

# **PRADIS**

**РАЗРАБОТКА ПГО НА ФОРТРАНЕ**

**ПРОГРАММНЫЙ КОМПЛЕКС ДЛЯ АВТОМАТИЗАЦИИ  
МОДЕЛИРОВАНИЯ НЕСТАЦИОНАРНЫХ ПРОЦЕССОВ В  
МЕХАНИЧЕСКИХ СИСТЕМАХ И СИСТЕМАХ ИНОЙ  
ФИЗИЧЕСКОЙ ПРИРОДЫ**

**ВЕРСИЯ 4.3**

# Содержание

<a href="#">Содержание.....</a>	<a href="#">2</a>
<a href="#">1. Описание общей схемы работы ПГО.....</a>	<a href="#">3</a>
<a href="#">2. Интерфейс ПГО.....</a>	<a href="#">4</a>
<a href="#">3. Графические операторы.....</a>	<a href="#">6</a>
<a href="#">4. Пример ПГО.....</a>	<a href="#">11</a>
<a href="#">5. Описание формата репозитория.....</a>	<a href="#">13</a>
<a href="#">6. Описание процесса добавления новой ПГО.....</a>	<a href="#">14</a>
6.1. Кратко:.....	14
6.2. Полно.....	14

## 1. Описание общей схемы работы ПГО

ПГО на Фортране для ПостПроцессора (ПП) разрабатываются в виде DLL библиотек подключаемых к ПП с использованием плагин механизма. ПП загружает нужную DLL, которая содержит процедуру ПГО. Информация о ПГО и DLL описывается в файле репозитория ПГО (PGO\_List.txt). Репозиторий ПГО содержит в своем списке не только ПГО, написанные на Фортране, но и ПГО, написанные на C++. Собственно говоря, основным языком разработки ПГО для ПП является C++, так как этот язык предоставляет доступ ко всем ресурсам графического ядра. На Фортране возможен доступ лишь к операциям создания геометрических объектов.

Когда требуется нарисовать объект или изменились его координаты, ПП вызывает соответствующую процедуру ПГО. В ПГО передаются все массивы, так же как и при вызове ПГО из решателя.

Графический объект ПГО на Фортране создается с помощью набора графических примитивов, входящих в соответствующую библиотеку, написанную на Фортране. Графические примитивы вызываются как подпрограммы с определенными параметрами. Эти параметры передаются в специальный объект ПП, который строит на их основе визуальный образ соответствующий примитиву и отображает его в 3D окне ПП. Параметры передаются в объект через специально выделенный массив.

Входные данные для ПГО берутся из массива А решателя, которые во время выполнения решателя записываются в специальный файл, имеющий имя такое же, как имя файла задания и расширение .DAT. В ПГО входные данные передаются через входные параметры, структура которых описана в документе **plugins.doc** и в параграфе 2 этого документа.

## 2. Интерфейс ПГО

ПГО теперь, в отличие от старого способа задания входных параметров, имеют строго регламентированный список входных параметров. Список параметров следующий:

- NAMEX: имя модели, связанной с ПГО. Массив пробелов, при значениях параметров паспорта VPS=0 и EXT=0 (неподвижный графический образ). Имеет тип CHARACTER \* 8 при реализации вызова на FORTRAN.
- I: вектор сил (моментов) для элемента. Имеет тип REAL \* 8 при реализации вызова на FORTRAN.
- X: вектор перемещений узлов модели, связанной с ПГО, размерности EXT. Не используется при значении параметров паспорта VPS=0 и EXT=0, а также при значении параметра UNV>0 (в этом случае используется вектор INNER). Имеет тип REAL \* 8 при реализации вызова на FORTRAN.
- V: вектор скоростей узлов модели, связанной с ПГО, размерности EXT. Не используется при значении параметров паспорта VPS=0 и EXT=0, а также при значении параметра UNV>0 (в этом случае используется вектор INNER). Имеет тип REAL \* 8 при реализации вызова на FORTRAN.
- A: вектор ускорений узлов модели, связанной с ПГО, размерности EXT. Не используется при значении параметров паспорта VPS=0 и EXT=0, а также при значении параметра UNV>0 (в этом случае используется вектор INNER).
- INNER: вектор вещественных чисел имеющих смысл степеней свободы модели элемента, связанной с ПГО. Не используется при значении параметров паспорта VPS=0 и EXT=0, а также при значении параметра UNV=0 (в этом случае используются вектора X,V,A).
- EXT: количество степеней свободы модели элемента, связанной с ПГО (длина INNER). Имеет тип INTEGER \* 4 при реализации вызова на FORTRAN.
- PARX: вектор параметров модели, связанной с ПГО. Не используется при значении параметров паспорта VPS=0 и EXT=0. Имеет тип REAL \* 8 при реализации вызова на FORTRAN. Нулевой элемент этого массива содержит количество параметров.
- WRKX: рабочий вектор модели, связанной с ПГО. Не используется при значении параметров паспорта VPS=0 и EXT=0. Имеет тип REAL \* 8 при реализации вызова на FORTRAN.
- PAR: вектор параметров ПГО. Имеет тип REAL \* 8 при реализации вызова на FORTRAN.
- WRK: рабочий вектор ПГО. Имеет тип REAL \* 8 при реализации вызова на FORTRAN.
- PARLR2: вектор параметров текущего слоя изображения. Имеет тип REAL \* 8 при реализации вызова на FORTRAN.

Соответственно заголовок ПГО и описание входных параметров во всех ПГО должны быть идентичными и выглядеть следующим образом:

```
SUBROUTINE TSTPGO (  
  ,      NAMEX,  
  ,      I,  
  ,      X_,  V_,  A_,  
  ,      INNER, EXT,  
  ,      PARX, WRKX,
```

```
,      PAR, WRK,  
,      PARLR2 )
```

**C    Формальные параметры**

```
CHARACTER*8   NAMEX
```

```
REAL * 8    I (1)
```

```
REAL * 8    X_(6), V_(6), A_(6)
```

```
REAL * 8    INNER(1), PARX(1), WRKX(1), PAR(1), WRK(1), PARLR2(1)
```

```
INTEGER * 4 EXT
```

Так же в ПГО передается из решателя неименованный блок COMMON, значения которого доступны через включение в текст ПГО файла **common.inc** входящего в состав комплекса. Он включается добавлением в текст программы следующей строки:

```
include 'common.inc'
```

### 3. Графические операторы.

Библиотека графических операторов находится в файле **graphs.inc** и подключается в ПГО добавлением в текст программы следующей строки:

```
include 'graphs.inc'
```

В состав библиотеки входят следующие операторы:

#### **С Функция задания цвета рисования**

С ПАРАМЕТРЫ:

С COL - цвет рисования объекта

- С 1        желтый
- С 2        яркозеленый
- С 3        светлокоричневый
- С 4        розовокрасный
- С 5        светлосиний
- С 6        белоголубой
- С 7        темнокрасный
- С 8        грязнозеленый
- С 9        фиолетовый
- С 10       темнозеленый
- С 11       светлокрасный
- С 12       синий
- С 13       серый
- С 14       белый

```
SUBROUTINE COLOR(COL)
```

```
REAL * 8 COL
```

#### **С Функция рисования цилиндра**

С ПАРАМЕТРЫ:

