SCAD Structure

Библиотека SCAD++ API

2021

ОГЛАВЛЕНИЕ

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 этом 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\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 в директории "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
libimalloc.dll
cilkrts20.dll
libchkp.dll
libiomp5md.dll
libiompstubs5md.dll
libirngmd.dll
libmmd.dll

libmmdd.dll
svml_dispmd.dll
mkl_avx.1.dll
mkl_avx2.1.dll
mkl_avx512.1.dll
mkl_blacs_ilp64.1.dll
mkl_blacs_intelmpi_ilp64.1.dll
mkl_blacs_intelmpi_lp64.1.dll
mkl_blacs_lp64.1.dll
mkl_blacs_mpich2_ilp64.1.dll
mkl_blacs_mpich2_lp64.1.dll
mkl_blacs_msmpi_ilp64.1.dll
mkl_blacs_msmpi_lp64.1.dll
mkl_cdft_core.1.dll
mkl_core.1.dll
mkl_def.1.dll
mkl_intel_thread.1.dll
mkl_mc.1.dll
mkl_mc3.1.dll
mkl_pgi_thread.1.dll
mkl_rt.1.dll
mkl_scalapack_ilp64.1.dll
mkl_scalapack_lp64.1.dll
mkl_sequential.1.dll
mkl_sycl.1.dll
mkl_sycld.1.dll
mkl_tbb_thread.1.dll
mkl_tbb_threadd.1.dll
mkl_vml_avx.1.dll
mkl_vml_avx2.1.dll
mkl_vml_avx512.1.dll
mkl_vml_cmpt.1.dll
mkl_vml_def.1.dll
mkl_vml_mc.1.dll
mkl_vml_mc2.1.dll
mkl_vml_mc3.1.dll

При использовании жесткостных характеристик из сортамента металлопроката нужны также файлы с расширение *.prf.

SCAD API реализован в виде библиотеки с интерфейсами языка C. Поставляемые библиотеки созданы с помощью компилятора Microsoft Visual C++ 2019. Поэтому для работы библиотек на компьютере должен быть установлен Redistributable Package Microsoft Visual C++ 2019 для платформы x64, который автоматически устанавливается при инсталляции SCAD Office (или может быть загружен с https://support.microsoft.com/en-us/topic/the-latest-supported-visual-c-downloads-2647da03-1eea-4433-9aff-95f26a218cc0). При разработке приложений с использованием SCAD API можно использовать любую среду программирования.

2. Создание и корректировка расчетных схем

2.1. Обработка ошибок

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

APICode_OK – успешное завершение;
APICode_InvalidHandle – неверен указатель на объект API;
APICode_InternalError – программная ошибка;
APICode_FatalError – программная ошибка;
APICode_IndexError – 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 ApiSetUnits(ScadAPICH lpAPI, const UnitsAPI *Un);*

Назначение[1]

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).

[1] Необходимо помнить, что работа со всеми величинами в проекте (длины, нагрузки, жесткости и т.п.) будет выполняться в заданных единицах измерения. Если данная функция не используется, то необходимо проверить заданные при работе с проектом «Единицы измерения для протокола расчета и экспорта».

APICode ApiReadProject(ScadAPICH handle, LPCSTR Name);[2]

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.

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 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.

---

[2] После чтения проекта необходимо проверить и, при необходимости, переопределить систему единиц работы с ним.

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.

APICode ApiSetCompany(ScadAPI lpAPI, LPCSTR NameCompany);

Назначение

Задание имени компании.

Параметры

lpAPI – указатель на объект API;
NameCompany – имя компании.

Возвращаемое значение

Код возврата.

LPCSTR ApiGetCompany(ScadAPI lpAPI);

Назначение

Чтение имени компании.

Параметры

lpAPI – указатель на объект API.

Возвращаемое значение

Указатель на строку с именем компании.

APICode ApiSetCustomer(ScadAPI lpAPI, LPCSTR NameCustomer);

Назначение

Задание имени заказчика.

Параметры

lpAPI – указатель на объект API;
NameCustomer – имя заказчика.

Возвращаемое значение

Код возврата.

LPCSTR ApiGetCustomer(ScadAPI lpAPI);

Назначение

Чтение имени заказчика.

Параметры

lpAPI – указатель на объект API.

Возвращаемое значение

Указатель на строку с именем заказчика.

APICode ApiSetObject(ScadAPI lpAPI, LPCSTR NameObject);

Назначение

Задание наименования объекта.

Параметры

lpAPI – указатель на объект API;
NameObject – наименование объекта.

Возвращаемое значение

Код возврата.

LPCSTR ApiGetObject (ScadAPI lpAPI);

Назначение

Чтение наименования объекта.

Параметры

lpAPI – указатель на объект API.

Возвращаемое значение

Указатель на строку с наименованием объекта.

APICode ApiSetExecutor(ScadAPI lpAPI, LPCSTR NameExecutor);

Назначение

Задание имени исполнителя.

Параметры

lpAPI – указатель на объект API;
NameExecutor – имя исполнителя.

Возвращаемое значение

Код возврата.

LPCSTR ApiGetExecutor(ScadAPI lpAPI);

Назначение

Чтение имени исполнителя.

