
* DOPUNA98.PRG

* 25-05-2003

* c_godina KLJUC za poslovnu godinu
* **FUNCTION** BOJE kolor sistem programa
* **FUNCTION** PAPIR pozadinska maska sa hederom i futerom
* **FUNCTION** POZADINA pozadinska maska sa hederom i futerom
* **FUNCTION** VERZIJA verzija i sistemski datum u hederu
* **FUNCTION** DAT_STOP GODINA LIMITA RADA PROGRAMA - FIKSNA UPISANA OVDE
* **FUNCTION** UPUTSTVO pregled i stampa TXT fajla uputstva
* **FUNCTION** STAMPA_1 kontrola printera - pregled i stampa TXT fajla
* **FUNCTION** LISTAJ pregled TXT fajla preko eksternog **LIST.COM** programa

***FUNCTION** C_godina

*

* FUNKCIJA:

* Zastita programa - dozvolom rada samo u odredjenoj kalendarskoj

* godini koja se realizuje samo uz sadejstvo sa funkcijom:

* **C_stop**(expD) -> expL, kojom se vrsi provera upisanog datuma rada

* sa tastature.

* SINTAKSA:

* **C_godina**(expN,expC) -> expN1

* PARAMETRI:

* expN - 0 ili 1

* expC - EXE_direktorijum na kome se nalaze EXE/DLL programa

* (to je uvek **WORKDIR**() direktorijum), na primer:

*

* C:\BAST\ - EXE_direktorijum programa

*

* i ovaj direktorijum na svakoj radnoj stanici moze biti

* drugacijeg naziva,

* za razliku od SERVER_direktorijuma na kome se nalazi

* baza podataka, na samo jednom od racunara, a koji moze

* biti:

*

* C:\BAZA\ - SERVER_direktorijum

* C:\BAZA\01\ - direktorijum firme 01

* C:\BAZA\01\1999\ - baza podataka za 1999 godinu

*

* a ostale radne stanice ga vide i poznaju preko mapiranog

* diska kao:

*

* E:\BAZA\ - SERVER_direktorijum

* E:\BAZA\01\ - direktorijum firme 01

* E:\BAZA\01\1999\ - baza podataka za 1999 godinu

* Ovaj SERVER_direktorijum MORA da postoji pre poziva funkciji

* C_godina, a kreira se iz instalacionog programa aplikacije.

*

* expN1 - numerik godine, na primer 1999, 2000, 2001...

* Poziv funkcije sa parametrom expN = 0

```

* formira MEM fajl na disku na podrucju EXE_direktorijum.
* bilo da ovog fajla tu vec ima, bilo da ga nema.
* Naziv MEM fajla je "_____", zatim se zadaje godina
* dozvoljenog rada i upisuje u MEM fajl kao SIFROVANI BROJ.
* U postupku zadavanja godine, sa (Esc) ili sa godinom
* manjom od 1999 vrsi se brisanje MEM fajla.
*
* Poziv funkcije sa parametrom expN = 1,2,3...
* proverava postoji li na disku na podrucju EXE_direktorijum
* MEM fajl sa nazivom "_____" ;
* Ako ne postoji funkcija vraća expN1 = 1999
* Ako postoji funkcija iz njega učitava sifrovani broj,
* desifruje ga i vraća dobijenu godinu kao expN1.
*
* NAPOMENA:
* Kod rada u mrezi, umesto MEM fajla koristiti DBF fajl,
* jer MEM fajl ne može da se stiti od istovremenog pristupa dva
* korisnika !!!
*
*****
* FUNKCIJA BOJE
*
*      Kolor sistem programa
*
* SINTAKSA boje()
* Postavlja nazive boja kao PUBLIC varijable za ceo program:
* Nazivi se koriste sa SETCOLOR(naziv)
* Poziva se na pocetku svakog programa.
*
*****
*
FUNCTION boje
*
PUBLIC b_gtrake,b_dtrake,b_gmeni,b_bazne,b_dbedit,b_dbupis,b_dbmeni,;
      boja_0,boja_1,bojane

IF ISCOLOR().T.
boja_0="BG+"      && boja neaktivne knjige - glavni izbor rada
boja_1="GR+/B"    && boja aktivne knjige - glavni izbor rada
bojane="W/N"      && NEMA BOJE = SET COLOR TO
b_gtrake="W+/R"
b_dtrake="BG/R"
b_gmeni="W/N,W+/R"
b_bazne="BG+/R,R/W"
b_dbedit="B/W,W+/G,B/W,B/W"
b_dbupis="B/W,W+/R,B/W,B/W"
b_dbmeni="GR+/B,W+/G,B/W,B/W"
ELSE
boja_0="W+/N"
boja_1="N/W"
bojane="W/N"      && NEMA BOJE = SET COLOR TO
b_gtrake="N/W"
b_dtrake="N/W"
b_gmeni="W/N,N/W"
b_bazne="N/W,N/W"
b_dbedit="N/W,W+/N,N/W,N/W"
b_dbupis="N/W,W+/N,N/W,N/W"
b_dbmeni="N/W,W+/N,N/W,N/W"

