

**GDI/GDI+**

Maja Štula

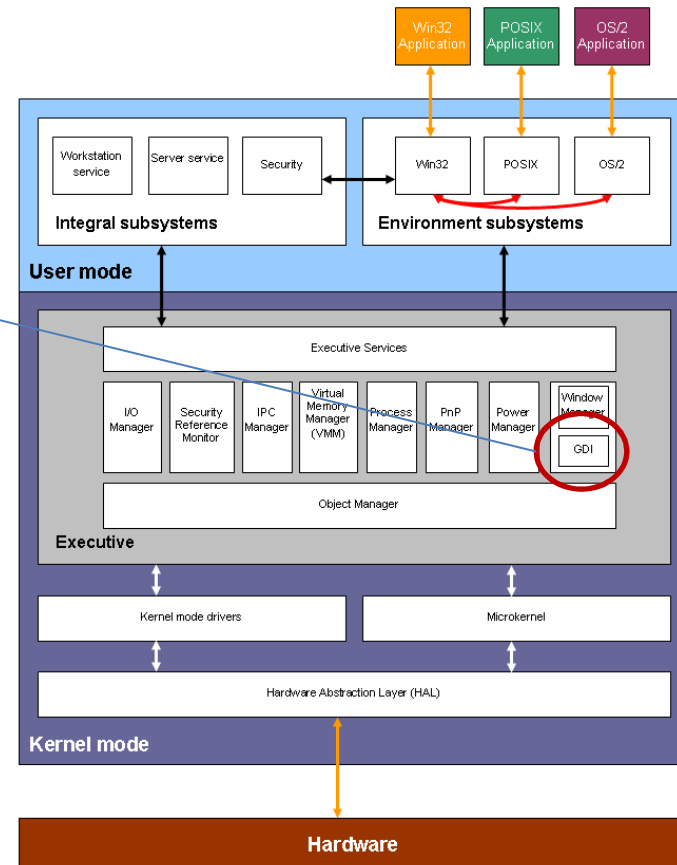
Ak. God. 2011/2012

# GDI

- Dio operacijskog sustava koji upravlja prikazom djela GUI sučelja koji nisu prozor (sadržaj u prozoru - tekst i grafika) se naziva GDI (*Graphics Device Interface*) .
- DC (*Device Context*) je struktura koju GDI koristi da bi pohranio informacije o grafičkim objektima koji upravljaju izgledom teksta i grafike na monitoru ili na printeru. Tip olovke, slova, boje ili bitmapa koja će se koristiti za prikaz na određenom uređaju je pohranjena u DC.

# GDI

- GDI je modul operacijskog sustava koji omogućava iscrtavanje sadržaja unutar prozora.
- Prozore iscrtava modul Window Manager, a sadržaj unutar prozora GDI modul.



# DirectX

- Pored GDI mehanizma za iscrtavanje sadržaja unutar prozora se može koristiti i DirectX mehanizam. On je potpuno odvojen, ima zasebni SDK-a i nije uvijek bio standardni dio Windowsa.



# GDI

- Svaki put kada se prozor pomakne, kada se maksimizira, kada ga prekrije neki drugi prozor itd. prozor se ponovo iscrtava na način da windowsi šalju prozoru poruku WM\_PAINT.
- Klijentsko područje prozora se defaultno iscrtava s predefiniranom pozadinom prozora.
- Ako želimo dodatno iscrtavanje u klijentskom području (npr. neki tekst) moramo obraditi poruku WM\_PAINT u kodu aplikacije i ako želimo zadržati iscrtani sadržaj dodati ponovo iscrtavanje sadržaja u prozoru.



# GDI

- Svi pozivi funkcija za crtanje u GDI-u se vrše preko DC objekta.
- DC nije ovisan o uređaju na koji se crta. Koristi se za crtanje na ekran, na printer ili u metafile (*Metafile* je skup struktura u koje je pohranjena slika neovisno o formatu uređaja –za razliku od bitmape).
- Windowsi podržavaju čitav niz alata za crtanje koje se mogu koristiti sa DC-om. To su olovke (*pen*) za crtanje linija, kistovi (*brushes*) za punjenje oblika i slova (*fonts*) za crtanje teksta.

# GDI+

- GDI+ je novija, poboljšana verzija GDI-a koja dolazi uključena u Windows XP i Server 2003 (gdiplus.dll). Može se instalirati naknadno i na neke starije verzije. GDI+ je objektno orijentiran (*class-based*) API za C/C++ programere (MSDN) za razvoj Windows desktop aplikacija.
- + znači da ta novija i poboljšana verzija uključuje određene funkcionalnosti koje ne postoje kod GDI-a.