Параметры

lpAPI – указатель на объект API.

Возвращаемое значение

Указатель на строку с именем исполнителя.

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. Координатные оси

UINT ApiGetQuantityAxesGroup(ScadAPI lpAPI);

Назначение

Чтение числа групп координатных линий.

Параметры

lpAPI – указатель на объект API.

Возвращаемое значение

число групп координатных линий.

APICode DeleteAxesGroup(ScadAPI lpAPI, UINT Num);

Назначение

Удаление группы координатных осей.

Параметры

lpAPI – указатель на объект API;
Num – номер группы координатных осей.

Возвращаемое значение

Код возврата.

*const ApiCAxesGroup * ApiGetAxesGroup (ScadAPI lpAPI, UINT Num);*

Назначение

Чтение группы координатных осей.

Параметры

lpAPI – указатель на объект API;
Num – номер группы координатных осей.

Возвращаемое значение

Указатель указатель на структуру *ApiAxesGroup*.

```
struct ApiSAxis
{
public:
       char       Name[16]; // Имя оси
       double Pos;    // Привязка
};
```

```
struct ApiAxesGroup
{
        char      *  Name;
        UINT         QuantityX;
        ApiSAxis *   X;
        UINT         QuantityY;
        ApiSAxis *   Y;
        UINT         QuantityZ;
        ApiSAxis *   Z;
        double       Point[3];    //  координаты точки привязки группы осей
        double       Data[2];     //  Дополнительные данные. Углы поворота и т.д.
        BYTE         Type;        //  тип системы координат
        BYTE         InActive;    //
        BYTE         InVisibleFoo;    //  НЕ ИСПОЛЬЗУЕТСЯ !
        BYTE         TypeLine;        //  тип
        BYTE         ThicknessLine;   //  толщина
        BYTE         BeginLineX;      //  начало линии x
        BYTE         EndLineX;        //  конец линии x
        BYTE         BeginLineY;      //  начало линии y
        BYTE         EndLineY;        //  конец линии y
        BYTE         StartZ;          //  номер группы осей для привязеки отметок уровня
        BYTE         Res[2];          //  резерв
};
```

APICode ApiAppendAxesGroup(ScadAPI lpAPI, LPApiCAxesGroup LPAG);

Назначение

Задание группы координатных осей.

Параметры

lpAPI – указатель на объект API;
LPAG – Указатель указатель на структуру *ApiAxesGroup*.

Возвращаемое значение

2.4.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 добавленого 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 добавленого 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 добавленого 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, BYTE 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

востановление 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.

2.5.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 добавленого 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 об 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 добавленого 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 ApiDeleteElem(ScadAPI lpAPI, UINT Num);

Назначение

Удаление элемента.

Параметры

lpAPI – указатель на объект API;
Num – номер элемента.

Возвращаемое значение

Код возврата.

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

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 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

Состановление 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.

2.6.Жесткости

UINT ApiGetQuantityRigid(ScadAPI lpAPI);

Назначение

Получение числа типов жесткостей.

Параметры

lpAPI – указатель на объект API.

Возвращаемое значение

Число типов жесткостей.

*APICode ApiGetRigid(ScadAPI lpAPI, UINT NumRgd, LPSTR Text, UINT MaxLenText, UINT * Qnt, const UINT ** ListElem);*

Назначение

Получение жесткостных характеристик.

Параметры

lpAPI – указатель на объект API;
NumRgd – номер жесткостных характеристик;
Text – текст с жесткостными характеристиками;
MaxLenText – размер массива *Text;*
Qnt – указатель на число элементов с данным типом жесткости;
ListElem – указатель на адрес списка элементов группы.

Возвращаемое значение

Код возврата.

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 жестоскости 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 характеристикю

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.

2.7. 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.

2.8. 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.

2.9. Связи

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

SgDirectX	*SgDirectAX*	*SgDirectCX*
SgDirectY	*SgDirectAY*	*SgDirectCY*
SgDirectZ	*SgDirectAZ*	*SgDirectCZ*
SgDirectUX	*SgDirectBX*	
SgDirectUY	*SgDirectBY*	
SgDirectUZ	*SgDirectBZ*	

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.

2.10. 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.

2.11. Системы координат элементов

Используются следующие обозначения для идентификации типов систем координат:

1. стержни (главные оси):

ApiRodCornerInDegrees – угол в градусах.
ApiRodCornerInRadians – угол в радианах.
ApiRodPointFocusAxecY – координаты точки. Принимается за ось Y_1 проекция на ортогональную оси X_1 плоскость вектора, проведенного из первого узла элемента в заданную точку.
ApiRodVectorFocusAxecY – направление вектора, проекция которого на ортогональную оси X_1 плоскость принимается за ось Y_1.
ApiRodPointFocusAxecZ – координаты точки. Принимается за ось Z_1 проекция на ортогональную оси X_1 плоскость вектора, проведенного из первого узла элемента в заданную точку.
ApiRodVectorFocusAxecZ – направление вектора, проекция которого на ортогональную оси X_1 плоскость принимается за ось Z_1.

