



Библиотека SCAD++ API

2015

ОГЛАВЛЕНИЕ

ОГЛАВЛЕНИЕ	2
1. Назначение	3
2. Создание и корректировка расчетных схем	4
2.1. Обработка ошибок	4
2.2. Инициализация объекта	5
2.3. Узлы	8
2.4. Элементы	11
2.5. Жесткости	16
2.6. Группы узлов и элементов	18
2.7. Блоки укрупненных элементов	21
2.8. Связи	23
2.9. Жесткие вставки	25
2.10. Системы координат элементов	27
2.11. Шарниры	31
2.12. Нагрузки	32
2.13. Упругое основание	38
2.14. Комбинации	39
2.15. Расчетные сочетания усилий	40
2.16. Арматура	44
2.17. Пример создания проекта	47
2.18. Пример чтения проекта	50
3. Анализ результатов расчета	51
3.1. Назначение	51
3.2. Инициализация	51
3.2. Наличие результатов расчета	52
3.3. Информация о типах результатов расчета и чтение их характеристик	53
3.4. Динамика	56
3.5. Премещения	56
3.6. Реакции в связях и от фрагмента схемы	57
3.7. Усилия и напряжения	59
3.8. РСУ и РСД	59
3.9. Арматура	61

1. Назначение

Для работы с проектами вычислительного комплекса SCAD++ создан специальный объект **ScadAPI**, с помощью которого можно создавать проекты, корректировать их, анализировать результаты расчетов. Описанная в данном документе библиотека предоставляет методы доступа к этому объекту.

Для работы с объектом рекомендуется ознакомиться с Языком архивации данных в книге "SCAD OFFICE Вычислительный комплекс SCAD", изданий 2011 и более поздних.

Необходимо помнить, что:

- данные и результаты расчетов в SCAD++ хранятся в **системе единиц СИ** (исключение составляют единицы измерения углов, для которых используются градусы, а не радианы). Для того, чтобы работать в других единицах измерения, необходимо воспользоваться соответствующими функциями преобразования;
- нумерация узлов, элементов, списков и т.п. всегда начинается с **ЕДИНИЦЫ**.
- контроль списков не производится;
- допускается одновременная работа нескольких объектов, работающими с разными задачами.

После инсталляции программы SCAD++ в директории, в которую была произведена инсталляция (например, "C:\SCAD Soft\SCAD Office") появится поддиректория API, в которой содержатся

- в поддиректории C:\SCAD Soft\SCAD Office\API\Include — необходимые заголовочные файлы (для использования функций и констант API и макросов достаточно добавить в исходный файл при помощи директивы `#include "ScadAPIX.hxx"`);
- в поддиректории C:\SCAD Soft\SCAD Office\API\Lib\32\ — библиотека, необходимая для линковки для платформы Win32;
- в поддиректории C:\SCAD Soft\SCAD Office\API\Lib\64\ — библиотека, необходимая для линковки для платформы x64;
- в поддиректории C:\SCAD Soft\SCAD Office\API\Demo — демонстрационный проект;

Для работы созданного пользователем приложения нужны также следующие динамически загружаемые библиотеки (которые можно найти для соответствующих платформ в директориях "C:\SCAD Soft\SCAD Office\32" и "C:\SCAD Soft\SCAD Office\64"):

SCADAPIX.dll
SCADAlien.dll
ProfileXXI.dll
SCADFemBase.dll
SprFEM.dll
SprResult.dll
SprSchema.dll
SprTools.dll
SprSchemaRc1033.dll
SprSchemaRc1049.dll
boost_thread-mt.dll
mkl_core.dll
mkl_intel_thread.dll

2. Создание и корректировка расчетных схем

2.1. Обработка ошибок

При работе программ API могут возникать различные ошибки, и необходимо анализировать коды возврата функций:

<i>APICode_OK</i>	– успешное завершение;
<i>APICode_InvalidHandle</i>	– неверен указатель на объект API;
<i>APICode_InternalError</i>	– программная ошибка;
<i>APICode_FatalError</i>	– программная ошибка;
<i>APICode_IndexError</i>	– ошибка индекса.

Если функция возвращает целочисленное значение, нуль может означать как ошибку, так и отсутствие данных. Аналогичная ситуация возникает и при возврате указателя на данные.

В этом случае надо воспользоваться для анализа ситуации функцией:

APICode ApiGetLastError (ScadAPI lpAPI);

Назначение

Анализ программной ошибки.

Параметры

lpAPI – указатель на объект API.

Возвращаемое значение

Код возврата.

При обнаружении ошибки, в некоторых случаях, можно получить сообщение о причине ошибки. Для получения сообщений необходимо воспользоваться функциями:

UINT ApiGetQuantityPhrase (ScadAPI lpAPI);

Назначение

Получить число выданных сообщений.

Параметры

lpAPI – указатель на объект API.

Возвращаемое значение

Число сообщений API.

LPCSTR ApiGetPhrase (ScadAPI lpAPI, UINT Num);

Назначение

Получение сообщения API.

Параметры

lpAPI – указатель на объект API;

Num – номер сообщения (1,2...).

Возвращаемое значение

Указатель на текст сообщения API.

2.2. Инициализация объекта

APICode ApiCreate(ScadAPI * lpAPI);

Назначение

Инициализация объекта **ScadAPI**.

Параметры

lpAPI – адрес указателя на объект API.

Возвращаемое значение

Код возврата.

APICode ApiSetLanguage (ScadAPI lpAPI, int Lang);

Назначение

Установка языка выдачи сообщений **ScadAPI**.

Параметры

lpAPI – адрес объекта API;

Lang – 0 - русский, 1- английский.

Возвращаемое значение

Код возврата.

APICode ApiClear(ScadAPI lpAPI);

Назначение

Создание нового проекта **Scad++**.

Параметры

lpAPI – указатель на объект API.

Возвращаемое значение

Код возврата.

APICode ApiSetUnits(ScadAPICH lpAPI, const UnitsAPI *Un);

Назначение

Установка системы единиц работы с проектом **SCAD++**.

Параметры

lpAPI – указатель на объект API;

Units – массив из трех структур UnitsAPI:

```
struct UnitsAPI
{
    char Name[10];
    float coef;
};
```

Единицы измерения задаются в следующем порядке:

- имя единицы измерения линейных размеров и коэффициент перевода ее в метры. Например для см – см и 100;
- имя единицы измерения размеров сечения стержней и коэффициент перевода ее в метры;
- имя единицы измерения сил и коэффициент перевода ее в Т.

Например:

```
UnitsAPI Un[3] = { { "m", 1 }, { "cm", 100 }, { "кГ", 1000 } };
```

Возвращаемое значение

Код возврата.

LPCUnitsAPI ApiGetUnits(ScadAPICH handle);

Назначение

Получение системы единиц работы с проектом **Scad++**.

Параметры

lpAPI – указатель на объект API.

Возвращаемое значение

Указатель на массив структур UnitsAPI (см. выше).

APICode ApiReadProject(ScadAPICH handle, LPCSTR Name);

Назначение

Чтение проекта **Scad++**.

Параметры

lpAPI – указатель на объект API;

Name – полное имя файла проекта. Если задано NULL, то откроется стандартный диалог чтения файла.

Возвращаемое значение

Код возврата.

После чтения проекта необходимо проверить и при необходимости переопределить систему единиц работы с ним.

APICode ApiWriteProject(ScadAPICH handle, LPCSTR Name);

Назначение

Запись проекта на устройство.

Параметры

lpAPI – указатель на объект API;

Name – путь к файлу проекта. Если задано NULL, то откроется стандартный диалог сохранения файла.

Возвращаемое значение

Код возврата.

UINT ApiGetTypeSchema(ScadAPI lpAPI);

Назначение

Чтение типа данных схемы.

Параметры

lpAPI – указатель на объект API.

Возвращаемое значение

Тип данных:

- 1 – расчетная схема SCAD;
- 2 – препроцессор FORUM;
- 3 – вариация моделей;
- 4 – монтаж.

UINT ApiGetTypeSystem(ScadAPI lpAPI);

Назначение

Чтение признака расчетной схемы. По умолчанию система общего вида – 5.

Параметры

lpAPI – указатель на объект API.

Возвращаемое значение

Тип схемы.

APICode ApiSetTypeSystem(ScadAPI lpAPI, UINT Num);

Назначение

Задание признака расчетной схемы. По умолчанию система общего вида – 5.

Параметры

lpAPI – указатель на объект API;

Num – тип схемы.

Возвращаемое значение

Код возврата.

APICode ApiSetName(ScadAPI lpAPI, LPCSTR Name);

Назначение

Задание имени проекта.

Параметры

lpAPI – указатель на объект API;

Name – имя проекта.

Возвращаемое значение

Код возврата.

LPCSTR ApiGetName(ScadAPI lpAPI);

Назначение

Чтение имени проекта.

Параметры

lpAPI – указатель на объект API.

Возвращаемое значение

Указатель на строку с именем проекта.

LPCSTR ApiGetProjectNameFile(ScadAPI lpAPI);

Назначение

Чтение имени файла проекта (для прочитанных).

Параметры

lpAPI – указатель на объект API.

Возвращаемое значение

указатель на имя файла проекта.

APICode ApiSetWorkCatalog(ScadAPI lpAPI, LPCSTR Txt);

Назначение

Задание полного имени рабочего каталога с результатами расчета.

Параметры

lpAPI – указатель на объект API;

Txt – полное имя каталога.

Возвращаемое значение

Код возврата.

2.3. Узлы

UINT ApiGetQuantityNode(ScadAPI lpAPI);

Назначение

Получить число узлов расчетной схемы.

Параметры

lpAPI – указатель на объект API.

Возвращаемое значение

Число узлов расчетной схемы.

LPCNodeApi ApiGetNode(ScadAPI lpAPI, UINT NumNode);

Назначение

Чтение координат узла расчетной схемы.

Параметры

lpAPI – указатель на объект API;

NumNode – номер узла.

Возвращаемое значение

Указатель на структуру *CNodeApi* с координатами узла:

```
struct CNodeApi
{
    LPCSTR    Text;    // имя узла
    double    x, y, z; // координаты
    BYTE      Flag;    // специальные флаги
};
```

UINT ApiNodeAddSize(ScadAPI lpAPI, UINT Qnt);

Назначение

Добавление узлов расчетной схемы.

Параметры

lpAPI – указатель на объект API;

Qnt – число добавляемых узлов с нулевыми координатами.

Возвращаемое значение

Номер первого добавленного узла. Если он равен нулю, то добавление не выполнено.

UINT ApiNodeAdd(ScadAPI lpAPI, UINT Qnt, const CNodeApi *ck);

Назначение

Добавление узлов расчетной схемы.

Параметры

lpAPI – указатель на объект API;

Qnt – число добавляемых узлов;

ck – указатель на массив структур *CNodeApi* с координатами узлов и их именами.

Возвращаемое значение

Номер первого добавленного узла. Если он равен нулю, то добавление не выполнено.

UINT ApiNodeAddOne(ScadAPI lpAPI, double x, double y, double z);

Назначение

Добавление узла расчетной схемы.

Параметры

lpAPI – указатель на объект API;

x, y, z – координаты узла.

Возвращаемое значение

Номер добавленного узла. Если он равен нулю, то добавление не выполнено.

APICode ApiNodeUpdate(ScadAPI lpAPI, UINT NumNode, double x, double y, double z);

Назначение

Корректировка координат узла расчетной схемы.

Параметры

lpAPI – указатель на объект API;

NumNode – номер узла;

x, y, z – координаты узла.

Возвращаемое значение

Код возврата.

APICode ApiNodeSetName(ScadAPI lpAPI, UINT NumNode, LPCSTR Text);

Назначение

задание имени узла.

Параметры

lpAPI – указатель на объект API;

NumNode – номер узла;

Text – имя узла.

Возвращаемое значение

Код возврата.

LPCSTR ApiNodeGetName(ScadAPI lpAPI, UINT Num);

Назначение

Чтение имени узла.

Параметры

lpAPI – указатель на объект API;

Num – номер узла.

Возвращаемое значение

Указатель на строку с именем узла.

APICode ApiDeleteNode(ScadAPI lpAPI, UINT Num, BOOL YesDeleteElem);

Назначение

Удаление узла.

Параметры

lpAPI – указатель на объект API;

Num – номер узла;

YesDeleteElem – удалить (TRUE) конечные элементы, примыкающие к данному узлу.

Возвращаемое значение

Код возврата.