С X1, Y1, Z1 - координаты центра основания цилиндра

С X2, Y2, Z2 - координаты центра верхней части цилиндра

С DIAM - диаметр цилиндра

```
SUBROUTINE CYLINDER(X1, Y1, Z1, X2, Y2, Z2, DIAM)
```

```
REAL * 8 X1, Y1, Z1, X2, Y2, Z2, DIAM
```

#### **С Функция рисования сферы**

С ПАРАМЕТРЫ:

С X1, Y1, Z1 - координаты центра сферы

С DIAM - диаметр сферы

```
SUBROUTINE SPHERE(X1, Y1, Z1, DIAM)
```

```
REAL * 8 X1, Y1, Z1, DIAM
```

#### **С Функция рисования прямой линии**

С ПАРАМЕТРЫ:

С X1, Y1, Z1 - координаты первой точки линии

С X2, Y2, Z2 - координаты второй точки линии

```
SUBROUTINE LINE(X1, Y1, Z1, X2, Y2, Z2)
```

```
REAL * 8 X1, Y1, Z1, X2, Y2, Z2
```

### С Функция рисования круга

С ПАРАМЕТРЫ:

С X1, Y1, Z1 - координаты центра круга

С X2, Y2, Z2 - координаты точки, лежащей на оси круга

С DIAM - диаметр круга

С FACE - если = 0, то рисуется контур

С - если = 1, то рисуется поверхность

```
SUBROUTINE CIRCLE(X1, Y1, Z1, X2, Y2, Z2, DIAM, FACE)
```

```
REAL * 8 X1, Y1, Z1, X2, Y2, Z2, DIAM, FACE
```

### С Функция рисования четырехугольной призмы

С ПАРАМЕТРЫ:

С X1, Y1, Z1, X2, Y2, Z2,

С X3, Y3, Z3, X4, Y4, Z4 - координаты четырех точек основания призмы

С X5, Y5, Z5 - координаты верхней точки призмы, лежащей на том же ребре,

С - что и точка X1, Y1, Z1

```
SUBROUTINE BOX(X1, Y1, Z1, X2, Y2, Z2,
```

```
' X3, Y3, Z3, X4, Y4, Z4, X5, Y5, Z5)
```

```
REAL * 8 X1, Y1, Z1, X2, Y2, Z2,
```

```
' X3, Y3, Z3, X4, Y4, Z4, X5, Y5, Z5
```

### С Функция рисования дуги

С ПАРАМЕТРЫ:

С X1, Y1, Z1 - координаты центра дуги

С X2, Y2, Z2 - координаты точки, лежащей на оси дуги

С ANG1, ANG2 - углы, между которыми рисуется дуга

С DIAM - диаметр дуги

```
SUBROUTINE ARC(X1, Y1, Z1, X2, Y2, Z2, ANG1, ANG2, DIAM)
```

```
REAL * 8 X1, Y1, Z1, X2, Y2, Z2, ANG1, ANG2, DIAM
```

### С Функция загрузки графического файла

С ПАРАМЕТРЫ:

С X1, Y1, Z1 - координаты расположения центра загружаемого образа

С NOMB - номер файла

С SCALE - масштабный коэффициент

```
SUBROUTINE LOADGM(X1, Y1, Z1, NOMB, SCALE)
```

```
REAL * 8 X1, Y1, Z1, NOMB
```

Данная функция загружает графический файл, созданный каким либо средством трехмерного моделирования и сохраненный в формате BRep (формат OpenCASCADE). Этот файл должен находиться в той же папке, что и файл DAT и иметь точно определенное имя: <имя файла задания>.<номер файла = NOMB>.img. Например, для задания с именем **swing** этот файл может иметь следующие имена: **swing.1.img**, **swing.2.img**, .... **swing.100.img** и т.п. Нулевая точка образа из файла помещается в заданную точку X1, Y1, Z1. Размер образа изменяется на SCALE масштабный коэффициент. Если SCALE =1, то размер не изменяется.

### **С Функция рисования конуса**

С ПАРАМЕТРЫ:

С X1, Y1, Z1 - координаты центра основания конуса

С X2, Y2, Z2 - координаты точки, лежащей на оси конуса

С R1 - диаметр основания конуса (может быть равен нулю)

С R2 - диаметр верхней части конуса (может быть равен нулю)

С H - высота конуса

SUBROUTINE CONE(X1, Y1, Z1, X2, Y2, Z2, R1, R2, H)

REAL \* 8 X1, Y1, Z1, X2, Y2, Z2, R1, R2, H

### **С Функция рисования треугольника**

С ПАРАМЕТРЫ:

С X1, Y1, Z1 - координаты первой точки треугольника

С X2, Y2, Z2 - координаты второй точки треугольника

С X3, Y3, Z3 - координаты третьей точки треугольника

С FACE - если = 0, то рисуется контур

С - если = 1, то рисуется поверхность

SUBROUTINE TRIANGLE(X1, Y1, Z1, X2, Y2, Z2, X3, Y3, Z3, FACE)

REAL \* 8 X1, Y1, Z1, X2, Y2, Z2, X3, Y3, Z3, FACE

### **С Функция рисования четырехугольника**

С ПАРАМЕТРЫ:

С X1, Y1, Z1 - координаты первой точки четырехугольника

С X2, Y2, Z2 - координаты второй точки четырехугольника

С X3, Y3, Z3 - координаты третьей точки четырехугольника

С X4, Y4, Z4 - координаты четвертой точки четырехугольника

С FACE - если = 0, то рисуется контур

С - если = 1, то рисуется поверхность

SUBROUTINE RECTANGLE(X1, Y1, Z1, X2, Y2, Z2,

' X3, Y3, Z3, X4, Y4, Z4, FACE)

REAL \* 8 X1, Y1, Z1, X2, Y2, Z2,

' X3, Y3, Z3, X4, Y4, Z4, FACE

### **С Функция рисования триэдра (3D системы координат)**

С ПАРАМЕТРЫ:

С X1,Y1,Z1 - начальные координаты точки А (центра подвижного базиса)

С по осям OX, OY, OZ;

С X2,Y2,Z2 - начальные координаты точки В, определяющей совместно

С с точкой А начальное положение локальной оси Z'

С подвижного базиса (ось Z' направлена от А к В)

С X3,Y3,Z3 - начальные координаты точки С, определяющей совместно

С с точками А и В плоскость расположения оси X';

С LENGTH - длина осей изображаемого базиса.

