

# Računarska grafika

(20ER7002)

## Font

## Vežbe



# Termini

- **glyph** – pojedinačni znak
- **font** – kolekcija znakova sa istim dizajnom
- **font family** – skup fontova koji dele isti osnovni dizajn
- **typeface** – dizajn fonta (debljina linija, serif,...)
- **pitch** – širina pojedinačnog znaka
  - **fixed pitch** – svi znaci imaju istu širinu
  - **variable pitch** – neki karakteri su širi a neki už
- **serif** – crtice na krajevima karaktera

Proportional  
Monospaced

AaBbCc Sans serif  
AaBbCc Serif  
AaBbCc

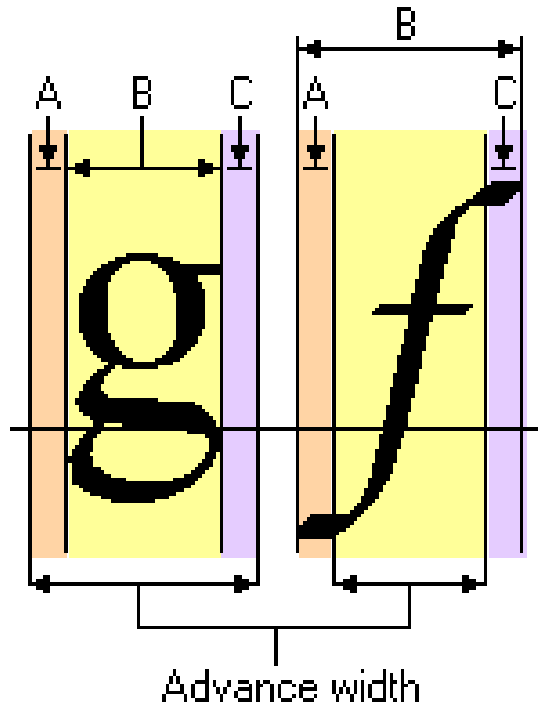
# Tipovi fonta

- **rasterski** – kreiraju se pomoću bitmapa (.FON)
- **vektorski** – definisani primitivama (linijskim segmentima), nezavisni od uređaja, ali se sporo iscrtavaju (.FON)
- **TrueType** – najfleksibilniji, sačinjeni od linija i krivih i dodatnih informacija o svakom znaku (.FON i/ili .TTF)
- **OpenType** – proširenje TrueType fontova tako da podržavaju Adobe PostScript Compact File Format
- **ClearType** – Microsoft implementacija subpikselskog iscrtavanja fontova (prilagođen LCD ekranima)

# Familija fontova

- **Decorative** – novi, ukrasni fontovi.
- **Dontcare** – generički fontovi, ili informacija o familiji ne postoji ili je irelevantna.
- **Modern** – fontovi sa fiksnom širinom (fixed pitch) sa ili bez serif-a, npr. Courier New.
- **Roman** – fontovi promenljive širine sa serif-om, npr. Times New Roman.
- **Script** – fontovi nalik pisanim slovima.
- **Swiss** – fontovi fiksne širine (fixed pitch) bez serif-a, npr. Arial.

# Metrika fonta – širina



- "A" – praznina koja se dodaje pre postavljanja znaka
- "B" - širina samog znaka
- "C" – praznina koja se dodaje iza znaka
- Ukupna širina je  $A+B+C$ .
- Znak može biti širi od njegove ukupne širine
  - A i C mogu biti i negativni brojevi

# Očitavanje ABC parametara fonta

- Metode za očitavanje širine znakova selektovanog fonta

`BOOL CDC::GetCharABCWidths( UINT nFirstChar, UINT nLastChar, LPABC lpabc )`

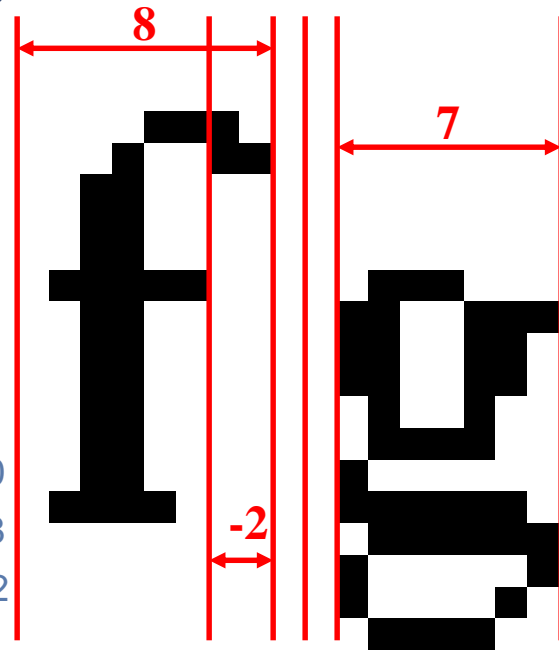
`BOOL CDC::GetCharWidth( UINT nFirstChar, UINT nLastChar, LPINT lpBuffer )`