```

ENDIF**RETURN(0)**

*

*bb

*bb

* FUNKCIJA PAPIR

*

* Postavljanje pozadine svakog izbora rada

*

* Da bi funkcija imala podatke za svoj frad: VER, P_NAZIV,P_GODINA, mora se

* prethodno definisati promenljiva **VER** (string), a mora se prethodno pozvati* funkcija **DOZ_RADA(1)** da bi definisala P_GODINA i **P_NAZIV** (globalne).

*

* SINTAKSA: **PAPIR("NASLOV PROGRAMA",n)**

* Parametar n:

* n=1 ili 0 menja boju pozadine - papira

*

* Funkcija uz NASLOV PROGRAMA dodaje i godinu rada programa iz INSTAL.**DBF**

* odnosno string promenljivu P_GODINA koja se kao GLOBALNA promenljiva formira

* iz DOZVOLA.PRG u BAZNE.LIB.

* U donju lajsnu ekrana funkcija upisuje P_NAZIV globalnu promenljivu iz

* INSTAL.**DBF** koja se takodje formira iz DOZVOLA.PRG u BAZNE.LIB.

* Ova funkcija na kraju rada poziva funkciju VERZIJA koja u gornju lajsnu

* **ekrana** (levo) upisuje oznaku verzije programa koristeći globalnu promenljivu* VER koja se definise na pocetku svakog programa, **a** (levo) upisuje trenutni

* sistemski datum.

*

*

FUNCTION PAPIR

*

PARAMETERS PRG_NASLOV,PRG_IZBOR**PRIVATE** PRG_NASLOV, PRG_FIRMA, PRG_IZBOR

PRG_FIRMA=P_NAZIV + + " " + "(" + P_GODINA + ")"

PRG_NASLOV=PRG_NASLOV

*-----

* postavljanje pozadine izbora rada

IF PRG_IZBOR=0**setcolor**(b_gtrake)**ELSE****set color to** /W**ENDIF**

*

@ 01,00 **clear to** 23,79@ 01,00 **say replicate**("Ä",80)@ 23,00 **say replicate**("Ä",80)**setcolor**(bojane)

*

POZADINA(PRG_NASLOV,PRG_FIRMA) && bazne lib**VERZIJA**() && oznaka verzije programa**setcolor**(bojane)**RETURN(0)**