2. стержни (конструктивные оси):

ApiRodPointFocusGeomAxecY – координаты точки. Принимается за конструктивную ось Y_2 проекция на ортогональную оси X_1 плоскость вектора, проведенного из первого узла элемента в заданную точку.
ApiRodVectorFocusGeomAxecY – направление вектора, проекция которого на ортогональную оси X_1 плоскость принимается за конструктивную ось Y_2.
ApiRodPointFocusGeomAxecZ – координаты точки. Принимается за конструктивную ось Z_2 проекция на ортогональную оси X_1 плоскость вектора, проведенного из первого узла элемента в заданную точку.

ApiRodVectorFocusGeomAxecZ – направление вектора, проекция которого на ортогональную оси X_1 плоскость принимается за конструктивную ось Z_2.

3. пластины:

ApiPlateAxecX – направление вектора в глобальной системе координат, проекция которого на плоскость элемента принимается за ось X_1.

ApiPlatePointFocusAxecX – координаты точки. Принимается за ось X_1 проекция на плоскость элемента вектора, проведенного из первого узла элемента в заданную точку.

ApiPlatePointFocusCentersAxecX – координаты точки. Принимается за ось X_1 проекция на плоскость элемента вектора, проведенного из центра элемента в заданную точку.

ApiPlateAxecY – направление вектора в глобальной системе координат, проекция которого на плоскость элемента принимается за ось Y_1.

ApiPlatePointFocusAxecY – координаты точки. Принимается за ось Y_1 проекция на плоскость элемента вектора, проведенного из первого узла элемента в заданную точку.

ApiPlatePointFocusCentersAxecY – координаты точки. Принимается за ось X_1 проекция на плоскость элемента вектора, проведенного из центра элемента в заданную точку.

4. объемные элементы:

ApiVolumeAxecXY – направления осей X_1, Y_1. Если заданы неортогональные векторы, то направление оси Y_1 корректируется в заданной плоскости.

ApiVolumeAxecXZ – направления осей X_1, Z_1. Если заданы неортогональные векторы, то направление оси Z_1 корректируется в заданной плоскости.

ApiVolumeAxecYZ – направления осей Y_1, Z_1. Если заданы неортогональные векторы, то направление оси Z_1 корректируется в заданной плоскости.

ApiVolumeCornersInDegrees – углы Эйлера в градусах

ApiVolumeCornersInRadians – углы Эйлера в радианах

ApiVolumeCylinderTwoPointsZ – цилиндр. Задаются две точки на оси Z_1.

ApiVolumeCylinderPointsAndVectorZ – цилиндр. Задаются координаты точки на оси цилиндра и направление оси Z_1.

ApiVolumeSphereCentreAndPointY – сфера. Задаются координаты точки центра сферы и точка, на которая позволяет ориентировать ось Y_1 в плоскости X_1Y_1.

ApiVolumeSphereCentreAndVectorY – сфера. Задаются координаты точки центра сферы и направление которая позволяет ориентировать ось Y_1 в плоскости X_1Y_1.

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.

2.12. Шарниры

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

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);

Value –адрес указателя на 6-ть чисел с упругими характеристиками шарниров по соответствующим направлениям.

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, double *Value);*

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).
Value –указатель на 6-ть чисел с упругими характеристиками шарниров по соответствующим направлениям. Может быть NULL.

Возвращаемое значение

Код возврата.

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

Назначение

Удаление шарниров в узлах элементов.

Параметры

lpAPI – указатель на объект API;
NumElem – номер элемента;
NumNodeElem – порядковый номер узла (нумерация с 1).

Возвращаемое значение

Код возврата.

*APICode ApiGetRodJoint(ScadAPI lpAPI, UINT NumElem, WORD *JointElastic, WORD * JointInsert , double * DataJointElastic, double * DataJointInsert);*

Назначение

Информация о шарнирах в узлах стержневых элементов.

Параметры

lpAPI – 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.

DataJointElastic, Data JointInsert – указатели на массивы (каждый из которых имеет размер 12), в которые будут переданы данные о жесткостях соответствующих шарниров).

Возвращаемое значение

Код возврата.

2.13. Нагрузки

2.13.1. Идентификация видов нагрузки и их направления их действия

Используются следующие обозначения для идентификации видов нагрузки:

5. узловая:

ApiForceNode – узловая.

6. заданные перемещения:

ApiForceNodeDisplace – заданные перемещения узла;
ApiForceNodeSpecial – заданные перемещения узла в специальной системе координат;
ApiForceNullElem – задание перемещения узла через нуль-элемент;

7. в местной системе координат элемента:

ApiForcePointLocal – сосредоточенная в точке элемента;
ApiForceEvenlytLocal – равномернораспределенная для стерженя по оси, для пластин по – площади, для, для трехмерных элементов – по объему или грани
ApiForceTrapezLocal – стерженя 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;