SUBROUTINE TRIHEDRON(X1, Y1, Z1, X2, Y2, Z2,

' X3, Y3, Z3, LENGTH)

REAL \* 8 X1, Y1, Z1, X2, Y2, Z2,

' X3, Y3, Z3, LENGTH

### **С Функция рисования эллипсоида**

С ПАРАМЕТРЫ:

С X1,Y1,Z1 - начал. координаты точки А

С X2,Y2,Z2 - начал. координаты точки В



C X3,Y3,Z3 - начал. координаты точки C  
 C X4,Y4,Z4 - начал. координаты точки D  
 C R1,R2,R3 - полуоси эллипсоида, R1, R2, R3 (>0).  
 C DEGREE - степень эллипсоида (p>1)  
 SUBROUTINE ELLIPSOID(X1, Y1, Z1, X2, Y2, Z2,  
 ' X3, Y3, Z3, X4, Y4, Z4,  
 ' R1, R2, R3, DEGREE)  
 REAL \* 8 X1, Y1, Z1, X2, Y2, Z2,  
 ' X3, Y3, Z3, X4, Y4, Z4,  
 ' R1, R2, R3, DEGREE

### **С Функция рисования вектора**

С ПАРАМЕТРЫ:

С X1, Y1, Z1 - координаты начальной

С X2, Y2, Z2 - координаты конечной точки

С LEN - размер стрелки

SUBROUTINE VECTOR( X1, Y1, Z1, X2, Y2, Z2, LEN)  
 REAL \* 8 X1, Y1, Z1, X2, Y2, Z2, LEN

### **С Функция возвращающая минимальное значение расстояния между двумя точками**

REAL \* 8 FUNCTION MINLEN()

Данная функция используется для проверки правильности параметров задаваемых в графические операторы. Если какие либо точки находятся ближе друг к другу, чем значение возвращаемого этой функцией значения, то эти точки считаются идентичными и расстояние между ними равно нулю.

### **С Функция возвращающая код ошибки**

С Коды:

С 0 - нет ошибок

С 1 - расстояние между двумя точками равно нулю

С 2 - диаметр меньше или равен нулю

С 3 - масштабный коэффициент меньше или равен нулю

С 4 - высота меньше или равна нулю

С 5 - длинна меньше или равна нулю

С 6 - степень эллипсоида меньше или равна единице

С 7 – неверный код цвета

С 8 – превышено предельное количество примитивов

INTEGER \* 4 FUNCTION ERRCODE()

Эту функцию следует вызывать после каждого графического оператора для проверки правильности входных параметров. Если функция возвращает код, отличный от нуля, значит входные параметры были заданы неправильно и никакого рисования не происходит.

Ниже приведены примеры вызовов этих операторов:

CALL SPHERE(X1,X2,X3, DIAM \* 2.)  
 CALL CYLINDER(X1,X2,X3, X4,X5,X6, DIAM)  
 CALL SPHERE(X4,X5,X6, DIAM \* 2.)  
 CALL LINE(X1,X2,X3, X4,X5,X6)

```

CALL CIRCLE(X1,X2,X3, X4,X5,X6, DIAM * 5, FACE)
CALL BOX(X1,X2,X3, X4,X5,X6, X7,X8,X9, X10,X11,X12,
'      X13,X14,X15)
CALL ARC(X1,X2,X3, X4,X5,X6, ANG1, ANG2, DIAM * 8)
CALL LOADGM(X4,X5,X6, NOMB, SCALE)
CALL CONE(X1,X2,X3, X4,X5,X6, R1, R2, H)
CALL TRIANGLE(X1,X2,X3, X4,X5,X6, X7,X8,X9, FACE)
CALL RECTANGLE(X1,X2,X3, X4,X5,X6, X7,X8,X9, X10,X11,X12, FACE)
CALL TRIHEDRON(X1,X2,X3, X4,X5,X6, X7,X8,X9, LENGTH)
CALL ELLIPSOID(X1,X2,X3, X4,X5,X6, X7,X8,X9, X10,X11,X12,
'      R1, R2, R3, DEGREE)
CALL VECTOR( X1,X2,X3, X4,X5,X6, LEN)
ERRC=ERRCODE()
MINLN=MINLEN()

```

## 4. Пример ПГО

Далее приведен пример текста простой ПГО:

**C IMAGE TSTPGO:EXT=6, PAR=2**

**C**

**C HELP Образ элемента сжимающегося цилиндрического**

**C HELP стержня с шарами на концах**

**C**

**C           Дата создания**

**04/09/2006 11:52am**

**C           Дата последней корректировки**

**05/09/2006 03:55pm**

**include 'init.inc'**

**include 'graphs.inc'**

```
      SUBROUTINE TSTPGO (  
      ,      NAMEX,  
      ,      I,  
      ,      X_, V_, A_,  
      ,      INNER, EXT,  
      ,      PARX, WRKX,  
      ,      PAR, WRK,  
      ,      PARLR2 )
```

**!DEC\$ ATTRIBUTES DLLEXPORT::TSTPGO**

**include 'common.inc'**

**C     Формальные параметры**

**CHARACTER\*8 NAMEX**

**REAL \* 8 I (1)**

**REAL \* 8 X\_(6), V\_(6), A\_(6)**

**REAL \* 8 INNER(1), PARX(1), WRKX(1), PAR(1), WRK(1), PARLR2(1)**

**INTEGER \* 4 EXT, ERRC, COL, ERRCODE**

**REAL \* 8 X1, X2, X3, X4, X5, X6,**

**REAL \* 8 DIAM, MINLN, MINLEN**

**X1 = PARX(1) + X\_(1)**

**X2 = PARX(2) + X\_(2)**

**X3 = PARX(3) + X\_(3)**

**X4 = PARX(4) + X\_(4)**

**X5 = PARX(5) + X\_(5)**

**X6 = PARX(6) + X\_(6)**

**DIAM = PAR(1)**

**COL = 4.**

**CALL COLOR(COL)**

**MINLN=MINLEN()**

**L = SQRT((X1-X4)\*(X1-X4)+(X2-X5)\*(X2-X5)+(X3-X6)\*(X3-X6))**

```
IF(L .LT. MINLN) GOTO 1  
IF(DIAM .LT. MINLN) GOTO 1
```

```
CALL SPHERE(X1,X2,X3, DIAM * 2.)  
ERRC=ERRCODE()  
IF(ERRC .GT.0) GOTO 1
```

```
CALL CYLINDER(X1,X2,X3, X4,X5,X6, DIAM)  
IF(ERRC .GT.0) GOTO 1
```

```
CALL SPHERE(X4,X5,X6, DIAM * 2.)  
IF(ERRC .GT.0) GOTO 1
```

```
1  RETURN  
   END
```

## 5. Описание формата репозитория.

Репозиторий ПГО хранится в текстовом файле PGO\_List.txt. Файл хранится в том же каталоге, где находится выполняемый файл ПП Postprocessor.exe. Для каждой ПГО список содержит три параметра, записанные через запятую:

- имя ПГО,
- имя вызываемой функции ПП,
- имя DLL библиотеки.

Для ПГО на Фортране параметр «имя вызываемой функции ПП» всегда должно быть FORTRAN, а параметр «имя DLL библиотеки» должно быть всегда PGO. Имя же самой DLL библиотеки, содержащей данную ПГО, должно полностью совпадать с именем ПГО. Например, для ПГО с именем **TSTPGO** имя библиотеки должно быть **TSTPGO.DLL**, а строка в репозитории должна быть следующая: **“TSTPGO, FORTRAN, PGO”**.

Внимание. В будущем предполагается изменение формата репозитория.

