

# **PRADIS**

**UTILITY of arm**

**PROGRAM SET FOR THE AUTOMATION OF THE  
SIMULATION OF NONSTATIONARY PROCESSES IN THE  
MECHANICAL SYSTEMS AND THE SYSTEMS OF OTHER  
PHYSICAL NATURE**

**VERSION 4.3**

## • Content

<a href="#">0 Content.....</a>	<a href="#">2</a>
<a href="#">0 Designation.....</a>	<a href="#">3</a>
<a href="#">0 Requirements.....</a>	<a href="#">3</a>
<a href="#">0 Use of a utility.....</a>	<a href="#">3</a>
4.1key n: the creation of new catalog.....	4
4.2keys? and *: obtaining reference information.....	4
4.3keys! and: addition and the removal of the reference information of built-in HELP.....	6
4.4keys!, #, +: the start of the programs of user.....	7
4.5key p: automatic addition [PGO].....	10
4.6key u: the addition of user function.....	10
4.7key -: the removal of the programs of user.....	10
<a href="#">0 The further news about the addition of the programs of user.....</a>	<a href="#">12</a>
General information, the principle of processing “built in” libraries.....	12
Variable COMMON of the regions of [reshatelja], the files of common.inc, init.inc.....	12
Arguments of call the procedure- component of user (model, [PGO], [PRVP]).....	13
Arguments of the call of the model of element.....	13
Arguments of call [PGO].....	14
Arguments of call [PRVP].....	14
<a href="#">0 Fixing in MSVC ' of the programs of user.....</a>	<a href="#">15</a>
6.1 creation of the project of library.....	15
5.1to add into the project file with the initial code of component.....	17
6.3registration of component with the aid of arm.....	20
6.4to dispose in the project use FORTRAN of runtime as multiflow library.....	21
6.5to add in the design of way to the switch oned files of init.inc and common.inc.....	22
6.6to add into the project binding with the library of itgdl.lib.....	22
6.7to dispose the way of the resulting file of assembling in % DINSYS %/dinama/of pradis32/.....	23
6.8to establish cantilever solver as the check-out application for the library, and also arguments of its call.....	24
6.9use of a diagnostic routine.....	25
6.10retention of project and [peresozdanie] of the library of component from arm.....	26

## • Designation.

It is possible with the aid of the procedure of the maintenance the system catalog PRADIS ARM:

- to obtain brief information about the current composition of the libraries of complex;
- to obtain operational reference information about different components of complex;
- to obtain the built-in reference information on some additional themes (not relating to the components of libraries);
- to add and to move away reference information on the themes and the components;
- to add modules into the libraries of complex and to exclude them from the composition of the libraries of complex;
- to create the file of new, empty system catalog PRADIS.

## • Requirements.

- Windows 2000, or Windows XP;
- MS Of visual Of studio 6.0 SP3 (C/Of c++ compiler, service of pack e it is required). With the installation must be established the variable environments, necessary for the starting in the batch mode (in particular, LIB, MSDevDir). The medium of development is necessary only for adding the modules to the libraries of complex;
- Digital Of visual Of fortran 6.0a (it is integrated by c MS Of visual Of studio with the installation). Compiler is necessary only for adding the modules to the libraries of complex;
- not less than 100 mb. of vacant place on the hard disk;
- must be correctly established and work Win32 PRADIS in the configuration “cantilever solver”, or “window pradis32”. Must be established the variable of environment DINSYS, and also all necessary resources and usable codes.

## • Use of a utility.

After installation PRADIS, the utility ARM.EXE is located in % DINSYS % \ of dinama \ of pradis32, where % DINSYS % - the value of the variable of environment DINSYS, the name of the disk, where is established BY PRADIS, for example c:. With the starting without the parameters the utility issues brief information about the possible modes of operation (first line it contains prompt with the call from the command line of session DOS):

```
the c:\dinama\pradis32 of >arm.exe
Use: arm [of <[kljuch]> of <[imja]1> [of <[imja]2> [of [imja]3...
[of [imja]N]]]
```

Procedure of working binary catalog PRADIS.

```
<[kljuch]>
? information on the components, which are contained in
the binary catalog is derived
+ it includes components in binary catalog and builds
dynamic [plugin]- libraries, if it is possible.
```

```

        If is not assigned <[imja]1...N>, then it attempts to
connect the templates
        from the file of templet.txt in the current catalog.
p automatically builds the dynamic of [plugin]
        library include components in binary catalog.
u adds functions into the user library of user.lib.
        # simply are built dynamic [plugin]- libraries, if it
is possible
        ! are included components in the binary catalog
        If is not assigned <[imja]1...N>, then it attempts to
connect the templates
        from the file of templet.txt in the current catalog.
- it excludes components from the binary catalog
* is derived contents of the built-in aid
        <[imja]1...N> not is applicable to this key
n of [sozdajot] empty binary catalog in the current
directory
        <[imja]1...N> not is applicable to this key
<[imja]1...N>
        the names of the inquired components

```

As can be seen from information, the first required parameter with the starting of utility must be the key of the regime, which can assume one of six values. Further (for the keys?, +, #! and) can follow the names of the inquired components or additional themes of reference information. Detailed description of the regimes of use is given in the following points.

#### 4.1 key n: the creation of new catalog.

With the call with the key n, the utility of arm creates in the current directory the file of empty binary catalog PRADIS, called armctlg. If file with this name already exists in the current directory, will be given out communication about the error. This key can prove useful only to very moved developers of the libraries of complex. The created with its aid file of armctlg can be placed into the catalog % OF DINSYS % \ of dinamа \ of sysarm \. Utility ARM.EXE does extract reference information from this file; therefore after this replacement information will prove to be empty (keys? and \*). After replacement it will be possible to fill catalog “from zero”, with the aid of ARM.EXE, causing it with the keys + and! (see further). In this case the utility ARM.EXE will place the reference information into the file % OF DINSYS % \ of dinamа \ of sysarm \ of armctlg.

#### 4.2 keys? and \*: obtaining reference information.

For obtaining the brief information about the current composition of complex it is possible to use commands with the keys? and \*. Upon command

```
> ARM?
```

at the display screen appears brief information about all included in the composition of complex modules - models of elements, programs of the calculation of output variables, programs of the realization of graphic means and programs of mapping. Besides output to screen the information

is always duplicated up in the text file SYSPRINT.TXT of the current catalog. This file can be processed by the standard means of operating system (for example, by text editor).

Let us give a example. To user it is necessary to find, as is called the model of the element, which realizes girder ideally elastic element, and to obtain more detailed reference information about this model. We obtain information about all included in system catalog modules. The fragment of file SYSPRINT.TXT, which interests user in this case, appears as follows:

.....

**Contents of the catalog of the models of the elements:**

**And m I to r and t to about e n and z n AU and e**

.....

**BAL3DK three-dimensional elastic girder element ("short" beam)**

**BALKA girder ideally elastic element.**

**BELT the characteristic of belt, assigned tabular**

.....

It is evident from the obtained information that the name of the interesting us element - BALKA. For obtaining the further information with this model, let us introduce the command

**> ARM? BALKA**

On the screen and into the file SYSPRINT.TXT in this case will be brought out the further information about the model of element BALKA.

Besides obtaining of information with th modules e included in complex is a possibility to obtain the index of that build it Of hELP'a (name all possible themes, on which it is possible to obtain reference information). Upon command

**> ARM \***

is derived the list by the fact build it Of hELP'[a]. the fragment of this list it appears as follows in the file SYSPRINT.TXT:

.....

**The list of themes, on which there is a reference information:**

.....

**ATRC BAL3DJ BAL3DK BALKA BELT**

**BELTV BLOK BORDER BRK BUKA**

**C CIL3DC CMASS COS3E CYLDR**

.....

In this list, besides the enumeration of all module- component, included in composition PRADIS and having additional reference information, are the titles of reference information, also, about some other questions, for example, a information about the program of integration SHTERM, a information about the regime DEBUG, etc.

Additional reference information, for example, about the program of integration, can be obtained, after introducing the command

> ARM? SHTERM

### 4.3 keys! and: addition and the removal of the reference information of built-in HELP.

As it was said above, in the information ARM besides information about the module-components, included in composition PRADIS and which have additional reference information, is a information, also, about some other questions. Information on separate additional questions can be moved away from the information by causing ARM with the key -, for example

> ARM - SHTERM

After this call the information about the theme SHTERM is moved away from the information and ceases to be accessible for the commands ARM \* and ARM?. In the latter case (if with the key? inquires the already remote theme) will be brought out communication, that the information about the theme (for example, SHTERM) is absent from the catalog.

In order to add information on certain additional question into the information, it is necessary to create text file with the added information with the expansion HLP. By the first line in the file must go the name of the added theme (coinciding with the name of file), preceded by the substring "of C SYSTEM "(it is required gap between SYSTEM and name of theme). The name of theme must be up to the moment of addition not occupied, i.e., there must not reveal upon command ARM \*. Further must follow the lines of information, anticipated by the substring C HELP. None of the lines in the file must exceed 72 symbols. Text in the Russian must be to coding WIN-1251. Example of the contained file (SHTERM.HLP):

```
C SYSTEM SHTERM
C HELP
C HELP THE NAME: Program of the integration of the system of the
differential
C HELP of the equations of the second order by Stormer's method.
C HELP
C HELP of the key parameters enumerated below required
C HELP appears only the parameter END. Values of the rest
C HELP of the parameters can be accepted on silence.
C HELP
C HELP THE TABLE OF THE KEY PARAMETERS
C HELP -----T- T -
C HELP of | of | of | the value of |
C HELP of | the name of | the assignment of the parameter of | on
|
C HELP of | of | of | to silence of |
C HELP +-+--+ +
C HELP of | END of | the time of the end of the integration of |
0. |
C HELP +-+--+ +
C HELP of | OUT of | the minimum step of the conclusion of the
results of calculating | 0. |
C HELP +-+--+ +
.....
```

```

C HELP +--+-- +
C HELP of | IGNORE of | indicates to the program of integration,
| of 0 |
C HELP of | of | to consider/0/, or there is no/1/recommended |
of |
C HELP of | of | by the models of elements the value of the step
of | of |
C HELP of | of | of the integration of | of |
C HELP L--+-- +
C HELP
C HELP THE ADDITIONAL REFERENCES:
C HELP DEBUG

```

After file with the expansion HLP is formed, it is necessary to cause ARM with the key!, which follows the name of file HLP without the expansion. A call ARM must be accomplished, being located in the directory, where is located BY HLP file. Example of the call:

**> ARM! SHTERM**

If call passed successfully, information from the file will be added into the information, and it is brought out on the screen. After this, it will be possible to it inquire, for example by the command

**> ARM? SHTERM**

If utility ARM does not be able to add reference information, will be brought out only communication about the error, with the description of its reasons.