8. в общей системе координат элемента:
9. *ApiForcePointGlobal* – сосредоточенная в точке элемента;
10. *ApiForceEvenlytGlobal* – равномерно распределенная для стерженя по оси, для пластин по – площади, для, для трехмерных элементов – по объему или грани
11. *ApiForceTrapezGlobal* – стерженя по отрезку на оси, для пластин по – площади, для трехмерных элементов – по объему
12. *ApiForceLineEvenlyLocal* – равномернораспределенная по линии, соединяющей два узла элемента;
13. *ApiForceLineTrapezGlobal* – трапециевидная по линии, соединяющей два узла элемента;
14. *ApiForceLineTrapezGlobal* – трапециевидная по линии, соединяющей два узла элемента;

15. *ApiForcePointPartGlobal* – сосредоточенная в точке элемента в долях длины для стержней;
16. *ApiForceEvenlyGlobalIns* – равномерно распределенная по стержню с учетом жестких вставок;
17. *ApiForceTrapezPartGlobal* – трапециевидная по отрезку стержня, заданного в долях длины;
18. собственный вес:

ApiForceWeight – коэффициент к весу;
ApiForceWeightIns – коэффициент к весу с учетом жестких вставок стержня;

ApiFORCE_TEMPERATURE_UNIF = 88,

19. массы:

ApiForcePointMass – массы, сосредоточенные по всем направлениям;

ApiForceEvenlyMass – равномерно распределенные массы по всем направлениям;

20. температурная:

ApiForceTempLocal – температура;
ApiForceTempUnif – унифицированная температурная нагрузка для плоских элементов.

21. нагрузки на области:

ApiForceLineEvenlyAreaGlobal – равномерно распределенные нагрузки по линии и полилинии;
ApiForceLineTrapezAreaGlobal – трапециевидные нагрузки по линии;
ApiForceEvenlyAreaGlobal – равномерно распределенные нагрузки по произвольному многоугольнику, заданному контуром;
ApiForceTrapezAreaGlobal – трапециевидные нагрузки по треугольнику и четырехугольнику.

Для направления действия нагрузок введены обозначения:
SgUnknown, *SgForceX*, *SgForceY*, *SgForceZ*, *SgForceUX*, *SgForceUY*, *SgForceUZ*.

2.13.2. Загружения

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 загружения.

Параметры

lpAPI – указатель на объект API;
Num – номер загружения;

Text – имя загружения.

Возвращаемое значение

Код возврата.

LPCSTR ApiGetLoadName(ScadAPI lpAPI, UINT Num);

Назначение

Получение имени загружения.

Параметры

lpAPI – указатель на объект API;
Num – номер загружения.

Возвращаемое значение

Указатель на строку с именем загружения.

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

Назначение

Задание характеристик загружения.

Параметры

lpAPI – указатель на объект API;
Num – номер загружения;
Text – характеристики загружения. 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 загружение
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 загружений в массы.

Параметры

lpAPI – указатель на объект API;
Num – номер загружения;
Qnt – число коэффициентов.
Coef – указатель на массив коэффициентов загружений. Coef[2] коэффициент преобразования загружения 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 загружения;
Qnt – число элементов;
ListElem – указатель на список номеров элементов;
W –коэффицент к нагрузке "Собственный вес;"
One – удалить ранее заданные данные нагрузки на указанные элементы в загружении (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 загружения.

Возвращаемое значение

Число нагрузок на узлы.

UINT ApiGetQuantityForceElem(ScadAPI lpAPI, UINT Num);

Назначение

Число нагрузок на элементы.

Параметры

lpAPI – указатель на объект API;
Num – номер загружения.

Возвращаемое значение

Число нагрузок на элементы.

UINT ApiGetQuantityForceArea(ScadAPI lpAPI, UINT Num);

Назначение

Число нагрузок на области.

Параметры

lpAPI – указатель на объект API;
Num – номер загружения.

Возвращаемое значение

Число нагрузок на области.

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

Назначение

Заданные нагрузки на области нагрузок.

Параметры

lpAPI – указатель на объект API;
Num – номер загружения;
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 линии/контура области, на которую задается нагрузка;
List – указатель на адрес списка узлов линии/контура области, на которую 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 загружения;
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 загружения;
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 загружения;
QntList – число элементов;

ListElem – указатель на список номеров элементов.

Возвращаемое значение

Код возврата.

APICode ApiDeleteForceArea(ScadAPI lpAPI, UINT Num, UINT NumForce);

Назначение

Удаление нагрузки на область.

Параметры

lpAPI – указатель на объект API;
Num – номер загружения;
NumForce – номер нагрузки.

Возвращаемое значение

Код возврата.

APICode ApiDeleteLoad(ScadAPI lpAPI, UINT Num);

Назначение

Удаление загружения.

Параметры

lpAPI – указатель на объект API;
Num – номер загружения.

Возвращаемое значение

Код возврата.

APICode ApiClearLoading(ScadAPI lpAPI, UINT Num);

Назначение

Удаление всех характеристик и нагрузок загружения.

Параметры

lpAPI – указатель на объект API;
Num – номер загружения.

Возвращаемое значение

Код возврата.

APICode ApiClearLoad(ScadAPI lpAPI);

Назначение

Удаление всех загружений.

Параметры

lpAPI – указатель на объект API.

Возвращаемое значение

Код возврата.

2.14. Группы нагрузок

UINT ApiGetQuantityLoadGroup(ScadAPI lpAPI);

Назначение

Получить число групп нагрузок.