Ниже приведен текст репозитория ПП для версии 4.0 (7.09.2006):

DOTD, DOTD, PGO  
AERHT, AERHT, PGO  
GSV, GSV, PGO  
GRET, GRET, PGO  
PRUG, PRUG, PGO  
AKLAB, AKLAB, PGO  
D3LAB, D3LAB, PGO  
GROT2, GROT2, PGO  
GROT3, GROT3, PGO  
TRTER, TRTER, PGO  
KN3EFV, KN3EFV, PGO  
KN3FFV, KN3FFV, PGO  
EL3DP, EL3DP, PGO  
LSK3D, LSK3D, PGO  
POINT, Point, PGO  
PRLGRM, PRLGRM, PGO  
RECTD, RECTD, PGO  
LSK, LSK, PGO  
LINED, Lined, PGO  
LINEV, Linev, PGO  
CIL3DC, CIL3DC, PGO  
GCYL, GCYL, PGO  
AMORT, AMORT, PGO  
ARROW, ARROW, PGO  
CMASS, CMASS, PGO  
GNIRS, GNIRS, PGO  
HS2VS, HS2VS, PGO  
OPORA, OPORA, PGO  
OPORAD, OPORAD, PGO  
SILUET, SILUET, PGO  
KONTUR, KONTUR, PGO  
PRUZS, PRUZS, PGO  
**TSTPGO, FORTRAN, PGO**  
ELP3D, ELP3D, PGO

## 6. Описание процесса добавления новой ПГО

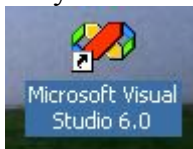
### 6.1. Кратко:

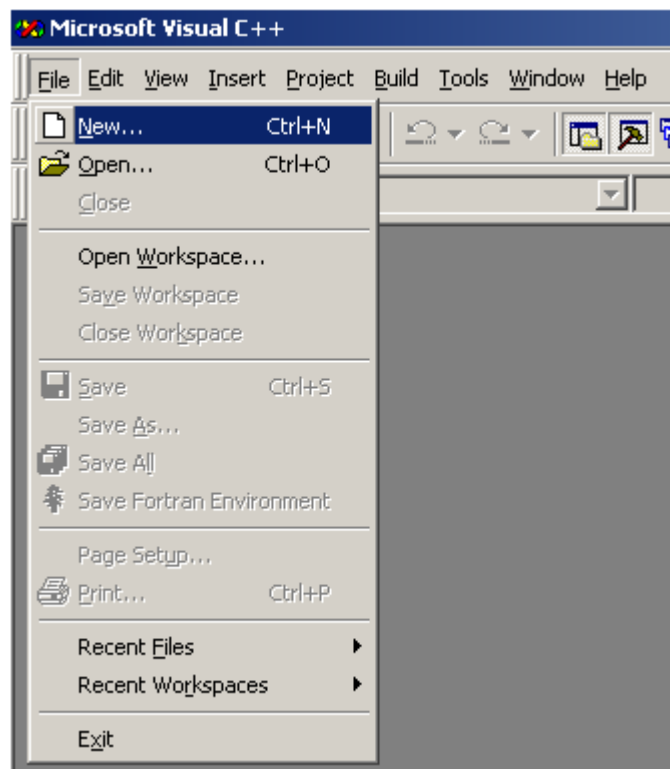
1. Создать в VC 6 новый проект динамической библиотеки с именем проекта полностью соответствующим имени создаваемой ПГО (например **TSTPGO**).
2. Выбрать пункт Project -> Settings... В появившемся диалоговом окне выбрать закладку Fortran. Выбрать пункт Run Time в верхнем поп-ап меню с названием Category. Сделать не выбранным (убрать галочку) пункт Array & String Bounds. Нажать кнопку Ok.
3. Включить в проект следующие файлы: init.inc, common.inc, graphs.inc.
4. Создать файл с именем, соответствующим имени создаваемой ПГО и расширением .FOR (например **TSTPGO.FOR**). Заполнить этот файл текстом программы ПГО по вышеприведенным правилам и руководствуясь вышеприведенным примером ПГО, а так же документом **plugins.doc**, в котором подробно описан механизм плагинов для Прадис.
5. Откомпилировать проект. Полученную динамическую библиотеку (например **TSTPGO.DLL**) поместить в каталог **%DINSYS%\dinama\post** или в другое место, доступное для загрузки динамических библиотек.
6. Включить новую ПГО в репозиторий ПП добавлением в файл **PGO\_List.txt** новой строки (например **"TSTPGO, FORTRAN, PGO"**)
7. Включить новую ПГО в системный каталог Прадис. Для этого надо скопировать файл с текстом ПГО (например **TSTPGO.FOR**) в отдельный каталог, удалить из файла программы ПГО (например **TSTPGO.FOR**) все вычисления и вызовы графических операторов, а так же строку **"include 'graphs.inc'"**, оставив подпрограмму пустой, и запустить команду ARM с ключом "+" и именем ПГО (например **%DINSYS%\dinama\pradis32\ARM + TSTPGO**). Более подробно порядок работы с системным каталогом и утилитой ARM описан в документе **description\_slang.doc**.
8. Запустить Прадис решатель на расчет задания, содержащего вызов новой ПГО.
9. После завершения расчета запустить на выполнение ПП, загрузить полученный в результате расчетов DAT-файл. ПГО должна отображаться в 3D выювере ПП.

### 6.2. Полно

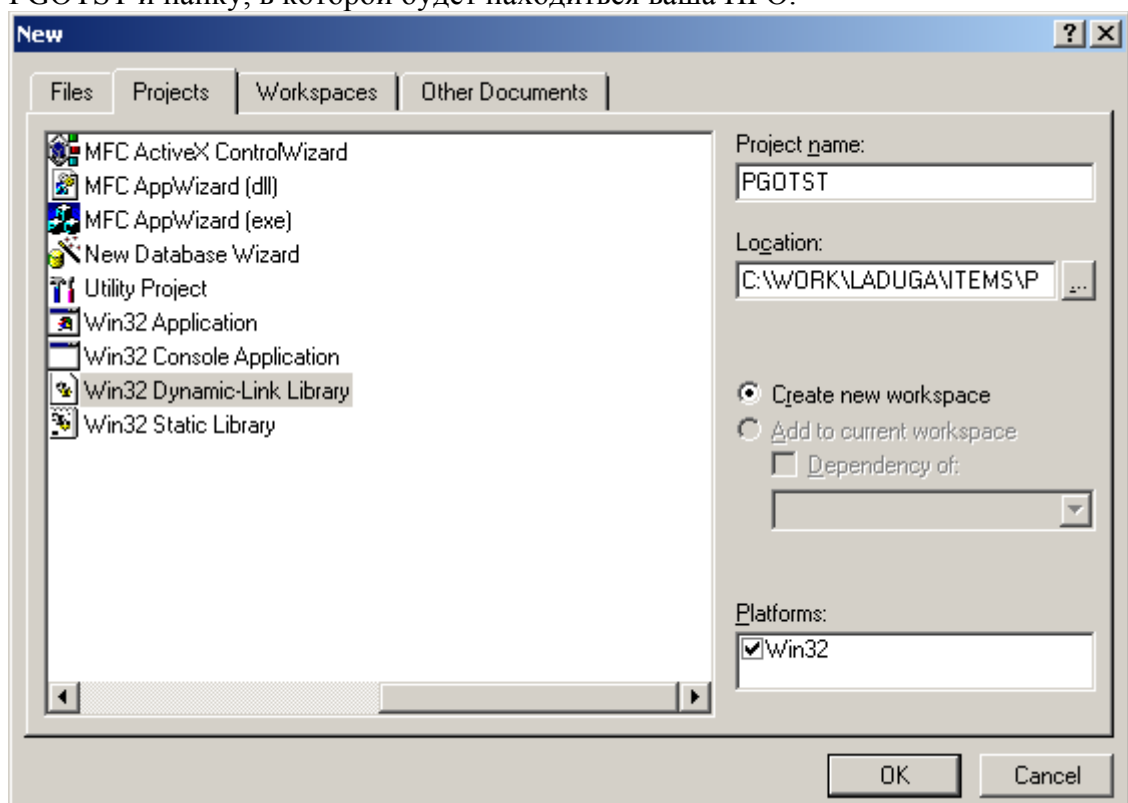
Допустим мы захотели создать новую ПГО на фортране под именем «PGOTST». Для этого нам потребуется выполнить следующее:

1. Запустите Microsoft Visual Studio. Как правило такой значок на рабочем столе:

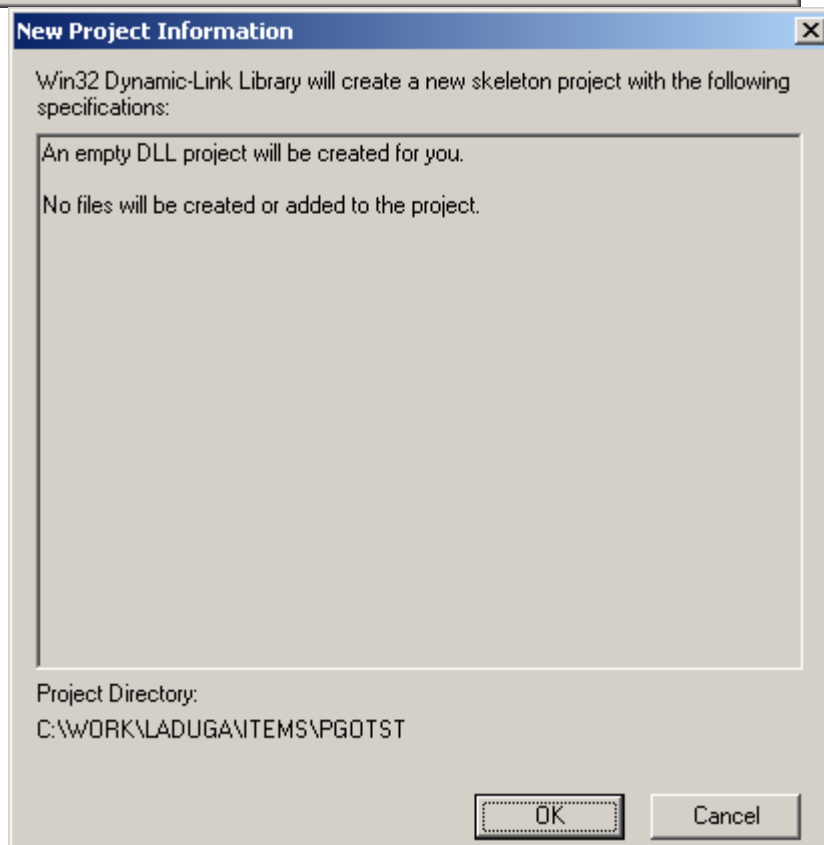
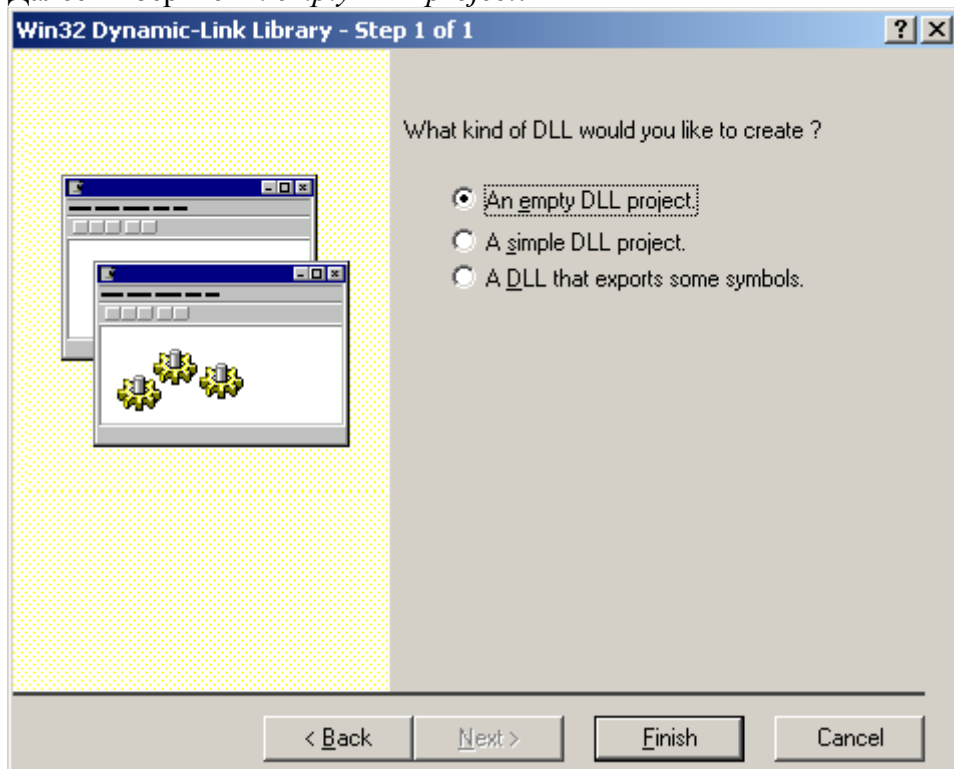




2. Выберите пункт меню *File / New*.
3. Выберите в закладке *Projects* пункт *Win32 Dynamic-Link Library*. Введите имя ПГО PGOTST и папку, в которой будет находиться ваша ПГО.



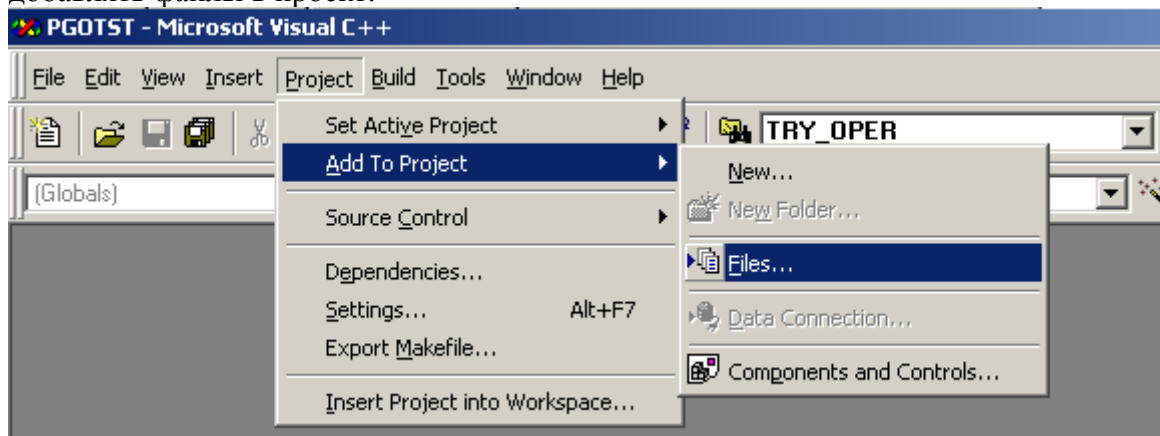
4. Далее выберите *An empty DLL project*.