APICode ApiDeleteNodeList(ScadAPI lpAPI, UINT Qnt, const UINT * List, BOOL YesDeleteElem);

Назначение

Удаление узлов.

Параметры

lpAPI – указатель на объект API;

Qnt – число удаляемых узлов;

List – указатель на номера узлов,

YesDeleteElem – удалить (TRUE) конечные элементы, примыкающие к заданным узлам.

Возвращаемое значение

Код возврата.

APICode ApiUnDeleteNode(ScadAPI lpAPI, UINT Num);

Назначение

Восстановление удаленного узла.

Параметры

lpAPI – указатель на объект API;

Num – номер узла.

Возвращаемое значение

Код возврата.

APICode ApiUnDeleteNodeList(ScadAPI lpAPI, UINT Qnt, const UINT * List);

Назначение

восстановление удаленных узлов.

Параметры

lpAPI – указатель на объект API;

Qnt – число восстанавливаемых узлов;

List – указатель на номера узлов,

Возвращаемое значение

Код возврата.

UINT ApiIsNodeDeleted(ScadAPI lpAPI, UINT Num);

Назначение

Проверка удаленности узла.

Параметры

lpAPI – указатель на объект API;

Num – номер узла.

Возвращаемое значение

1 - удален, 0 – нет.

2.4. Элементы

UINT ApiGetElemQuantity(ScadAPI lpAPI);

Назначение

Получить число элементов расчетной схемы.

Параметры

lpAPI – указатель на объект API.

Возвращаемое значение

Число элементов расчетной схемы.

UINT ApiElemAdd(ScadAPI lpAPI, UINT Qnt);

Назначение

Добавление элементов расчетной схемы.

Параметры

lpAPI – указатель на объект API;

Qnt – число добавляемых элементов с нулевыми данными.

Возвращаемое значение

Номер первого добавленного элемента. Если он равен нулю, то добавление не выполнено.

**APICode ApiElemGetData(ScadAPI lpAPI, UINT Num, UINT * Type,
UINT * NumRgd, UINT * QntNd, const UINT ** ListNode);**

Назначение

Чтение данных о элементе.

Параметры

lpAPI – указатель на объект API;

Num – номер элемента;

Type – указатель на тип элемента;

NumRgd – указатель на номер типа жесткости;

QntNd – указатель на число узлов элемента;

ListNode – адрес указателя на номера узлов.

Возвращаемое значение

Код возврата.

APICode ApiElemGetInf(ScadAPI lpAPI, UINT Num, CElemInfApi * ElemInf);

Назначение

Чтение данных о элементе.

Параметры

lpAPI – указатель на объект API;

Num – номер элемента;

ElemInf – указатель на структуру *CElemInfApi* с координатами узла:

struct CElemInfApi

```
{
    LPCSTR  Text;           // имя узла
    UINT     QuantityNode;  // число узлов
    UINT     * Node;        // номера узлов
    WORD     TypeElem;      // тип элемента
    BYTE     IsDeletet;     // признак удаленного элемента
    UINT     TypeRigid;     // тип жесткости
    UINT     NumInsert;     // номер списка жестких вставок
}
```

```

    UINT    NumSysCoord;    // номер списка системы коорд. жесткостей
    UINT    NumSysCoordEffors; // система коорд. вычисления усилий/напряжений
    UINT    NumBed;         // номер списка коэффициентов постели.
    UINT    NumStress;      // номер списка преднапряжения
    struct {
        UINT Quantity;    // число отверстий
        UINT * Pointer;    // порядковые номера начала отверстий в списке узлов элемента
    } * Hole;            // отверстия при их наличии
};

```

Возвращаемое значение

Код возврата.

UINT ApiElemAddData(ScadAPI lpAPI, UINT Type, UINT QuantityNode, const UINT * ListNode);

Назначение

Добавление элемента.

Параметры

lpAPI – указатель на объект API;

Type – тип элемента;

QuantityNode – число узлов элемента;

ListNode – указатель на массив номеров узлов.

Возвращаемое значение

Номер первого добавленного элемента. Если он равен нулю, то добавление не выполнено.

APICode ApiElemUpdate(ScadAPI lpAPI, UINT Num, UINT Type, UINT QuantityNode, const UINT * ListNode);

Назначение

Корректировка элемента.

Параметры

lpAPI – указатель на объект API;

Num – номер элемента;

Type – тип элемента;

QuantityNode – число узлов элемента;

ListNode – указатель на массив номеров узлов.

Возвращаемое значение

Код возврата.

APICode ApiElemSetName(ScadAPI lpAPI, UINT Num, LPCSTR Text);

Назначение

Задание имени элемента.

Параметры

lpAPI – указатель на объект API;

Num – номер элемента;

Text – имя элемента.

Возвращаемое значение

Код возврата.

LPCSTR ApiElemGetName(ScadAPI lpAPI, UINT Num);

Назначение

Чтение имени элемента.

Параметры

lpAPI – указатель на объект API;

Num – номер элемента;

Возвращаемое значение

Указатель на строку с именем элемента

APICode ApiElemSetType(ScadAPI lpAPI, UINT Type, UINT ElemBegin, UINT ElemEnd=0);

Назначение

Задание типа элементов.

Параметры

lpAPI – указатель на объект API;

Type – тип элемента;

ElemBegin – начальный номер элемента, которому задается тип;

ElemEnd – конечный номер.

Возвращаемое значение

Код возврата.

APICode ApiDeleteElem(ScadAPI lpAPI, UINT Num);

Назначение

Удаление элемента.

Параметры

lpAPI – указатель на объект API;

Num – номер элемента.

Возвращаемое значение

Код возврата.

APICode ApiDeleteElemList(ScadAPI lpAPI, UINT Qnt, const UINT * List);

Назначение

Удаление элементов.

Параметры

lpAPI – указатель на объект API;

Qnt – число элементов;

List – указатель на массив номеров элементов.

Возвращаемое значение

Код возврата.

APICode ApiUnDeleteElem(ScadAPI lpAPI, UINT Num);

Назначение

Составление удаленного элемента.

Параметры

lpAPI – указатель на объект API;

Num – номер элемента.

Возвращаемое значение

Код возврата.

APICode ApiUnDeleteElemList(ScadAPI lpAPI, UINT Qnt, const UINT * List);

Назначение

Восстановление удаленных элементов.

Параметры

lpAPI – указатель на объект API;

Qnt – число элементов;

List – указатель на массив номеров элементов.

Возвращаемое значение

Код возврата.

UINT ApiIsElemDeleted(ScadAPI lpAPI, UINT NumElem);

Назначение

Проверка удаленности элемента.

Параметры

lpAPI – указатель на объект API;

NumElem – номер элемента.

Возвращаемое значение

1 - удален, 0 – нет.

UINT ApiElemGetQuantityHole(ScadAPI lpAPI, UINT Num);

Назначение

Получить число внутренних контуров в укрупненном элементе.

Параметры

lpAPI – указатель на объект API;

Num – номер элемента.

Возвращаемое значение

Число внутренних контуров.

**APICode ApiElemGetHole(ScadAPI lpAPI, UINT Num, UINT NumContur,
UINT *QuantityNode, const UINT ** ListNode);**

Назначение

Получить внутренний контур в укрупненном элементе.

Параметры

lpAPI – указатель на объект API;

Num – номер элемента;

NumContur – номер контура (с 1);

QuantityNode – указатель на число узлов контура;

ListNode – адрес указателя на номера узлов контура.

Возвращаемое значение

Код возврата.

**APICode ApiElemSetHole(ScadAPI lpAPI, UINT Num, UINT QuantityHole,
const UINT * ListBeginNodeHole);**

Назначение

Задание внутренних контуров в укрупненном элементе.

Параметры

lpAPI – указатель на объект API;

Num – номер элемента;

QuantityHole – число внутренних контуров;

ListBeginNodeHole – указатель на список порядковых номеров узлов элемента, с которых начинаются внутренние контура (нумерация с 1).

Возвращаемое значение

Код возврата.

2.5. Жесткости

UINT ApiGetQuantityRigid(ScadAPI lpAPI);

Назначение

Получение числа типов жесткостей.

Параметры

lpAPI – указатель на объект API.

Возвращаемое значение

Число типов жесткостей.

UINT ApiSetRigid(ScadAPI lpAPI, LPCSTR Text);

Задание жесткостных характеристик.

Параметры

lpAPI – указатель на объект API;

Text – текст с жесткостными характеристиками. Рекомендуется использовать вывод в текстовый формат данных SCADa в соответствии с "Языком архивации данных". Например:

```
Text = "S0 3.52e+006 20 25 NU 0.2 RO 2.5 TMP 1e-005 Shift 493.004 61488.2 61548.5";  
// прямоугольное пользовательское сечение в "Т", "см" и "м".  
Text = "GE 2.1e+007 0.3 0.1 RO 7.85 TMP 1.2e-005 1.2e-005";  
// характеристики плиты в "Т" и "м".
```

Возвращаемое значение

номер типа жесткости или 0 при наличии ошибок.

UINT ApiChangeRigid(ScadAPI lpAPI, UINT Num, LPCSTR Text);

Назначение

Изменение жесткостных характеристик.

Параметры

lpAPI – указатель на объект API;

Num – номер жесткостных характеристик;

Text – текст с жесткостными характеристиками.

Возвращаемое значение

номер типа жесткости или 0 при наличии ошибок.

LPCSTR ApiGetRigidName(ScadAPI lpAPI, UINT Num);

Назначение

Получить имя типа жесткости.

Параметры

lpAPI – указатель на объект API;

Num – номер жесткостных характеристик

Возвращаемое значение

Указатель на строку с именем типа жесткости.

APICode ApiSetRigidName(ScadAPI lpAPI, UINT Num, LPCSTR Name);

Назначение

Задание имени типа жесткости.

Параметры

lpAPI – указатель на объект API;

Num – номер жесткостных характеристик.

Name – имя жесткостных характеристик.

Возвращаемое значение

Код возврата.

APICode ApiSetRigidElem(ScadAPI lpAPI,UINT Num,UINT Qnt,const UINT * ListElem);

Назначение

Задание жесткостных характеристик элементам.

Параметры

lpAPI – указатель на объект API;

Num – номер типа жесткости.

Qnt – число элементов;

ListElem – указатель на массив номеров элементов.

Возвращаемое значение

Код возврата.

2.6. Группы узлов и элементов

Используются следующие обозначения для идентификации типов групп элементов:

```
typedef enum : BYTE {  
    ApiGroupUndefined    = 0,  
    ApiGroupRod           = 1,  
    ApiGroupPlate        = 2,  
    ApiGroupVolume       = 3,  
    ApiGroupSpecial      = 4,  
    ApiGroupAxecymmetric = 5,  
} ApiGroupType;
```

UINT ApiGetQuantityGroupElem(ScadAPI lpAPI);

Назначение

число групп элементов.

Параметры

lpAPI – указатель на объект API.

Возвращаемое значение

число групп элементов.

**APICode ApiGetGroupElem(ScadAPI lpAPI, UINT Num, UINT *Type,
 UINT *Qnt, const UINT **ListElem, LPCSTR * Text);**

Назначение

Получение информация о группе элементов.

Параметры

lpAPI – указатель на объект API.

Num – номер группы;

Type – указатель на тип группы;

Qnt – указатель на число элементов группы;

ListElem – указатель на адрес списка элементов группы;

Text – указатель на имя группы.

Возвращаемое значение

Код возврата.

**UINT ApiSetGroupElem(ScadAPI lpAPI, LPCSTR Text, BYTE Type, UINT Qnt, const
UINT *ListElem);**

Назначение

Формирование группы элементов.

Параметры

lpAPI – указатель на объект API.

Text – имя группы;

Type – тип группы.

Qnt – число элементов группы;

ListElem – список элементов группы.

Возвращаемое значение

Номер добавленной группы.

APICode ApiSetNameGroupElem(ScadAPI lpAPI, UINT Num, LPCSTR Name);

Назначение

Задание имени группы элементов.

Параметры

lpAPI – указатель на объект API.

Num – номер группы;

Name – имя группы.

Возвращаемое значение

Код возврата.

LPCSTR ApiGetNameGroupElem(ScadAPI lpAPI, UINT Num);

Назначение

имя группы элементов.

Параметры

lpAPI – указатель на объект API.

Num – номер группы.

Возвращаемое значение

Указатель на строку с именем группы.

UINT ApiGetQuantityGroupNode(ScadAPI lpAPI);

Назначение

Получить число групп узлов.