Параметры

lpAPI – указатель на объект API.

Возвращаемое значение

Число групп нагрузок.

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

Назначение

Задание имени группы нагрузок.

Параметры

lpAPI – указатель на объект API;
Num – номер группы нагрузок;
Text – имя группы нагрузок.

Возвращаемое значение

Код возврата.

LPCSTR ApiGetLoadNameGroup(ScadAPI lpAPI, UINT Num);

Назначение

Получение имени группы нагрузок.

Параметры

lpAPI – указатель на объект API;
Num – номер группы нагрузок.

Возвращаемое значение

Указатель на строку с именем группы нагрузок.

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

Назначение

Задание собственного веса на элементы в группу нагрузок.

Параметры

lpAPI – указатель на объект API;
Num – номер группы нагрузок;
Qnt – число элементов;
ListElem – указатель на список номеров элементов;
W –коэффицент к нагрузке "Собственный вес;"
One – удалить ранее заданные данные нагрузки на указанные элементы в загружении (TRUE);
IsInsert – не задавать вес жестких вставок (FALSE).

Возвращаемое значение

Код возврата.

UINT ApiGetQuantityForceNodeLoadGroup(ScadAPI lpAPI, UINT Num);

Назначение

Число нагрузок на узлы в группе нагрузок.

Параметры

lpAPI – указатель на объект API;
Num – номер группы нагрузок.

Возвращаемое значение

Число нагрузок на узлы в группе нагрузок.

UINT ApiGetQuantityForceElemLoadGroup(ScadAPI lpAPI, UINT Num);

Назначение

Число нагрузок на элементы группы нагрузок.

Параметры

lpAPI – указатель на объект API;
Num – номер группы нагрузок.

Возвращаемое значение

Число нагрузок на элементы группы нагрузок.

UINT ApiGetQuantityForceAreaLoadGroup (ScadAPI lpAPI, UINT Num);

Назначение

Число нагрузок на области в группе нагрузок.

Параметры

lpAPI – указатель на объект API;
Num – номер группы нагрузок.

Возвращаемое значение

Число нагрузок на области в группе нагрузок.

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

Назначение

Заданые нагрузки на узлы группы загрузок.

Параметры

lpAPI – указатель на объект API;
Num – номер группы нагрузок;
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 ApiGetForceElemLoadGroup(ScadAPI lpAPI, UINT Num, UINT NumPP, BYTE * Qw, BYTE * Qn, UINT * QntData, const double ** Data, UINT * QntList, const UINT ** List);*

Назначение

Заданые нагрузки на элементы группы загрузок.

Параметры

lpAPI – указатель на объект API;
Num – номер группы нагрузок;
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 ApiGetForceAreaLoadGroup (ScadAPI lpAPI, UINT Num, UINT NumPP, BYTE * Qw, BYTE * Qn, UINT * QntData, const double ** Data, UINT * QntList, const UINT ** List);*

Назначение

Заданные нагрузки на области в группе нагрузок.

Параметры

lpAPI – указатель на объект API;
Num – номер загружения;
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 линии/контура области, на которую задается нагрузка;
List – указатель на адрес списка узлов линии/контура области, на которую задается нагрузка.

Возвращаемое значение

Код возврата.

*APICode ApiAppendForceLoadGroup(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 группы нагрузок.

Параметры

lpAPI – указатель на объект API;
Num – номер группы нагрузок;
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 ApiDeleteForceNodeLoadGroup(ScadAPI lpAPI, UINT Num, UINT QntList, const UINT * ListNode);*

Назначение

Удаление нагрузок на узлы группы нагрузок.

Параметры

lpAPI – указатель на объект API;
Num – номер группы нагрузок;
QntList – число узлов;
ListNode – указатель на список номеров узлов.

Возвращаемое значение

Код возврата.

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

Назначение

Удаление нагрузок на элементы группы нагрузок.

Параметры

lpAPI – указатель на объект API;
Num – номер группы нагрузок;
QntList – число элементов;
ListElem – указатель на список номеров элементов.

Возвращаемое значение

Код возврата.

APICode ApiDeleteForceAreaLoadGroup(ScadAPI lpAPI, UINT Num, UINT NumForce);

Назначение

Удаление нагрузки на область в группе нагрузок.

Параметры

lpAPI – указатель на объект API;
Num – номер загружения;
NumForce – номер нагрузки.

Возвращаемое значение

Код возврата.

APICode ApiDeleteLoadGroup(ScadAPI lpAPI, UINT Num);

Назначение

Удаление группы нагрузок.

Параметры

lpAPI – указатель на объект API;
Num – номер группы нагрузок.

Возвращаемое значение

Код возврата.

APICode ApiClearLoadingGroup(ScadAPI lpAPI, UINT Num);

Назначение

Удаление всех характеристик и нагрузок группы нагрузок.

Параметры

lpAPI – указатель на объект API;
Num – номер группы нагрузок.

Возвращаемое значение

Код возврата.

APICode ApiClearLoadGroup(ScadAPI lpAPI);

Назначение

Удаление всех групп нагрузок.