# GDI+

- Funkcionalnost GDI+ je podijeljena u tri osnovna dijela:
  - Vektorska grafika - obuhvaća crtanje i manipuliranje geometrijskim i sličnim prikazima (linije, kombinacije linija, zaobljeni i četverostrani oblici,...) kao skupinom točaka na ekranu (ili drugom uređaju za crtanje). GDI+ je objektno orijentiran (što nije prepreka korištenju u WIN32 API aplikaciji) tj. sadrži različite klase koje pružaju potrebnu funkcionalnost.
  - Oslikavanje (*Imaging*) – obuhvaća rad sa složenim slikama koje je teško predstavljati vektorskom grafikom.
  - Tipografija (*Typography*) - obuhvaća kreiranje, manipuliranje i općenito rad sa slovima aplikacije.

# GDI+

- Kod rada sa GDI-jem najprije je potrebno dohvatiti handle na DC (npr. funkcijom GetDC).
- Svi pozivi funkcija za crtanje idu preko handle na DC uz koji su vezani alati potrebni za crtanje poput olovke (pen), fonta, itd.
- Kod GDI+ za crtanje se koristi klasa koja se zove *Graphics*.

# GDI+ klasa *Graphics*

- Graphics klasa sadrži različite metode potrebne za crtanje linija, likova, slika i teksta.
- Objekt tipa klase *Graphics* sadrži attribute uređaja za prikaz na kojem se radi crtanje (monitor, printer, ...) i attribute elemenata koji se crtaju.
- Klasa *Graphics* nasljeđuje klasu *GdiplusBase*.

# GDI+ C++ konstruktori klase

## *Graphics*

- Konstruktori klase:
- `Graphics::Graphics (Image*)` – Kreira *Graphics* objekt povezan s objektom *Image*.
- `Graphics::Graphics (HDC)` – Kreira *Graphics* objekt vezan uz određeni DC.
- `Graphics::Graphics (HDC, HANDLE)` - Kreira *Graphics* objekt vezan uz određeni DC i određeni uređaj.
- `Graphics::Graphics (HWND, BOOL)` - Kreira *Graphics* objekt vezan uz određeni prozor.

# GDI+ C#

- U .NET-u pristup osnovnim GDI+ funkcionalnostima je moguć preko osnovnog imenskog prostora `System.Drawing`.
- Naprednije funkcionalnosti (gradijentni kist i sl.) nalaze se u imenskim prostorima `System.Drawing.Drawing2D`, `System.Drawing.Imaging` i `System.Drawing.Text`.

# GDI+ C#

- Klasa Graphics sadrži osnovne metode za crtanje na neki uređaj za prikaz.
- Kada imamo Graphics objekt na njemu pozivamo različite metode kojima iscrtavamo sadržaj.
- Uz Graphics objekt je vezan objekt tipa Pen za crtanje ravnih i zakrivljenih linija, tipa Brush (tj. klase koje su naslijeđene iz osnovne klase Brush za ispunjavanje) za crtanje zatvorenih oblika s bojom, tipa Font za ispis teksta....
- Često se koriste i klase Rectangle i Point.

# GDI+ C#

- Klasa Graphics je povezana s određenim DC-om.
- Pristup instanci klase može se napraviti na više načina:
  1. Pozivom CreateGraphics metode u kontrolama ili na formi.
  2. Preko događaja Paint i argumenta event handlera tipa klase PaintEventArgs koja sadrži Graphics svojstva.
  3. Kreiranjem objekta Graphics iz postojećeg objekta koji nasljeđuje klasu Image.

# GDI+ C#

1. Pozivom CreateGraphics metode u kontrolama ili na formi.

```
Graphics g;
```

```
g = this.CreateGraphics();
```

- this treba biti forma ili kontrola!!