- nFirstChar*** – kod prvog karaktera iz opsega trenutno selektovanog fonta čije ABC parametre očitavamo
- nLastChar*** – kod poslednjeg karaktera iz opsega
- lpabc*** – pointer na niz ABC struktura, koji mora imati bar onoliko elemenata koliko je specificirano opsegom (***nLastChar - nFirstChar + 1***)
- lpBuffer*** – pointer na niz intedžera, koji mora imati bar onoliko elemenata koliko je specificirano opsegom (***nLastChar - nFirstChar + 1***)

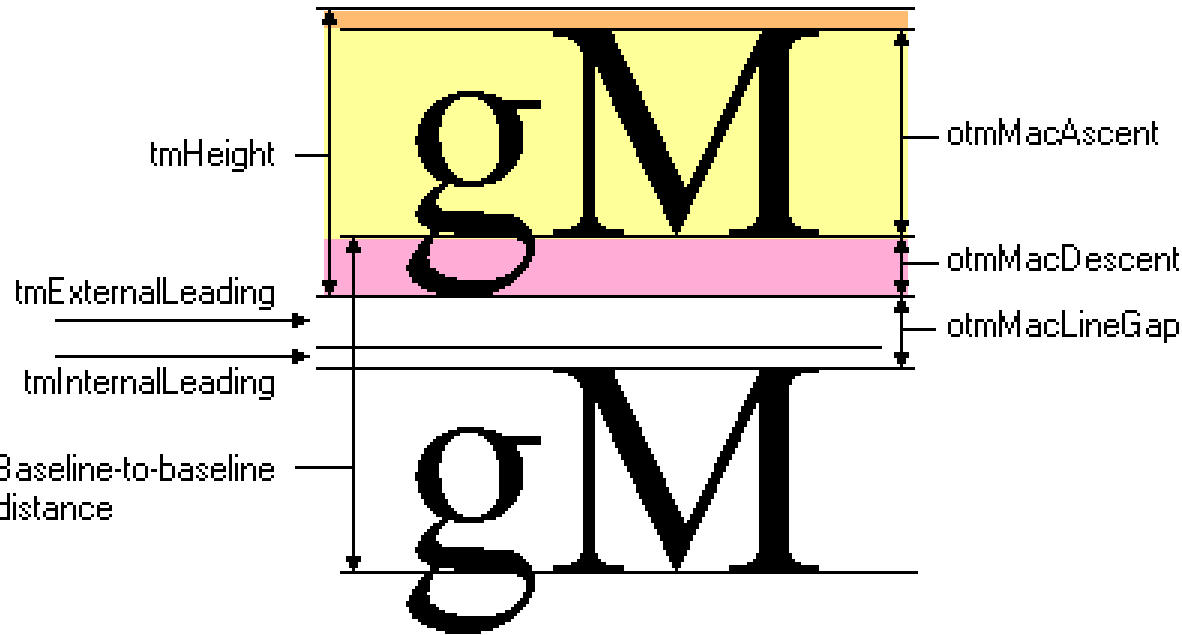
# Primer očitavanja ABC parametara fonta

```
CFont font;  
font.CreateFont(20,0,0,0,0,0,0,0,0,0,0,0,0,0,"Times New Roman");  
CFont* pOldFont = (CFont*)pDC->SelectObject(&font);  
int widths[2];  
BOOL suc = pDC->GetCharWidth('f','g',widths);  
ABC abcArray[2];  
suc = pDC->GetCharABCWidths('f','g',abcArray);  
pDC->TextOut(20,20,"f");  
pDC->TextOut(30,20,"g");  
pDC->SelectObject(pOldFont);
```

```
Widths[0] = 6  
Widths[1] = 8  
abcArray[0].abcA = 0  
abcArray[0].abcB = 8  
abcArray[0].abcC = -2  
abcArray[1].abcA = 1  
abcArray[1].abcB = 7  
abcArray[1].abcC = 0
```