Параметры

lpAPI – указатель на объект API.

Возвращаемое значение

Число групп узлов.

**APICode ApiGetGroupNode(ScadAPI lpAPI, UINT Num, UINT *Qnt,
const UINT **ListNode, LPCSTR * Text);**

Назначение

Получить информацию о группе узлов.

Параметры

lpAPI – указатель на объект API.

Num – номер группы;

Qnt – указатель на число элементов группы;

ListElem – указатель на адрес списка элементов группы;

Text – указатель на имя группы.

Возвращаемое значение

Код возврата.

**UINT ApiSetGroupNode(ScadAPI lpAPI, LPCSTR Text,
UINT Qnt, const UINT *ListNode);**

Назначение

Формирование группы узлов.

Параметры

lpAPI – указатель на объект API.

Text – имя группы;

Qnt – число элементов группы;

ListElem – список элементов группы.

Возвращаемое значение

Номер добавленной группы.

APICode ApiSetNameGroupNode(ScadAPI lpAPI, UINT Num, LPCSTR Name);

Назначение

Задание имени группы узлов.

Параметры

lpAPI – указатель на объект API.

Num – номер группы;

Name – имя группы.

Возвращаемое значение

Код возврата.

LPCSTR ApiGetNameGroupNode(ScadAPI lpAPI, UINT Num);

Назначение

Получение имени группы узлов.

Параметры

lpAPI – указатель на объект API.

Num – номер группы.

Возвращаемое значение

Указатель на строку с именем группы.

2.7. Блоки укрупненных элементов

UINT ApiGetQuantityBlock(ScadAPI lpAPI);

Назначение

Получить число блоков элементов.

Параметры

lpAPI – указатель на объект API.

Возвращаемое значение

Число блоков элементов.

**APICode ApiGetBlock(ScadAPI lpAPI, UINT Num, UINT *Qnt,
const UINT **ListElem, LPCSTR * Text, COLORREF *Col);**

Назначение

Получить информацию о блоке.

Параметры

lpAPI – указатель на объект API.

Num – номер блока;

Qnt – число элементов блока;

ListElem – указатель на список номеров элементов группы;

Text – имя блока;

Col – цвет блока.

Возвращаемое значение

Код возврата.

**UINT ApiSetBlock(ScadAPI lpAPI, LPCSTR Text, COLORREF Col,
UINT Qnt, const UINT *ListElem);**

Назначение

Создание блока элементов.

Параметры

lpAPI – указатель на объект API.

Text – имя блока;

Col – цвет блока;

Qnt – число элементов блока;

ListElem – указатель на список номеров элементов блока.

Возвращаемое значение

Номер добавленного блока.

APICode ApiSetNameBlock(ScadAPI lpAPI, UINT Num, LPCSTR Name);

Назначение

Задание имени блока.

Параметры

lpAPI – указатель на объект API.

Num – номер блока;

Name – имя блока

Возвращаемое значение

Код возврата.

LPCSTR ApiGetNameBlock(ScadAPI lpAPI, UINT Num);

Назначение

Получение имени блока.

Параметры

lpAPI – указатель на объект API.

Num – номер блока.

Возвращаемое значение

Указатель на строку с именем блока.

APICode ApiDeleteBlock(ScadAPI lpAPI, UINT Num);

Назначение

Удаление блока.

Параметры

lpAPI – указатель на объект API.

Num – номер блока.

Возвращаемое значение

Код возврата.

2.8. Связи

Для работы со связями введены следующие определения для задания связей по соответствующим направлениям:

<i>SgDirectX</i>	<i>SgDirectAX</i>	<i>SgDirectCX</i>
<i>SgDirectY</i>	<i>SgDirectAY</i>	<i>SgDirectCY</i>
<i>SgDirectZ</i>	<i>SgDirectAZ</i>	<i>SgDirectCZ</i>
<i>SgDirectUX</i>	<i>SgDirectBX</i>	
<i>SgDirectUY</i>	<i>SgDirectBY</i>	
<i>SgDirectUZ</i>	<i>SgDirectBZ</i>	

UINT ApiGetBound(ScadAPI lpAPI, UINT NumNode);

Назначение

Получение заданных на узел связей.

Параметры

lpAPI – указатель на объект API.

NumNode – номер узла.

Возвращаемое значение

Маска связей, наложенных на узел. Например, для определения наличия связи по X надо выполнить операцию *<маска> & SgDirectX*.

APICode ApiSetBound(ScadAPI lpAPI, WORD Mask, UINT QntNode, const UINT * ListNode, BOOL YesClear);

Назначение

Задание связей на узлы.

Параметры

lpAPI – указатель на объект API;

Mask – маска связей. *Например: SgDirectX | SgDirectZ | SgDirectUY;*

QntNode – число узлов;

ListNode – указатель на список номеров узлов;

YesClear – добавить к существующим (FALSE) или удалить предыдущие назначения и установить указанные (TRUE);

Возвращаемое значение

Код возврата.

UINT ApiGetQuantityBoundUnite(ScadAPI lpAPI);

Назначение

Получить число объединений перемещений.

Параметры

lpAPI – указатель на объект API.

Возвращаемое значение

Число объединений перемещений.

LPCSTR ApiGetBoundUniteName(ScadAPI lpAPI, UINT Num);

Назначение

Получение имени объединения перемещений.

Параметры

lpAPI – указатель на объект API;

Num – номер объединения.

Возвращаемое значение

Указатель на строку с именем объединения перемещений.

APICode ApiSetBoundUniteName(ScadAPI lpAPI, UINT Num, LPCSTR Name);

Назначение

Задание имени объединения перемещений.

Параметры

lpAPI – указатель на объект API;

Num – номер объединения;

Name – имя объединения.

Возвращаемое значение

Код возврата.

APICode ApiGetBoundUnite(ScadAPI lpAPI, UINT Num, WORD *Mask, UINT *QntNode, const UINT **ListNode);

Назначение

Получить информацию об объединении перемещений.

Параметры

lpAPI – указатель на объект API;

Num – номер объединения.

Mask – указатель на маску объединения перемещений;

QntNode – указатель на число узлов;

ListNode – указатель на адрес списка узлов.

Возвращаемое значение

Код возврата.

UINT ApiSetBoundUnite(ScadAPI lpAPI, WORD Mask, UINT QntNode, const UINT * ListNode);

Назначение

Задание объединения перемещений.

Параметры

lpAPI – указатель на объект API;

Mask – маска объединения связей. *Например, SgDirectX | SgDirectZ;*

QntNode – число узлов;

ListNode – указатель на список номеров узлов.

Возвращаемое значение

Номер добавленного объединения.

APICode ApiDeleteBoundUnite(ScadAPI lpAPI, UINT Num);

Назначение

Удаление объединения перемещений.

Параметры

lpAPI – указатель на объект API.

Num – номер объединения перемещений.

Возвращаемое значение

Код возврата.

2.9. Жесткие вставки

UINT ApiGetQuantityInsert(ScadAPI lpAPI);

Назначение

Получить число типов жестких вставок.

Параметры

lpAPI – указатель на объект API.

Возвращаемое значение

Число типов жестких вставок.

**BOOL ApiGetNumInsert(ScadAPI lpAPI, UINT Num, BYTE *Type,
UINT *QntSize, double * Size, UINT *QntEl, const UINT ** ListEl);**

Назначение

Получить информацию о жесткой вставке.

Параметры

lpAPI – указатель на объект API;

Num – номер жесткой вставки;

Type – указатель на тип системы координат жесткой вставки;

QntSize – указатель на число данных;

Size – указатель на адрес данных (по три значения проекций на соответствующие оси на каждый узел элемента);

QntEl – указатель на число КЭ, у которых заданы жесткие вставки;

ListEl – указатель на адрес списка элементов, у которых заданы жесткие вставки.

**BOOL ApiGetInsert(ScadAPI lpAPI, UINT Num, BYTE *Type,
UINT *QntSize, double * Size);**

Назначение

Получить информацию о жесткой вставке.

Параметры

lpAPI – указатель на объект API;

Num – номер элемента;
Type – указатель на тип системы координат жесткой вставки;
QntSize – указатель на число данных;
Size – указатель на адрес данных (по три значения проекций на соответствующие оси на каждый узел элемента).

Возвращаемое значение

Есть (TRUE) или нет (FALSE) жесткая вставка.

**BOOL ApiGetInsertNode(ScadAPI lpAPI, UINT NumElem,
UINT NumNodeElem, BYTE *Type, double * Size);**

Назначение

Получить информацию о жесткой вставке на элементе.

Параметры

lpAPI – указатель на объект API;

NumElem – номер элемента;

NumNodeElem – порядковый номер узла (с 1);

Type – указатель на тип системы координат жесткой вставки;

Size – указатель на адрес данных (три значения).

Возвращаемое значение

Есть (TRUE) или нет (FALSE) жесткая вставка.

**UINT ApiSetInsert(ScadAPI lpAPI, UINT Group, BYTE Type, UINT QntSize,
const double * Size, UINT QntElem, const UINT * ListElem);**

Назначение

Задание жесткой вставки.

Параметры

lpAPI – указатель на объект API;

Group – тип элемента (пока только *ApiGroupRod*) ;

Type – тип системы координат жесткой вставки;

QntSize – число данных;

Size – адрес данных (по три значения на каждый узел элемента).

QntElem – число элементов;

ListElem – указатель на список номеров элементов.

Возвращаемое значение

номер заданных данных.

**APICode ApiClearInsert(ScadAPI lpAPI, UINT QntElem,
const UINT * ListElem);**

Назначение

Удаление жесткой вставки.

Параметры

lpAPI – указатель на объект API;

QntElem – число элементов;

ListElem – указатель на список номеров элементов.

Возвращаемое значение

Код возврата.

2.10. Системы координат элементов

Используются следующие обозначения для идентификации типов систем координат:

- стержни, координатные оси:
ApiRodCornerInDegrees – угол в градусах
ApiRodCornerInRadians – угол в радианах
ApiRodPointFocusAxecY – точка, на которую ориентирована гл. ось Y
ApiRodVectorFocusAxecY – вектор, на которую ориентирована гл. ось Y
ApiRodPointFocusAxecZ – точка, на которую ориентирована гл. ось Z
ApiRodVectorFocusAxecZ – вектор, на которую ориентирована гл. ось Z
- стержни, главные оси:
ApiRodPointFocusGeomAxecY – точка, на которую ориентирована геом. ось Y
ApiRodVectorFocusGeomAxecY – вектор, на которую ориентирована геом. ось Y
ApiRodPointFocusGeomAxecZ – точка, на которую ориентирована геом. ось Z
ApiRodVectorFocusGeomAxecZ – вектор, на которую ориентирована геом. ось Z
- плиты:
ApiPlateAxecX – направление X в глобальной СК
ApiPlatePointFocusAxecX – направление X на точку
ApiPlatePointFocusCentersAxecX – направление X на точку из центра
ApiPlateAxecY – направление Y в глобальной СК
ApiPlatePointFocusAxecY – направление Y на точку
ApiPlatePointFocusCentersAxecY – направление Y на точку из центра
- трехмерные элементы:
ApiVolumeAxecXY – направление X, Y
ApiVolumeAxecXZ – направление X, Z
ApiVolumeAxecYZ – направление Y, Z
ApiVolumeCornersInDegrees – углы Эйлера в градусах
ApiVolumeCornersInRadians – углы Эйлера в радианах
ApiVolumeCylinderTwoPointsZ – цилиндр. Задаются две точки на оси от и до
ApiVolumeCylinderPointsAndVectorZ – цилиндр. Задаются точка и направление
ApiVolumeSphereCentreAndPointY – сфера. Задаются точка и точка, на которую ориентирована ось Y
ApiVolumeSphereCentreAndVectorY – сфера. Задаются точка и направление оси Y
При этом, координаты точек и векторов задаются тремя координатами по направлениям соответствующих осей выбранной системы.

UINT *ApiGetQuantitySystemCoordElem(ScadAPI lpAPI);*

Число систем координат элементов (ориентация местных осей стержней и жесткостных характеристик пластин).

Назначение

Параметры

lpAPI – указатель на объект API.

Возвращаемое значение

Число систем координат.

**BOOL ApiGetSystemCoordElemOne(ScadAPI lpAPI, UINT NumElem,
BYTE *Type, UINT *QntSize, double * Size);**

Назначение

Система координат элемента (ориентация местных осей стержней и осей ортотропии или анизотропии материала).