## 4.4 keys!, #, +: the start of the programs of user.

In the complex PRADIS is realized the technology of the load of the built in ([plugin]) components of [reshatelja]: the models of elements, programs of the calculation of output variables and programs of graphic means. This means that the user can according to the specific rules create dynamic library with the realization of component (model/[PGO]/[PRVP]) and register it in the complex. After this component it will become accessible for the use in the complex on the level with the components entering the standard delivery. At the present moment the complex makes it possible to automatically [sgenerirovat] and to add the dynamic library of component (model/[PGO]/[PRVP]) on the basis of initial FORTRAN of the file, which contains the realization of the procedure of component. The starts in the complex of user component of initial FORTRAN of file it is carried out with the aid of the call ARM.EXE with the key +. As the argument is transferred the name FORTRAN of file (without the expansion) with the initial code of the procedure of component. File must be located in the current directory. Example of the call:

**> ARM + OF BALKA**

Let us transfer briefly requirements for FOR to file, necessary for the creation and the start in the complex of the component (additional details they are given in the subsequent points of this document):

- File must begin from the special commentary of the describing type component (model/[PGO]/[PRVP]), additional descriptive parameters (for example, a quantity of

degrees of freedom, etc), and massif of the lines of the reference information (it will be included in the information of complex on the component). In more detail the creation of commentary is described in the files of documentation from pradis \ of res \ of lsv\_pradis \ of pradis \ of docs \ of include \. Example of the commentary:

```
C MODEL BALKA: EXT=6, PAR=9, WRK=16, STR =1, ADR=1, IGN=2
```

```
C
```

```
C program is created 03/14/91 10:11 am
```

```
C the date of last correction 02/06/96 11:48 pm
```

```
C
```

```
C
```

```
C HELP girder ideally elastic element.
```

```
C HELP THE NAME: Girder ideally elastic
```

```
C HELP element with the small deformations.
```

```
C
```

```
C HELP THE FIELD OF APPLICATION: Mechanics.
```

```
.....
```

```
C HELP OF DEGREE OF FREEDOM:
```

```
C HELP 1- is progressive in the direction of the axis OX of  
the point A of element.
```

```
C HELP THE SPECIAL SITUATIONS:
```

```
C HELP if in the course of calculations the instantaneous  
length of the element
```

```
C HELP becomes equal to 0, is achieved emergency exit from
```

```
C HELP of model.
```

- Directly the commentary must follow FORTRAN the directive of the start of the title file of init.inc. With the installation of complex this initial file is copied into the catalog % OF DINSYS % \ of dinama \ of sysarm \ of plugin, where % DINSYS % - the value of the variable of environment DINSYS, the name of the disk, where is established BY PRADIS, for example c:. The start of file is necessary for the transfer to the dynamic library of the component variable COMMON of the regions of [reshatelja], described in the files of documentation from pradis \ of res \ of lsv\_pradis \ of pradis \ of docs \ of include \. At assembling of the library of component by utility ARM.EXE the file of init.inc will be found in the directory of the installation of complex. Example of the directive of the start of init.inc:

```
include "of init.inc"
```

- Further must follow the title of procedure the realizing user component (model/[PGO]/[PRVP]). In the current version of complex for each type of component (model/[PGO]/[PRVP]) is its strictly fixed set of the parameters of procedure. Examples of the titles of the procedures of components, and the descriptions of the types of their parameters:

Model:

```
      SUBROUTINE BALKA (THE I, Y, THE X, THE V, A, PAR, NEW,  
OLD, WRK)
```

```
.....
```

```
      REAL * OF 8 I, Y, THE X, THE V, A, PAR, NEW, OLD, WRK
```

[PGO]:

```
      SUBROUTINE AKLAB (NAMEX, THE I, THE X, THE V, A, INNER,  
EXT, PARX, WRKX,  
      , PAR, WRK, PARLR2)
```

```
.....
```

```
      CHARACTER * OF 8 NAMEX
```

```
      REAL * OF 8 I, THE X, THE V, A, INNER, PARX, WRKX, PAR,  
WRK, PARLR2
```



```

      INTEGER * OF 4 EXT
[PRVP]:
      SUBROUTINE OF THE X (XOUT, PAR, NEW, OLD, WRK, A,
DOFADDR, NDOF)
      .....
      REAL * OF 8 XOUT, PAR, NEW, OLD, WRK, A
      INTEGER * OF 4 DOFADDR, NDOF

```

In more detail the sense of the arguments of the procedures of components is described in the files of documentation from pradis \ of res \ of lsv\_pradis \ of pradis \ of docs \ of include \ and it is below in this document.

- Immediately after the title of procedure (before the description of the types of the arguments of procedure, commentaries and any other directives!) the special DIGITAL- specific commentary, which ensures the export of the procedure of component from the dynamic library, must follow. The procedure cannot be caused will be without this commentary. Example of the commentary:  
**!DEC\$ ATTRIBUTES DLLEXPORT::BALK**
- After the title of the procedure of component and special commentary, in the division of the description of arguments and variables of procedure, must be located BY FORTRAN the directive of the start of the title file of common.inc. With the installation of complex this initial file is copied into the catalog % OF DINSYS % \ of dinama \ of sysarm \ of plugin, where % DINSYS % - the value of the variable of environment DINSYS, the name of the disk, where is established BY PRADIS, for example c:. The start of file is necessary for the correct description of the types variable COMMON of the regions of [reshatelja], described in the files of documentation from pradis \ of res \ of lsv\_pradis \ of pradis \ of docs \ of include \. At assembling of the library of component by utility ARM.EXE the file of common.inc will be found in the directory of the installation of complex. Similar of means, in the current version of complex it is not possible to describe the variables of COMMON- regions independently, and it is possible to only use them into the tele- procedures. All necessary variables are already described into common.inc. Example of the directive of the start of common.inc:  
**include "of common.inc"**
- Should be focused attention that the dynamic library of [plugin] of component will be gathered from only FORTRAN of file, without the binding with any user auxiliary libraries. If for the procedure of component are necessary any subprograms (with exception of standard, from runtime FORTRAN), the code of all these subprograms should be put into the same initial FORTRAN the file, where the procedure of component is located.

Thus, after call ARM with the key + and the names of initial .FOR of files as the arguments (for example, [imja]1:. of [imja]N), ARM will assemble from the initial files the separate dynamic libraries of components. Assembling is accomplished by a call MS Of visual Of studio ' OF SP3 and Digital FORTRAN 6.0a. These programs must be established. Also must be determined the variable of the environment Of mSDevDir, value of which is way to the catalog of bin, where is located the usable file of msdev (for example, the C:\Program Of files \ Of microsoft Of visual Of studio \ Of common \ Of mSDev98). This variable is established with the installation MS Of visual Of studio 6.0, if we at the end to its installation confirm the installation of the variables of environment. In the case of successful assembling, library of [imja]1.dll... of [imja]N.dll they will be copied into the catalog % OF DINSYS % \ of dinama \ of pradis32. In that flowing of directory will appear the files of [imja]1.log... of [imja]N.log, th media during the assembling e containing conclusion. To the period of assembling library that flowing of directories re- establishes in % DINSYS % \ of dinama \ of sysarm \ of plugin. During the unsuccessful assembling in this of directory will remain all files, generated by medium during assembling of library, which will make it possible to diagnose error.

Besides assembling of libraries on the key + OF ARM.EXE is produced the necessary registration of component and its library in its configurative files. Also in the information remains information from initial .FOR of file (from the special commentary in the beginning of file). After successful completion ARM on the key + this information will be brought out on the screen, and it will also become accessible with the call ARM with the keys \* and?. In the case of unsuccessful completion ARM instead of this will be brought out the communication about the error.

If with the work on the the user of [plugin] by component, no longer it is necessary to re-create from the initial file dynamic library, but it is necessary to only re-register reference information, or the descriptive parameters (for example, a quantity of degrees of freedom, etc), it is possible to use ARM with the key!. Should be focused attention what to cause ARM with the key! for FORTRAN of file with the code of component it follows only if to this ARM already successfully it was caused with the key + for the same file! I.e., key! let us use only, if it is necessary to anew report to complex contents of the special commentary in the beginning of file, when the library was already assembled and registered and the code on FORTRAN does not require changes. Example of call ARM with the key!:

**> ARM! BALK**

Finally, if it does not be necessary to anew include the information part FORTRAN of file from the special commentary, but it is necessary to only [peresobrat] dynamic library after a change in the code on FORTRAN, it is possible to use ARM with the key #. Should be focused attention that to cause ARM with the key # for FORTRAN of file with the code of component follows only if to this ARM already successfully it was caused with the key + for the same file! Sequential call for the initial file ARM with the keys! and # it will not have the same effect as call ARM with the key +! With the work with the code of the procedure of component always one should begin from the call ARM with the key +! Example of call ARM with the key #:

**> ARM # OF BALK**

## **4.5 key p: automatic addition [PGO].**

With the aid of this key it is possible to add into the complex of [plugin] [PGO], having only initial file [PGO] on FORTRAN. All necessary operations on the generation of dynamic libraries and the start of components in binary catalog are performed automatically.

For the correct work of this key system variable DINAMA\_FORTRAN it must contain the complete name (from [putjom]) of the compiler of FORTRAN.

## **4.6 key u: the addition of user function.**

With the aid of this key it is possible to add into the complex its eigenfunctions for the use in [pluginakh], having only initial file on FORTRAN. Functions are added to the library DINAMA \ of lib \ of user.lib.

## **4.7 key -: the removal of the programs of user.**

With the aid of the call ARM with the key of – it is possible not only to exclude from the information of complex information about additional questions (as it is described above), but also to completely exclude from the information and the configuration the program- components (models/[PGO]/[PRVP]) of user, which were previously added with the aid of the key +. After call ARM with the key - does cease to be accessible the information about the component on the keys \* and?, and will also cause error the attempt to use the excluded component in the text of task for PRADISlang. Only of what does not make WITH ARM – it do not move away from the catalog % OF DINSYS % \ of dinamа \ of pradis32 the dll- files of the dynamic libraries of the excluded components. Libraries themselves already more be caused complex will not be, and if necessarily user can remove them by hand. Example of call ARM with the key -:

**> ARM - BALKA**

- **The further news about the addition of the programs of user.**

In this point is given a certain extended information, which can be useful with the start in the complex of user [plugin] of components (models/[PGO]/[PRVP]). In essence are described the principles of working by the complex of [plugin] of the libraries, which contain the added components. Understanding that how complex it works from [plugin] by components, can prove to be useful with the permission of the problems of the user programs appearing with the start.