Параметры

lpAPI – указатель на объект API.

Возвращаемое значение

Код возврата.

2.15. 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).

2.16. Комбинации

UINT ApiGetQuantityComb(ScadAPI lpAPI);

Назначение

Получение числа комбинаций загружений.

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 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.

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

Назначение

Задание комбинации загружений.

Параметры

lpAPI – указатель на объект API.
QntData – число данных
Data – коэффициенты комбинаии. 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 комбинаии. 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.

2.17. 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 сочетаний 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 загружения.

Параметры

lpAPI – указатель на объект API;
NumStr – номер загружения,
Rsu – указатель на структуру ***APIRsuNew*** с характеристиками строки.

```
struct APIRsuNew {
    WORD     TypeLoad;          // тип загружения
    WORD     ModeLoad;          // вид загружения
    WORD     Sign;              // знакопеременность
    WORD     Crane;             // номер крана
    WORD     CraneRegime;       // Группа режимов работы крана, 1-8
    WORD     NoActive;          // признак активности загружения
    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 загружения,
Rsu – характеристики строки.

Возвращаемое значение

Код возврата.

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

Назначение

Задание одновременно действующих загружений в расчетных сочетаний усилий.

Параметры

lpAPI – указатель на объект API;
Num – номер загружения,
Qnt – число одновременно действующих загружений с данным;
List – массив структур *ApiRSU_ADD* одновременно действующих загружений:

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

Все загружения из данного списка в случае создания комбинации принимаются при вычислении коэффициентов как одно загружение. Но загружения, 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 загружений в расчетных сочетаний усилий.

Параметры

lpAPI – указатель на объект API;
Num – номер загружения,
Qnt – число взаимоисключающих загружений с данным;
List – список взаимоисключающих загружений.

Возвращаемое значение

Код возврата.

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

Назначение

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

Параметры

lpAPI – указатель на объект API;
Num – номер загружения,
Qnt – число "отцов" для данного загружения;
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 * ListE-lem);

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.

2.18. Арматура

2.18.1. Группы армирования

UINT ApiGetQuantityConcrete(ScadAPI lpAPI);

Назначение

Получение числа групп армирования.

Параметры

lpAPI – указатель на объект API.

Возвращаемое значение

Число групп армирования.

APICode ApiGetConcrete(ScadAPI lpAPI, UINT Num, const ApiConcreteElem ** Concrete, UINT *Qnt, const UINT **ListElem);

Назначение

Получение данных о группе армирования.

Параметры

lpAPI – указатель на объект API.
Num – номер группы;
Concrete – указатель на адрес структуры ***ApiConcreteElem;***
QntElem – указатель на число элементов;
ListElem – указатель на список элементов

Возвращаемое значение

Код возврата.

Структура ***ApiConcreteElem*** содержит информацию о группе:

```
struct  ApiConcreteElem
{    //  группа
        BYTE   Modul;          // номер модуля:     103 - оболочки, 104 - балки-стенки, 105 - пластины,
                               //         107 - изгибаемые стержни, 108 - сжато-изогнутые (растянутые) стержни
        BYTE   Type;           // 1 - статически определимая конструкция, 0 - статически неопределимая конструкция
        BYTE   CrackResisting;  // анализировать (1) и нет (0) трещиностойкость
        BYTE   MinArmatur;      // данные по минимальной арматуре. 0 - нет, 1 -   есть
        BYTE   YesExpert;       //  резерв
        BYTE   YesLengthOfFactor;  //  информация о расчетных длинах, 1 - задаются расчетные длины, 0 - задаются
коэффициенты расчетных длин
        BYTE   NbCalc; // используется только при расчетах по СП 52-101-03. Если это поле равно 1, то для
внецентренно сжатых элементов при расчете значения продольной силы Nb по п.3.52 Пособия к СП 52-101-2003
учитывается площадь сечения арматуры

        BYTE   OldCode;        // резерв

        double Range[4];       // расстояния до центра тяжести арматур (Range[0]=a1, Range[1]=a2 для стержней и пластин;
Range[2]= a3, Range[3]= a4 для пластин). Задаются в см.
        double EffectiveLength[2];  //  расчетные длины в плоскостях X1OZ1, X1OY1 (при YesLengthOfFactor=1)
        double FactorEffectiveLength[2];  //  коэффициенты расчетных длин в плоскостях X1OZ1, X1OY1 (при
YesLengthOfFactor=0)
        double Displacement[2]; //  случайные эксцентриситеты по осям Z1, Y1. Задаются в см.
        double SeismFactor[2];  //  коэффициенты учета сейсмического воздействия (нормальных сечений, наклонных
сечений)

        double m_GammaN;       // коэффициент надежности по ответственности (первое предельное состояние)
        double m_GammaN2;      // коэффициент надежности по ответственности (второе предельное состояние)
//  бетон
        BYTE   TypeBeton;      //  вид бетона. 0 - тяжелый бетон, 1 – мелкозернистый А, 2 - мелкозернистый Б, 2 -
мелкозернистый В, 4 - легкий
        BYTE   ConditionsHardening; // условия твердения (1 – Естественное, 2 - В пропарочных камерах, 3 - Автоклавная
обработка)
        BYTE   Filler;         // заполнитель легкого бетона
        BYTE   Stiffener;      // 1 - ребро плиты (используется только для стержневых конструктивных элементов)
        char   ClassBeton[16];  //  класс бетона
        char   SortBeton[16];   //  марка легкого бетона по средней плотности
        double FactorHardening; // коэффициент условий твердения
        double FactorForce;     // коэффициент учета нагрузок длительного действия γb2 (при расчетах по СНиП), γb1 (при
расчетах по СП)
        double FactorTotal;     // результирующий коэффициент условий работы бетона (при расчетах по СНиП)
        double ResD2[2]; // резерв