# Metrika fonta – visina



```
typedef struct tagTEXTMETRIC { // tm
    LONG tmHeight;
    LONG tmAscent;
    LONG tmDescent;
    LONG tmInternalLeading;
    LONG tmExternalLeading;
    LONG tmAveCharWidth;
    LONG tmMaxCharWidth;
    LONG tmWeight;
    LONG tmOverhang;
    LONG tmDigitizedAspectX;
    LONG tmDigitizedAspectY;
    BCHAR tmFirstChar;
    BCHAR tmLastChar;
    BCHAR tmDefaultChar;
    BCHAR tmBreakChar;
    BYTE tmItalic;
    BYTE tmUnderlined;
    BYTE tmStruckOut;
    BYTE tmPitchAndFamily;
    BYTE tmCharSet;
```

```
} TEXTMETRIC;
```

**BOOL CDC::GetTextMetrics( LPTEXTMETRIC lpMetrics )**



# Veličina ispisa teksta

- Metode za očitavanje veličine teksta

`CSize CDC::GetTextExtent( const CString& str )`

- Računa veličinu (širinu i visinu jedne teksta ispisanog tekućim fontom.
  - Povratna vrednost je instanca klase **CSize** koji ima dva atributa **cx** (sadrži širinu) i **cy** (sadrži visinu)
- Širina linije teksta ne mora biti jednaka zbiru širina pojedinih znakova
- str*** – tekst čija se veličina želeli odrediti

# Atributi fonta

- *visina (height),*
- *širina (width),*
- *iskošenost (escapement),*
- *debljina (weight)*
- *Character Set,*
- *pitch*
- *familija*
- *...*

# Visina i širina fonta

## ■ visina

- $> 0$  – bira se font najpribližnije visine u logičkim jedinicama
- $= 0$  – bira se font podrazumevane visine
- $< 0$  – bira se font sa visinom jednakom apsolutnoj vrednosti zadate, ali u fizičkim koordinatama

## ■ širina

- $> 0$  – bira se font najpribližnije širine u logičkim jedinicama
- $= 0$  – bira se font podrazumevane širine

## Debljina fonta

|                  |            |                |            |
|------------------|------------|----------------|------------|
| FW_DONTCARE      | 0          | FW_SEMIBOLD    | 600        |
| FW_THIN          | 100        | FW_DEMIBOLD    | 600        |
| FW_EXTRALIGHT    | 200        | <b>FW_BOLD</b> | <b>700</b> |
| FW_ULTRALIGHT    | 200        | FW_EXTRABOLD   | 800        |
| FW_LIGHT         | 300        | FW_ULTRABOLD   | 800        |
| <b>FW_NORMAL</b> | <b>400</b> | FW_HEAVY       | 900        |
| FW_REGULAR       | 400        | FW_BLACK       | 900        |
| FW_MEDIUM        | 500        |                |            |

# Character Set

- **ANSI\_CHARSET** – podrazumevani (0), najčešće se koristi
- **DEFAULT\_CHARSET** – koristi se često da aplikacija ne bi imala neočekivane rezultate ukoliko se izabere font koji ne postoji (1)
- **OEM\_CHARSET** – koristi se prvenstveno za konzolne aplikacije, sličan je ANSI setu, ali zavisi od sistema, retko se koristi (zavisan je od sistema)
- **SYMBOL\_CHARSET** – koristi se za prikaz specijalnih simbola, npr. matematičkih formula (2).

# Širina znaka (*pitch*) i familija

## Širina znaka

- **DEFAULT\_PITCH** – bira se širina znaka na osnovu ostalih atributa
- **FIXED\_PITCH** – kreira se font fiksne širine
- **VARIABLE\_PITCH** – kreira se font promenljive širine

