

```
NB. Чтение файла JSON в программу
NB. Алгоритм
NB. 1. Поиск и ввод файла с данными в формате JSON
NB. 2. Вывод на экран меню набора и комментариев к набору
NB. 3. Или отказ или чтение в переменную всего файла
NB. 4. Вывод "меню" файла; Номер строки, комментарий.
NB. 5. Выбор значений в качестве результата функции
NB. =====
NB. _____
NB. _____
```

```
NB. IDE=: 1
IDE=: 0
NB.showide''
fjs_wctrl_fkey=: showide
```

```
NB. Головная формашowide=: 3 : 0
showide=: 3 : 0
if. IDE do.
  IDE=: 0
  wd'ide hide'
else.
  IDE=: 1
  wd'ide show'
  wd'psel fjs;pactive'
end.
empty''
)
```

```
FJSON=:0 : 0
pc fjs escsclose closeok; pn Модуль ввода-вывода JSON;
menupop "&Файлы";
menu wcat "&Определить рабочий каталог";
menu tinfile "&Открыть текстовый файл с данными";
menu toutfile "&Сохранить текстовый файл с данными";
menusep;
menu jfile "&Открыть файл JSON";
menu tojfile "&Создать файл JSON";
menusep;
menu jexit "Завершить программу";
menupopz;
menupop "&Описание программы" ;
menu descr "&Читать";
menupopz;
menupop "&О программе";
menu priv "&Права";
menu auth "&Авторы";
menupopz;
rem _____;
bin v;
bin h;
cc hh static; cn НЦ ИНИС НИЯУ МИФИ;
bin zv;
groupbox;
```

```

cc t00 static; cn Ввод-вывод JSON ;
set t00 font "Lucida Console" 15;
bin h;
groupbox Начало;
cc tf01 static; cn Распределить файлы в меню <Файлы>;
groupboxend;
groupbox "Системные переменные";
cc ibfin static; cn Для ввода текста "FTREAD";
cc infout static; cn Для вывода текста "FTWRITE";
groupboxend;
bin zv;

rem bin s;
groupboxend;
pshow;
pcenter;
rem pmove 600 600 500 300;

cc workdir static;
bin s;
bin zh;
cc jexit button; cn Выход;
bin s;
bin z;
)

authtext=: 0 : 0
INIS-JSON
В. М. Куприянов, А. Купин
Программа разработана в Национальном центре
ИНИС МАГАТЭ
для хранения фактологических данных,
извлеченных из открытой информации
Международной ядерной информационной системы
ИНИС МАГАТЭ
Назначение программы; - стандартизованное
представление ядерно- технологических данных с целью
предоставления достуав пользователей к хранилищу данных
для предприятий Росатома, в первую очередь,
в обеспечение моделирования задач замкнутого топливного цикла АЭ
)
fjs_auth_button=: 3 : 0
wd 'pc textd dialog escclose closeok; pn Описание (ESC - закрыть)'
wd 'cc win editm'
wd 'set win text *', authtext
wd 'pshow; pcenter'
''
)

fjs_run=: 3 : 0
wd FJSON
wd 'set t00 text Ввод-вывод JSON'

```

```

wd 'set t00 font "Lucida Console" 18'
wd 'set workdir visible 0'
wd 'set jexit font "Lucida Console" bold 12'
wd 'set tf01 text *', 'Распределить файлы в меню <Файлы>' '
wd 'setp wh 500 300'

FTWRITE=:'' NB. Идентификатор текстоваого файла для запис
)
NB. =====
fjs_descr_button=: 3 : 0
wd 'pc textd dialog escsclose closeok; pn Описание (ESC - закрыть)'
wd 'cc win editm'
wd 'set win text *', descrt
wd 'set win wh 200 50'
wd 'pshow; pcenter'
''
)
descrt=: 0 : 0
Вход-выход в систему J - ctrl+w
Выход из текущего окна ESC
Для ввода-вывода произвольных текстов
использовать системные переменные
)

NB. =====
NB. Окончание программы
fjs_jexit_button=: 3 : 0
wd 'pclose'
)
NB. =====
NB. Ввод данных из файла JSON
fjs_jfile_button=: 3 : 0
FJread ''
)
NB. =====
NB. Чтение текстового файла в рабочую область
fjs_tinfile_button=:3 : 0
ftfile=:wd 'mb open1 "" "" "" '
FTREAD=: fread ftfile
wd 'mb info *Ввод Файл прочтен в переменную',LF,LF, 'FTREAD''
NB. ftread
FTREAD
)

NB. =====
fjs_toutfile_button=:3 : 0
NB. Запись текстового файла из переменной FTWRITE
wd 'mb info *Будет записана переменная', ' FTWRITE" в текстовый файл в
каталог',LF,LF,workd
p=. 'Text (*.txt)|All Files (*.*)'
NB.wd 'mb save "Сохранить файл" "',(jpath '~install'),' " "',p, '""
foutf=:wd 'mb save "Сохранить файл" "',(workd),' " "',p, '""
tft=: ' из переменной "FTWRITE" '