## **General information, the principle of processing “built in” libraries.**

The technology of dynamic incorporation into the solver PRADIS was developed for the possibility to add the new models of elements, [PGO], [PRVP] without a change in the existing code of [reshatelja], and without its recompilation. The configuration of the built in libraries of complex is contained in the files % OF DINSYS % \ of dinama \ of sysarm \ of armctlg (binary catalog) and % DINSYS % \ of dinama \ of sysarm \ of plugin\_repository.xml ([repozitarij] of plugin). In the file of catalog for all components of complex to be contained reference information, and also different descriptive parameters of components (such as a quantity of degrees of freedom, etc). [Repozitarij] in simple XML size contains the collection of the descriptions of the built in dynamic libraries (plugin), and component (models of elements, [PGO], [PRVP]), which in them are realized.

The principle of working the built in components is the following. Prior to the start of calculation the solver loads the file of [repozitarija] of plugin. Then is carried out load into the address space of the process of [reshatelja] of dll of the files of the dynamic libraries of components, described in [repozitarii]. The files of dll of libraries search for in % DINSYS % \ of dinama \ of pradis32. Further for the described in [repozitarii] models, [PGO] and [PRVP] are built the tables of the correspondence of the numerical identifiers of components (addresses of records from armctlg), and the structures, which describe these components. The most important member in each structure of – the address of the global procedure, which realizes component (model of element, [PGO] or [PRVP]). During reading from [repozitarija] of identifier and name of the function of component, the address of the corresponding function searches for in the module of that of already loaded of dll of library, and it is recorded in necessary type table (model/[PGO]/[PRVP]), with its identifier. In the process of calculation before the turning to the model of element, [PGO], or [PRVP] is carried out the search in the appropriate table of the loaded components on the identifier. If the corresponding to identifier component is found – it is caused procedure with the appropriate address.

## **Variable COMMON of the regions of [reshatelja], the files of common.inc, init.inc.**

As described in the files documentations from pradis \ of res \ of lsv\_pradis \ of pradis \ of docs \ of include \, component of complex (model/[PGO]/[PRVP]) can use variable COMMON of the regions of [reshatelja] (not determined, NOTAT, GRCONF), which can be required in the course of computation. The dynamic library of [reshatelja] and the dynamic libraries of the components of – different modules, and have its own versions of global variables (in particular COMMON of regions). Because of this it is not possible by the usual method to determine in the initial file of the procedure of the component COMMON of the regions, analogous TO COMMON to the

regions of [reshatelja], and to expect that their values variables will change [reshatelem] on the course of computation. Solver works with COMMON by the regions (memory), located in its module.

In order to ensure in the libraries of the components of complex the access to variable COMMON of the regions of [reshatelja] is used the following method. In the initial code the components are determined BY COMMON of the regions, analogous to the regions of [reshatelja], but which consist of the indicators to the variables (FORTRAN POINTER). Before beginning calculation for each dynamic library of components the solver causes the special initializing function realized in it, into which it transfers to its variable COMMON of regions. Inside the function the transmitted variables “tie” to the indicators from COMMON of the regions, declared the initial code of component. Practically after this the variable- indicators COMMON of the region of [plugin] of library begin to indicate the memory, in which are located the variables of the corresponding COMMON region of [reshatelja]. Now variable COMMON of regions it is possible the usual method to use inside the procedure of component.

In order to free user from the need for adding into the initial code of each component of the declaration COMMON of regions from the indicators, and their also initializing functions are realized the files of init.inc and common.inc. These files are copied into the catalog % OF DINSYS % \ of dinam \ of sysarm \ of plugin with the installation of complex. To user it suffices to include init.inc in the initial code of the procedure of component to its title, and in its library will declared COMMON of region from the indicators, and appear their standard initializing function, which the solver will cause. In order to use variable COMMON of fields into tele- procedures, it is necessary to add in the field of describing the variables the start of the file of common.inc. Are there correctly described the variable- indicators COMMON of regions. This file helps to avoid errors.

## **Arguments of call the procedure- component of user (model, [PGO], [PRVP]).**

As it was spoken above, the procedures of the components of each type complex (model, [PGO], [PRVP]) have their, strictly fixed collection of arguments. Let us transfer arguments, their types and sense in the procedures of components.

Arguments of the call of the model of element.

- The I: the vector of forces (moments) for the element.
- Y: the jacobian of the model of element.
- The X: the vector of displacements of the units of dimensionality EXT+ENT. It is not used with ADR=2, ADR=3.
- The V: the velocity vector of the units of dimensionality EXT+ENT. It is not used with ADR=3.
- A: the G-vector of the units of dimensionality EXT+ENT.
- PAR: the massif of the parameters of model.
- NEW: the vector “new state” of model.
- OLD: the vector “old state” of model.
- WRK: working massif for the model of element.

All arguments have a type REAL \* 8. It is in more detail about the value of arguments (with exception of the X, the V, A), and the parameters of the passport of the model of element (EXT,

ENT, ADR) see documentation on the development of components in pradis, located into pradis \ of res \ of lsv\_pradis \ of pradis \ of docs \ of include \.

Arguments of call [PGO].

- NAMEX: the name of model or [PRVP], connected with [PGO]. Massif of gaps, with the values of the parameters of passport VPS=0 and EXT=0 (fixed graphic means). Has type CHARACTER \* 8.
- The I: the vector of forces (moments) for the element. Has type REAL \* 8.
- The X: the vector of displacements of the units of the model, connected with [PGO], dimensionality EXT. It is not used with the value of the parameters of passport VPS=0 and EXT=0, or with the value of the parameter UNV>0 (in this case it is used vector INNER). Has type REAL \* 8.
- The V: the velocity vector of the units of the model, connected with [PGO], dimensionality EXT. It is not used with the value of the parameters of passport VPS=0 and EXT=0, or with the value of the parameter UNV>0 (in this case it is used vector INNER). Has type REAL \* 8.
- A: the G-vector of the units of the model, connected with [PGO], dimensionality EXT. It is not used with the value of the parameters of passport VPS=0 and EXT=0, or with the value of the parameter UNV>0 (in this case it is used vector INNER). Has type REAL \* 8.
- INNER: the vector of the real numbers of the making sense degrees of freedom of the model of element, connected with [PGO]. It is not used with the value of the parameters of passport VPS=0 and EXT=0, or with the value of the parameter UNV=0 (in this case they are used the vector of the X, the V, A). Has type REAL \* 8.
- EXT: a quantity of degrees of freedom of the model of element, connected with [PGO] (length INNER). Has type INTEGER \* 4.
- PARX: the vector of the parameters of the model, connected with [PGO]. It is not used with the value of the parameters of passport VPS=0 and EXT=0. Has type REAL \* 8.
- WRKX: the working vector of the model, connected with [PGO]. It is not used with the value of the parameters of passport VPS=0 and EXT=0. Has type REAL \* 8.
- PAR: the vector of the parameters [PGO]. Has type REAL \* 8.
- WRK: working vector [PGO]. Has type REAL \* 8.
- PARLR2: the vector of the parameters of the current layer of image. Has type REAL \* 8.

It is in more detail about the value of arguments (with exception of the X, the V, A), and the parameters of passport [PGO] see documentation on the addition of components to pradis, located into pradis \ of res \ of lsv\_pradis \ of pradis \ of docs \ of include \.

Arguments of call [PRVP].

- XOUT: the calculated output variable or the vector of the calculated output variables. Has type REAL \* 8.
- PAR: the massif of the parameters [PRVP]. Has type REAL \* 8.
- NEW: the vector “new state” OF [PRVP]. Has type REAL \* 8.
- OLD: the vector “old state” OF [PRVP]. Has type REAL \* 8.
- WRK: working massif for [PRVP]. Has type REAL \* 8.
- A: the common vector of the real variables of [reshatelja]. Has type REAL \* 8.
- DOFADDR: the massif of addresses in the vector A, along which are located necessary [PRVP] of the value of the degrees of freedom - displacement, speed, or acceleration. If the number of unit for [PRVP] into PRADISlang is assigned as [nomer]\_[uzla] of – of

displacement, if it is assigned as [nomer]\_[uzla]' - speed, and if it is assigned as [nomer]\_[uzla] " - acceleration. Has type INTEGER \* 4.

- NDOF: the size of vector DOFADDR. Has type INTEGER \* 4.