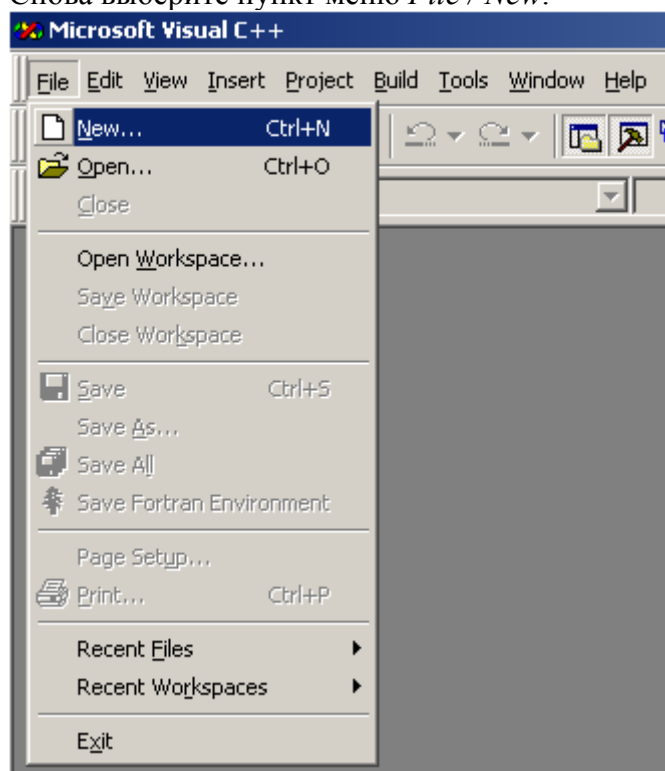
5. Жмите OK.
6. Скопируйте файлы `init.inc`, `common.inc`, `graphs.inc` в только что созданную папку проекта, указанную в окне выше.



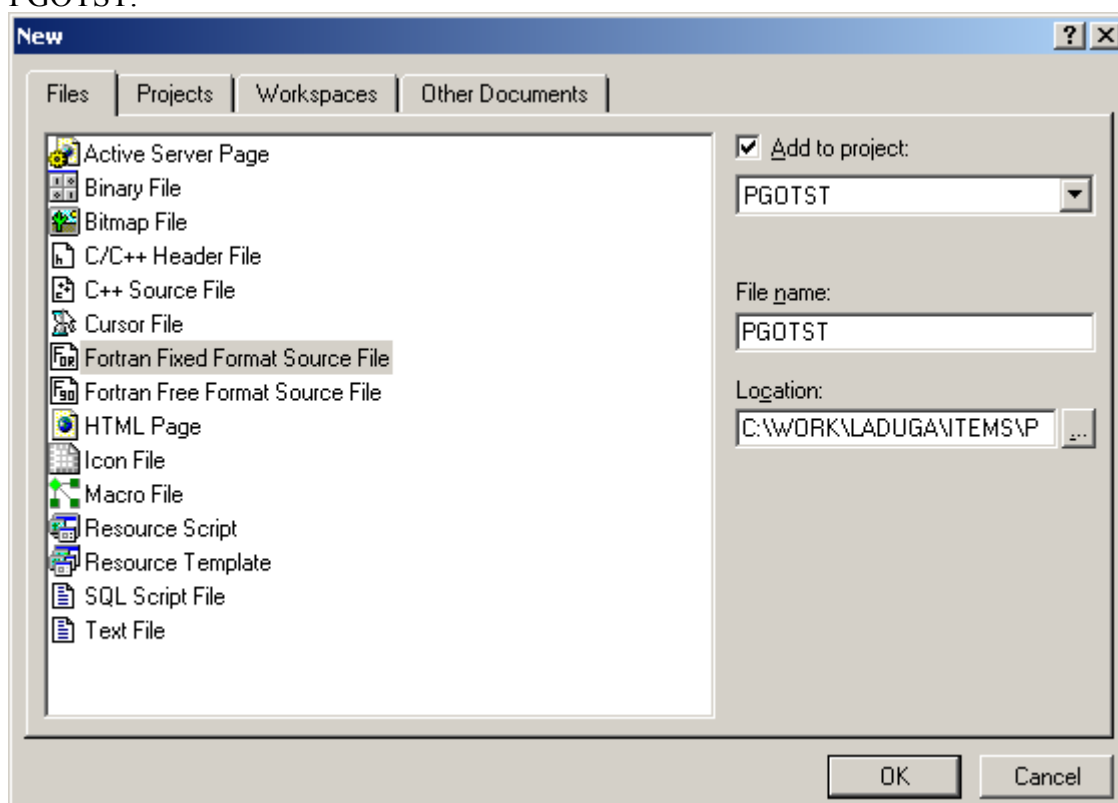
7. Выберите пункт меню *Project / Add to project / Files*. Этот пункт меню позволит Вам добавлять файлы в проект.



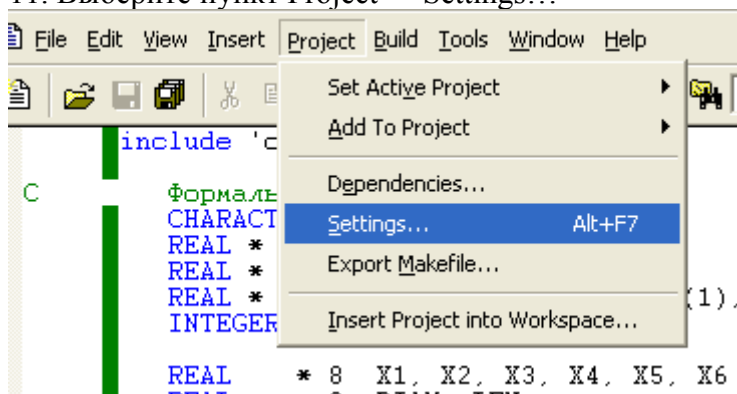
8. Добавьте файлы *init.inc*, *common.inc*, *graphs.inc* из папки проекта.
9. Снова выберите пункт меню *File / New*.