Параметры

lpAPI – указатель на объект API;

NumElem – номер элемента;

Type – указатель на тип;

QntSize – указатель на число данных;

Size – адрес данных для сохранения (максимальная размерность – 6).

Возвращаемое значение

Задана (TRUE) или не задана (FALSE).

**BOOL ApiGetSystemCoordElem(ScadAPI lpAPI, UINT Num, BYTE *Type,
UINT *QntSize, const double ** Size, UINT * QntList, const UINT ** ListElem);**

Назначение

Система координат элементов (ориентация местных осей стержней и осей ортотропии или анизотропии материала).

Параметры

lpAPI – указатель на объект API;

Num – номер системы координат;

Type – указатель на тип;

QntSize – указатель на число данных;

Size – адрес указателя данных;

QntElem – указатель на число элементов;

ListElem – указатель на адрес списка элементов.

Возвращаемое значение

Задана (TRUE) или не задана (FALSE).

**UINT ApiSetSystemCoordElem(ScadAPI lpAPI, BYTE Group, BYTE Type,
UINT QntSize, const double * Size, UINT QntElem, const UINT * ListElem);**

Назначение

Задание системы координат элементов (ориентация местных осей стержней и осей ортотропии или анизотропии материала).

Параметры

lpAPI – указатель на объект API;

Group – тип элемента (*ApiGroupRod* и др.) ;

Type – тип;

QntSize – число данных;

Size – указатель на данные;

QntElem – число элементов;

ListElem – адрес списка элементов.

Возвращаемое значение

номер добавленных данных.

**APICode ApiClearSystemCoordElem(ScadAPI lpAPI, UINT QntElem,
const UINT * ListElem);**

Назначение

Удаление системы координат элементов (ориентация местных осей стержней и осей ортотропии или анизотропии материала).

Параметры

lpAPI – указатель на объект API;

QntElem – число элементов;

ListElem – адрес списка элементов.

Возвращаемое значение

Код возврата.

UINT ApiGetQuantitySystemCoordEffors(ScadAPI lpAPI);

Назначение

Число систем координат элементов для вычисления напряжений в пластинах и объемных элементах.

Параметры

lpAPI – указатель на объект API.

Возвращаемое значение

число систем координат.

**BOOL ApiGetSystemCoordEfforsOne(ScadAPI lpAPI, UINT NumElem,
BYTE *Type, UINT *QntSize, double * Size);**

Назначение

Система координат элемента для вычисления напряжений для пластин и объемных элементов.

Параметры

lpAPI – указатель на объект API;

NumElem – номер элемента;

Type – указатель на тип;

QntSize – указатель на число данных;

Size – адрес данных для сохранения (максимальная размерность – 6).

Возвращаемое значение

Задана (TRUE) или не задана (FALSE).

**BOOL ApiGetSystemCoordEffors(ScadAPI lpAPI, UINT Num, BYTE *Type,
UINT *QntSize, const double ** Size, UINT * QntList, const UINT ** ListElem);**

Назначение

Система координат элементов для вычисления напряжений для пластин и объемных элементов.

Параметры

lpAPI – указатель на объект API;

Num – номер системы координат;

Type – указатель на тип;

QntSize – указатель на число данных;

Size – адрес указателя данных;

QntElem – указатель на число элементов;

ListElem – указатель на адрес списка элементов.

Возвращаемое значение

Задана (TRUE) или не задана (FALSE).

**UINT ApiSetSystemCoordEffors(ScadAPI lpAPI, BYTE Group, BYTE Type,
UINT QntSize, const double * Size, UINT QntElem, const UINT * ListElem);**

Назначение

Задание системы координат элементов для вычисления напряжений для пластин и объемных элементов.

Параметры

lpAPI – указатель на объект API;

Group – тип элемента (*ApiGroupRod* и др.) ;

Type – тип;

QntSize – число данных;

Size – указатель на данные;

QntElem – число элементов;

ListElem – адрес списка элементов.

Возвращаемое значение

Номер типа добавленных данных.

**APICode ApiClearSystemCoordEffors(ScadAPI lpAPI, UINT QntElem,
const UINT * ListElem);**

Назначение

Удаление системы координат элементов для вычисления напряжений для пластин и объемных элементов.

Параметры

lpAPI – указатель на объект API;

QntElem – число элементов;

ListElem – адрес списка элементов.

Возвращаемое значение

Код возврата.

2.11. Шарниры

**UINT ApiGetJoint(ScadAPI lpAPI, UINT NumElem, UINT NumNodeElem,
BYTE * Place);**

Назначение

Информация о шарнирах в узлах элементов.

Параметры

lpAPI – указатель на объект API;

NumElem – номер элемента;

NumNodeElem - порядковый номер узла (с 1);

Place – указатель на положение шарнира : узел(1) или гибкая часть(0);

Возвращаемое значение

Маска шарниров: для определения наличия шарнира вокруг оси X надо выполнить операцию <маска> & SgDirectUX.

**APICode ApiSetJoint(ScadAPI lpAPI, WORD Mask, UINT NumElem,
UINT NumNodeElem, BYTE Place);**

Назначение

Задание шарниров в узлах элементов.

Параметры

lpAPI – указатель на объект API;

Mask – маска связей. Например, SgDirectX | SgDirectZ | SgDirectUY;

NumElem – номер элемента;

NumNodeElem – порядковый номер узла (нумерация с 1);

Place – положение шарнира : узел (1) или гибкая часть (0).

Возвращаемое значение

Код возврата.

APICode ApiDeleteJoint(ScadAPI IpAPI, UINT NumElem, UINT NumNodeElem);

Назначение

Удаление шарниров в узлах элементов.

Параметры

IpAPI – указатель на объект API;

NumElem – номер элемента;

NumNodeElem – порядковый номер узла (нумерация с 1).

Возвращаемое значение

Код возврата.

APICode ApiGetRodJoint(ScadAPI IpAPI, UINT NumElem, WORD *JointElastic, WORD *JointInsert);

Назначение

Информация о шарнирах в узлах стержневых элементов.

Параметры

IpAPI – указатель на объект API;

NumElem – номер элемента;

JointElastic – указатель на шарниры на гибкой части;

JointInsert – указатель на шарниры в узлах элемента при наличии жестких вставок.

JointElastic и *JointInsert* являются битовыми масками, в которых первые 6-ть битов описывают шарниры 1-го узла, а следующие – второго.

Возвращаемое значение

Код возврата.

2.12. Нагрузки

Используются следующие обозначения для идентификации видов нагрузки:

- узловая:
ApiForceNode – узловая
- заданные перемещения:
ApiForceNodeDisplace – заданные перемещения узла;
ApiForceNodeSpecial – заданные перемещения узла в специальной системе координат;
ApiForceNullElem – задание перемещения узла через нуль-элемент;
- в местной системе координат элемента:
ApiForcePointLocal – сосредоточенная в точке элемента;
ApiForceEvenlyLocal – равномернораспределенная для стержня по оси, для пластин по – площади, для, для трехмерных элементов – по объему или грани
ApiForceTrapezLocal – стержня по отрезку на оси, для пластин по – площади, для трехмерных элементов – по объему
ApiForceLineEvenlyLocal – равномернораспределенная по линии, соединяющей два узла элемента;
ApiForceLineTrapezLocal – трапециевидная по линии, соединяющей два узла элемента;
ApiForcePointPartLocal – сосредоточенная в точке элемента в долях длины для стержней;
ApiForceEvenlyLocalIns – равномернораспределенная по стержню с учетом жестких вставок;
ApiForceTrapezPartLocal – трапециевидная по отрезку стержня, заданного в долях длины;
- в общей системе координат элемента:
- *ApiForcePointGlobal* – сосредоточенная в точке элемента;

- *ApiForceEvenlyGlobal* – равномерно распределенная для стержня по оси, для пластин по – площади, для, для трехмерных элементов – по объему или грани
- *ApiForceTrapezGlobal* – стержня по отрезку на оси, для пластин по – площади, для трехмерных элементов – по объему
- *ApiForceLineEvenlyLocal* – равномерно распределенная по линии, соединяющей два узла элемента;
- *ApiForceLineTrapezGlobal* – трапециевидная по линии, соединяющей два узла элемента;
- *ApiForceLineTrapezGlobal* – трапециевидная по линии, соединяющей два узла элемента;
- *ApiForcePointPartGlobal* – сосредоточенная в точке элемента в долях длины для стержней;
- *ApiForceEvenlyGlobalIns* – равномерно распределенная по стержню с учетом жестких вставок;
- *ApiForceTrapezPartGlobal* – трапециевидная по отрезку стержня, заданного в долях длины;
- собственный вес:
ApiForceWeight – коэффициент к весу;
ApiForceWeightIns – коэффициент к весу с учетом жестких вставок стержня;
ApiFORCE_TEMPERATURE_UNIF = 88,
- массы:
ApiForcePointMass – массы, сосредоточенные по всем направлениям;
ApiForceEvenlyMass – равномерно распределенные массы по всем направлениям;
- температурная:
ApiForceTempLocal – температура;
ApiForceTempUnif – унифицированная температурная нагрузка для плоских элементов.

Для	направления	действия	нагрузок	введены	обозначения:
	<i>SgUnknown</i> ,	<i>SgForceX</i> ,	<i>SgForceY</i> ,	<i>SgForceZ</i>	
	<i>SgForceUX</i> ,	<i>SgForceUY</i> ,	<i>SgForceUZ</i> .		

UINT ApiGetQuantityLoad(ScadAPI lpAPI);

Назначение

Получить число загружений линейной задачи.

Параметры

lpAPI – указатель на объект API.

Возвращаемое значение

Число загружений.

UINT ApiGetQuantityLoadDyn(ScadAPI lpAPI);

Назначение

Получить число динамических загружений линейной задачи.

Параметры

lpAPI – указатель на объект API.

Возвращаемое значение

Число динамических загружений.

APICode ApiSetLoadName(ScadAPI lpAPI, UINT Num, LPCSTR Text);

Назначение

Задание имени нагружения.

Параметры

lpAPI – указатель на объект API;

Num – номер загрузки;

Text – имя загрузки.

Возвращаемое значение

Код возврата.

LPCSTR ApiGetLoadName(ScadAPI lpAPI, UINT Num);

Назначение

Получение имени загрузки.

Параметры

lpAPI – указатель на объект API;

Num – номер загрузки.

Возвращаемое значение

Указатель на строку с именем загрузки.

APICode ApiSetLoadDescription(ScadAPI lpAPI, UINT Num, LPCSTR Text);

Назначение

Задание характеристик загрузки.

Параметры

lpAPI – указатель на объект API;

Num – номер загрузки;

Text – характеристики загрузки. Рекомендуется использовать вывод в текстовый формат данных SCADa в соответствии с "Языком архивации данных". Например:

Text = "Type=0 Mode=1 LongTime=1 ReliabilityFactor=1.05";

// статическое нагружение

Text = " Type=2 ReliabilityFactor=1.1 21 5 1 1 3 0 0 0 5 18 1 0 0.3 1";

// динамический ветер

Возвращаемое значение

Код возврата.

APICode ApiSetLoadMass(ScadAPI lpAPI, UINT Num, UINT Qnt, const double * Coef);

Назначение

Задание преобразования статических нагружений в массы.

Параметры

lpAPI – указатель на объект API;

Num – номер загрузки;

Qnt – число коэффициентов.

Coef – указатель на массив коэффициентов нагружений. Coef[2] коэффициент преобразования нагружения 2;

Возвращаемое значение

Код возврата.

**APICode ApiSetWeight(ScadAPI lpAPI, UINT Num, UINT Qnt,
const UINT * ListElem, double W, BOOL One, BOOL IsInsert);**

Назначение

Задание собственного веса на элементы.

Параметры

lpAPI – указатель на объект API;

Num – номер загрузки;

Qnt – число элементов;

ListElem – указатель на список номеров элементов;

W – коэффициент к нагрузке "Собственный вес;"

One – удалить ранее заданные данные нагрузки на указанные элементы в загрузке (TRUE);

IsInsert – не задавать вес жестких вставок (FALSE).

Возвращаемое значение

Код возврата.

UINT ApiGetQuantityForceNode(ScadAPI lpAPI, UINT Num);

Назначение

Число нагрузок на узлы.

Параметры

lpAPI – указатель на объект API;

Num – номер загрузки.

Возвращаемое значение

Код возврата.

UINT ApiGetQuantityForceElem(ScadAPI lpAPI, UINT Num);

Назначение

Число нагрузок на элементы.

Параметры

lpAPI – указатель на объект API;

Num – номер загрузки.