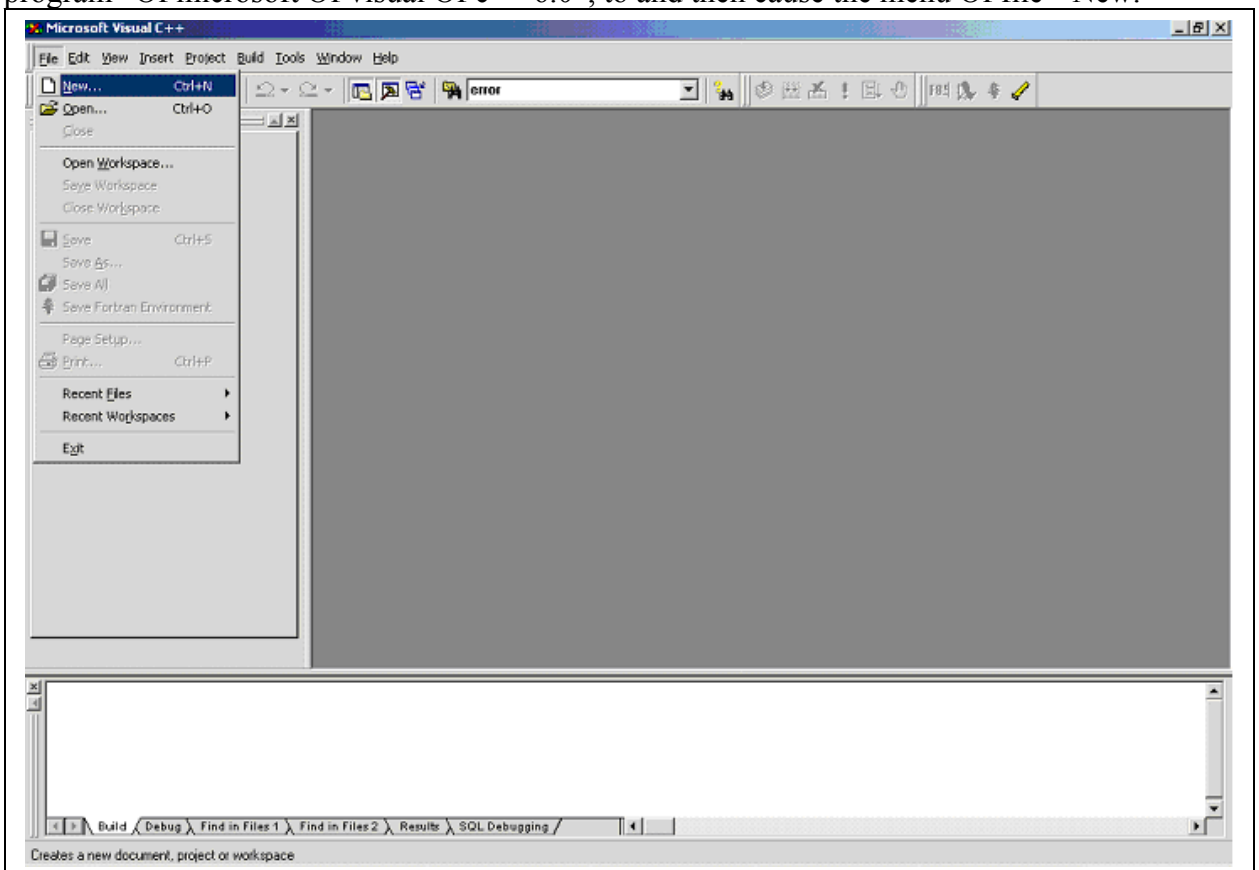
It is in more detail about the value of arguments (with exception A, DOFADDR, NDOF), and the parameters of passport [PRVP] see documentation on the addition of components to pradis, located into pradis \ of res \ of lsv\_pradis \ of pradis \ of docs \ of include \.

## • Fixing in MSVC ' of the programs of user.

In this point is described the list of the steps, which must be carried out user - the developer of the components of complex PRADIS in order to use MSVC ' for the more convenient search for the errors of compilation, and also fixing of its procedure of component into debug regime.

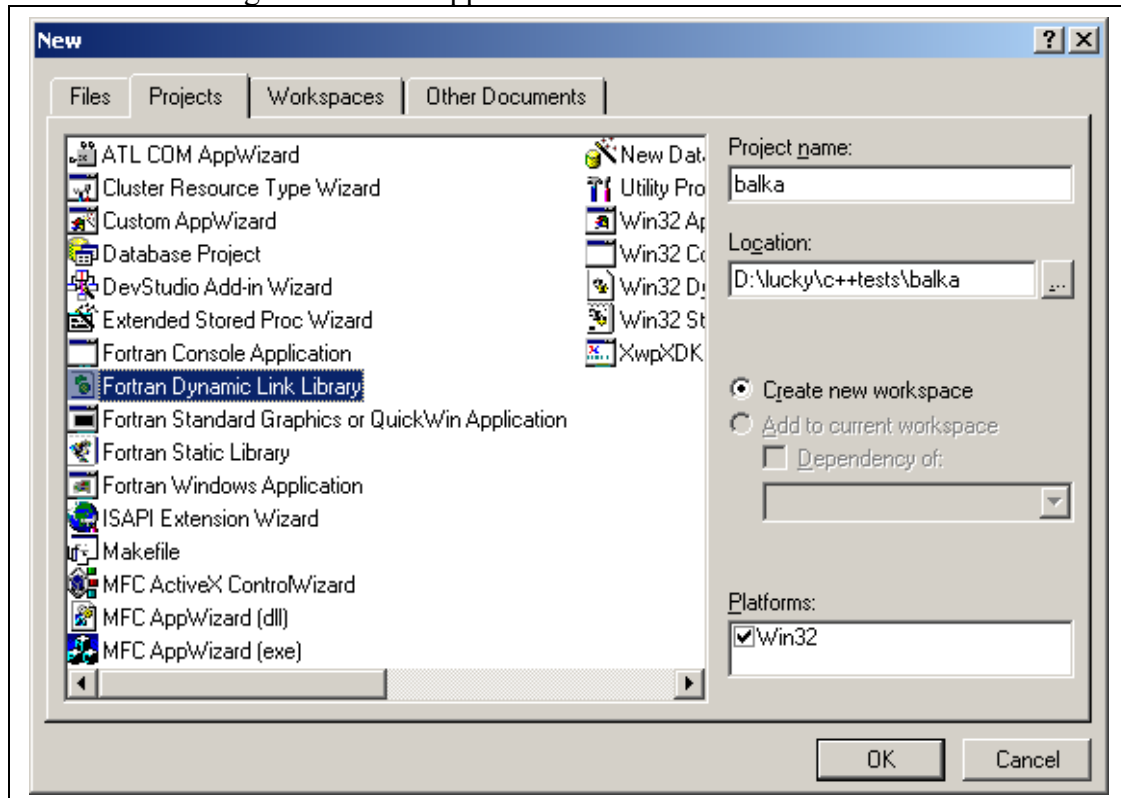
### 6.1 creation of the project of library.

For the work on the procedure of component in the medium MSVC ' should be, first of all, created the project of the dynamic component of library. For this it is necessary to open the program "Of microsoft Of visual Of c++ 6.0", to and then cause the menu Of file->New.



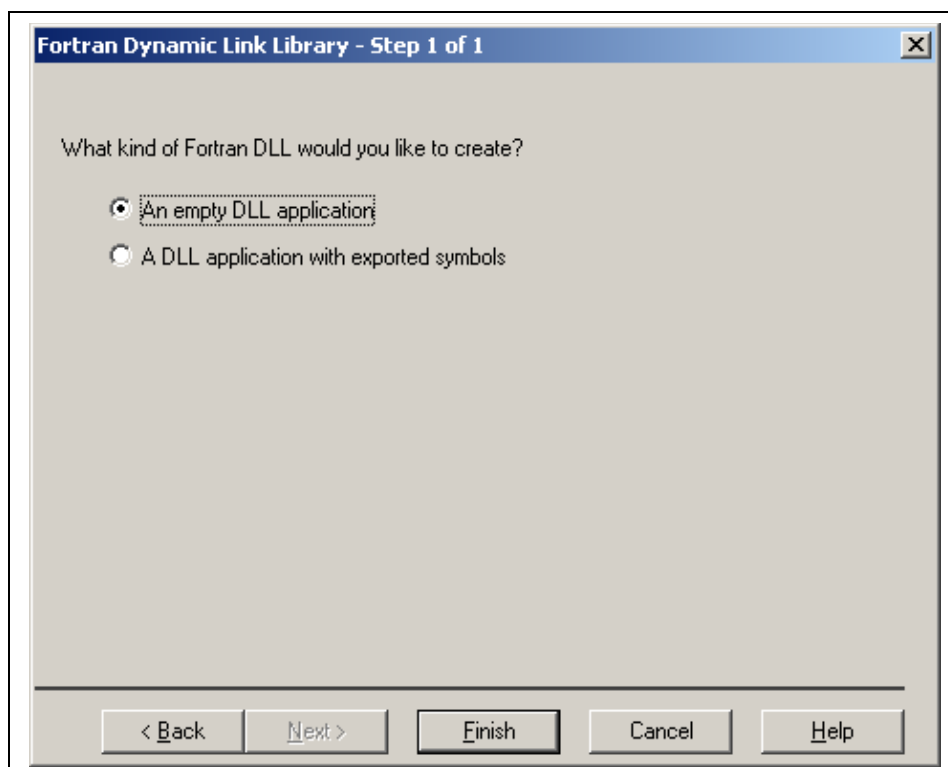
In the appeared dialog "New", in the supplementary sheet "Of projects" should be selected the point "Of fortran Of dynamic Of link Of library". In the window of the introduction "Of location" on the button "..." to select the directories, where project will be located. In the window of the introduction "Of project of name" to introduce the name of the component of complex. Name should be selected by the same, as will be called the created component

(model/[PGO]/[PRVP]), and under which it will be caused into PRADISlang. After the actions indicated the dialog “New” must appear thus:



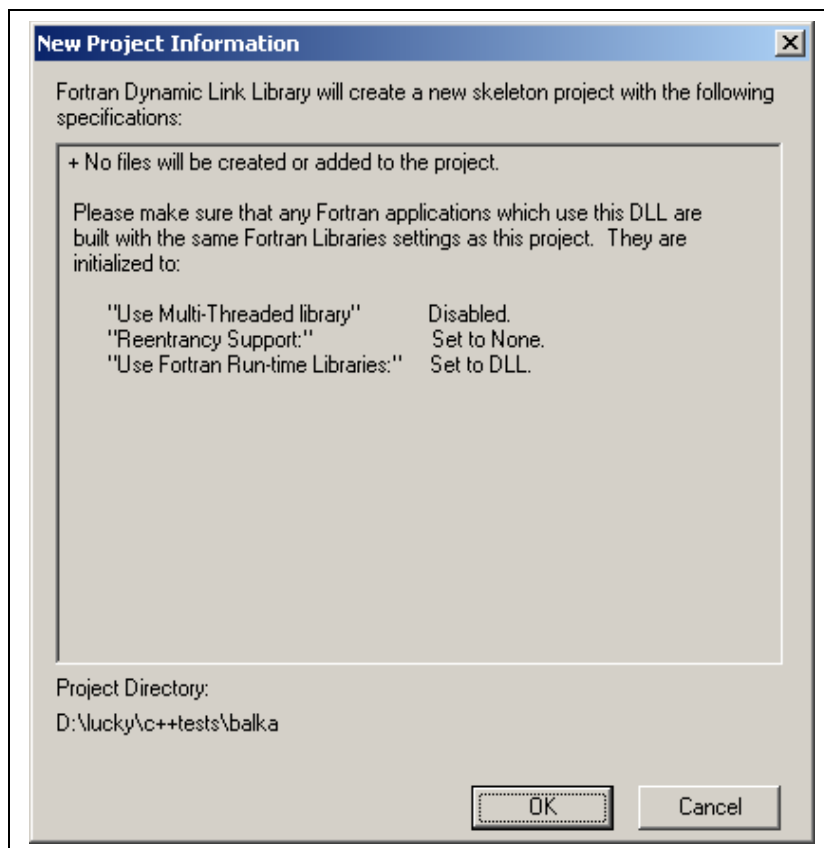
If in the supplementary sheet “Of projects” there is no point “Of fortran Of dynamic Of link Of library”, means it is not established correctly Digital Of fortran 6.0a, and it should be established.

