

Notiuni teoretice

Fișierul .class trebuie să se afle în același director cu fișierul HTML, altfel nu este suficient să se scrie doar numele său la atributul CODE. Clasa principală a unui *applet* trebuie neapărat să extindă clasa predefinită Applet. Metoda paint() primește ca parametru o referință a unui obiect de tip Graphics (clasă predefinită în java.awt), creat automat în momentul lansării în execuție a *applet*-ului.

Un obiect Graphics poate realiza operații de desenare într-o porțiune a ecranului. După ce se operează modificări într-un *applet*, el trebuie recompilat, rezultând un fișier .class modificat. Pentru ca în pagina web, care include *applet*-ul respectiv, să fie vizibile modificările, este necesară acționarea butonului *Reload* al *browser*-ului în timp ce se ține apăsată tasta *Shift*.

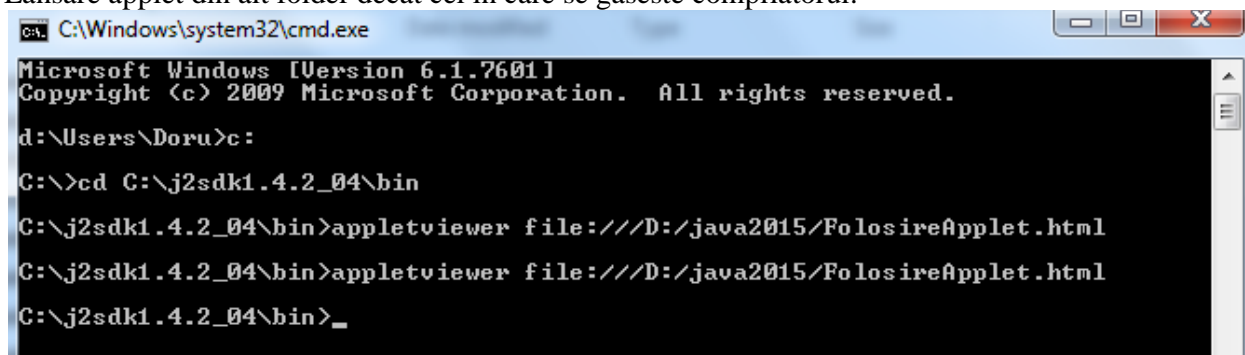
Folosind clasele Font și Color din pachetul java.awt, se poate face *applet*-ul mai atractiv. Cu ajutorul obiectelor Color se pot stabili culori pentru umplerea diverselor forme geometrice sau pentru text. O culoare este specificată cu ajutorul a trei numere întregi din intervalul [0-255] care indică nivelul de roșu, verde și albastru (RGB) prin combinația cărora va rezulta culoarea dorită. Spre exemplu, pentru a scrie textul cu violet, *applet*-ul va deveni:

```
import java.awt.*;
import java.applet.*;
public class PrimulApplet extends Applet{
public void paint (Graphics g){
Color c=new Color(180,10,120);
g.setColor(c);
g.drawString("Hello World", 20, 20);
}
}
```

Există un set de obiecte de tip Color definite ca membri statici ai clasei Color:

Color.black (negru), Color.blue (albastru), Color.darkGray (gri închis), Color.gray (gri), Color.green (verde), Color.red (roșu), Color.white (alb), Color.yellow (galben) etc.

Lansare applet din alt folder decat cel in care se gaseste compilatorul:



```
C:\Windows\system32\cmd.exe
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

d:\Users\Doru>c:
C:\>cd C:\j2sdk1.4.2_04\bin
C:\j2sdk1.4.2_04\bin>appletviewer file:///D:/java2015/FolosireApplet.html
C:\j2sdk1.4.2_04\bin>appletviewer file:///D:/java2015/FolosireApplet.html
C:\j2sdk1.4.2_04\bin>_
```

Pentru specificarea unui anumit font pentru caracterele textului se utilizează clasa Font cu precizarea a trei parametri:

- tipul fontului ("TimesRoman");
- stilul, sub forma unor constante predefinite: Font.PLAIN (caractere normale), Font.BOLD (caractere îngroșate) sau Font.ITALIC (caractere înclinate).