# GDI+ C#

## 2. Preko događaja Paint

```
private void Form1_Paint(object sender,  
    System.Windows.Forms.PaintEventArgs pe)  
{ Graphics g = pe.Graphics; }
```

## 3. Kreiranjem objekta Graphics iz postojećeg objekta koji nasljeđuje klasu Image

```
Bitmap myBitmap = new Bitmap(@"C:\Documents and  
    Settings\Joe\Pics\myPic.bmp");  
Graphics g = Graphics.FromImage(myBitmap);
```

# GDI+ C#

- Primjer iscrtavanja teksta i pravokutnika na formi:

```
//Kreiranje Graphics objekta pozivom CreateGraphics  
metode na formi  
Graphics forma = this.CreateGraphics();  
//Kreiranje Font objekta za ispis teksta  
Font slova = new Font("Arial", 14, FontStyle.Italic);  
//Ispis teksta sa definiranim fontom i predefiniranim kistovima  
forma.DrawString("tekst", slova, Brushes.Red, 30, 30);  
//Iscrtavanje pravokutnika sa predefiniranim olovkama  
forma.DrawRectangle(Pens.Cyan, 160, 160, 120, 180);
```

# GDI+

- Nove funkcionalnosti u odnosu na GDI su:
  - Gradijentni kist (*Gradient Brushes*)
  - Osnovni odlomci (*Cardinal Splines*)
  - Neovisni grafički objekti (*Independent Path Objects*)
  - Transformacije i Matrica objekt (*Transformations and the Matrix Object*)
  - Skalabilna područja (*Scalable Regions*)
  - Alfa stapanje (*Alpha Blending*) (AlphaBlend WIN32 API)
  - Podrška za različite formate slike (*Support for Multiple Image Formats*)

# Gradijentni kist

- Gradijenti kist se koristi za crtanje linija, ispunjavanje oblika i područja itd. Kod crtanja gradijentnim kistom boja se postepeno mijenja. Gradijent promjene može biti linearni (dijagonalno, gore, dolje, lijevo, desno) ili definiran grafičkim objektima (*path gradient*).



# Gradijentni kist

//Kreiranje Graphics objekta pozivom CreateGraphics  
metode na formi

Graphics forma = this.CreateGraphics();

//Kreiranje linearnog gradijentnog kista

LinearGradientBrush dva = new

LinearGradientBrush(new Point(50, 150), new

Point(150, 450), Color.Red, Color.Purple);

//Is crtavanje pravokutnika sa kreiranim kistom

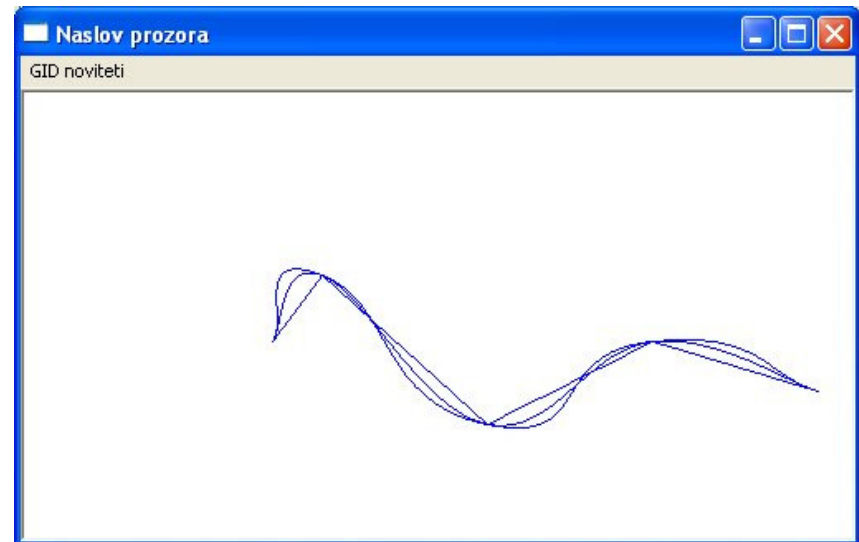
forma.FillRectangle(dva, new Rectangle(350, 150, 250, 80));

# Gradijentni kist

```
//Kreiranje Graphics objekta pozivom CreateGraphics metode na formi
Graphics forma = this.CreateGraphics();
//Kreiranje GraphicsPath (putanja) objekta
GraphicsPath path = new GraphicsPath();
//Dodavanje elemenata u putanju
path.AddRectangle(new Rectangle(50, 50, 250, 80));
//Kreiranje gradijentnog kista preko objekta putanje
PathGradientBrush pthGrBrush = new PathGradientBrush(path);
//Definiranje boje u centru kista definiranog preko putanje
pthGrBrush.CenterColor = Color.FromArgb(255, 255, 0, 0);
//Definiranje boje na granicama kista
pthGrBrush.SurroundColors = new Color[] { Color.FromArgb(255, 255, 255, 20) };
//Is crtavanje pravokutnika sa kreiranim kistom
forma.FillRectangle(pthGrBrush, new Rectangle(50, 50, 250, 80));
```

# Osnovni odlomci

- GDI+ podržava osnovne odlomke (*cardinal splines*), koji nisu podržani u GDI-u. Osnovni odlomak je sekvenca pojedinačnih zakrivljenih linija koje zajedno tvore veću zakrivljenu liniju. Odlomak je određen nizom točaka kroz koje prolazi krivulja. Prolazak kroz linije je zaglađen tako da se dobiva prava zakrivljena linija.