```

```

FTWRITE fwrite foutf
NB. wd 'mb info *Будет записан текстовый файл с именем',LF,LF,'
"ftwrite.txt" ',tft
NB. foutame=:wd 'mb save "" "" "text (*.txt|*.)" '
NB. wd 'mb seve "" ""
)
NB. =====
NB. Ввод данных из файла JSON
NB. Определение рабочего каталога
fjs_wcat_button=: 3 : 0
wjcat=:wd 'mb dir "Рабочий каталог" '
1!:44 wjcat
workd=:1!:43 ''
NB. wd 'cc workdir static'
wd 'set workdir visible 1'
wd 'set workdir font "Lucida Console" 14'
wd 'set workdir text Рабочий каталог: ',workd
)

NB. =====
NB. Вывод данных в файл JSON
fjs_tojfile_button=: 3 : 0
FormBl ''
)
NB. =====
NB. inpnew from 82
NB. Конвертер текстового файла UTF-8 к формату J
os2j=: '_' ('-' ss ])} ] & ( ' ' (LF ss ] ) } ] ) NB. аргумент - текстовая
строка из файла
j2os=: ('-' ('_' ss ])} ] )
NB. =====
NB. Формирование файла JSON
NB. 1. Создается заголовок - требуется имя набора
NB. 2. Формируется блок - строка 1(аргумент) 'имя';Значения
NB. 3. Запрашивается комментарий к строке 1
NB. 4. Запрашивается завершение блока 1
NB. 4.1 Если "да"
NB. 5. Запрашивается комментарий к набору
NB. 6. Если "нет"
NB. 6.1 Повторить пп 2...4.1
NB. =====
FormBl=: 3 : 0
wd 'pc fb dialog escsclose closeok; pn Формирование JSON объекта'
wd 'cc t01 static; cn Заголовок набора'
wd 'cc head0 edit'
wd 'cc h0 button; cn Принять имя набора'
wd 'set head0 text *','Имя набора'
wd 'cc exit button; cn Завершить?'
wd 'set exit visible 0'
nobl=:0
wd 'cc noset static'
wd 'set noset text *','Набор № ', ": nobl NB. =: >: nobl
wd 'set noset visible 0'
wd 'pshow; pcenter'

```

```

wd 'setp wh 200 100'
NB. ch
bset=: ('Dataset');head0
''
)

fb_h0_button=:3 : 0
wd 'set head0 enable 0'
NB. !!!!!!!!!!!!!
NB. wd 'set fbnofin enable 0'      NB. Отключение выхода
name0=:head0
NB. bset=: (<nam0) (<0 1) } bset  NB. Изменено
bset=: 'Dataset ';name0
wd 'set h0 visible 0'
wd 'cc hlt static; cn Ввод набора данных'
wd 'bin zh'
NB.
wd 'cc fbnofin button; cn Сохранить набор'
wd 'set fbnofin font "Lucida Console" 10 '
wd 'bin ss'
wd 'cc h1 button; cn Ввод данных'
wd 'set h1 font "Lucida Console" 10 '  NB. 27_04
NB. wd 'set h1 focus'  NB. 27_01
NB. wd 'cc fbno button; cn Дальше'
wd 'bin zv'

NB. !!!!!!!!!!!!!
NB. wd 'set fbnofin enable 0'      NB. Отключение выхода
if. nobl > 0 do. wd 'set fbnofin enable 1'
end.
NB. new
NB. wd 'set h1 text Ввод данных ',": nobl
)

fb_h1_button=: 3 : 0
wd 'pc fbno dialog escsclose closeok ptop; pn Ввод набора'
wd 'cc tno static'
wd 'set tno text *', 'Набор номер: ', ": nobl=:>: nobl
cname=: 'Имя числовой переменной или в апострофах строка чисел'
wd 'cc tn0 static; cn Определить имя переменной'
wd 'cc fedname edit'
wd 'set fedname text Наименование переменной'
wd 'cc tn01 static; cn Идентификатор переменной с данными'
wd 'cc frdent edit'
wd 'set frdent focus'
wd 'set frdent text "Имя переменной, числа или текст в апострофах" '
wd 'set frdent cursor 1'
wd 'set frdent select'
wd 'cc tnosom static; cn Примечание к переменной '
wd 'cc fbcomm edit'
wd 'cc fbnook0 button; cn Запомнить'
NB. wd 'set frdent text *', cname wd 'set frdent focus'
NB. wd 'set frdent text Число'
NB. wd 'set frdent cursor 1'

