+60,MID+55,TOP+60);
//дясна ръка
page.drawLine(MID-
20,TOP+5,MID+20,TOP+5);//периферия
page.fillRect(MID-15,TOP-20,30,25);//шапка}}
```

Въвеждане на примера в HTML код

```
<HTML>
<HEAD>
<TITLE>Снежен човек</TITLE>
</HEAD>
<BODY>
<APPLET CODE="Snowman.class" WIDTH=300 HEIGHT=225>
</APPLET>
</BODY>
</HTML>
```

Събития и слушатели

- Listeners
- Event Handlers

Събития

Събитие е обект, който представлява интересна случка за потребителя.

Примери:

- натискане на бутон на мишката;
- натискане на клавиш от клавиатурата;
- натискане на графичен бутон или плъзгач, генерирани от GUI

Програмата трябва да разпознае и да отговори на събитията.

Клас java.util.EventObject
супер клас, чиито наследници представят събития.

- public Object getSource()

Връща обекта- източник на събитието

Слушатели

Слушател е обект, който чака да се случи събитие и да генерира подходящ отговор.

1.Създаване на обект- слушател за даденото събитие

- Чрез слушателски интерфейс- предефинират се всички методи на интерфейса
- Чрез вътрешен адаптерен клас, реализиращ слушателския интерфейс- предефинират се само необходимите методи на адаптерния клас
- Чрез анонимен вътрешен клас, наследяващ адаптерен клас

II. Добавяне на слушателя към графичната компонента, която може да генерира събитието

Когато се случи събитието, автоматично се извиква подходящият метод на слушателя.

Методът получава като параметър обект, представящ събитието

Listener	Тип на събитието	Пример
`ActionListener`	Action events	Button clicks
`AdjustmentListener`	Adjustment events	Scroll bar moves
`ChangeListener`	Change events	Slider is repositioned
`FocusListener`	Keyboard focus events	Text field gains or loses focus
`ItemListener`	Item events	Check box changes status
`KeyListener`	Keyboard events	Text is entered
`MouseListener`	Mouse events	Mouse clicks
`MouseMotionListener`	Mouse movement events	Mouse rolls
`WindowListener`	Window events	Window closes

Събития, генерирани при използване на мишка- клас **MouseEvent** (java.awt.event)

Методи на класа MouseEvent:

point getPoint()

Връща координатите на мястото, в което се е случило събитието с мишката.

int getX()

Int getY()

Връща координатите на мястото, в което се е случило събитието с мишката.

int getClickCount()

Връща броя на бързите последователни натискания, представени със събитието с мишка

1.**Събития при натискане бутон на мишка**

Слушателски интерфейс

java.awt.event.MouseListener

Адаптерен клас MouseAdapter (java.awt.event)- **реализира** MouseListener

public void addMouseListener(MouseListener I)

Добавя слушател на събития с мишка към компонентата.

- **натиснат бутон на мишка надолу;**

  void mousePressed (MouseEvent event)

- **освободен бутон на мишка;**

  void mouseReleased (MouseEvent event)

- натиснат и освободен бутон на мишката без преместване;

  void mouseClicked (MouseEvent event)

- преместване на указателя на мишката над компонента;

  void mouseEntered (MouseEvent event)

- преместване на указателя на мишката извън компонентата

  void mouseExcited (MouseEvent event)

2.Събития с движение на мишка

Слушателски интерфейс java.awt.event.MouseMotionListener

Адаптерен клас MouseMotionAdapter (java.awt.event), който реализира MouseMotionListener

public void addMouseMotionListener (MouseMotionListener l)

добавя подходящ слушател на събития с движения на мишка към компонентата

- преместване на мишка над компонента;

  void mouseMoved (MouseEvent event)

- влачене на мишка над компонента

  void mouseDragged (MouseEvent event)

II. Събития от клавиатурата- клас java.awt.event.KeyEvent

Метод на класа KeyEvent:

public int getKeyCode()

Връща кода на натиснатия клавиш.

Слушателски интерфейс java.awt.event.KeyListener

Адаптерен клас java.awt.event.KeyAdapter, който реализира KeyListener

public void addKeyListener (KeyListener I)

Добавя слушател на събития от клавиатурата

1. Натиснат клавиш надолу- KEY_PRESSED:

- VK_LEFT- стрелка наляво
- VK_RIGHT- стрелка надясно
- VK_UP- стрелка нагоре
- VK_DOWN- стрелка надолу

void keyPressed (KeyEvent I)

2. Освободен клавиш- KEY_RELEASED

void keyReleased (KeyEvent I)

3. Натиснат символ- натиснат клавиш или клавишна комбинация, произвеждаща символ KEY_TYPED

void keyTyped (KeyEvent I)

public class <име_на_клас> extends JApplet implements **MouseListener**{

….

<някакъв_обект>.addMouseListener(this);

//аплетът е слушател на събитието

….

public void mouseClicked (MouseEvent e){

//реализира слушател на събитието

….

}

//Празни дефиниции на методи

public void mousePressed (MouseEvent e){}

public void mouseReleased (MouseEvent e){}

public void mouseEntered (MouseEvent e){}

public void mouseExited (MouseEvent e){}

Вътрешен клас наследяващ адаптерния клас