//    арматура
        char   ClassArm[2][16]; // классы продольной (ClassArm[0]) и поперечной (ClassArm[1]) арматур
        double FactorWork[2];   // коэффициенты условий работы продольной и поперечной арматуры
```

double MaxDiam; // максимальный диаметр углового стержня в миллиметрах (при подборе)
double MaxProcent; // максимальный процент армирования (при подборе)
double MaxKolUg; // резерв
double ResD3[1]; // резерв

// трещиностойкость
BYTE Category; // категория трещиностойкости (1 - отсутствие трещин или 3 - ограниченная ширина раскрытия трещин)
BYTE ConditionsOperation; // условия эксплуатации
BYTE RegimeBeton; // режим влажности бетона (1 - Естественная влажность, 2 - Водонасыщение и высушивание, 3 - Водонасыщенное состояние)
BYTE Dampness; // влажность воздуха окружающей среды (1 - 40-75%, 2 - менее 40%, 3 - более 75%)
BYTE YesSeicmRSU; // 1 - учитывать РСУ с сейсмикой при анализе ширины раскрытия трещин
BYTE Stress; // напряженное состояние: одноосное - 0, косой изгиб - 1
BYTE IsUserArm; // 1 - учитывать заданное армирование при подбора арматуры в пластинах
BYTE Res3; // резерв
double DiamRod[2]; // при подборе по трещиностойкости — диаметры стержней продольной и поперечной арматур (в миллиметрах)
double WidthCrack[2]; // максимально допустимая ширина непродолжительного и продолжительного раскрытия трещин (в миллиметрах)
double Interval; // резерв
BYTE bFibModel; // резерв
BYTE IsContrElem; // признак конструктивного элемента
BYTE IsMinArmPercent; // учитывать минимальный процент армирования при подборе
BYTE CmMode; // резерв - всегда задавать значение 0
BYTE ArbatVersion; // всегда задавать значение 3
BYTE Tr2003; // При расчетах по СП: требования к ширине раскрытия трещин выбираются: из условия сохранности арматуры - 0, из условия ограничения проницаемости конструкций - 1
BYTE SlaveGroup; // 1 - дополнительная группа
BYTE Res4[1]; // резерв
double Gb_Damage; // коэффициент учета характера разрушения γ_{b2} (при расчетах по СП)
double Gb_VertPos; // коэффициент учета вертикального положения при бетонировании γ_{b3} (при расчетах по СП)
double Gb_Freezing; // коэффициент учета замораживания/оттаивания и отрицательных температур γ_{b5} (при расчетах по СП 63.13330.2012) или γ_{b4} (при расчетах по СП 52-101-03)

BYTE DisplacementCheck[10]; // признаки использования ограничения по прогибам и перемещениям
float DisplacementLimit_L[10]; // ограничения по прогибам и перемещениям в долях длины элемента
BYTE Use_YesLengthOfFactor_New; // всегда 1 (использовать новую схему задания расчетных длин)
BYTE reserved2[5]; // резерв
float DisplacementLimit_Abs[10]; // абсолютные ограничения по прогибам и перемещениям
double m_GammaN_A; // коэффициент надежности по ответственности (аварийное состояние)
double m_SpecialConcreteCoef;// коэфициент условий работы бетона при особых (не сейсмических) воздействиях
double m_SpecialArmCoef; // коэфициент условий работы арматуры при особых (не сейсмических) воздействиях
double m_SpecialLargeSpanCoef; // Коэффициент понижающий расчетное сопротивление
BYTE DeflectSNIPCheck[6]; // признак использования ограничения по прогибу (см. ниже)
float DeflectSNIPLimit_L[6]; // ограничение по прогибу относительно длины элемента (см. ниже)
float DeflectSNIPLimit_Abs[6]; // абсолютное ограничение по прогибу (см. ниже)
BYTE DisplacementSNIPCheckEx[2]; // признак использования ограничения по прогибу (см. ниже)
float DisplacementExSNIPLimit_L[2]; // ограничение по перемещениям относительно длины элемента (см. ниже)
float DisplacementExSNIPLimit_Abs[2]; // абсолютное ограничение по перемещениям (см. ниже)
BYTE DeflectECCheck[6]; // признак использования ограничения по прогибу (EC)
float DeflectECLimit_L[6]; // ограничение по прогибу относительно длины элемента (Eurocode) (см. ниже)
float DeflectECLimit_Abs[6]; // абсолютное ограничение по прогибу (Eurocode) (см. ниже)