*

*bb

* FUNKCIJA POZADINA

```

*                pozadinska maska programa
* SINTAKSA:
* POZADINA('TEKST U HEDERU','TEKST U FUTERU')
*
*****
FUNCTION POZADINA
PARAMETERS tekst_gore, tekst_dole
PRIVATE boja_gore,boja_dole
*
set color to
if iscolor()=.t.
boja_gore="w+/r"
boja_dole="gr+/r"
else
set color to
boja_gore=" /w"
boja_dole=" /w"
endif
* gornja traka - heder
set color to &boja_gore
@ 00,00 say space(80)
centar(tekst_gore,0)
* donja traka - futer
set color to &boja_dole
@ 24,00 say space(80)
centar(tekst_dole,24)
set color to
RETURN(0)

*bbbbb
FUNCTION VERZIJA
*
PRIVATE boja,datum
SET CENTURY ON
boja=setcolor()
set color to &b_dtrake
@ 00,01 say ver && promenljiva ver je PUBLIC u glavnom programu
datum=dtoc(date())
set color to &b_gtrake
@ 00,69 say datum
setcolor(boja)
SET CENTURY OFF
*
RETURN(0)
*bbbbb

*bbbbb
* FUNKCIJA dat_stop
*
*                Kontrola godine u kojoj je dozvoljen rad
*
* SINTAKSA: IF dat_stop(datum)=.T.
*                * prihvata se upisani datum DATUM1 kao ispravan
*                ELSE
*                * ne prihvata se upisani datum DATUM1 kao ispravan
*                ENDIF

```

FUNCTION dat_stop

PARAMETERS datum_rada && datum koji korisnik upise sa tastature

SET CENTURY ON

PUBLIC god_roka && godina dozvoljenog rada fiksirana u bazne.lib

PRIVATE god_rada && godina dobijena iz datuma rada

PRIVATE vrati

vrati=.F.

god_roka=**C_godina**(1,"A') && godina dozvoljenog rada u fajlu _____.____

god_rada=**year**(datum_rada)

*

* kontrola godine u kojoj je dozvoljen rad...

*

* **IF STR**(god_rada,4,0) > **STR**(god_roka,4,0)

IF ROUND(god_rada,0) > **ROUND**(god_roka,0)

vrati=.F.

c_greska("Moguc rad samo sa datumima iz '**STR**(god_roka,4,0)+' godine')

ELSE

vrati=.T.

ENDIF

*

RETURN(vrati)

FUNCTION C_godina

//

// **FUNCTION** C_godina(nFlag,sFolder) --> nGodina

//

// **VAZNO**: MEM fajlovi Xbase++ i Clipper-a nisu kompatibilni. Zbog toga ovaj fajl koji pravi Xbase ima jednu crticu manje u nazivu (7 umesto 8 kao kod Clipper-a), da bi na istom podrucju mogli da ih koriste Clipper aplikacije i Xbase aplikacije.

// Formira MEM fajl 7+3 odnosno: _____.____, za razliku od Clipper 87' ciji je MEM fajl 8+3 _____.____

// iz koga učitava ili u koga upisuje sifrovanu poslovnu godinu od 2000 - 2999. Ako se upise druga godina

// briše se _____.____ fajl. Ako se iz GET moda za upis godine izadje sa Esc, briše se _____.____ fajl.

// Ako nema fajla _____.____ funkcija vraća poslovnu godinu 2000.

// Funkcija prima 2 parametra: C_godina(citaj,EXE_direktorijum) --> numerik (godina 2003)

// citaj = 0 upisuje godinu u _____.____ fajl, ako fajla nema formira ga i dozvoljava upis

// godina od 2000 do 2999, svaki drugi upis ili izlaz sa Esc briše fajl sa diska.

// citaj = 1 učitava godinu iz _____.____ fajla, ako fajla nema učitava 2000.

//

// EXE_direktorijum je folder u kome se fajl _____.____ formira i traži. Ako ovaj folder nije zadat sa

// više od tri slova, tada se podrazumeva da je _____.____ fajl na aktuelnom direktorijumu CURDIR().