Возвращаемое значение

Код возврата.

**APICode ApiGetForceNode(ScadAPI lpAPI, UINT Num, UINT NumPP,
BYTE * Qw, BYTE * Qn, UINT * QntData, const double ** Data,
UINT * QntList, const UINT ** List);**

Назначение

Заданные нагрузки на узлы

Параметры

lpAPI – указатель на объект API;

Num – номер загрузки;

NumPP – номер списка;
Qw – вид нагрузки;
Qn – направление нагрузки;
QntData – указатель на число данных, описывающих нагрузку;
Data – указатель на адрес данных, описывающих нагрузку
QntList – указатель на число узлов, на которые задается нагрузка;
List – указатель на адрес списка узлов, на которые задается нагрузка.

Возвращаемое значение

Код возврата.

**APICode ApiGetForceElem(ScadAPI lpAPI, UINT Num, UINT NumPP,
BYTE * Qw, BYTE * Qn, UINT * QntData, const double ** Data,
UINT * QntList, const UINT ** List);**

Назначение

Заданные нагрузки на элементы

Параметры

lpAPI – указатель на объект API;
Num – номер загрузки;
NumPP – номер списка;
Qw – вид нагрузки;
Qn – направление нагрузки;
QntData – указатель на число данных, описывающих нагрузку;
Data – указатель на адрес данных, описывающих нагрузку
QntList – указатель на число элементов, на которые задается нагрузка;
List – указатель на адрес списка элементов, на которые задается нагрузка.

Возвращаемое значение

Код возврата.

**APICode ApiAppendForce(ScadAPI lpAPI, UINT Num, BYTE Qw, BYTE Qn,
UINT QntData, const double * Data, UINT QntList, const UINT * List);**

Назначение

Задание нагрузки на узлы и элементы.

Параметры

lpAPI – указатель на объект API;
Num – номер загрузки;
Qw – вид нагрузки;
Qn – направление нагрузки;
QntData – число данных, описывающих нагрузку;
Data – указатель на данные, описывающих нагрузку
QntList – число узлов/элементов, на которые задается нагрузка;
List – указатель на список номеров узлов/элементов, на которые задается нагрузка.

Возвращаемое значение

Код возврата.

**APICode ApiDeleteForceNode(ScadAPI lpAPI, UINT Num, UINT QntList,
const UINT * ListNode);**

Назначение

Удаление нагрузок на узлы.

Параметры

lpAPI – указатель на объект API;

Num – номер загрузки;

QntList – число узлов;

ListNode – указатель на список номеров узлов.

Возвращаемое значение

Код возврата.

**APICode ApiDeleteForceElem(ScadAPI lpAPI, UINT Num, UINT QntList,
const UINT * ListElem);**

Назначение

Удаление нагрузок на элементы.

Параметры

lpAPI – указатель на объект API;

Num – номер загрузки;

QntList – число элементов;

ListElem – указатель на список номеров элементов.

Возвращаемое значение

Код возврата.

APICode ApiDeleteLoad(ScadAPI lpAPI, UINT Num);

Назначение

Удаление загрузки.

Параметры

lpAPI – указатель на объект API;

Num – номер загрузки.

Возвращаемое значение

Код возврата.

APICode ApiClearLoading(ScadAPI lpAPI, UINT Num);

Назначение

Удаление всех характеристик и нагрузок загрузки.

Параметры

lpAPI – указатель на объект API;

Num – номер загрузки.

Возвращаемое значение

Код возврата.

APICode ApiClearLoad(ScadAPI lpAPI);

Назначение

Удаление всех загрузок.

Параметры

lpAPI – указатель на объект API.

Возвращаемое значение

Код возврата.

2.13. Упругое основание

UINT ApiGetQuantityBed(ScadAPI lpAPI);

Назначение

Получение числа типов упругого основания.

Параметры

lpAPI – указатель на объект API.

Возвращаемое значение

Число типов упругого основания.

BOOL ApiGetBedElem(ScadAPI lpAPI, UINT NumElem, BYTE *Type, UINT *QntSize, double * Size);

Назначение

Получение информации об упругом основании на элементе.

Параметры

lpAPI – указатель на объект API;

NumElem – номер элемента;

Type – тип упругого основания: 'I' – изотропное, 'O' – ортотропное, 'A' – анизотропное;

QntSize – число данных;

Size – данные. Не более 6-ти чисел.

Возвращаемое значение

Код возврата.

TRUE при наличии упругого основания на элементе.

UINT ApiSetBed(ScadAPI lpAPI, BYTE Group, BYTE Type, UINT QntSize, const double * Size, UINT QntElem, const UINT * ListElem);

Назначение

Задание информации об упругом основании.

Параметры

lpAPI – указатель на объект API;

Group – тип элементов: *ApiGroupType* (стержни или пластины);

Type – тип упругого основания: 'I' – изотропное, 'O' – ортотропное, 'A' – анизотропное;

QntSize – число данных;

Size – данные;
QntElem – число элементов;
ListElem – список элементов
Возвращаемое значение
Номер типа упругого основания.

**BOOL ApiGetBed(ScadAPI lpAPI, UINT Num, BYTE *Type, UINT *QntSize,
const double ** Size, UINT * QntElem, const UINT ** ListElem);**

Получение информации об упругом основании.

Параметры

lpAPI – указатель на объект API;
Num – номер списка данных о упругом основании;
Type – адрес типа упругого основания: 'I' – изотропное, 'O' – ортотропное, 'A' – анизотропное;
QntSize – указатель на число данных;
Size – указатель адреса массива данных;
QntElem – указатель на число элементов;
ListElem – указатель адреса списка элементов

Возвращаемое значение

Информация существует (TRUE).

2.14. Комбинации

UINT ApiGetQuantityComb(ScadAPI lpAPI);

Назначение

Получение числа комбинаций загрузжений.

Параметры

lpAPI – указатель на объект API.

Возвращаемое значение

Число комбинаций.

**APICode ApiGetComb(ScadAPI lpAPI, UINT Num, UINT *QntData,
const double **Data);**

Назначение

Чтение данных комбинации загрузжений.

Параметры

lpAPI – указатель на объект API;
Num – номер комбинации;
QntData – число данных
Data – коэффициенты комбинации. *Data*[n-1] – коэффициент n-ого загрузжения.

Возвращаемое значение

Код возврата.

UINT ApiSetComb(ScadAPI lpAPI, UINT QntData, const double *Data);

Назначение

Задание комбинации загружений.

Параметры

lpAPI – указатель на объект API.

QntData – число данных

Data – коэффициенты комбинации. *Data*[n-1] – коэффициент n-ого загружения.

Возвращаемое значение

Номер комбинации загружений.

APICode ApiChangeComb(ScadAPI lpAPI, UINT Num, UINT QntData, const double *Data);

Назначение

Корректировка комбинации загружений.

Параметры

lpAPI – указатель на объект API.

Num – номер комбинации;

QntData – число данных

Data – коэффициенты комбинации. *Data*[n-1] – коэффициент n-ого загружения.

Возвращаемое значение

Код возврата.

APICode ApiDeleteRsn(ScadAPI lpAPI);

Назначение

Удаление комбинаций загружений.

Параметры

lpAPI – указатель на объект API.

Возвращаемое значение

Код возврата.

2.15. Расчетные сочетания усилий

APICode ApiSetRsu(ScadAPI lpAPI);

Назначение

Добавление/удаление строк расчетных сочетаний усилий по умолчанию в соответствии с изменением числа загружений. Инициализация нового документа по данным загружений.

Параметры

lpAPI – указатель на объект API.

Возвращаемое значение

Код возврата.

APICode ApiDeleteRsu(ScadAPI lpAPI);

Назначение

Удаление данных о расчетных сочетаниях усилий.

Параметры

lpAPI – указатель на объект API.

Возвращаемое значение

Код возврата.

APICode ApiSetNoCombRsu(ScadAPI lpAPI, BYTE Yes);

Назначение

Задание учета комбинаций загружений в расчетных сочетаниях усилий.

Параметры

lpAPI – указатель на объект API;

Yes – да (TRUE) или нет.

Возвращаемое значение

Код возврата.

BOOL ApiGetNoCombRsu(ScadAPI lpAPI);

Назначение

Чтение признака учета комбинаций загружений в расчетных сочетаниях усилий.

Параметры

lpAPI – указатель на объект API.

Возвращаемое значение

да (TRUE) или нет (FALSE).

APICode ApiSetRsuStr(ScadAPI lpAPI, UINT NumStr, APIRsuNew * Rsu);

Назначение

Задание характеристик загрузки.

Параметры

lpAPI – указатель на объект API;

NumStr – номер загрузки,

Rsu – указатель на структуру **APIRsuNew** с характеристиками строки.

```
struct APIRsuNew {  
    WORD    TypeLoad;        // тип загрузки  
    WORD    ModeLoad;        // вид загрузки  
    WORD    Sign;            // знакопеременность  
    WORD    Crane;           // номер крана  
    WORD    CraneRegime;     // Группа режимов работы крана, 1-8  
    WORD    NoActive;         // признак активности загрузки  
    double  CoeffSafetyFactor; // коэффициент надежности по нагрузке  
    double  LongTimeLoadComponent; // доля длительной составляющей  
    double  Coeff[16];        // коэффициенты РСУ  
};
```

Возвращаемое значение

Код возврата.

APICode ApiGetRsuStr(ScadAPI lpAPI, UINT NumStr, APIRsuNew * Rsu);

Назначение

Параметры

lpAPI – указатель на объект API;

NumStr – номер загрузки,

Rsu – характеристики строки.

Возвращаемое значение

Код возврата.

**APICode ApiSetListAddRsu(ScadAPI lpAPI, UINT Num, UINT Qnt,
const ApiRSU_ADD * List);**

Назначение

Задание одновременно действующих нагрузок в расчетных сочетаниях усилий.

Параметры

lpAPI – указатель на объект API;

Num – номер загрузки,

Qnt – число одновременно действующих нагрузок с данным;

List – массив структур *ApiRSU_ADD* одновременно действующих нагрузок:

```
struct ApiRSU_ADD {  
    BYTE Type; // признак включения в комбинацию  
    UINT NumNagr; // номер загрузки  
};
```

Все нагрузки из данного списка в случае создания комбинации принимаются при вычислении коэффициентов как одно нагружение. Но нагрузки, у которых Type=0 должны в обязательном порядке присутствовать в комбинации, а в противном случае могут отсутствовать.

Возвращаемое значение

Код возврата.

**APICode ApiSetListExclusionRsu(ScadAPI lpAPI, UINT Num, UINT Qnt,
const UINT * List);**

Назначение

Задание взаимоисключающих нагрузок в расчетных сочетаниях усилий.

Параметры

lpAPI – указатель на объект API;

Num – номер загрузки,

Qnt – число взаимоисключающих нагрузок с данным;

List – список взаимоисключающих нагрузок.

Возвращаемое значение

Код возврата.

**APICode ApiSetListFatherRsu(ScadAPI lpAPI, UINT Num, UINT Qnt,
const UINT * List);**

Назначение

Задание "отцов" для загрузки, без которых оно не может быть включено в комбинацию.

Параметры

lpAPI – указатель на объект API;

Num – номер загрузки,

Qnt – число "отцов" для данного загрузки;

List – список "отцов".

Возвращаемое значение

Код возврата.

UINT ApiGetQuantityUnificationRsu(ScadAPI lpAPI);

Назначение

Получение числа унифицированных групп.

Параметры

lpAPI – указатель на объект API.

Возвращаемое значение

Число унифицированных групп.

**UINT ApiSetUnificationRsu(ScadAPI lpAPI, BYTE Type , WORD Mask , UINT QntElem,
const UINT * ListElem);**

Назначение

Задание группы унификации.

Параметры

lpAPI – указатель на объект API.

Type – тип унификации;

Mask – номер группы унификации;

QntElem – число элементов блока;

ListElem – указатель на список номеров элементов группы;

Возвращаемое значение

Номер группы унификации.

**APICode ApiGetUnificationRsu(ScadAPI lpAPI, UINT NumStr, BYTE *Type, WORD
*Mask, UINT *QntElem, const UINT ** ListElem);**

Назначение

Получение данных о группе унификации.

Параметры

lpAPI – указатель на объект API.

NumStr – номер блока;

Type – указатель на тип унификации;

Mask – указатель на номер группы унификации;
QntElem – указатель на число элементов группы;
ListElem – адрес указателя на список элементов группы;
Возвращаемое значение
 Код возврата.