```
public class<име_на_клас>extends Japplet{
….
<някакъв_обект>.addMouseListener (new MyAdapter());
….
//вътрешен клас
class MyAdapter extends MouseAdapter{
public void mouseClicked (MouseEvent e){
//реализира слушател на събитието
….
}}}
```

Чрез анонимен вътрешен клас, наследяващ адаптерния клас

```
public class <име_на_клас>
extends JApplet{

….

<някакъв
обект>.addMouseListener(new
MouseAdapter(){

public void
mouseClicked(MouseEvent e){
// реализира слушател на
събитието

…..

}

});

…

}
```

Пример:
Аплет, който изчертава синя точка при натискане бутона на мишката (събитие “натиснат бутон на мишката надолу” MouseEvent чрез анонимен клас, който реализира метода mousePressed на адаптерния клас MouseAdapter).
Клас Point (java.awt)
Представя точка с координати (x,y)

```
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
 public class Dot extends JApplet{
private Point clickPoint=null;
private static final int RADIUS=6;
public void init(){
addMouseListener(new
MouseAdapter(){
public void mousePressed (MouseEvent
event){
clickPoint=event.getPoint();//точка, в
която е натисната мишката
repaint();
}});
setBackground(Color.BLACK);
}
public void paint(Graphics page){
page.clearRect(0,0,this.getWidth(),this.ge
tHeight());
page.setColor(Color.blue);
if(clickPoint!= null)//запълва кръга
page.fillOval
(clickPoint.x-RADIUS, clickPoint.y-
RADIUS,RADIUS*2,RADIUS*2);
}}
```

Пример за аплет, който изчертава зелена гъвкава линия чрез натискане и влачене бутона на мишката. Началната точка се определя от събитието “натискане бутона на мишката надолу” **MouseEvent** (чрез анонимен клас, който реализира метода **mousePressed** на адаптерния клас **MouseAdapter**).
Крайната точка се определя от събитието “влачене на мишката” **MouseEvent** (чрез анонимен клас, който реализира метода **mouseDragged** на адаптерния клас **MouseMotionAdapter**), пречертава се всеки път и линията се разтяга

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
public class Line extends JApplet{
private Point point1=null; //начална точка
private Point point2=null;// крайна точка
private static final int APPLET_WIDTH=200;
//широчина на аплета
private static final int APPLET_HEIGHT=200;
//височина на аплета
public void init (){
setBackground(Color.BLACK) ;
//фонов цвят
setSize(APPLET_WIDTH,APPLET_HEIGHT);
//размер на аплета
addMouseListener(new MouseAdapter(){
public void mousePressed(MouseEvent event){
point1=event.getPoint();//точка, в която е натисната мишката
}
});
```

```
addMouseMotionListener(new MouseMotionAdapter(){
public void mouseDragged(mouseEvent event){
point2=event.getPoint();//точка при влачене на мишката
repaint();
}
});
}
public void paint(Graphics page){
page.clearRect(0,0,this.getWidth(),this.getHeight());
page.setColor(Color.BLUE);
if(point1!=null && point2!=null) //изчертава линия
page.drawLine(point1.x,point1.y,point2.x,point2.y);
}}
```