PARAMETERS citaj,EXE_direktorijum

PRIVATE godina,godina_s,godina_d,fajl,boja_stara,prozor

boja_stara=**SETCOLOR**()

save **screen to** prozor

M->godina=2000

* ako je upisan folder C:\COBA2000\
* tada kljuc godine fajl _____.____ (7+3 crtice, stari clipper87 8+3crtice)
* trazi u tom folderu, a ako nije
* upisan folder, tada ga trazi na **CURDIR()**

```
IF LEN(EXE_direktorijum)>3  
fajl=EXE_direktorijum+'_____.____'  
ELSE  
fajl='_____.____'  
ENDIF
```

```
IF citaj=0  
* upisi godinu:  
@ 11,37 TO 13,42 DOUBLE  
@ 12,38 GET M->godina PICT "9999"  
READ  
CLEAR GETS  
restore screen from prozor  
release prozor
```

```
IF LASTKEY()=27  
    DELETE FILE &fajl  
    RETURN(2000)  
ENDIF  
IF godina<2000 .or. godina>2999  
    c_greska('Dozvoljene godine od 2000 do 2999')  
    DELETE FILE &fajl  
    RETURN(2000)  
ENDIF
```

```
godina_s=godina^3  
godina_s=godina_s-99999999999999
```

```
C_poruka('Sifrovano '+str(godina_s))
```

```
* Snimanje:  
SAVE TO &fajl ALL LIKE godina_s  
RELEASE godina_string,godina_s  
* Kontrola:  
RESTORE FROM &fajl ADDITIVE
```

```
godina_s=godina_s+99999999999999  
godina_d=godina_s^(1/3)
```

```
C_poruka('DeSifrovano '+str(godina_d))
```

```
M->godina=godina_d
```

```
ELSE  
* procitaj godinu:
```

```
IF .NOT. FILE(fajl)  
    * godina=2000  
ELSE
```

```

RESTORE FROM &fajl ADDITIVE
godina_s=godina_s+99999999999999
godina_d=godina_s^(1/3)
M->godina=godina_d
ENDIF

```

```

ENDIF

```

```

M->godina=round(M->godina,0)
SETCOLOR(boja_stara)
RETURN(M->godina)

```

```

*bbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbb

```

```

*bbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbb

```

```

*
```

```

* FUNKCIJA UPUTSTVO

```

```

*
```

```

* SINTAKSA:

```

```

* UPUTSTVO('naslov boksa za izbor rada','naziv txt fajla uputstva')

```

```

* UPOTREBA:

```

```

* poziva se za pregled ili stampu tekst fajla uputstva za rad. Koristi dve

```

```

* funkcije iz BAZNE.LIB i to: Stampa_1('naslov boksa za izbor rada') i

```

```

* LISTAJ('naziv txt fajla uputstva')

```

```

* Vraca vrednost GLOBALNE promenljive STAMPA = "DA" ili STAMPA="NE"

```

```

* Uputstvo stampa komandom TYPE uputstvo.txt TO PRINTER

```

```

*
```

```

FUNCTION UPUTSTVO

```

```

PARAMETERS naslov,txtfile

```

```

*=====

```

```

IF stampa_1(naslov)=.t.

```

```

IF STAMPA="NE"

```

```

    listaj(txtfile)

```

```

ELSE

```

```

    IF FILE(txtfile)

```

```

        SET CONSOLE OFF

```

```

        TYPE &txtfile TO PRINTER

```

```

        SET CONSOLE ON

```

```

    ENDIF

```

```

ENDIF

```

```

ENDIF

```

```

*=====

```

```

RETURN(0)

```

```

*bbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbb

```

```

*****

```

```

* FUNKCIJA STAMPA_1

```

```

*
```

```

*      Odluka da li se nesto stampa ili prikazuje na ekranu
*      iz boksa za izbor rada: stampa, prikaz, kraj
*

```

```

*      Sintaksa: IF STAMPA_1('TEKST NASLOVA BOKSA')=.F.