```

```

NB. wd 'set frdent select'
NB. wd 'pcenter'          NB. 27_02
NB. wd 'psel fbno'
NB. wd 'set frdent select'
wd 'pmove 500 200 200 200'
wd 'pshow'
wd 'set frdent focus'
NB. wd 'pmove 900 334 200 200'ё
NB. wd 'set frdent text *', 'Имя переменной или Текст в апострофах '
''
)

fb_fbnofin_button=: 3 : 0
NB. Запись комментария
fb_fbnook_button ''
NB. wd 'pclose'
)

fbno_fbnook0_button=: 3 : 0
NB. Формирование очередного блока данных
NB. wd 'set fbnook0 enable 0'
namevar=:1 2 $ 'Имя';fedname
bsetno=: 1 2 $ 'Значения'; ". frdent NB. Перевод в число
fednamer=:fedname
fbcommr=:fbcomm
bsetcomm=:1 2 $ ('Примечание'); fbcommr
NB. =====Формирование раздела
bset=:bset, namevar, bsetno,bsetcomm
NB. -----

NB. wd 'psel fb' NB. Возврат к новой переменной
if. nobl = 1 do.wd 'cc setno static' end.

''
wd 'pclose'
)

NB. fbno_fbnow_button=: 3 : 0

NB. )

NB. fbno_fbnook_button=: 3 : 0
fb_fbnook_button=: 3 : 0
fjson=: enc_pjson_bset

wd 'pc fj dialog escfclose closeok ptop maxbutton; pn записать файл JSON'
wd 'bin v'
wd 'cc ttl static; cn Примечание к набору'
NB. wd 'bin ss'
wd 'cc setcomm editm'
NB. wd 'bin z'
NB. wd 'pshow'
NB. wd 'pcenter'
wd 'pmove 500 200 220 200'

```

```

wd 'setp wh 220 100'
NB. wd 'pmove 370 150 200 100'      NB.27_03
NB. wd 'pset wh wh 100 120'
NB. wd 'bin v'
wd 'set setcomm wh 200 50'
wd 'cc jfok button; cn Запомнить'
NB. wd 'bin s'
wd 'bin z'
wd 'pshow'
''
)
\

fj_jfok_button=: 3 : 0
cres=: setcomm
setc=: 1 2 $ ('Примечание к набору') ; setcomm
bset=:bset, setc
fjson=:enc_pjson_bset
fjname=:wd 'mb save "" "" "J (*.json)" '
fjname=:fjname, '.json'
fjson fwrite fjname
wd 'pclose'
NB. wd 'pclose'
wd 'psel fb'
wd 'mb info *Записан файл JSON', LF,LF,fjname
NB. wd 'set mb 200 100 100 100'
wd 'pclose'
)

FJread=: 3 : 0
wd 'pc fjr dialog escfclose closeok; rn Чтение файла JSON'
wd 'cc t0 static; cn Ввести файл'
wd 'cc fileopen button; cn Открыть файл JSON'
wd 'cc fcat combolist'
wd 'cc fcatok button; cn Посмотреть'
wd 'cc fcsel edit'
wd 'cc rout static; cn Результат'
wd 'set rout visible 0'      NB. Управление
wd 'pshow'
)

fjr_fileopen_button=: 3 : 0
p=. 'Data (*.json)|All Files (*.*)'
fjdat=: wd 'mb open1 "Открыть файл данных" "",(jpath '~1'),' " "',p,'"'
fwork=:freads fjdat
outfile=:dec_pjson_fwork
flist=: {. |: outfile
wd 'set fcat items *', ; > dquote each flist
)

fjr_fcatok_button=: 3 : 0
ftype=: >(" . fcat_select) { {. |: outfile
if. */'Значения' e. ftype do.
nsize=: $fenty=:>2{   {: |: outfile