## Familija

- **FF\_DECORATIVE**
- **FF\_DONTCARE**
- **FF\_MODERN**
- **FF\_ROMAN**
- **FF\_SCRIPT**
- **FF\_SWISS**

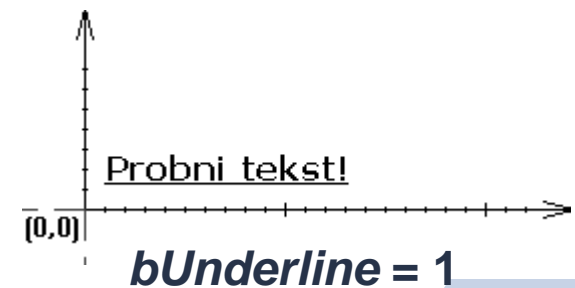
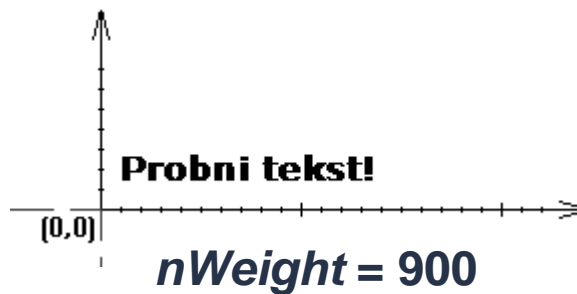
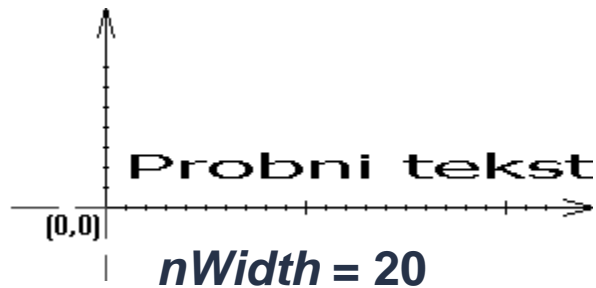
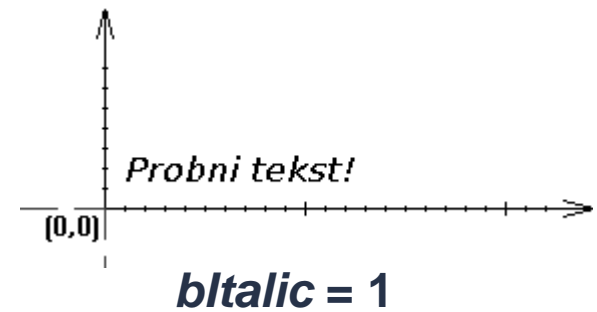
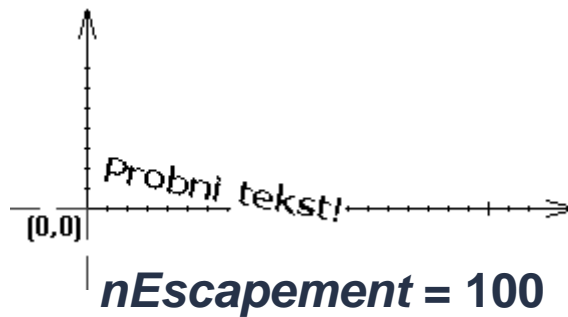
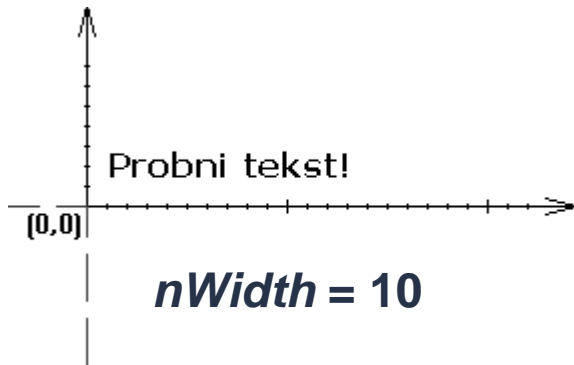
## Primer

- **DEFAULT\_PITCH | FF\_SWISS**

# Kreiranje fonta

```
BOOL CFont::CreateFont(  
    int nHeight,           // visina u logičkim jedinicama  
    int nWidth,            // srednja širina karaktera u logičkim jedinicama  
    int nEscapement,       // ugao esc. u 10-tim delovima stepena  
    int nOrientation,      // orijentacija osnovne linije (base-line)  
    int nWeight,           // debljina  
    BYTE bltalic,          // italic (kurziv)  
    BYTE bUnderline,       // podvlačenje  
    BYTE cStrikeOut,       // precrtavanje  
    BYTE nCharSet,         // identifikator skupa karaktera (character set)  
    BYTE nOutPrecision,    // preciznost ispisa  
    BYTE nClipPrecision,   // preciznost odsecanja  
    BYTE nQuality,         // kvalitet ispisa  
    BYTE nPitchAndFamily,  // širina znaka i familija  
    LPCTSTR lpszFacename  // typeface naziv fonta  
);
```