# Osnovni odlomci

- Primjer iscrtavanja zakrivljene linije na formi:  
//Kreiranje Graphics objekta pozivom CreateGraphics  
metode na formi  
Graphics forma = this.CreateGraphics();  
//Definiranje točaka kroz koje linija prolazi  
Point[] tocke = { new Point(120, 120), new Point(240,  
240), new Point(300, 340) };  
//Definiranje zakrivljenosti linije  
forma.DrawCurve(Pens.Crimson, tocke, 1.4F);

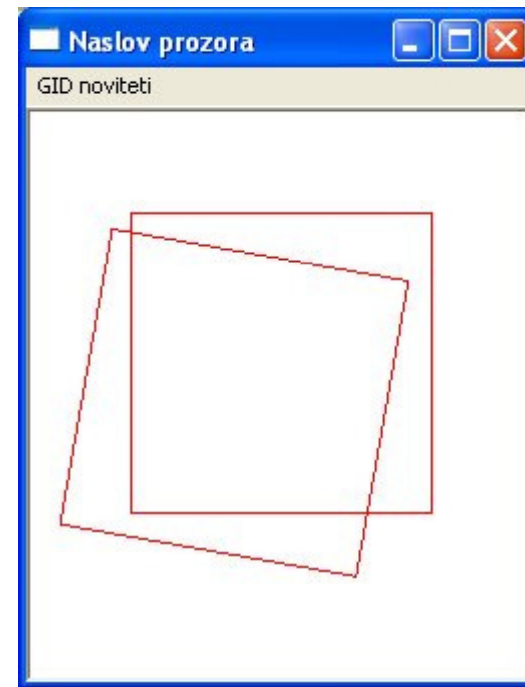


# Neovisni grafički objekti

- Putanja (path) je sekvenca grafičkih primitiva (poput linija, pravokutnika, zakrivljenih linija, teksta i sl.) kojom se može manipulirati i koja se može iscrtavati kao jedan element. Putanja se može podijeliti na likove (*figures*) koji mogu biti otvoreni ili zatvoreni. Lik se može sastojati od nekoliko grafičkih primitiva .
- Kod GDI-a putanja je vezana uz DC i uništava se prilikom iscrtavanja.
- Kod GDI+ crtanje se izvodi preko *Graphics* objekta preko kojeg se može kreirati i održavati nekoliko *GraphicsPath* objekata. Objekt tipa *GraphicsPath* se ne uništava prilikom crtanja i može se koristiti za iscrtavanje putanje više od jednog puta.

# Transformacije i *Matrix* objekt

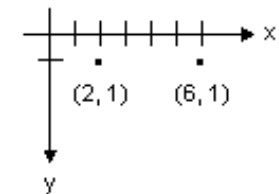
- GDI+ ima objekt *Matrix* koji olakšava transformacije grafičkih elemenata (poput rotacija, translacija, skaliranja i sl.). To je matrica 3x3 dimenzija.
- Jedna matrica 3x3 može sadržavati jednu transformaciju ili sekvencu transformacija.
- Objekt *Matrix* je povezan s grafičkim objektom koji se transformira.
- Na primjer, *GraphicsPath* objekt ima metodu *Transform* koja kao argument ima adresu *Matrix* objekta.



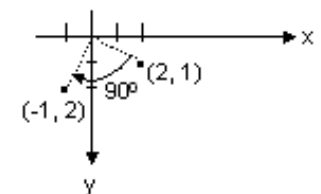
# GDI+ Transformacije i *Matrix* objekt

- Objekt Matrica (*Matrix*) predstavlja matricu veličine  $3 \times 3$  koja predstavlja afinu (linearnu) transformaciju.
- U objektu je pohranjeno samo 6 od 9 brojeva jer sve matrice veličine  $3 \times 3$  koje opisuju affine transformacije imaju istu treću kolonu (0, 0, 1).
- Npr. transformacija poput skaliranja (pomicanja točke) u ravnini može se izvesti množenjem dvodimenzionalnog vektora koji opisuje točku sa matricom veličine  $2 \times 2$ , također i rotacija te refleksija oko jedne osi ravnine.