```

```

wd 'set fcsel text *', 'Числовой массив, размер ;',": nsize
wd 'cc yesread button; cn Получить'
fl=:(<: ". fcat_select) { { : | : outfile
else.
fl=:(". fcat_select) { { : | : outfile
wd 'set fcsel text *', > fl
end.
)

fjr_yesread_button=: 3 : 0
NB. Получить выбранные численные значения из файла JSON
NB. (": fenty) 1!:2 <2
wd 'set rout visible 1'
wd 'set rout caption *', 'Имя ', (>fl), ' - Значения помещены в
переменную FROMJSON'
wd 'set rout font "Lucida Console" 12'
FROMJSON=: fenty
wd 'cc fexit button; cn Закреть программу'
wd 'set yesread visible 0'

)
fjr_fexit_button=:3 : 0
wd 'pclose'
)
NB. Модуль ввода-вывода для доступа к архивам JSON
cocurrent 'pjson'
fmtnum=: ,@('d<null>'&(8!:2))
fmtnums=: ' ' -.~ }.@, @(' , ' ,. >@{.@('d<null>'&(8!:1))@,.)
fmtint=: ,@('d<null>0'&(8!:2))
fmtints=: ' ' -.~ }.@, @(' , ' ,. >@{.@('d<null>0'&(8!:1))@,.)

sep=: }.@;@:(', '&,each)
bc=: '{' , '}' ,~ ]
bk=: '[' , ']' ,~ ]

ESC=: _2[\(CR;'r';LF;'n';TAB;'t';'"';'\';'\/';'\\/')
decesc=: rplc&(1|. "1 ESC)
encesc=: rplc&ESC
remq=: ]`({.@})@.('"' = {.)
isboxed=: 0 < L.
ischar=: 2=3!:0
isfloat=: 8=3!:0
isscalar=: 0 = #@$.
quotes=: '" '&, @(', &' "' )

false=: 0
true=: 1
NULL=: 0
cutcommas=: 3 : 0
y=. ' , ' ,y
m=. ~:/\y=' "'
m=. *./ (m < y=' , ' ) , 0 = _2 +/\ @ (-/)\ m <"1 ' {} [] '= /y
m <@dltb; . _1 y
)

```



```

dec=: 3 : 0
decl dltb ' ' (y I.@:e. TAB,CRLF)} y
)
decl=: 3 : 0
if. 0=#y do. '' return. end.
select. {. y
case. '{' do. dec_object y
case. '[' do. dec_array y
case. '"' do. dec_esc }.}:y
case. do. dec_num y
end.
)
dec_array=: 3 : 0
y=. dltb }.}:y
if. 0=#y do. $0 return. end.
if. -. y +./@:e. '"{[' do. ,dec_num y return. end.
decl each cutcommas y
)
dec_num=: 3 : 0
nms=. ;: 'false true null'
res=. 0 ". ' ' (I.y=',') } y
if. -. 1 e. ,nms E.&> <y do. return. end.
nos=. <;_1 ',',y -. ' '
'f t n'=. nos&(I.@:= <) each nms
res=. true t} false f} res
if. #n do. NULL n} res end.
)
dec_object=: 3 : 0
y=. }.}:y
if. 0=#y do. '' return. end.
dec_object1 &> a: -.~ cutcommas y
)
dec_object1=: 3 : 0
n=. y i. ':'
k=. remq dltb n {. y
v=. decl dltb (n+1) }. y
k;<v
)
enc=: 3 : 0
if. 1<#$y do.
  if. isboxed y do.
    enc_dict y
  else.
    bk sep <@enc"_1 y
  end.
elseif. isboxed y do.
  bk sep enc each y
elseif. ischar y do.
  enc_char y
elseif. isfloat y do.
  enc_num y
elseif. do.
  enc_int y
end.

```

```

)
enc_char=: quotes @ encesc
enc_num=: bk @ fmtnums`fmtnum @. isscalar
enc_int=: bk @ fmtints`fmtint @. isscalar
enc_dict=: 3 : 0
'rank>2 argument not supported' assert 2 = #y
'rank 2 argument must be a dictionary' assert (2 = {:$y) > 0 e. ischar &>
{"1 y
key=. "" ,each ({"1 y) ,each <"":'
val=. enc each {"1 y
rep=. ;key,.val ,each <',' ,LF
bc LF,(_2}.rep),LF
)
cocurrent 'base'

NB. ++++++++
NB. ++++++++
fjs_run ''

```