# Parametri fonta





# Funkcije za ispis teksta

## ■ Window Manager funkcije (USER.DLL)

- **DrawText**
- **TabbedTextOut**

## ■ GDI funkcije (GDI.DLL)

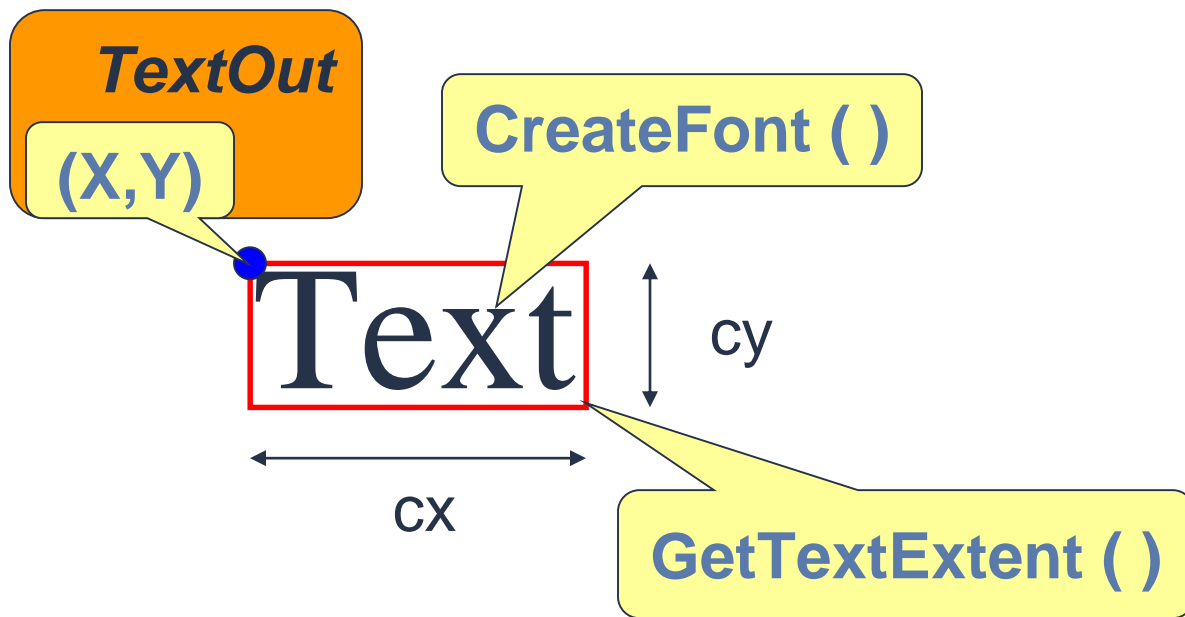
- **TextOut**
- **ExtTextOut**

## Ispisa teksta – TextOut

```
BOOL CDC::TextOut( int x, int y, const CString& str )  
virtual BOOL CDC::TextOut( int x, int y,  
                           LPCTSTR lpszString, int nCount )
```

- *x, y* – koordinate tačke odakle počinje ispis teksta
- *str* – tekst koji treba ispisati
- *lpszString* – pokazivač na niz karaktera koji treba ispisati
- *nCount* – dužina niza karaktera koji treba ispisati

# Ispisa teksta – TextOut



# Primer ispisa teksta – TextOut

```
void CGDI_FontsView::OnDraw(CDC* pDC) {  
    int nHeight          = 20; // OVO MORA DA SE NAVEDE  
    int nWidth           = 0;  
    int nEsc             = 0;  
    int nOrient          = 0;  
    int nWeight          = 0;  
    BYTE bltalic         = 0;  
    BYTE bUnderline      = 0;  
    BYTE cStrikeOut      = 0;  
    BYTE nCharSet        = 0;  
    BYTE nOutPrecision   = 0;  
    BYTE nClipPrecision  = 0;  
    BYTE nQuality        = 0;  
    BYTE nPitchAndFamily = 0;  
    CString sFaceName    = "Arial"; // OVO MORA DA SE NAVEDE
```