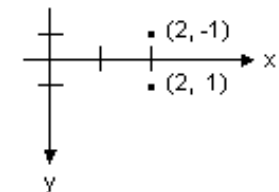
Scale  $\begin{bmatrix} 2 & 1 \end{bmatrix} \begin{bmatrix} 3 & 0 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 6 & 1 \end{bmatrix}$



Rotate 90°  $\begin{bmatrix} 2 & 1 \end{bmatrix} \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} = \begin{bmatrix} -1 & 2 \end{bmatrix}$

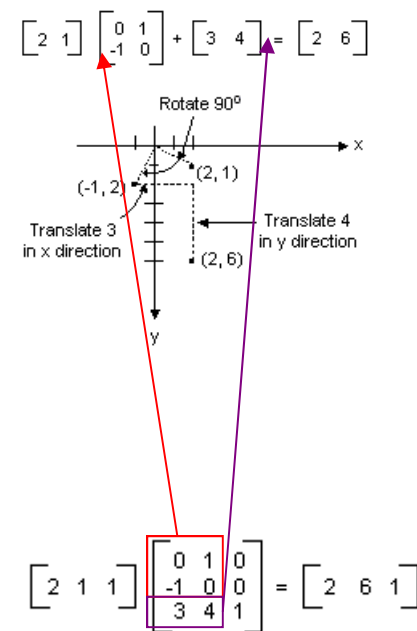


Reflect across x-axis  $\begin{bmatrix} 2 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} = \begin{bmatrix} 2 & -1 \end{bmatrix}$



# GDI+ Transformacije i *Matrix* objekt

- Nelinearne transformacije ne mogu se izraziti na prethodno opisan način. Npr. rotacija sa transformacijom zahtjeva množenje matrica i zbrajanje.
- Ova se operacija može izvesti i množenjem s jednom matricom veličine 3x3 u kojoj treća kolona nema funkciju u transformaciji.



# GDI+ Transformacije i *Matrix* objekt

- Klasa *Matrix* sadrži metode za složene transformacije:
  - Multiply,
  - Rotate,
  - RotateAt,
  - Scale,
  - Shear,
  - Translate.

# GDI+ Transformacije i *Matrix* objekt

## PRIMJER:

```
Graphics g = this.CreateGraphics();
Bitmap moj = new
    Bitmap(@"E:\predavanja\windows\c_sharp_primjeri\gdi_plus\gdi_plus\bin\Debug\wpf101_
    01.gif");
g.DrawImage(moj, 0, 0, moj.Width, moj.Height);
Matrix X = new Matrix();
X.Rotate(45, MatrixOrder.Append);
g.Transform = X;
g.DrawImage(moj, new Rectangle(205, 0, moj.Width, moj.Height), 0, 0, moj.Width, moj.Height,
    GraphicsUnit.Pixel);
Matrix Y = new Matrix();
Y.Shear(0.4F, 0.2F);
g.Transform = Y;
g.DrawImage(moj, new Rectangle(20, 20, moj.Width, moj.Height), 0, 0, moj.Width, moj.Height,
    GraphicsUnit.Pixel);
moj.Dispose();
```

# Skalabilna područja

- Kod GDI-a područja su označavana preko koordinata uređaja na koji se crta i jedina moguća transformacija područja bila je translacija.
- GDI+ područja označava s neovisnim koordinatama te omogućava bilo koju transformaciju područja koja se može pohraniti u transformacijskoj matrici.



# Alfa stapanje

- Kod alfa stapanja moguće je postaviti razinu prozirnosti boje kojom se ispunjava neko područje.
- Prozirna boja se stapa s bojom pozadine. Što je više postavljena razina prozirnosti boja se više stapa s pozadinom i više se pozadine vidi kroz boju.
- Kod GDI+ boja je 32-bitna vrijednost. 8 bita se odnosi na alfa vrijednost, zatim po 8 bita za crvenu, zelenu i plavu boju.
- Alfa vrijednost određuje transparentnost boje, odnosno koliko se promatrana boja stapa s bojom pozadine. Vrijednost 0 znači da je boja potpuno transparentna, a 255 da je boja neprozirna.