Pentru combinarea stilurilor se utilizează operatorul sau „|”;
— dimensiunea caracterelor, exprimată ca număr de puncte: 10, 12, 14, 18 etc.

Pentru a folosi fonturi *Times New Roman*, bold și cu dimensiunea de 18 puncte, exemplul se modifică astfel:

```
import java.awt.*;
import java.applet.*;
public class PrimulApplet extends Applet{
    public void paint (Graphics g){
        Color c=new Color(180,10,120);
        g.setColor(c);
        Font f=new Font("TimesRoman", Font.BOLD, 18);
        g.drawString("Hello World", 20, 20);
    }
}
```

Dacă într-un *applet* se execută operațiile `setColor` sau `setFont`, valorile stabilite de către acestea pentru culoare și font vor rămâne active până la sfârșitul programului sau până la o nouă modificare. Pentru a obține informații despre culoarea/fontul curente se pot folosi următoarele metode:

- `getColor` returnează un obiect de tip `Color` reprezentând culoarea curentă. Pentru a afla componentele RGB ale culorii se vor aplica metodele `getRed`, `getGreen` și `getBlue`.
- `getFont` care returnează referința unui obiect `Font`, reprezentând fontul curent.

Pe lângă afișarea de text, clasa `Graphics` include și metode de desenare a unor forme geometrice:

- `drawRect(int x, int y, int width, int height)`; desenează conturul unui dreptunghi cu colțul din stânga sus în punctul de coordonate (x,y), lungimea egală cu `width` și înălțimea egală cu `height`. Culoarea utilizată pentru contur este culoarea curentă;
- `drawLine(int x1, int y1, int x2, int y2)`; desenează un segment de dreaptă ale cărui capete au coordonatele (x1, y1), respectiv, (x2, y2). Culoarea utilizată pentru linie este culoarea curentă;
- `drawOval(int x, int y, int width, int height)`; desenează conturul unei elipse având cele două diametre principale paralele cu laturile *applet*-ului. (x, y) reprezintă coordonatele colțului din stânga sus ale dreptunghiului imaginar care circumscrie elipsa, iar `width` și `height` reprezintă lățimea, respectiv înălțimea elipsei. Culoarea utilizată pentru contur este culoarea curentă;
- `fillRect(int x, int y, int width, int height)`; colorează interiorul unui dreptunghi cu culoarea curentă. Semnificația parametrilor este aceeași ca și la `drawRect`;
- `fillOval(int x, int y, int width, int height)`; colorează interiorul unei elipse cu culoarea curentă. Semnificația parametrilor este aceeași ca și la `drawOval`.

a. Crearea applet-urilor ce includ desene

1) Afișarea unei imagini și a unui text încercuit.

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

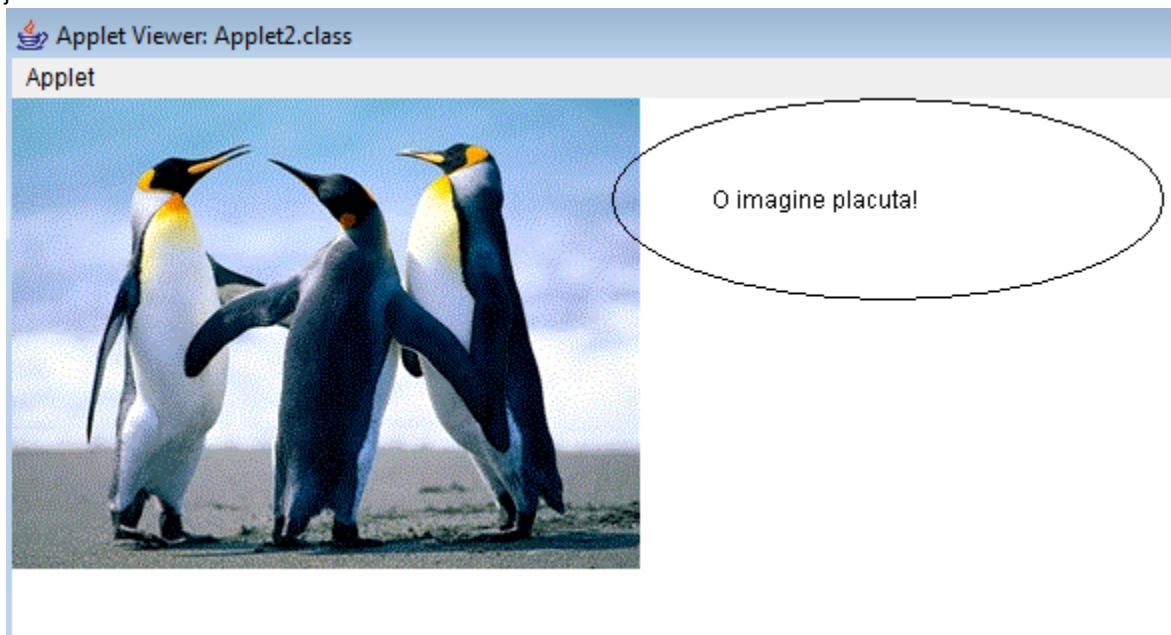
public class Applet2 extends Applet {
```