2.16. Арматура

UINT ApiGetQuantityArmElemRod(ScadAPI lpAPI);

Назначение
 Получение числа групп армирования стержней.
Параметры
lpAPI – указатель на объект API.
Возвращаемое значение
 Число групп армирования стержней.

APICode ApiGetArmElemRod(ScadAPI lpAPI, UINT NumStr, const ApiArmRod ** ArmRod);

Назначение
 Получение данных о группе армирования стержней.
Параметры
lpAPI – указатель на объект API.
NumStr – номер группы;
ArmRod – указатель на адрес структуры **ApiArmRod** с информацией о группе армирования:

```
struct ApiArmRod {
    LPSTR    Text;                // имя группы заданного армирования стержней
    UINT     Quantity;            // число элементов в группе
    UINT     * List;              // указатель на список номеров элементов
    UINT     QuantityArmRod;      // число участков армирования
    ApiArmElemRod * ArmRod;      // указатель на данные участков армирования
};

struct ApiArmElemRod
{
    UINT     PartNo;              // номер участка
    double   L_percent;          // длина участка в процентах от длины
    UINT     IsS1D2;              // TRUE означает, что S1 имеет два различных диаметра
    UINT     IsS2D2;              // TRUE означает, что S2 имеет два различных диаметра
    UINT     IsSw;                // TRUE означает, что есть поперечная арматура
    UINT     IsS34;              // TRUE означает, что есть арматура S3,S4
    UINT     dS1L1_1;            // первый диаметр S1 (в мм)
    UINT     nS1L1_1;            // количество стержней S1
    UINT     dS2L1_1;            // первый диаметр S2 (в мм)
    UINT     nS2L1_1;            // количество стержней S2
    UINT     dS1L1_2;            // второй диаметр S1 (IsS1D2==TRUE) (в мм)
    UINT     nS1L1_2;            // количество стержней S1 второго диаметра (IsS1D2==TRUE)
    UINT     dS2L1_2;            // второй диаметр S2 (IsS2D2==TRUE) (в мм)
    UINT     nS2L1_2;            // количество стержней S2 второго диаметра (IsS2D2==TRUE)
    UINT     dS3L1_1;            // диаметр S3 (в мм)
```

```

    UINT  nS3L1_1; // количество стержней S3
    UINT  dS4L1_1; // диаметр S4 (в мм)
    UINT  nS4L1_1; // количество стержней S4
    UINT  dSw;     // диаметр поперечной арматуры в плоскости Z (в мм)
    UINT  nSw;     // количество стержней (срезов) поперечной арматуры в плоскости Z
    double StepSw; // шаг поперечной арматуры в плоскости Z (в метрах)
    UINT  dSw2;    // диаметр поперечной арматуры в плоскости Y (в мм)
    UINT  nSw2;    // количество стержней (срезов) поперечной арматуры в плоскости Y
    double StepSw2; // шаг поперечной арматуры в плоскости Y (в метрах)
    BYTE  IsS1L2;  // TRUE означает, что S1 имеет два ряда
    BYTE  IsS2L2;  // TRUE означает, что S2 имеет два ряда
    double DeltaS1; // расстояние между рядами S1 (IsS1L2==SCTURE)
    double DeltaS2; // расстояние между рядами S2 (IsS2L2==SCTURE)
    UINT  dS1L2;   // диаметр S1 второго ряда (IsS1L2==SCTURE) (в мм)
    UINT  nS1L2;   // количество стержней S1 второго ряда (IsS1L2==SCTURE)
    UINT  dS2L2;   // диаметр S2 второго ряда (IsS2L2==SCTURE) (в мм)
    UINT  nS2L2;   // количество стержней S2 второго ряда (IsS2L2==SCTURE)
    char  reserved[sizeof(double)*14+sizeof(BYTE)*2+sizeof(UINT)*4];
};

```

**UINT ApiArmRodAppend(ScadAPI lpAPI, LPCSTR Text, UINT QntStr,
const ApiArmElemRod * Data, UINT Qnt, const UINT * Lst);**

Назначение

Задание арматуры на стержнях.

Параметры

lpAPI – указатель на объект API;

Text – имя задаваемой арматуры;

QnStr – число участков заданного армирования;

Data – указатель на массив данных, описывающих арматуру;

Qnt – число элементов, на которых задается арматура;

List – указатель на список номеров элементов, на которых задается арматура.

Возвращаемое значение

Номер группы армирования стержней.

**APICode ApiArmRodReplace(ScadAPI lpAPI, UINT Num,
LPCSTR Text, UINT QntStr, const ApiArmElemRod * Data,
UINT Qnt, const UINT * Lst);**

Назначение

Корректировка группы армирования стержней.

Параметры

lpAPI – указатель на объект API;

Num – номер корректируемой группы заданного армирования стержней;

Text – имя группы задаваемой арматуры;

QnStr – число участков заданной арматуры;

Data – указатель на массив данных, описывающих арматуру;

Qnt – число элементов, на которых задается арматура;

List – указатель на список номеров элементов, на которых задается арматура.

Возвращаемое значение

Код возврата.

UINT ApiGetQuantityArmElemPlate(ScadAPI lpAPI);

Назначение

Получение числа групп заданного армирования пластин.

Параметры

lpAPI – указатель на объект API.

Возвращаемое значение

Число групп заданного армирования пластин.

APICode ApiGetArmElemPlate(ScadAPI lpAPI, UINT NumStr, const ApiArmPlate ** ArmPlate);

Назначение

Получение данных о группе заданного армирования пластин.

Параметры

lpAPI – указатель на объект API.

NumStr – номер группы;

ArmPlate – указатель на адрес структуры **ApiArmPlate** с информацией о группе заданного армирования:

```
struct ApiArmPlate {
    LPSTR    Text;                // имя группы
    UINT      Quantity;           // число элементов в группе
    UINT      * List;             // указатель на список номеров элементов
    ApiArmElemPlate  ArmPlate;    // данные об арматуре
};

struct ApiArmElemPlate
{
    UINT      dS1;                // диаметр продольной арматуры S1 (в мм)
    double     StepS1;            // шаг продольной арматуры S1 (в метрах)
    UINT      dS2;                // диаметр продольной арматуры S2(в мм)
    double     StepS2;            // шаг продольной арматуры S2 (в метрах)
    UINT      dS3;                // диаметр продольной арматуры S3(в мм)
    double     StepS3;            // шаг продольной арматуры S3 (в метрах)
    UINT      dS4;                // диаметр продольной арматуры S4(в мм)
    double     StepS4;            // шаг продольной арматуры S4 (в метрах)
    UINT      dW;                // диаметр поперечной арматуры (в мм)
    double     StepWx;            // шаг поперечной арматуры по оси X (в метрах)
    double     StepWy;            // шаг поперечной арматуры по оси Y (в метрах)
    BOOL       NoUp;              // TRUE означает, что верхней арматуры нет
    BOOL       NoDown;            // TRUE означает, что нижней арматуры нет
    BOOL       NoTrans;           // TRUE означает, что поперечной арматуры нет
    char       reserved[sizeof(double)*16];
};
```

Возвращаемое значение

Код возврата.

```
UINT ApiArmPlateAppend( ScadAPI lpAPI, LPCSTR Text, const ApiArmElemPlate *
    Data, UINT Qnt, const UINT * Lst );
```

Назначение

Задание арматуры на пластинах.

Параметры

lpAPI – указатель на объект API;

Text – имя задаваемой арматуры;

Data – указатель на данные, описывающие арматуру;

Qnt – число элементов, на которых задается арматура;

List – указатель на **на список** номеров **элементов**, для которых задается арматура.

Возвращаемое значение

Номер группы армирования пластин.

```
APICode ApiArmPlateReplace( ScadAPI lpAPI, UINT Num, LPCSTR Text, const
    ApiArmElemPlate * Data, UINT Qnt, const UINT * Lst );
```

Назначение

Корректировка группы армирования пластин.

Параметры

lpAPI – указатель на объект API;

Num – номер корректируемой группы армирования пластин;

Text – имя группы **задаваемой арматуры**;

Data – указатель на данные, описывающие арматуру;

Qnt – число элементов, на которых задается арматура;

List – указатель на **список** номеров **элементов**, для которых задается арматура.

Возвращаемое значение

Код возврата.

2.17. Пример создания проекта

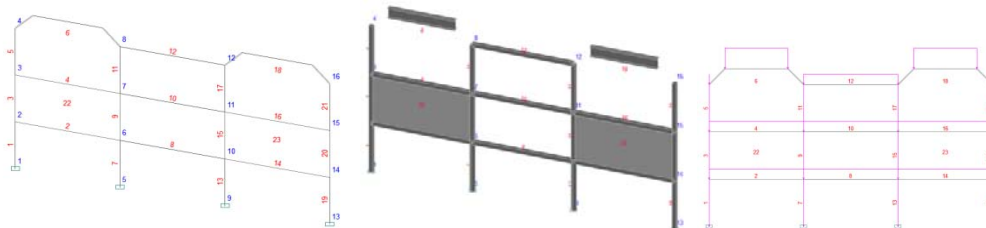


Рис 2-2. Тестовая расчетная схема со схемой загрузки "Собственный вес"

```
void APITestNewProject( )
{
    ScadAPI handle(NULL);
    UINT n, i, j, Node[4], Elem[4];
    double Size[6];
    const static UnitsAPI Un[3] = { { "m", 1 }, { "cm", 100 }, { "T", 1 } };
```

```

if ( ApiCreate(&handle) != APICode_OK ) ApiMsg("Error"); // создание объекта API и контроль
if ( ApiClear(handle) != APICode_OK ) ApiMsg("Error"); // после открытия можно не делать
if ( ApiSetLanguage (handle,1) != APICode_OK ) ApiMsg("Error");
ApiSetName(handle,"TestNewProject");
ApiSetUnits(handle,Un);
if ( ApiSetTypeSchema(handle,5) != APICode_OK ) ApiMsg("Error");

// узлы
ApiNodeAddSize(handle,16);
for ( i=0,n=0; i<4; i++ ) {
    for ( j=0; j<4; j++ ) {
        ApiNodeUpdate(handle,++n,i*6,0,j*3);
    }
}
ApiNodeSetName(handle,16,"Node 16");

// Элементы
ApiElemAdd(handle,21);
for ( i=0; i<4; i++ ) {
    n = i * 6 + 1;
    for ( j=0; j<3; j++ ) {
        Node[0] = 4*i+j+1; Node[1] = Node[0] + 1;
        ApiElemUpdate(handle,n++,5,2,Node);
        Node[0] = Node[1]; Node[1] = Node[0] + 4;
        if ( i < 3 ) ApiElemUpdate(handle,n++,5,2,Node);
    }
}

Node[0] = 2; Node[1] = 6; Node[2] = 3; Node[3] = 7;
n = ApiElemAddData(handle,41,4,Node);
Node[0] = 10; Node[1] = 14; Node[2] = 11; Node[3] = 15;
n = ApiElemAddData(handle,41,4,Node);
ApiElemSetName(handle,n,"Plate");

// жесткости
// пользовательское сечение
n = ApiSetRigid(handle,"S0 3.52e+006 20 25 NU 0.2 RO 2.5 TMP 1e-005 Shift 493.004 61488.2 61548.5");
for ( i=1; i<=21; i++ ) ApiSetRigidElem(handle,n,1,&i);
// металлопрокат
n = ApiSetRigid(handle,"STZ RUSSIAN p_wide_h 18 TMP 1.2e-005");
Elem[0] = 6; Elem[1] = 18;
ApiSetRigidElem(handle,n,2,Elem);
// жесткости пластин
n = ApiSetRigid(handle,"GE 2.1e+007 0.3 0.1 RO 7.85 TMP 1.2e-005 1.2e-005");
ApiSetRigidName(handle,2,"Plate");
Elem[0] = 22; Elem[1] = 23;
ApiSetRigidElem(handle,n,2,Elem);

// связи
Node[0] = 1; Node[1] = 5; Node[2] = 9; Node[3] = 13;
ApiSetBound(handle,SgDirectX | SgDirectZ | SgDirectUY,4,Node,TRUE);

// объединения связей
Node[0] = 4; Node[1] = 16;
n = ApiSetBoundUnite(handle,SgDirectX | SgDirectZ,2,Node);
ApiSetBoundUniteName(handle,n,"Union 1");

// шарниры
ApiSetJoint(handle,SgDirectUX | SgDirectUY,4,2,0);