After the selection indicated should be harvested “OK” in the dialog “New”. In the appeared then intermediate dialog should be selected point “An empty DLL of application” and harvested “Finish”.



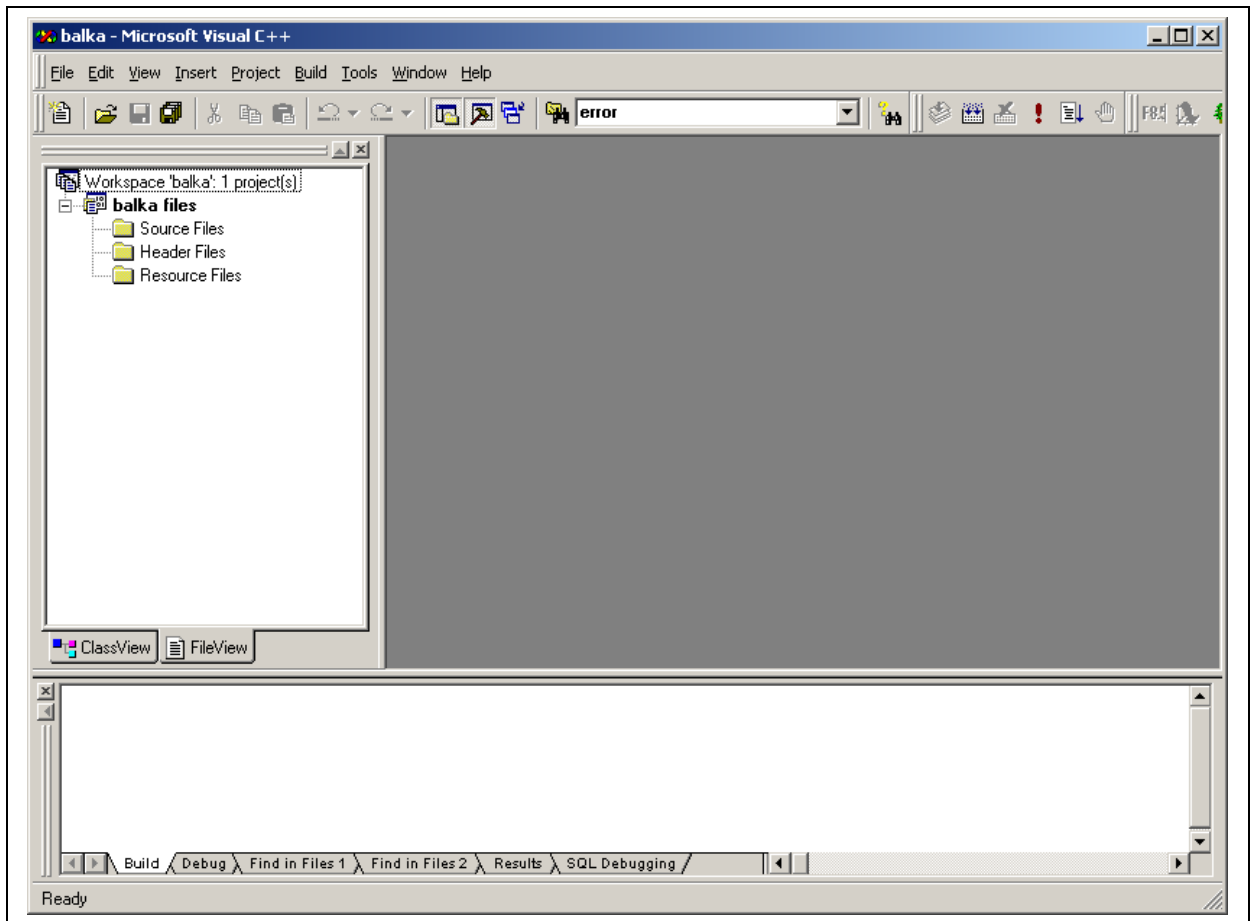


Then should be harvested “OK” in the dialog “New Of project Of information”.

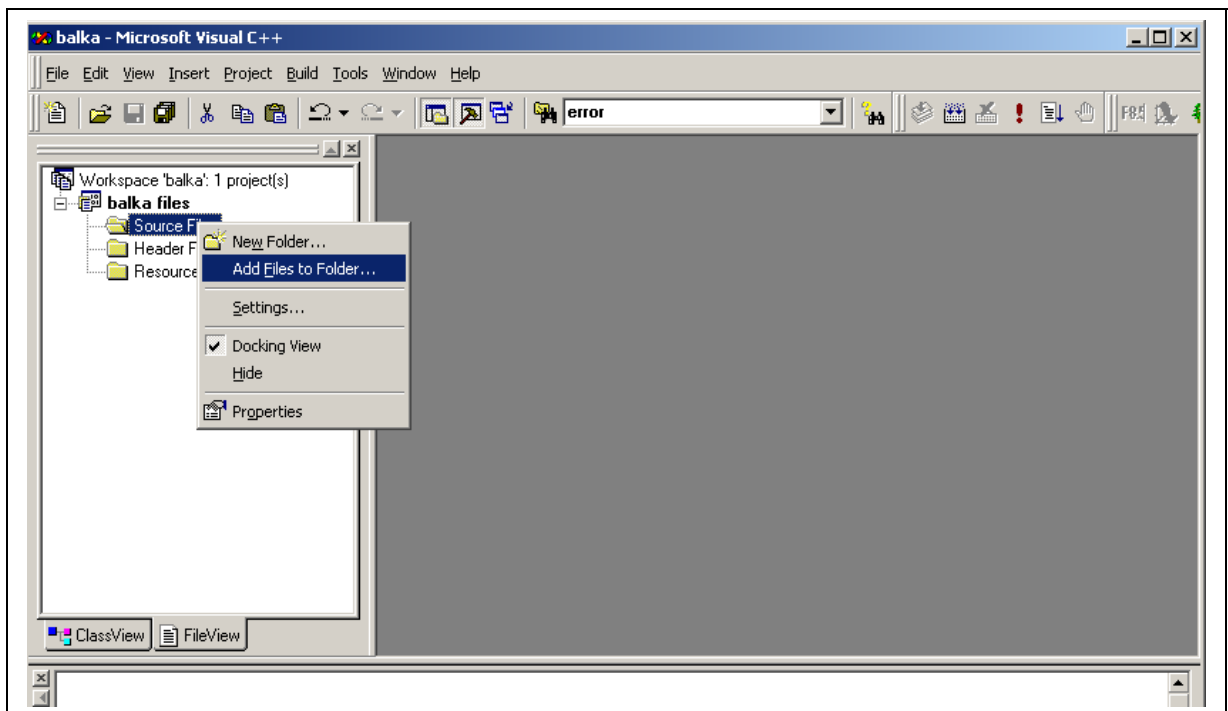


## 5.1 to add into the project file with the initial code of component.

To pass to the supplementary sheet “File Of view” of the in the form working space MS Of visual Of c++ (left cry of mouse on the inscription “Of file Of view”).

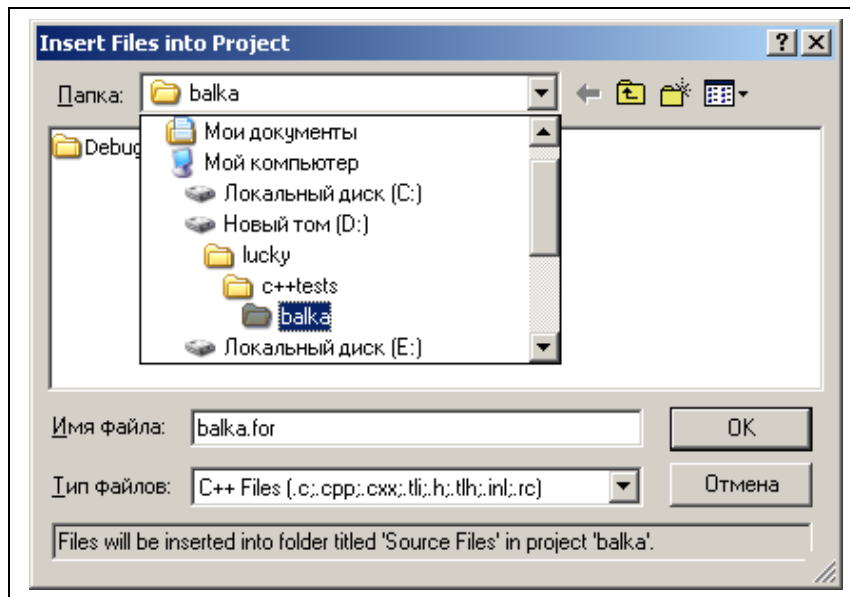


Then should be isolated by the left cry of mouse the unit “Of source Of files”, caused on it the contextual menu (right cry of mouse on the unit) and selected in it the point “Of add of files to of folder”.