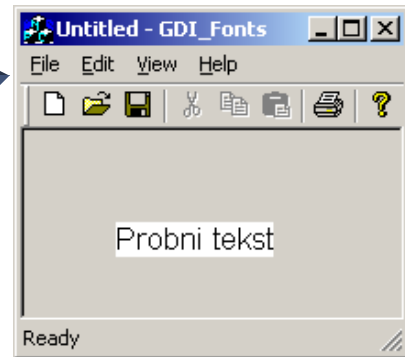
```
}
```

```
CFont font;  
font.CreateFont( nHeight, nWidth, nEsc, nOrient, nWeight,  
                bltalic, bUnderline, cStrikeOut, nCharSet,  
                nOutPrecision, nClipPrecision, nQuality,  
                nPitchAndFamily, sFaceName);
```

```
CFont* pOldFont = pDC->SelectObject(&font);
```

```
pDC->TextOut(50,50,"Probni tekst");
```

```
pDC->SelectObject(pOldFont);  
font.DeleteObject();
```



# Poravnanje teksta u funkciji TextOut

## Horizontalno poravnanje:

- **TA\_LEFT**
- **TA\_RIGHT**
- **TA\_CENTER**

## Vertikalno poravnanje

- **TA\_TOP**
- **TA\_BOTTOM**
- **TA\_BASELINE**

## Tekuća pozicija kursora

- **TA\_NOUPDATECP** - ne menja tekuću poziciju nakon poziva TextOut
- **TA\_UPDATECP** – Menja tekuću x-poziciju posle svakog poziva f-je i postavlja je na desnu stranu okvirnog pravougaonika ispisanog teksta. Ako je ovo postavljeno, koordinate prosledene TextOut se zanemaruju.

## Primer

```
pDC->SetTextAlign(TA_LEFT | TA_TOP |  
                  TA_UPDATECAP);
```

```
pDC->MoveTo(x, y)
```

```
pDC->TextOut(-1, -1, text);
```

# Ispisa teksta – DrawText

```
int CDC::DrawText( const CString& str,  
                  LPRECT lpRect, UINT nFormat)  
virtual int CDC::DrawText( LPCTSTR lpzString,  
                           int nCount, LPRECT lpRect, UINT nFormat)
```

- Ispis formatiranog teksta u zadatom pravougaoniku.
- Poravnava tekst u odnosu na ivice pravougaonika i vrši prelom u više redova
- str*** – tekst koji treba ispisati
- lpzString*** – pokazivač na niz karaktera koji treba ispisati
- nCount*** – dužina niza karaktera koji treba ispisati
- lpRect*** – pokazivač na pravougaonik u kom treba ispisati tekst
- nFormat*** – način formatiranja teksta

# Formatiranje ispisa teksta – DrawText

Vrednosti za formatiranje teksta mogu se kombinovati bitskim OR operatorom

**DT\_SINGLELINE** – Ispis teksta samo u jednoj liniji

- Specijalni simboli koji označavaju prelazak u novi red (carriage returns i linefeeds) ne prelamaju liniju.

**DT\_CENTER** – Horizontalno centrira tekst.

**DT\_LEFT** – Levo poravnanje.

**DT\_RIGHT** – Desno poravnanje.

**DT\_BOTTOM** – Poravnava sa donjom ivicom okvirnog pravougaonika.

- Ovaj format mora da ide u kombinaciji sa DT\_SINGLELINE.

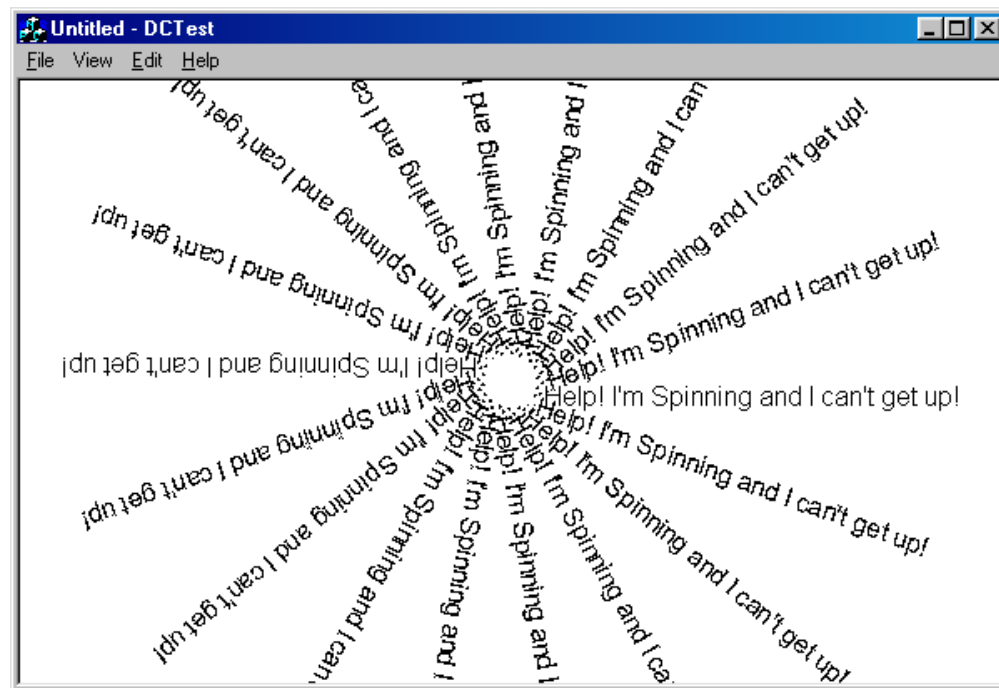
**DT\_TOP** – Poravnanje uz gornju ivicu (uz DT\_SINGLELINE).