```



```

ApiSetJoint(handle,SgDirectUX | SgDirectUY,16,1,0);

// жесткие вставки
Elem[0] = 6; Elem[1] = 18;
Size[0] = 1;      Size[1] = 0;      Size[2] = 1;      Size[3] = -1;      Size[4] = 0;      Size[5] = 1;
ApiSetInsert(handle,1,3,6,Size,2,Elem);

// коэффициенты постели
Size[0] = 0.1;      Size[1] = 1000;      Size[2] = 500;
Size[3] = 0.2;      Size[4] = 1500;      Size[5] = 700;
Elem[0] = 1; Elem[1] = 7; Elem[2] = 13; Elem[3] = 19;
ApiSetBed(handle,ApiGroupRod,0,6,Size,4,Elem);

// Группы элементов
Elem[0] = 22; Elem[1] = 23;
ApiSetGroupElem(handle,"Plate",2,2,Elem);
Elem[0] = 1; Elem[1] = 7; Elem[2] = 13; Elem[3] = 19;
ApiSetGroupElem(handle,"Rod",2,4,Elem);
Size[0] = 1200;      Size[1] = 300;
Elem[0] = 23;
ApiSetBed(handle,ApiGroupPlate,"T",6,Size,1,Elem);

// Группы узлов
Node[0] = 1; Node[1] = 5; Node[2] = 9; Node[3] = 13;
ApiSetGroupNode(handle,"Bound",4,Node);

// системы координат элементов
Size[0] = 45;
Elem[0] = 7; Elem[1] = 13;
ApiSetSystemCoordElem(handle,ApiGroupRod,ApiRodCornerInDegrees,1,Size,2,Elem);
Size[0] = 1;      Size[1] = 0;      Size[2] = 1;
Elem[0] = 22; Elem[1] = 23;
ApiSetSystemCoordEffors(handle,ApiGroupPlate,ApiPlateAxcX,2,Size,2,Elem);

// 4-е нагружения
ApiSetLoadDescription(handle,1,"Type=0 Mode=1 LongTime=1 ReliabilityFactor=1.05");
ApiSetLoadName(handle,1,"Узловые нагрузки");
ApiSetLoadDescription(handle,2,"Type=0 Mode=1 LongTime=1 ReliabilityFactor=1.05");
ApiSetLoadName(handle,2,"Распределенные нагрузки");
ApiSetLoadDescription(handle,3,"Type=0 Mode=1 LongTime=1 ReliabilityFactor=1.05");
ApiSetLoadName(handle,3,"Собственный вес");
ApiSetLoadDescription(handle,4,"Type=2 ReliabilityFactor=1.1 21 5 1 1 3 0 0 5 18 1 0 0.3 1");
ApiSetLoadName(handle,4,"Ветер");
// преобразование статических нагружений в массы
ZeroMemory(Size,sizeof(Size));
Size[3] = 1;
ApiSetLoadMass(handle,4,4,Size);

// нагрузки
Size[0] = 1.2;
Node[0] = 8; Node[1] = 12;
ApiAppendForce(handle,1,ApiForceNode,SgForceZ,1,Size,2,Node);
Size[0] = -0.5;
Node[0] = 2; Node[1] = 3; Node[2] = 4;
ApiAppendForce(handle,1,ApiForceNode,SgForceX,1,Size,3,Node);

Size[0] = 2.1;
Node[0] = 6; Node[1] = 12; Node[1] = 18;

```

```

    ApiAppendForce(handle,2,ApiForceEvenlyGlobal,SgForceZ,1,Size,3,Node);

    Size[0] = 2.1;
    Node[0] = 6; Node[1] = 12; Node[1] = 18;
    ApiAppendForce(handle,2,ApiForceEvenlyGlobal,SgForceZ,1,Size,3,Node);
    for ( i=1; i<=23; i++ ) ApiSetWeight(handle,3,1,&i,1.1,TRUE,FALSE);

    if ( ApiWriteProject(handle,"TestNewProject.spr") ) ApiMsg("Error Write");
    ApiRelease(&handle);
}

```

2.18. Пример чтения проекта

```

#include "stdafx.h"

#include "ScadAPIX.hxx"

//void ApiMsg( LPCSTR Text ) {    AfxMessageBox(Text,MB_OK|MB_ICONSTOP|MB_SYSTEMMODAL); }

void APITestReadProject( )
{
    const static UnitsAPI Un[3] = { { "m", 1 }, { "cm", 100 }, { "T", 1 } };
    ScadAPI handle(NULL);
    LPCSTR Text;
    UINT i;

    if ( ApiCreate(&handle) != APICode_OK ) ApiMsg("Error"); // создание объекта API и контроль
    if ( ApiReadProject(handle,NULL) != APICode_OK ) ApiMsg("Error"); // чтение объекта API и контроль

    // могут быть как сообщения об ошибках, так и предупреждения
    for ( i=1; i<=ApiGetQuantityPhrase(handle); i++ ) {
        Text = ApiGetPhrase(handle,i);
        if ( Text ) ApiMsg(Text);
    }

    if ( ApiSetLanguage (handle,1) != APICode_OK ) ApiMsg("Error");
    ApiSetUnits(handle,Un);
    // .....
    Text = ApiGetProjectNameFile(handle);
    if ( ApiWriteProject(handle,Text) ) ApiMsg("Error Write");
    ApiRelease(&handle);
}

```

3. Анализ результатов расчета

3.1. Назначение

Содержит методы для анализа работы результатов работы вычислительного комплекса SCAD.

Деформации можно получить для каждой сформированной вычислительным комплексом правой части. Для динамических нагрузок можно получить как деформации, соответствующие формам колебаний, которые вычислены в соответствии с нормативными документами, так и огибающие.

Число правых частей и информацию о содержимом каждой из них можно получить с помощью методов **GetQuantityString** и **GetInfString**.

Усилия выдаются для каждого конечного элемента сразу для всех правых частей и сечений (точек).

Расчетные сочетания усилий (PCY) будут выданы для всех сечений элемента. Но надо помнить, что выданные усилия могут быть вычислены для другого элемента или сечения при задании унификации.

3.2. Инициализация

Перед инициализацией необходимо прочитать анализируемый проект методом *ApiReadProject*. При этом нельзя устанавливать единицы измерения в нем методом *ApiSetUnits*.

**APICode ApiInitResult(ScadAPI lpAPI,
const UnitsAPI *Units, LPCSTR NameWorkCatalog);**

Назначение

Инициализация объекта для анализа результатов расчета.

Параметры

lpAPI – указатель на объект API;

Units – массив из двух структур UnitsAPI:

```
struct UnitsAPI {  
    char Name[10];  
    float coef;    };
```

Единицы измерения задаются в следующем порядке:

- имя единицы измерения линейных размеров и коэффициент перевода ее в метры. Например для см – см и 100;
- имя единицы измерения сил и коэффициент перевода ее в Т.

Все результаты будут выданы в производных единицах от заданных. Углы всегда будут выдаваться в радианах;

NameWorkCatalog – указатель на полное имя каталога рабочих файлов. Если данный параметр равен NULL, то откроется диалог поиска каталога.

Возвращаемое значение

Код возврата.

BYTE ApiTypeProcess (ScadAPI lpAPI);

Назначение

Проверка типа расчета.

Параметры

lpAPI – указатель на объект API.

Возвращаемое значение

FALSE – линейный расчет, TRUE – нелинейный.

3.2. Наличие результатов расчета

BOOL ApiYesModal (ScadAPI lpAPI);

Назначение

Проверка наличия в результатах расчета задачи динамики.

Параметры

lpAPI – указатель на объект API.

Возвращаемое значение

TRUE – вычислены формы колебаний, FALSE – нет.

BOOL ApiYesDisplace (ScadAPI lpAPI);

Назначение

Проверка наличия в результатах расчета вычисленных перемещений.

Параметры

lpAPI – указатель на объект API.

Возвращаемое значение

TRUE – перемещения вычислены, FALSE – нет.

BOOL ApiYesEffors (ScadAPI lpAPI);

Назначение

Проверка наличия в результатах расчета вычисленных усилий.

Параметры

lpAPI – указатель на объект API.

Возвращаемое значение

TRUE – усилия вычислены, FALSE – нет.

BOOL ApiYesComb (ScadAPI lpAPI);

Назначение

Проверка наличия в результатах расчета вычисленных комбинаций усилий.

Параметры

lpAPI – указатель на объект API.

Возвращаемое значение

TRUE – усилия вычислены, FALSE – нет.

BOOL ApiYesRSU (ScadAPI lpAPI);

Назначение

Проверка наличия в результатах расчета вычисленных РСУ.

Параметры

lpAPI – указатель на объект API.

Возвращаемое значение

TRUE – РСУ вычислены, FALSE – нет.

BOOL ApiYesRSD (ScadAPI lpAPI);

Назначение

Проверка наличия в результатах расчета вычисленных РСП.

Параметры

lpAPI – указатель на объект API.

Возвращаемое значение

TRUE – РСП вычислены, FALSE – нет.

3.3. Информация о типах результатов расчета и чтение их характеристик

При анализе результатов расчета могут быть доступны следующие результаты:

- *API_RESULT_LOAD* – перемещения и соответствующие им напряжения/усилия;
- *API_RESULT_LOAD_COMB* – перемещения и соответствующие им напряжения/усилия от комбинаций нагружений;
- *API_RESULT_MODE* – формы колебаний;
- *API_RESULT_STABIL* – формы потери устойчивости;
- *API_RESULT_STABIL_COMB* – формы потери устойчивости от комбинаций нагружений;
- *API_RESULT_MASS* – сформированные узловые массы для динамического нагружения;
- *API_RESULT_FRAGMENT* – реакции от фрагмента схемы;
- *API_RESULT_FRAGMENT_COMB* – реакции от фрагмента схемы для комбинаций нагружений;
- *API_RESULT_FRAGMENT_LINK* – реакции в связях;
- *API_RESULT_FRAGMENT_LINK_COMB* – реакции в связях от комбинаций нагружений.

Каждое нагружение или шаг сохранения нелинейного процесса может содержать несколько строк результатов: по числу сформированных для него нагрузок, вычисленных собственных векторов или форм потери устойчивости и т.д..

Для получения информации о строке результатов используется структура:

```
struct ApiLoadingData {  
    BYTE    TypeInf;           // тип анализируемых данных: API_RESULT_LOAD и т.д.;  
    BYTE    TypeLoad;          // номер модуля динамики для данных динамического расчета;  
    WORD    TypeEnvelope;      // тип информации, описанный ниже;  
    WORD    NumLoad;           // номер нагружения;  
    WORD    NumSchemUnite;     // номер задачи вариации  
    WORD    NumLoadSchemUnite; // номер нагружения в задаче вариации
```

```

WORD  NumStep;           // шаг для нелинейного процесса
WORD  NumFixedStep;      // номер п/п сохранения в шаговом процессе
UINT  QuantityForm;      // число вычисленных форм колебаний, правых частей,
                        // огибающих, форм потери устойчивости;
UINT  NumForm;           // номер формы колебаний, формы потери
                        // устойчивости, свертки динамических воздействий;
double Value;            // в зависимости от TypeInf – собст. значения, время для
                        // динамики, коэффициент потери устойчивости
float  ProcMassX;         // Для формы колебаний процент масс по направлению X
float  ProcMassY;         // Для формы колебаний процент масс по направлению Y
float  ProcMassZ;         // Для формы колебаний процент масс по направлению Z
LPCTSTR Name;            // указатель на имя загрузки
double * Comb;            // указатель на рассматриваемую комбинацию
                        // загружений линейной задачи: метод ApiGetQuantityLoad
                        // для определения числа загружений. В Comb[i] находится
                        // коэффициент загрузки i+1
};

```

В *ApiLoadingData* тип информации находится в *TypeEnvelope*:

- *ApiDYN_NO* – статика
- *ApiDYN_TIME* – интегрирование по времени;
- *ApiDYN_LS* – статическая составляющая ветровой нагрузки (только для нелинейности и монтажа);
- *ApiDYN_LN_LS* – нелинейность + статическая составляющая ветровой нагрузки;
- *ApiDYN_LB_LS* – базовое нагружение монтажа + статическая составляющая ветровой нагрузки;
- *ApiDYN_MASS* – массы;
- *ApiDYN_FORM* – формы колебаний;
- *ApiDYN_STAB* – формы потери устойчивости
- *ApiDYN_RD* – действительная часть гармонического воздействия;
- *ApiDYN_RI* – комплексная часть гармонического воздействия;
- *ApiDYN_FORMX* – направление X при шестикомпонентном воздействии
- *ApiDYN_FORMY* – направление X при шестикомпонентном воздействии
- *ApiDYN_FORMZ* – направление X при шестикомпонентном воздействии
- *ApiDYN_FORMUX* – направление X при шестикомпонентном воздействии
- *ApiDYN_FORMUY* – направление X при шестикомпонентном воздействии
- *ApiDYN_FORMUZ* – направление X при шестикомпонентном воздействии
- *ApiDYN_SD* – динамическая огибающая (сумма квадратов или по нормам) результатов, полученных по формам колебаний;
- *ApiDYN_SI* – динамическая огибающая от действительной и комплексной части гармонического воздействия
- *ApiDYN_LS_SD* – статическая + динамическая огибающая.
- *ApiDYN_LN_SD* – нелинейность + динамическая огибающая (251)
- *ApiDYN_LS_SI* – статическая + динамическая огибающая гармонического воздействия;
- *ApiDYN_LN_SI* – нелинейность+ динамическая огибающая гармонического воздействия;
- *ApiDYN_LB_SD* – базовое нагружение монтажа + динамическая огибающая
- *ApiDYN_LB_SI* – базовое нагружение монтажа + динамическая огибающая гармонического воздействия;
- *ApiDYN_LN_TIME* – нелинейность+интегрирование по времени
- *ApiDYN_LB_TIME* – базовое нагружение монтажа + интегрирование по времени

UINT *ApiGetResultQuantityLoad* (*ScadAPI* *IpAPI*);

Назначение

Число загружений.