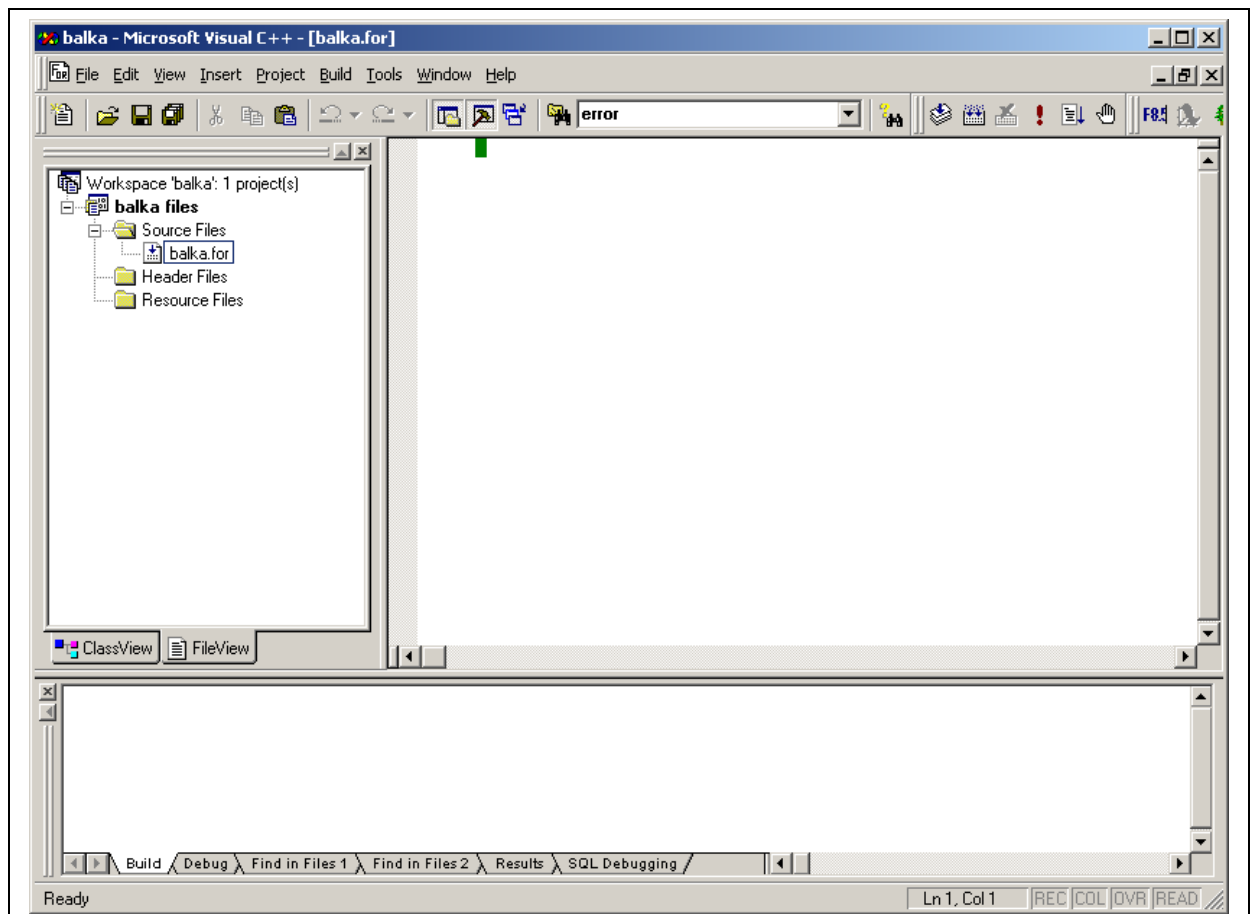


In the opened dialog of the selection of files should be introduced the name of file with the expansion of for. Name must coincide with the name of project. One should trace so that the file

would be created in the correct directory (where were located the remaining files of project), to and then harvest OK.



Further, if file did not exist earlier, will arise warning about the fact that the file does not exist. In it one should harvest “yes” (“Yes”). After this, the file will appear in the tree “Of file Of view” in the unit “Of source Of files”. Should be made on it a dual left cry mouse, and confirmed the proposal of studio to create file (if it did not exist earlier). The file will be opened after this, and studio will appear thus.

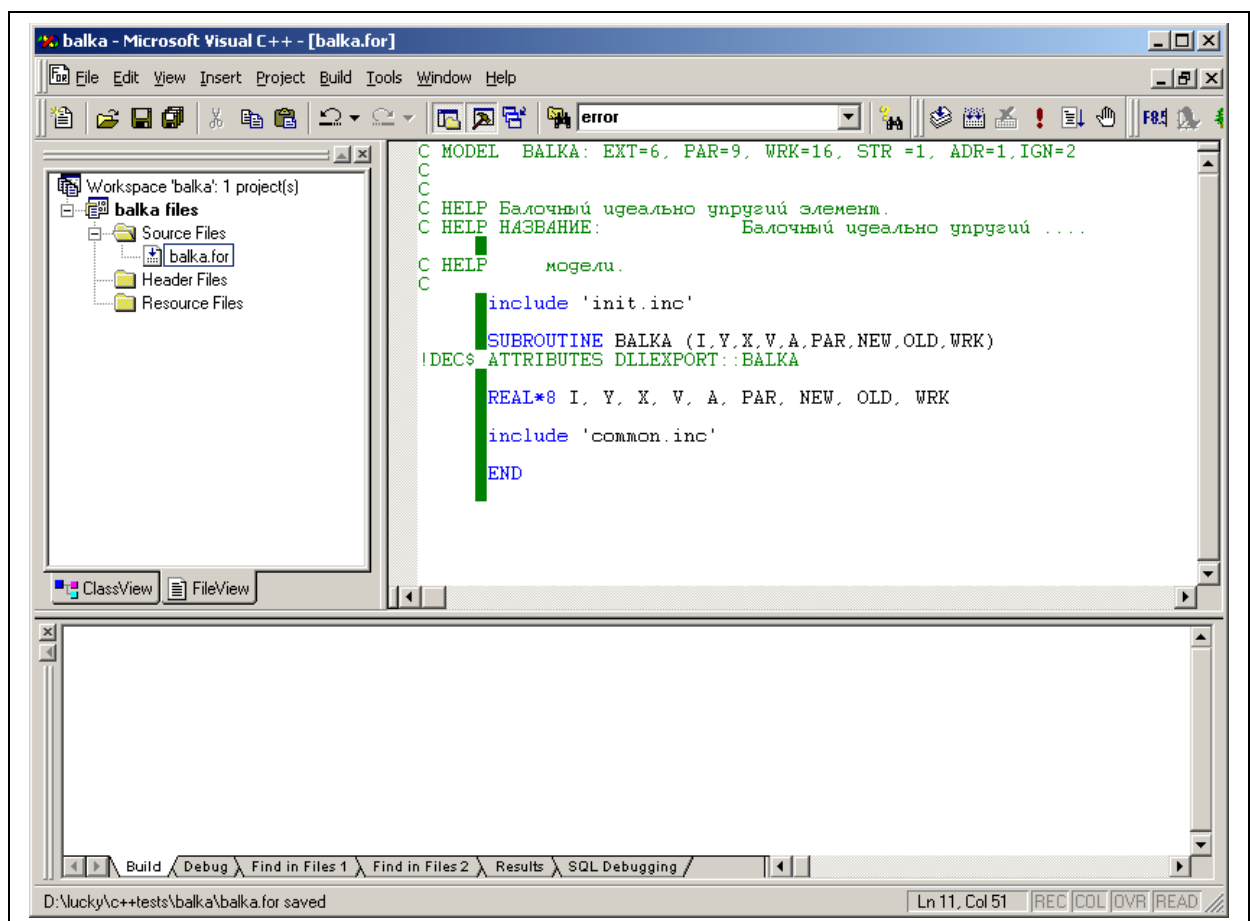


You will focus attention, that into the project one ought not to add no other initial files, except FOR of the file, whose name coincides with the name of the added component. Otherwise component, possibly, will not succeed in gathering and registering with the aid of the utility ARM, since the utility knows how to operate only with one initial file.

In accordance with the previously described point “keys!, #, +: the start of the programs of user.” one should introduce into the beginning of file

- the special commentary
- then the start of the file of init.inc
- then the title of the procedure of the component
- then the special commentary of the export
- then the description of the arguments of the procedure of the component
- then the start of the file of common.inc
- then the body of procedure and completing END

Then to preserve file by the combination of the keys For ctrl+S. In this case the studio will appear thus.



### 6.3 registration of component with the aid of arm.

This action is necessary in order to produce the correct registration of the created procedure of component in the configuration of complex. One should, for example, in the command line DOS pass into the catalog, where the initial file of the procedure of component was created. Then to cause arm with the key + and the name of initial file without the expansion as the argument. In the case of successful call the window of command line will appear thus.

```

C:\> Командная строка
Microsoft Windows XP [Версия 5.1.2600]
(C) Корпорация Майкрософт, 1985-2001.

C:\Documents and Settings\lucky>d:
D:\>cd "D:\lucky\c++tests\balka\"
D:\lucky\c++tests\balka>C:\dinama\pradis32\arm.exe + balka
1  C MODEL  BALKА: EXT=6, PAR=9, WRK=16, STR =1, ADR=1,IGN=2

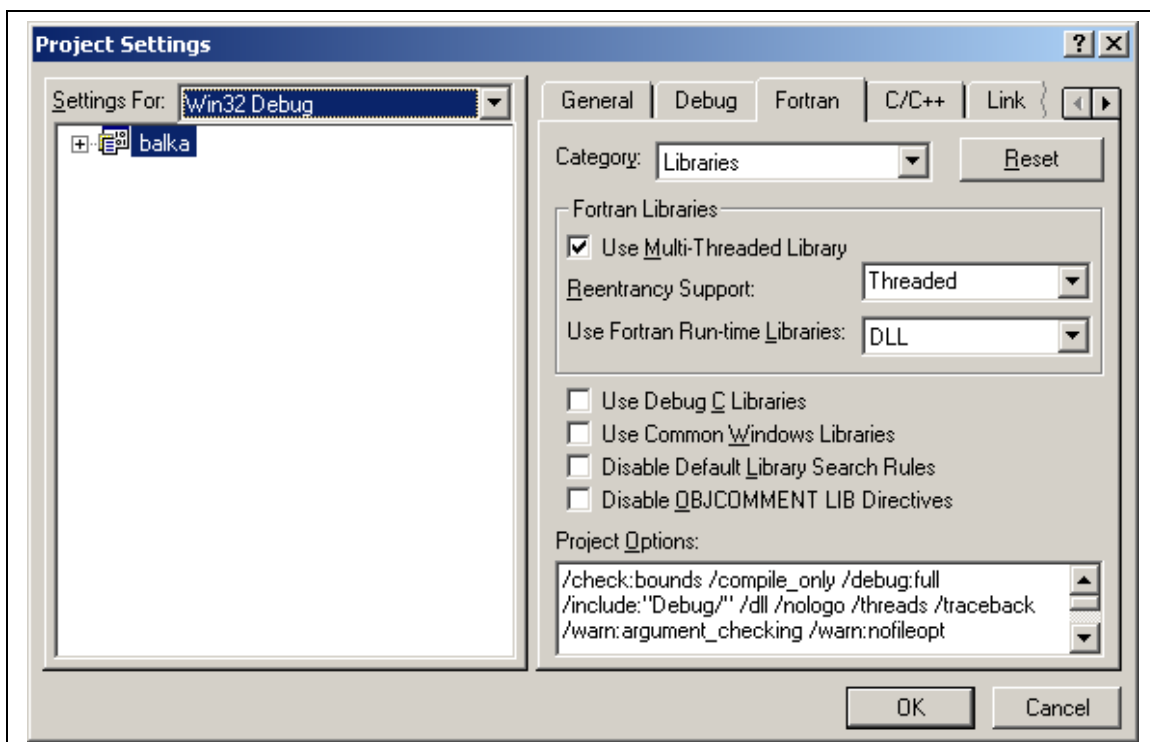
      Паспорт программы BALKА      добавлен в системный каталог

2  C
3  C
4  C HELP Балочный идеально упругий элемент.
5  C HELP НАЗВАНИЕ: Балочный идеально упругий ....
6
7  C HELP модели.
8  C
9  include 'init.inc'
D:\lucky\c++tests\balka>_