**DT\_VCENTER** – vertikalno centriran tekst (uz DT\_SINGLELINE).

**DT\_TABSTOP** – Postavlja dužinu tabulatora.

- Viši bajt nFormat definiše koliko karaktera je dugačak tab. Podrazumeva se 8 karaktera.

# Primer ispisa teksta



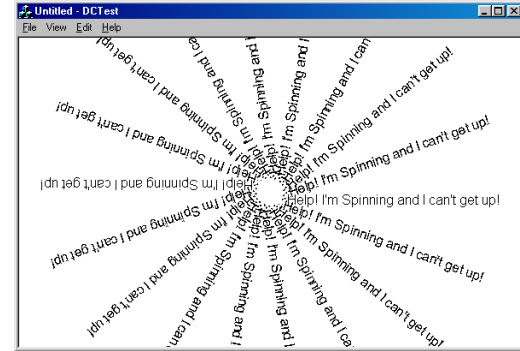


# Primer postavljanja transformacije

```
CRect rcClient;  
GetClientRect( rcClient );  
COLORREF clrOld = pDC->SetTextColor( m_clrChoice );  
int nOldMode = pDC->SetBkMode( TRANSPARENT );  
CString szMsg = " ...Help! I'm Spinning and I can't get up!";  
CFont  fntRotate;  
for( int nDegrees = 0; nDegrees < 3600; nDegrees += 200 )  
{  
  
}  
pDC->SetTextColor( clrOld );  
pDC->SetBkMode( nOldMode );
```



```
m_logFont.IfEscapement = nDegrees;  
fntRotate.CreateFontIndirect( &m_logFont );  
CFont* pOldFont = pDC->SelectObject( &fntRotate );  
pDC->TextOut( rcClient.Width()/2,  
              rcClient.Height()/2, szMsg );  
pDC->SelectObject( pOldFont );  
fntRotate.DeleteObject();
```



# Stock fontovi

- ANSI\_FIXED\_FONT – sistemski font fiksne širine karaktera
- ANSI\_VAR\_FONT – sistemski font promenljive širine karaktera
- DEVICE\_DEFAULT\_FONT – font zavistan od uređaja
  - Postoji samo na Windows NT
- DEFAULT\_GUI\_FONT – podrazumevani font namenjen ispisu teksta na korisničkom interfejsu (meni, kontrole u prozorima i slično)
- OEM\_FIXED\_FONT – font fiksne širine karaktera zavistan od OEM-a
- SYSTEM\_FONT – sistemski font

Primer

```
CFont* pOldFont = pDC->SelectStockObject(SYSTEM_FONT);
```

# Domaći zadatak

- Napisati program koji korisniku omogućuje unos proizvoljne poligonalne linije. Temena se zadaju **levim klikom** miša, a unos se prekida **desnim klikom**.
- Po završetku unosa **izlomljene linije** otvara se **dijalog** za unos stringa proizvoljne dužine.
- Nakon klika na dugme OK potrebno je string ispisati fontom “**Times New Roman**”, veličine **20**, **tamno plave boje** i sa **providnom pozadinom**, **duž najdužeg segmenta izlomljene linije**.
- Ukoliko je tekst duži od najdužeg segmenta upozoriti korisnika odgovarajućom **porukom** (MessageBox).

# Pitanja