Параметры

lpAPI – указатель на объект API.

Возвращаемое значение

число загрузок.

UINT ApiGetResultQuantityFixedStep (ScadAPI lpAPI);

Назначение

Число точек сохранения результатов в нелинейном процессе.

Параметры

lpAPI – указатель на объект API.

Возвращаемое значение

число точек сохранения результатов в нелинейном процессе.

UINT ApiGetResultQuantityLoadDynamic (ScadAPI lpAPI);

Назначение

Число динамических загрузок.

Параметры

lpAPI – указатель на объект API.

Возвращаемое значение

число динамических загрузок.

UINT ApiGetQuantityLoadStr (ScadAPI lpAPI, API_RESULT_DATA Type, UINT NumLoad);

Назначение

Число строк результатов для загрузки/шага сохранения нелинейного процесса.

Параметры

lpAPI – указатель на объект API.

Type – тип анализируемых данных: API_RESULT_LOAD и т.д.;

NumLoad – номер загрузки для линейного процесса и номер сохраненных результатов для нелинейного.

Возвращаемое значение

число строк результатов для загрузки/шага сохранения нелинейного процесса.

APICode ApiGetResultData (ScadAPI lpAPI, API_RESULT_DATA Type, UINT NumLoad, UINT NumStr, ApiLoadingData * APD);

Назначение

Информация о сформированной строке результатов указанного загрузки/шага нелинейного процесса.

Параметры

lpAPI – указатель на объект API.

Type – тип анализируемых данных: API_RESULT_LOAD и т.д.;

NumLoad – номер загрузки для линейного процесса и номер сохраненных результатов для нелинейного;

NumStr – номер строки результатов загрузки;

APD – указатель на структуру *ApiLoadingData*.

Возвращаемое значение

Код возврата.

3.4. Динамика

double * ApiGetMassa(ScadAPI lpAPI, UINT NumLoad, UINT NumNode);

Назначение

Приведенные узловые массы.

Параметры

lpAPI – указатель на объект API.

NumLoad – номер загрузки для линейного процесса и номер сохраненных результатов для нелинейного;

NumNode – номер узла.

Возвращаемое значение

Указатель на приведенные узловые массы в *i*-й позиции которого находится значение массы по направлению степени *i* + 1;

Если ошибки в данных, то указатель – NULL.

double * ApiGetMode(ScadAPI lpAPI, UINT NumLoad, UINT NumMode, UINT NumNode);

Назначение

Приведенные узловые массы.

Параметры

lpAPI – указатель на объект API.

NumLoad – номер загрузки для линейного процесса и номер сохраненных результатов для нелинейного;

NumMode – номер формы колебаний;

NumNode – номер узла.

Возвращаемое значение

Указатель на приведенные узловые массы в *i*-й позиции которого находится значение массы по направлению степени *i* + 1;

Если ошибки в данных, то указатель – NULL.

3.5. Премещения

double * ApiGetDisplace(ScadAPI lpAPI, UINT NumLoad, UINT NumStr, UINT NumNode);

Назначение

Перемещения узла.

Параметры

lpAPI – указатель на объект API.

NumLoad – номер загрузки для линейного процесса и номер сохраненных результатов для нелинейного;

NumStr – номер читаемой строки;

NumNode – номер узла.

Возвращаемое значение

Указатель на перемещения узла: в *i*-й позиции которого находится значение перемещения по направлению степени *i* +1.

Если ошибки в данных, то указатель – NULL.

double * ApiGetCombDisplace(ScadAPI lpAPI, UINT NumComb, UINT NumNode);

Назначение

Перемещения узла для комбинации нагрузок.

Параметры

lpAPI – указатель на объект API.

NumLoad – номер загрузки для линейного процесса и номер сохраненных результатов для нелинейного;

NumNode – номер узла.

Возвращаемое значение

Указатель на перемещения узла: в *i*-й позиции которого находится значение перемещения по направлению степени *i* +1.

Если ошибки в данных, то указатель – NULL.

3.6. Реакции в связях и от фрагмента схемы

double * ApiGetReak(ScadAPI lpAPI, UINT NumLoad, UINT NumStr, UINT NumNode);

Назначение

Реакции в связях в узле.

Параметры

lpAPI – указатель на объект API.

NumLoad – номер загрузки для линейного процесса и номер сохраненных результатов для нелинейного;

NumStr – номер читаемой строки;

NumNode – номер узла.

Возвращаемое значение

Указатель на реакции в связях в узле: в *i*-й позиции которого находится значение реакции по направлению степени *i* +1.

Если ошибки в данных, то указатель – NULL.

double * ApiGetCombReak(ScadAPI lpAPI, UINT NumComb, UINT NumNode);

Назначение

Реакции в связях в узле от комбинаций нагрузок.

Параметры

lpAPI – указатель на объект API.

NumLoad – номер загрузки для линейного процесса и номер сохраненных результатов для нелинейного;

NumStr – номер читаемой строки;

NumNode – номер узла.

Возвращаемое значение

Указатель на реакции в связях в узле от комбинаций загрузок: в *i*-й позиции которого находится значение реакции по направлению степени *i* + 1.

Если ошибки в данных, то указатель – NULL.

double * ApiGetReakFragment(ScadAPI lpAPI, UINT NumLoad, UINT NumStr, UINT NumNode);

Назначение

Реакции от фрагмента схемы в узле.

Параметры

lpAPI – указатель на объект API.

NumLoad – номер загрузки для линейного процесса и номер сохраненных результатов для нелинейного;

NumStr – номер читаемой строки;

NumNode – номер узла.

Возвращаемое значение

Указатель на реакции от фрагмента схемы в узле: в *i*-й позиции которого находится значение реакции по направлению степени *i* + 1.

Если ошибки в данных, то указатель – NULL.

double * ApiGetCombReakFragment(ScadAPI lpAPI, UINT NumComb, UINT NumNode);

Назначение

Реакции от фрагмента схемы в узле от комбинаций загрузок.

Параметры

lpAPI – указатель на объект API.

NumLoad – номер загрузки для линейного процесса и номер сохраненных результатов для нелинейного;

NumStr – номер читаемой строки;

NumNode – номер узла.

Возвращаемое значение

Указатель на реакции от фрагмента схемы в узле от комбинаций загрузок: в *i*-й позиции которого находится значение реакции по направлению степени *i* + 1.

Если ошибки в данных, то указатель – NULL.

3.7. Усилия и напряжения

Для чтения вычисленных усилий/напряжений используется структура:

```
struct ApiElemEffors {  
    UINT    NumElem;        // номер элемента  
    BYTE    QuantityUs;     // число усилий в точке  
    const BYTE * TypeUs;     // типы усилий  
    BYTE    QuantityPoint;   // число точек выдачи усилий  
    WORD    QuantityPointLayers; // число слоев выдачи усилий в точке (многослойные)  
    WORD    QuantityLoading; // число строк типа API_RESULT_LOAD  
    WORD    QuantityComb;    // число комбинаций  
    BYTE    IsComb;          // наличие комбинаций  
    size_t    MaxSizeUs;      //  
    size_t    QuantityDataUs; // размерность массива усилий  
    double * Us;              // указатель на усилия  
    size_t    MaxSizeUsComb; //  
    size_t    QuantityDataUsComb; // размерность массива усилий от комбинаций  
    double * UsComb;         // указатель на усилия от комбинаций загружений  
};
```

Размерности массивов Us и UsComb равны соответственно:

QuantityDataUs = QuantityPoint * QuantityLoading * QuantityPointLayers * QuantityUs;

QuantityDataUsComb = QuantityPoint * QuantityComb * QuantityPointLayers * QuantityUs.

При этом сначала располагаются усилия/напряжения в нижней точке слоя, затем последующих. И так для далее.

APICode ApiGetEffors(ScadAPI lpAPI, **UINT NumElem,
 ApiElemEffors ** Effors, **BYTE** TypeRead);**

Назначение

Информация о усилиях/напряжениях в элементе.

Параметры

lpAPI – указатель на объект API.

NumElem – номер элемента;

Effors – указатель на структуру ApiElemEffors.

TypeRead – тип чтения: 0 – читать все, 1 – не читать комбинации, 2 - читать только комбинации.

Возвращаемое значение

Код возврата.

3.8. РСУ и РСД

Для чтения вычисленных усилий/напряжений используются структуры:

```
struct ApiElemRsu {  
    UINT    NumElem;        // номер КЭ или группы унификации РСУ  
    BYTE    TypeUnif;       // тип унификации РСУ  
    WORD    GroupUnif;      // группа унификации РСУ  
    BYTE    TypeConstr;     // тип конструкции из группы в РСУ  
    BYTE    QuantityPoint;  // число точек  
    BYTE    QuantityUs;     // число усилий в точке
```

```

    UINT      LengthData; // число строк PCY/PCП
    UINT      Quantity;   //число комбинаций
    ApiElemRsuStr * Str;
};

struct ApiElemRsuStr {
    UINT      NumElem;      // номер элемента при унификации
    BYTE      NumPoint;     // номер точки
    BYTE      NumPointElem; // номер сечения элемента
    BYTE      NumColumn;    // номер столбца, по которым найдено
    BYTE      GroupRsu;     // 0 - расчетные, 1 - длительные
                          // расчетные, 2 - нормативная,
                          // 3 - длительная часть нормативной
    WORD      NumCrit;      // номер критерия
    WORD      QuantityCoef; // число коэффициентов в комбинации
    BYTE      YesSeism;     // признак наличия сейсмической нагрузки
    BYTE      YesSpec;      // признак наличия специальной
                          // не сейсмической нагрузки
    BYTE      YesCrane;     // признак наличия кранов
    BYTE      YesTransport; // наличие транспортных грузов
    BYTE      Res[4];
    float * Us;             // строка усилий/напряжений/перемещений
    WORD * NumLoad;         // указатель на номера загружений в комбинации
    float * Coef;           // указатель на коэффициенты загружений в
                          // комбинации
    float Value;           // значение критерия
};

```

APICode ApiGetRsu(ScadAPI lpAPI, UINT NumElem, ApiElemRsu * rsu);

Назначение

Информация о PCY в элементе.

Параметры

lpAPI – указатель на объект API.

NumElem – номер элемента;

rsu – указатель на структуру ApiElemRsu.

Возвращаемое значение

Код возврата.

APICode ApiGetRsd(ScadAPI lpAPI, UINT NumElem, ApiElemRsu * rsu);

Назначение

Информация о PCY в элементе.

Параметры

lpAPI – указатель на объект API.


NumElem – номер элемента;

rsu – указатель на структуру ApiElemRsu.

Возвращаемое значение

Код возврата.

3.9. Арматура

API не предоставляет доступ к результатам подбора арматуры; это, в частности, связано с тем, что элемент может одновременно принадлежать нескольким группам армирования (основной и дополнительным). Обойти данное ограничение можно преобразовав результаты подбора в заданное армирование (операция ) и используя методы, описанные в разделе 2.16.