```

## 6.4 to dispose in the project use FORTRAN of runtime as multiflow library.

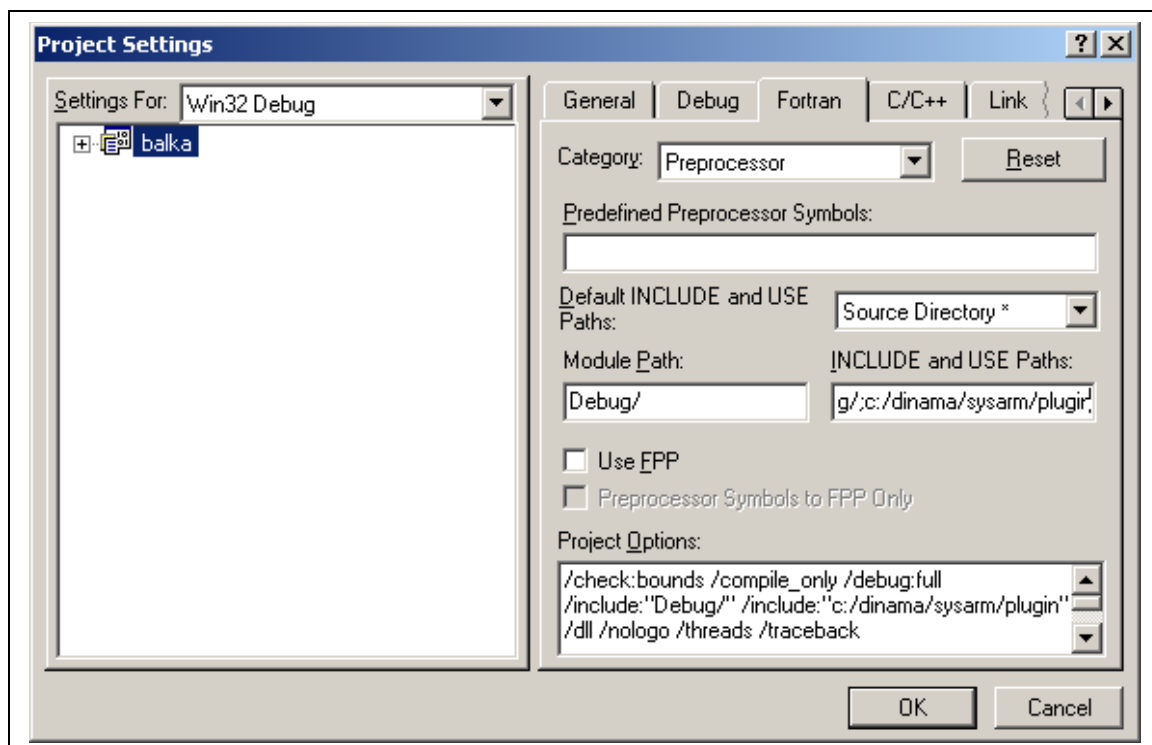
To make active a window Of visual Of c++ with the open project of component, and to cause the dialog of tuning project by the combination of the keys For alt+F7. Further to pass in the dialog “Project Of settings” to the supplementary sheet “Of fortran”, to select in [kombobokse] “Of category” the point “Of libraries”, to and then include [chekboks] “Of use Of multi-Threaded Of library”. One should trace so that the tuning would be established for the check-out configuration. In this case [komboboks] “Of settings of for” must be in the position “Of win32 Of debug”.



Further should be harvested “OK” in the dialog “Project Of settings”.

## 6.5 to add in the design of way to the switch oned files of init.inc and common.inc.

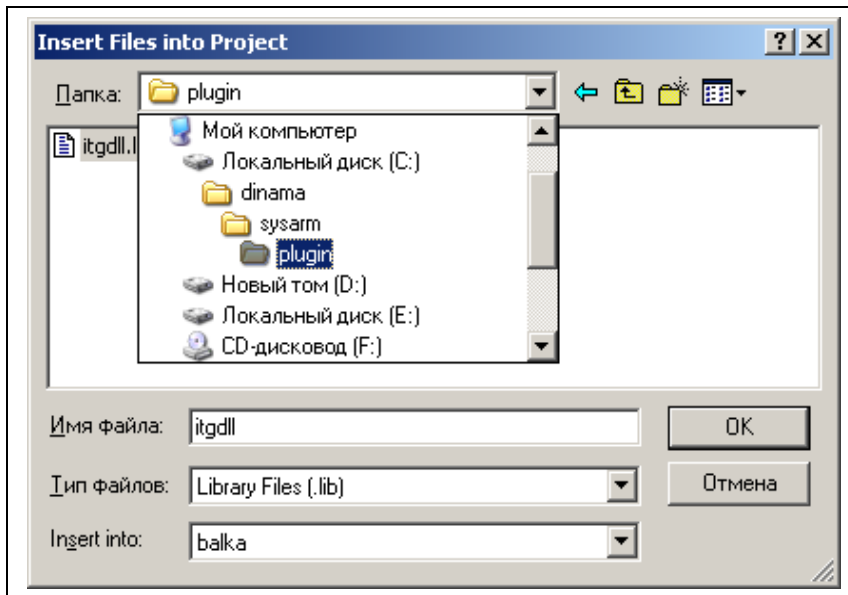
To make active a window Of visual Of c++ with the open project of component, and to cause the dialog of tuning project by the combination of the keys For alt+F7. Further to pass in the dialog “Project Of settings” to the supplementary sheet “Of fortran”, to select in [kombobokse] “Of category” the point “Of preprocessor”, and then into the end of the field of introduction “INCLUDE and USE Of paths:” to introduce substrng “;c:/dinama/of sysarm/of plugin”, where “c:” the value of the variable of environment DINSYS (disk, where it is established BY PRADIS). One should trace so that the tuning would be established for the check-out configuration. In this case [komboboks] “Of settings of for” must be in the position “Of win32 Of debug”.



Further should be harvested “OK” in the dialog “Project Of settings”.

## 6.6 to add into the project binding with the library of itgdll.lib.

This action is necessary, if Hollerith-coded the procedure of component it is intended to use auxiliary procedures of libraries PRADIS, such as the conclusion of error (for example, W\_ERROROUT) and the like it is necessary to cause the menu Of project->Add to of project->Files. In the appeared dialog “Insert of files Of into Of project” should be to establish “the type of the files” of – “Of library Of files (.lib)”, and then selected the file c:/dinama/of sysarm/plugin/of itgdll.lib, where “c:” the value of the variable of environment DINSYS (disk, where it is established BY PRADIS).

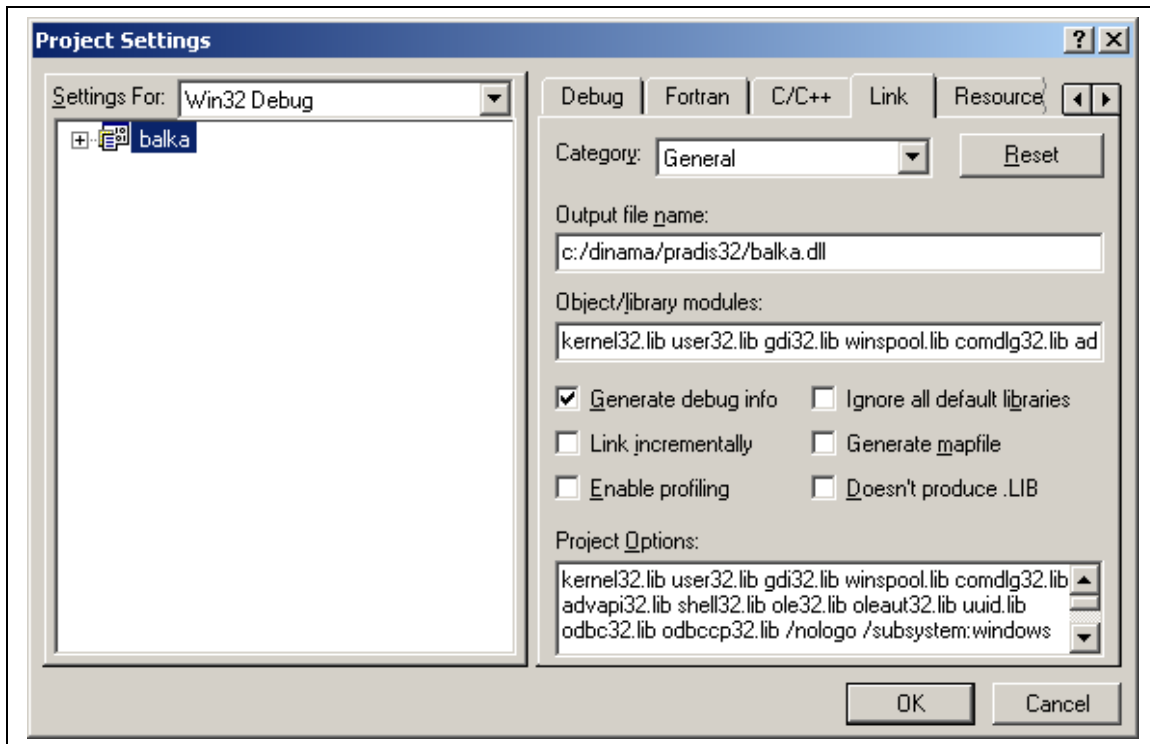


Further should be harvested “OK” in the dialog “Insert of files Of into Of project”.

You will focus attention, that into the project one ought not to add no library files, except itgdl.lib. Otherwise component, possibly, will not succeed in gathering and registering with the aid of the utility ARM, since the utility during assembling of the dynamic library of component is connected not with what other libraries.