# Alfa stapanje

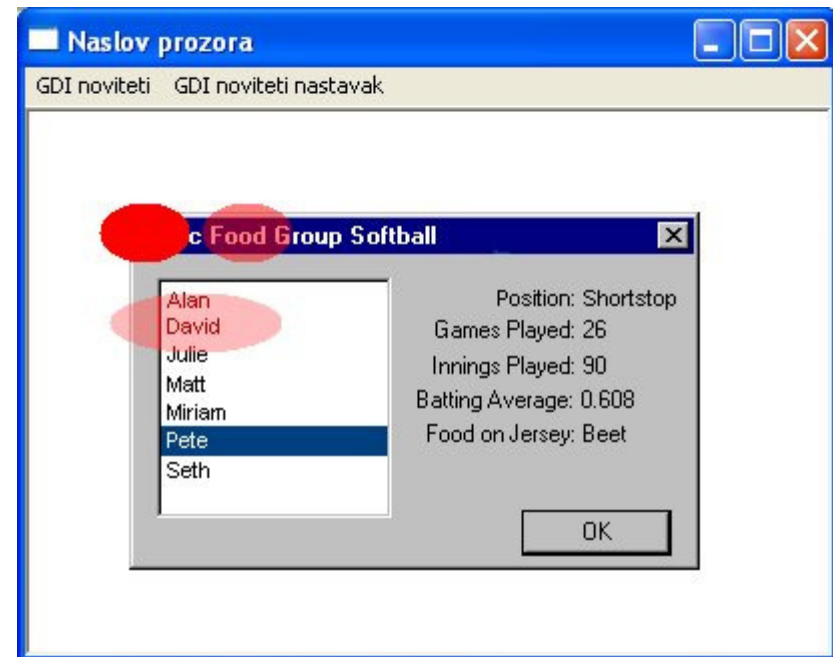
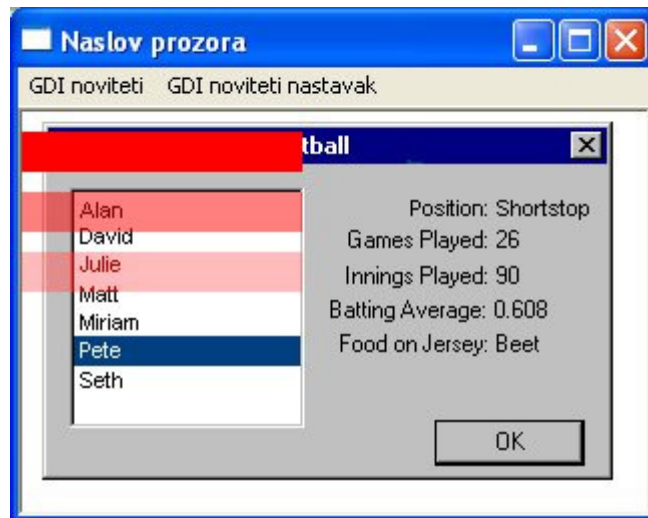
- Alfa stapanje se radi piksel po piksel.
- Svaka od tri komponente boje (crvena, zelena, plava) za određeno područje se stapa s odgovarajućom pozadinskom bojom prema formuli:

$$\text{displayColor} = \text{sourceColor} * \text{alpha} / 255 + \text{backgroundColor} * (255 - \text{alpha}) / 255$$

- Npr. ako je piksel crvene boje s jakošću 150, a komponenta crvene boje u pozadinskom pikselu je 100, alfa vrijednost piksela je 200, crvena boja u rezultirajućoj boji se dobije:

$$150 * 200 / 255 + 100 * (255 - 200) / 255 = 139$$

# Alfa stapanje



# Alfa stapanje

```
Graphics g = this.CreateGraphics();  
Bitmap moj = new Bitmap("slika.jpg");  
g.DrawImage(moj, 0, 0, moj.Width, moj.Height);  
LinearGradientBrush dva = new  
LinearGradientBrush(new Point(0, 0), new  
Point(300, 300), Color.FromArgb(200,Color.Red),  
Color.FromArgb(200,Color.Violet));  
g.FillRectangle(dva, 0, 0, 300, 300);  
moj.Dispose();
```

# Podrška za različite formate slike

- GDI+ sadrži klase Image, Bitmap, i Metafile, koje omogućuju učitavanje, pohranu i manipulaciju slikama različitih formata. Podržani formati su:
  - BMP
  - Graphics Interchange Format (GIF)
  - JPEG
  - Exif
  - PNG
  - TIFF
  - ICON
  - WMF
  - EMF