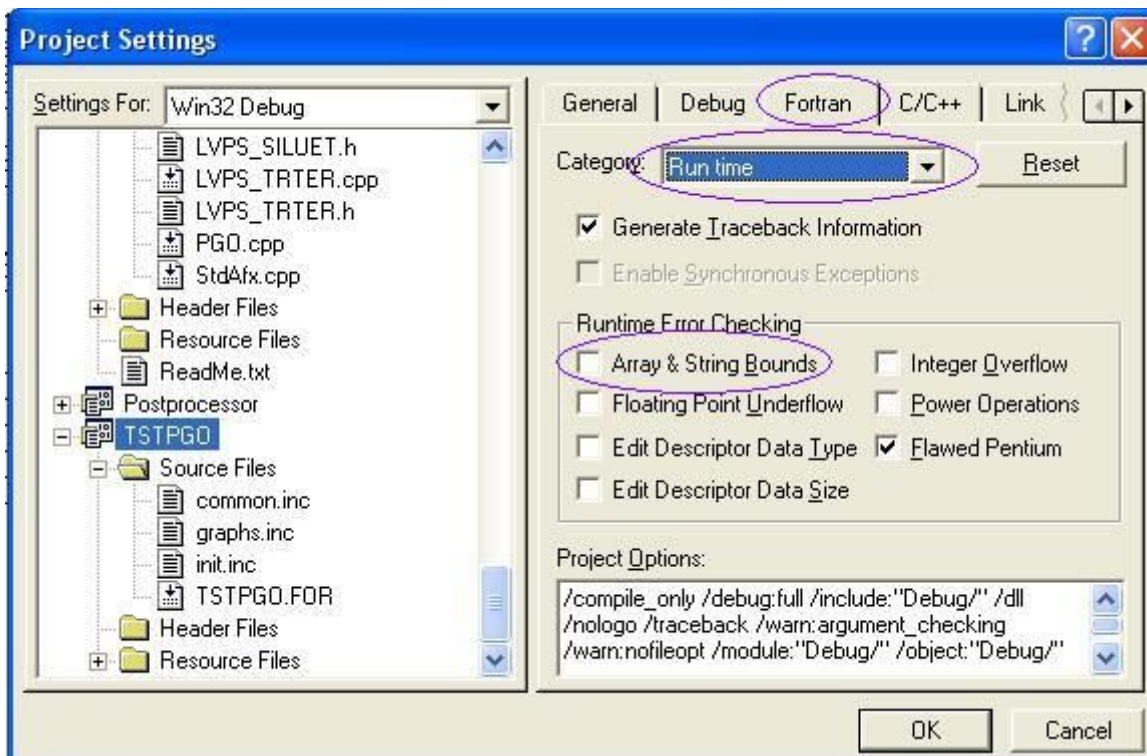
10. Во вкладке *Files* выберите пункт *Fortran Fixed Format Source File*. Введите имя файла PGOTST.



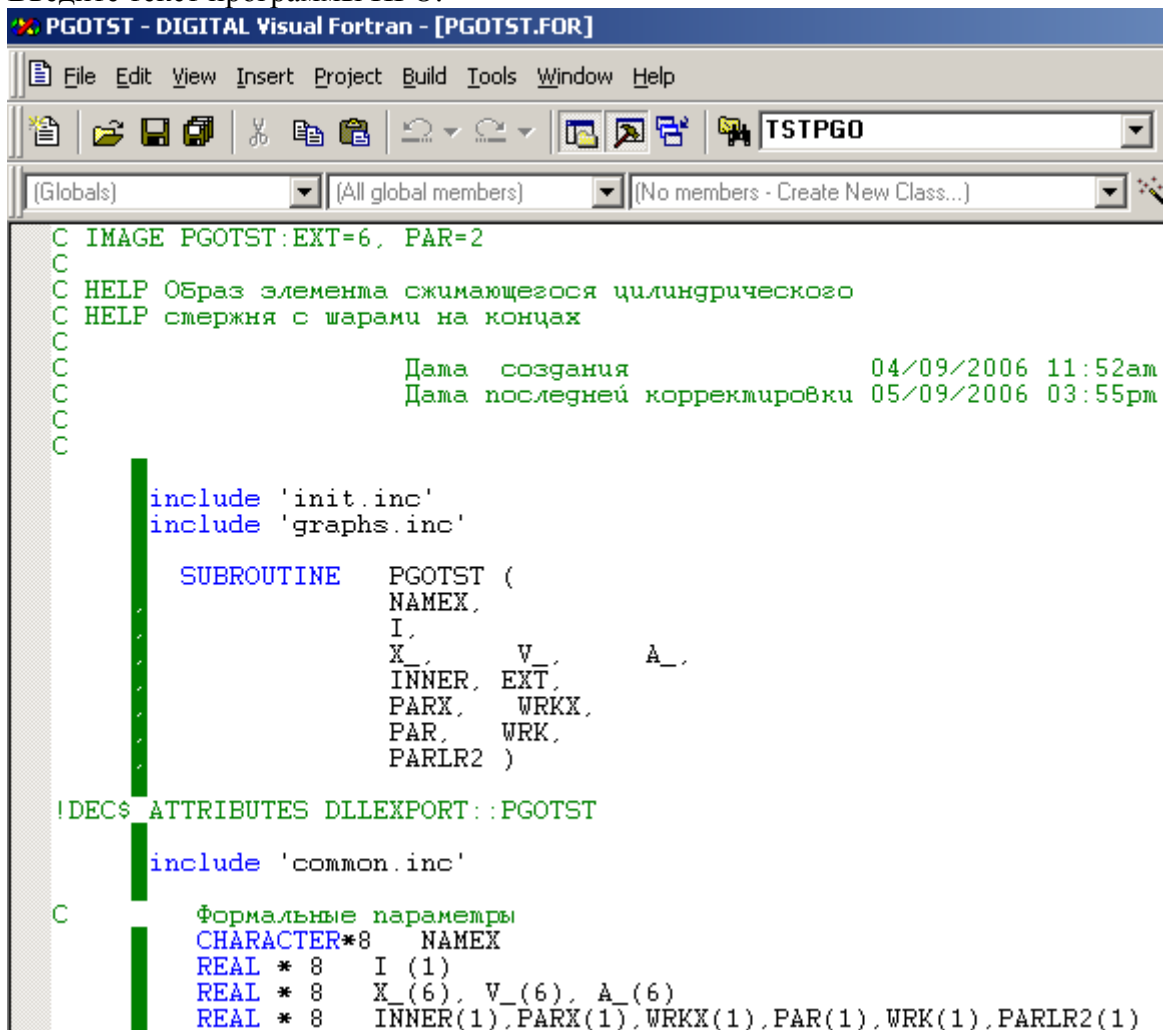
11. Выберите пункт Project -> Settings...



12. В появившемся диалоговом окне выберите закладку Fortran. Выберите пункт Run Time в верхнем поп-ап меню с названием Category. Сделайте не выбранным (уберите галочку) пункт Array & String Bounds. Нажмите кнопку Ok.