## 6.7 to dispose the way of the resulting file of assembling in % DINSYS %/dinama/of pradis32/.

To make active a window Of visual Of c++ with the open project of component, and to cause the dialog of tuning project by the combination of the keys For alt+F7. Further to pass in the dialog “Project Of settings” to the supplementary sheet “Of link”, to select in [kombobokse] “Of category” the point “Of general”, to and then introduce in the field of introduction “Output of file of name” substring “c:/dinama/of pradis32/” instead of the prefix “Debug/”, where “c:” the value of the variable of environment DINSYS (disk, where it is established BY PRADIS). One should trace so that the tuning would be established for the check-out configuration. In this case [komboboks] “Of settings of for” must be in the position “Of win32 Of debug”.



Further should be harvested “OK” in the dialog “Project Of settings”. Because of the executed action the project will place the resulting file of dynamic library after each of its assemblings there, from where it it will load complex. Now it is possible to compile project on the key F7. Preliminarily one should ascertain that the check-out configuration is active for the project. For this should be caused the menu “Of build->Set Of active Of configuration”, and established in the dialog “Set Of active Of project Of configuration” the configuration “Of win32 Of debug”. In the case of successful assembling on F7 (absence of the errors of – “of errors” in the lower window of studio), the dynamic library of component will be substituted in that established of directory.

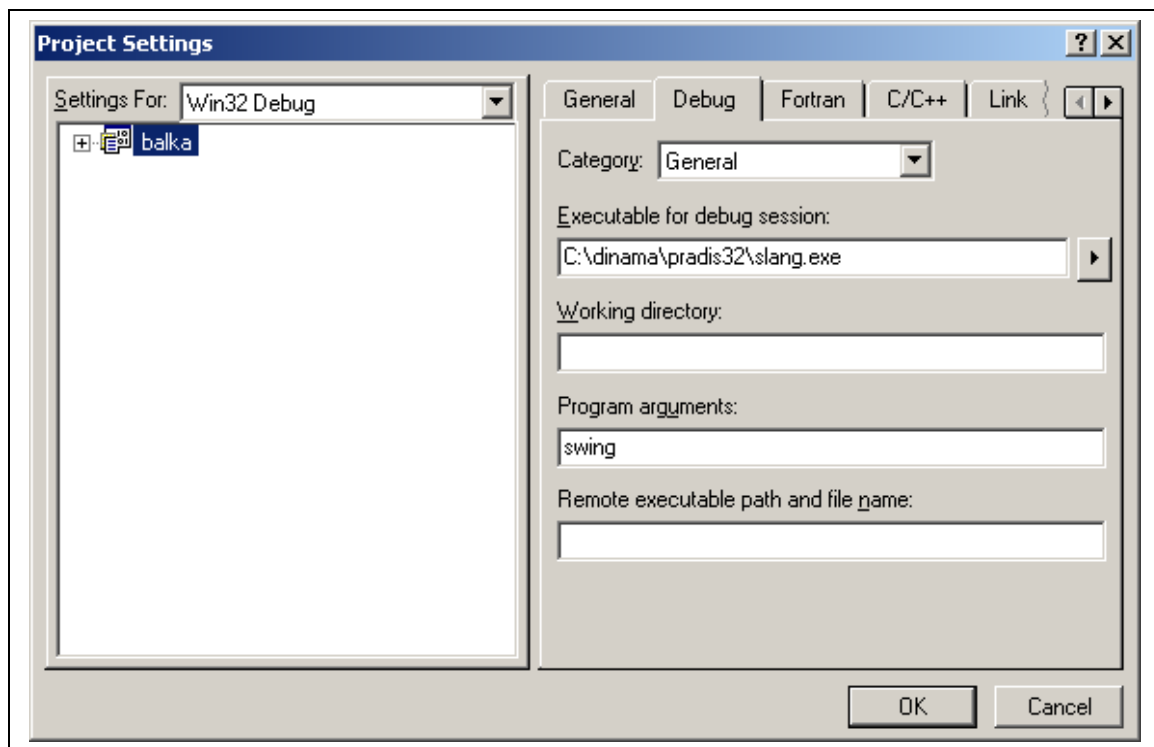
## 6.8 to establish cantilever solver as the check-out application for the library, and also arguments of its call.

In order to use the diagnostic routine of studio for the dynamic library, should be selected the application, which is must, after being neglected, to load library, to and then cause its procedure. In our case such application is cantilever the solver of slang. Furthermore, should be established the arguments of the command line of application. For slang this as the minimum name of the file of task for PRADISlang. It will not be neglected without this of slang, but only will give out communication about the error. Thus, should be created the file of task for PRADISlang (turning to the created component) and placed it in the directory, where the files of project are located. In our example this there will be the file “the D:\lucky\c of ++tests \ of balka \ of swing”, in which is present the turning to BALKKA.

Now should be made active a window Of visual Of c++ with the open project of component, and caused the dialog of tuning project by the combination of the keys For alt+F7. Further to pass in the dialog “Project Of settings” to the supplementary sheet “Of debug”, to select in [kombobokse] “Of category” the point “Of general”, to and then introduce in the field of introduction “Executable of for of debug of session” the name of the usable file of the cantilever of slang with the complete method and the expansion (exe). It is possible to harvest there to the button with “the pointer”, to select “Browse” and to find file with the aid of the standard dialog.



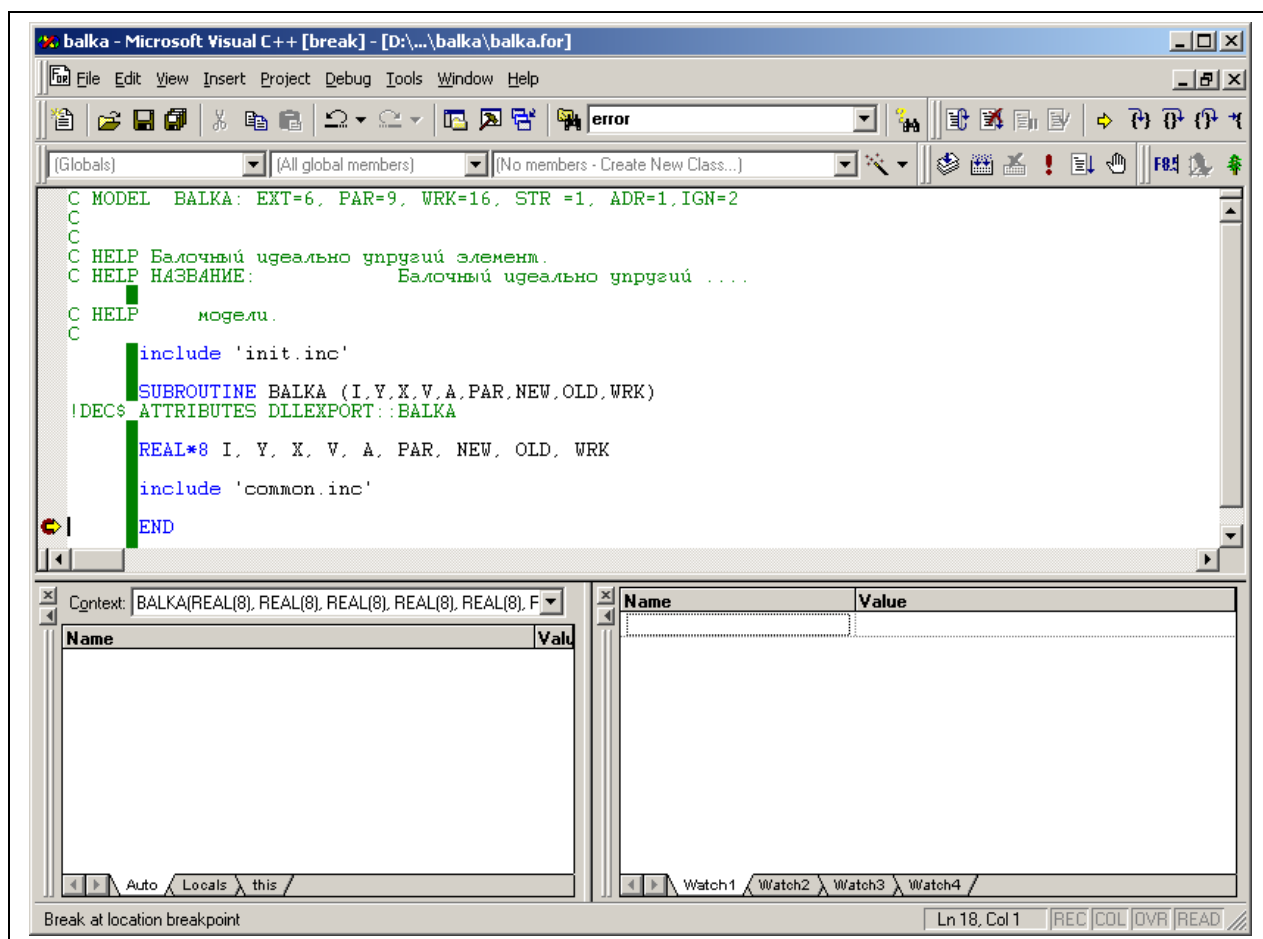
After this, in the field of the introduction “Of program of arguments” should be introduced the name of the calculated task “of swing”. One should trace so that the tuning would be established for the check-out configuration. In this case [komboboks] “Of settings of for” must be in the position “Of win32 Of debug”.



After pressure on “OK” in the dialog it will be possible to fix the code of procedure with the diagnostic routine of studio.

## 6.9 use of a diagnostic routine.

It is possible to use diagnostic routine, only if project successfully is compiled (see earlier). First of all, should be established Hollerith-coded point of discontinuity. For this in the source text of the procedure of component one should establish cursor on the desired line and harvest F9. In the beginning line must appear brown circle. Then for the starting of check-out application one should harvest F5. In our case will be neglected cantilever the solver with the task, indicated in the arguments of command line in the preceding point. If in the task there is a turning to the fixed component, and all the remaining steps of this point are executed correctly, then cantilever solver will load the check-out version of our dynamic library, and then fulfillment will interrupt, when program reaches the point of discontinuity.



Further it is possible to look the instantaneous values of variables, to make step-by-step performance and to use other similar possibilities of diagnostic routine.

## 6.10 retention of project and [peresozdanie] of the library of component from arm.

After the end of work with the check-out version, which consists in the editing of the initial code of procedure, recompilation and fixing of dynamic library, should be preserved project, after causing the menu "Of file->Save Of all". Then again it will be possible to use it, after opening with the aid of the studio the file of working space with the expansion of dsw. Further should be anew included component in complex from the initial file, after causing ARM with the key +, as it is described earlier in this point. As a result in [distributive] of complex instead of the check-out version of library again will prove to be its optimized Release assembling.