```

```

*      RETURN --- izlaz iz programa za stampu/prikaz

```

```

*
```

```

ENDIF

```

```

*
```

```

*      Odluka moze biti=.t. (nastavlja se program za prikaz/stampu)
*      Odluka moze biti=.f. (prekida se program za prikaz/stampu)
*      Posle izvršenja funkcije sa .T. formira se GLOBALNA promenljiva
*      STAMPA="DA" ili STAMPA="NE" na osnovu statusa printera, pa se
*      na osnovu nje odredjuje da li se ide na stampu ili na pregled.
*****

```

```

FUNCTION STAMPA_1
PARAMETERS tekst
PUBLIC STAMPA
PRIVATE boja, izbor, izlaz, tekst, prozor, boja1
save screen to prozor
STAMPA="NE"
    izlaz=.t.      && nastavlja se program posle izvršenja funkcije
* izlaz=.f.      && prekida se program posle izvršenja funkcije
if iscolor()=.f.
    boja1="N/W, W+/N"
else
    boja1="BG+/R, W+/N"
endif
boja=setcolor()
setcolor(boja1)
*
DO WHILE .T.
@ 08,29 clear to 15,49
@ 10,32 to 14,46
centar(tekst,9)
@ 11,33 prompt " Ekran      "
@ 12,33 prompt " Printer    "
@ 13,33 prompt " Izlaz      "
tone(700,1)
menu to izbor
if izbor=3 .or. izbor=0
    izlaz=.f.
    STAMPA="NE"
    exit      && prekida se program; izlaz=.f.; STAMPA="NE"
endif
if izbor=1
    izlaz=.t.
    STAMPA="NE"
    exit      && nastavlja se program; izlaz=.t.; STAMPA="NE"
endif
if izbor=2
    if isprinter()=.f.
        c_greska("PRINTER NIJE U REDU")
        STAMPA="NE"
        loop
    endif
    izlaz=.t.
    STAMPA="DA" && nastavlja se program; izlaz=.t.; STAMPA="DA"
    exit
endif
ENDDO
restore screen from prozor
release prozor
setcolor(boja)
*

```


RETURN(izlaz)

*

**bb*

* FUNKCIJA LISTAJ

* Listanje TXT fajla u ASCII kodu sa eksternim programom **LIST.COM**

* SINTAKSA:

* **LISTAJ**('naziv txt fajla')

*

FUNCTION LISTAJ

PARAMETERS ffajl

PRIVATE ffajl, fprozor

// I FAZA - pribavljanje COBA alatke:

IF .NOT. **FILE**("LIST.EXE")

IF **FILE**("C:\COBA2000\LIST.EXE")

COPY FILE C:\COBA2000\LIST.EXE **TO** LIST.EXE

ELSE

MsgBox("Nije nadjen fajl:" + **chr**(13) + **chr**(10) +;
 "C:\COBA2000\LIST.EXE", "STOP")

ENDIF

ENDIF

// II FAZA - pribavljanje WINDOWS alatke:

IF .NOT. **FILE**("LIST.EXE")

IF **FILE**("C:\WINDOWS\notepad.exe")

COPY FILE C:\WINDOWS\notepad.exe **TO** LIST.EXE

ELSE

MsgBox("Nije nadjen fajl:" + **chr**(13) + **chr**(10) +;
 "C:\WINDOWS\notepad.exe", "STOP")

ENDIF

ENDIF

IF .NOT. **FILE**("LIST.EXE")

 * **c_poruka**("Nije nadjen fajl LIST.EXE pa ekranski pregled nije moguc")

MsgBox("Nije nadjen fajl:" + **chr**(13) + **chr**(10) +;
 CURDIR() + "LIST.EXE" + **chr**(13) + **chr**(10) +;
 "pa ekranski pregled nije moguc", "STOP")

ELSE

```
IF .NOT. FILE(ffajl)
* c_poruka("Nije nadjen fajl "+ffajl+" pa ekranski pregled nije moguc")
  MsgBox("Nije nadjen fajl:"+chr(13)+chr(10)+;
    ffajl+chr(13)+chr(10)+;
    "pa ekranski pregled nije moguc","STOP")

ELSE
  fprozor=space(10)
  SAVE SCREEN TO fprozor
  RUNSHELL(ffajl,"LIST.EXE",.F.)
  RESTORE SCREEN FROM fprozor
  RELEASE fprozor
ENDIF

ENDIF

RETURN(0)
*****
```