```

Image img;
public void init() {
    img = getImage(getCodeBase(), "Penguins.gif");
}

public void paint (Graphics g) { //g=suprafata pe care se poate desena
    g.drawImage(img, 0, 0, this);
    g.drawOval(300,0,275,100);
    g.drawString("O imagine placuta!", 350, 55);
}
}

```



```

<HTML>
<HEAD>
<TITLE> Un applet simplu </TITLE>
</HEAD>
<APPLET CODE="Applet2.class" WIDTH=400 HEIGHT=350></APPLET>
</HTML>

```

b. Fire de execuție in applet-uri cu desene animate

Aplicatia Minge animata

Applet-ul urmator foloseste fire de executie si deseneaza o minge si o deplaseaza in interiorul unui dreptunghi ca in figura de mai jos.

AppletMinge.java

```
import java.applet.Applet;

import java.awt.Color;
import java.awt.Graphics;

public class AppletMinge extends Applet implements Runnable{

    private int x=7, xChange=7;
    private int y=2, yChange=2;
    private int diameter=20;
    private int rectLeftX=0, rectRightX=200;
    private int rectTopY=0, rectBottomY=200;
    private boolean ok;

    public void intt(){
        ok=true;
    }

    public void paint(Graphics g){
        super.paint(g);
        g.drawRect(rectLeftX, rectTopY, rectRightX-rectLeftX+10, rectBottomY-rectTopY+10);

        /*desenam cu culoarea fundalului pentru
        a acoperi mingea anterior desenata*/
        g.setColor(getBackground());

        //se sterge mingea
        g.fillOval(x, y, diameter, diameter);

        //daca se intalneste peretele, se schimba directia de miscare
        if(x+xChange<=rectLeftX)
            xChange=-xChange;
        if(x+xChange>=rectRightX)
            xChange=-xChange;
        if(y+yChange<=rectTopY)
            yChange=-yChange;
        if(y+yChange>=rectBottomY)
            yChange=-yChange;

        //se modifica noua pozitie a mingii
        x=x+xChange;
        y=y+yChange;

        //se stabileste culoarea rosie pentru desenare
        g.setColor(Color.red);
    }
}
```

```

//se deseneaza mingea
g.fillOval(x, y, diameter, diameter);
}

/*se apleleaza la distrugerea appletului
si are ca efect terminarea metodei paint ()*/
public void destroy(){
    ok=false;
}
public void start(){
    ok=true;
    new Thread(this).start();
}
public void stop(){
    ok=false;
}
public void run(){
    while (ok){
        repaint();
        try{
            Thread.sleep(100);
        }catch (InterruptedException e){}
    }
}
}

```

```

}

```

FolosireAppletMinge.http

```

<html>
<head>
<title>Minge animata</title>
</head>
<body>
<applet code=AppletMinge.class width=400 height=400>
</applet>
</body>
</html>

```