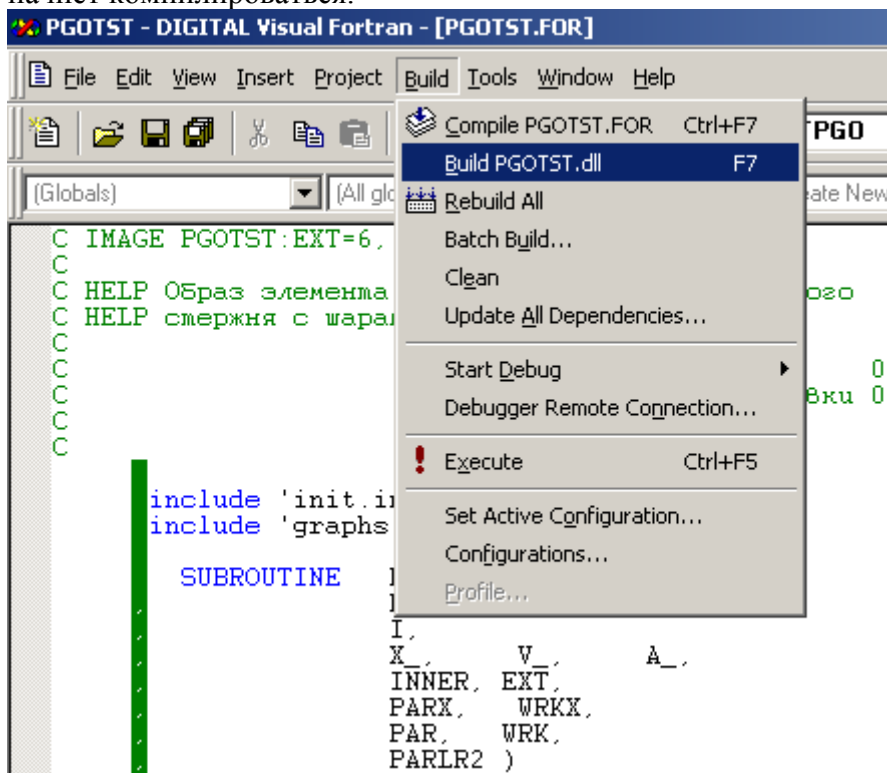


13. Введите текст программы ПГО.

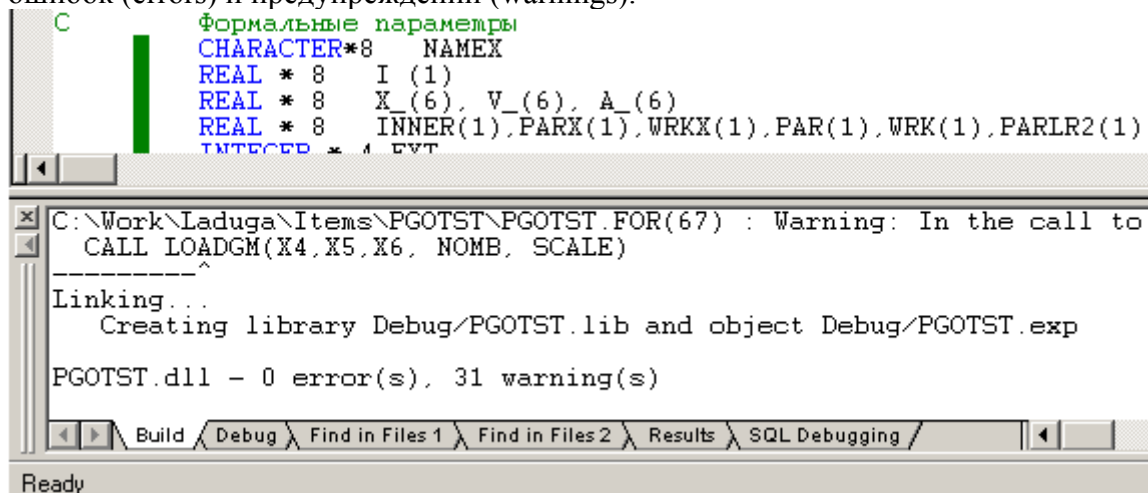


14. Нажмите «Ctrl-s», чтобы сохранить текст программы.

15. Выберите пункт меню *Build / Build PGOTST.dll* или просто нажмите F7. Программа начнёт компилироваться.



16. Дождитесь, когда компиляция закончится и внизу появится сообщение о количестве ошибок (errors) и предупреждений (warnings).



17. Если ошибок не нулевое количество (0 error(s) ), то вероятнее всего в вашей программе на фортране присутствуют ошибки. Исправьте их и вернитесь к пункту 12. Если ошибок ноль, то переходите к следующему пункту.
18. В папке проекта должна была быть создана папка Debug, в ней лежит только что созданная Вами библиотека PGOTST.dll. Скопируйте её в папку %DINSYS%\dinama\post.

19. В папке ПП лежит файл PGO\_list.txt. Добавьте в конец файла ещё одну строку:

```
GNIRS, GNIRS, PGO
HS2US, HS2US, PGO
OPORA, OPORA, PGO
OPORAD, OPORAD, PGO
SILUET, SILUET, PGO
KONTUR, KONTUR, PGO
PRUZZ, PRUZZ, PGO
TSTPGO, FORTRAN, PGO
ELP3D, ELP3D, PGO
PGOTST, FORTRAN, PGO
```

PGOTST, FORTRAN, PGO.

20. Скопируйте в папку *DINAMA/pradis32* файл PGOTST.FOR из папки проекта.  
21. Удалите из *DINAMA/pradis32/PGOTST.FOR* все вычисления и строку «include 'graphs.inc'». Остаётся что-то вроде пустой ПГО. Не забудьте, что первая строка должна быть вида: «C IMAGE PGOTST:EXT=6, PAR=2».

```
edit PGOTST.FOR - Far
C:\dinama\pradis32\PGOTST.FOR      Win   Line
C IMAGE PGOTST:EXT=6, PAR=2
C
C HELP Образ элемента сжимающегося цилиндрического
C HELP стержня с шарами на концах
C
C      Дата создания      04/09/2006 11:52am
C      Дата последней корректировки 05/09/2006 03:55pm
C
C
C      include 'init.inc'
C      include 'graphs.inc'
C
C      SUBROUTINE PGOTST <
C      NAMEX,
C      I,
C      X_, U_, A_,
C      INNER, EXT,
C      PARX, WRKX,
C      PAR, WRK,
C      PARLR2 >
C
C !DEC$ ATTRIBUTES DLLEXPORT::PGOTST
C
C      include 'common.inc'
C
C      Формальные параметры
C      CHARACTER*8 NAMEX
C      REAL * 8 I (1)
C      REAL * 8 X_(6), U_(6), A_(6)
C      REAL * 8 INNER(1), PARX(1), WRKX(1), PAR(1), WRK(1), PARLR2(1)
C      INTEGER * 4 EXT
C
C      REAL * 8 X1, X2, X3, X4, X5, X6
C      REAL * 8 DIAM, LEN
C
C      REAL * 8 ANG1, ANG2, NOMB, SCALE, R1, R2, R3, H, X7, X8, X9,
C      X10, X11, X12, X13, X14, X15, LENGTH, DEGREE
C
C
C      X1 = PARX(1) + X_(1)
C      X2 = PARX(2) + X_(2)
C      X3 = PARX(3) + X_(3)
```

22. Перейдите в *DINAMA/pradis32* и запустите «arm + PGOTST».

```
C:\dinama\pradis32>arm + PGOTST
1  C IMAGE PGOTST:EXT=6, PAR=2

      Паспорт программы PGOTST      добавлен в системный каталог

2  C
3  C HELP Образ элемента сжимающегося цилиндрического
4  C HELP стержня с шарами на концах
5  C
6  C                      Дата создания          04/09/2006 11:52am
7  C                      Дата последней корректировки 05/09/2006 03:55pm
8  C
9  C
10
11  include 'init.inc'

C:\dinama\pradis32>
1Left  2Right  3View.. 4Edit.. 5Print  6MkLink  7Find  8History 9Video 10Tr
```

23. Теперь Вы можете пользоваться новой ПГО PGOTST как любой другой.