BYTE check_ApplyECMinAcross; // Устанавливать минимальную поперечную арматуру (см. п. 6.2.1(4) EN 1992-1-1)
BYTE check_Rod_IgnoreECTorsion; // Не учитывать кручение (см. п. 6.3.1(2) EN 1992-1-1)
BYTE check_Plate_106; //Использовать формулу (8.106) СП 63.13330

BYTE check_Plate_LimitCrackDistance; //Ограничивать расстояние между трещинами величиной min(40 см, 40*диаметр арматуры)
BYTE check_Plate_SkipAcrossArm; //Не учитывать поперечную арматуру при малой интенсивности поперечного армирования
BYTE check_Rod_IncreaseAlongArm; //Увеличивать продольную арматуру при реализации п. 8.1.34 СП 63.13330
float reserved_dbl32_1;

double Seismic_Compressed_Zone; // Коэффициент снижения граничной относительной высоты сжатой зоны (для пластинчатых элеметов)
double EC_Gc_adjust; // коэффициент понижающий/повышающий коэффициент условий работы бетона (приложение А)
int EC_AgeDays; // возраст бетона в днях
int EC_Cement; // тип цемента
int EC_CreepAgeDays; //возраст (дни)
double EC_Temp_during_AgeDays; // дельта температур (EN 1992-1-1 B.10)
int EC_Temp_QntDays; // количество суток, когда температура EC_Temp_during_AgeDays преобладает (EN 1992-1-1 B.10)
double EC_TrVlagBetPercent; // относительная влажность %

BYTE check_SP_Phi_n_LowerBoundTension; // Учитывать ограничение по нижней границе коэффициента φ_n при растяжении
float SP_Phi_n_LowerBoundTension; // Нижняя граница коэффициента φ_n при растяжении
BYTE check_SP_Phi_n_LowerBoundCompression; // Учитывать ограничение по нижней границе коэффициента φ_n при сжатии
float SP_Phi_n_LowerBoundCompression; // Нижняя граница коэффициента φ_n при сжатии
BYTE check_SP_Phi_n_UpperBoundCompression; // Учитывать ограничение по верхней границе коэффициента φ_n при сжатии
float SP_Phi_n_UpperBoundCompression; // Верхняя граница коэффициента φ_n при сжатии
BYTE check_SP_Phi_n_TensionIgnore_8_55; // резерв
BYTE check_EC_Increase_Seismic_Shear_Combinations; // Увеличивать расчетные поперечные силы для сейсмических комбинаций п. 5.4.2.4 (7) EN 1998
BYTE check_EC_Normalized_Axial_Seismic_Load; // Для сейсмических комбинаций вычислять фактор по нормализованной осевой силе п. 5.4.3.4.1(2) EN 1998
BYTE check_SP_Slenderness; // Учитывать ограничение по предельной гибкости
float SP_Slenderness; // Ограничение по предельной гибкости
BYTE ResD4_byte;
double ResD4[7];

};

Поле *ClassBeton* зависит от выбранных норм проектирования. Допустимы следующие значения:

Нормы	
СНиП	"B7,5", "B10", "B12,5", "B15", "B20", "B25", "B30", "B35", "B40", "B45", "B50", "B55", "B60"
СП 52-101-03	"B10", "B15", "B20", "B25", "B30", "B35", "B40", "B45", "B50", "B55", "B60"
СП 63.13330	"B10", "B12,5", "B15", "B20", "B25", "B30", "B35", "B40", "B45", "B50", "B55", "B60", "B70", "B80", "B90", "B100"
Следует использовать символ "B" латинского алфавита.	

При расчетах по СП поле *ConditionsHardening* следует задавать равным 1. При расчетах по СНиП: 1 - Естественное, 2 - В пропарочных камерах, 3 - Автоклавная обработка.

При расчетах по СНиП поле *Dampness* следует устанавливать равным 1. При расчетах по СП: 1 - 40-75%, 2 - менее 40%, 3 - более 75%.

При расчетах по СП поле *RegimBeton* следует устанавливать равным 1. При расчетах по СНиП: 1- Естественная влажность, 2- Водонасыщение и высушивание, 3- Водонасыщенное состояние.

При расчетах по СП поле *ConditionsOperation* следует устанавливать равным 1. При расчетах по СНиП: 1 - В помещении, 2 - На открытом воздухе или в грунте, 3 - Грунт. Переменный уровень вод.

Поле *ClassArm* зависит от выбранных норм проектирования. В зависимости от класса арматуры программа допускает использовать соответствующие наборы диаметров арматуры (см. таблицу ниже).

Нормы	Классы арматуры	Допустимые диаметры арматуры
СНиП	"A-I"	6, 8, 10, 12, 14, 16, 18, 20, 22, 25, 28, 32, 36, 40
	"A-II"	10, 12, 14, 16, 18, 20, 22, 25, 28, 32, 36, 40
	"A-III"	6, 8, 10, 12, 14, 16, 18, 20, 22, 25, 28, 32, 36, 40
	"A-IV"	10, 12, 14, 16, 18, 20, 22
	"A-V"	10, 12, 14, 16, 18, 20, 22, 25, 28, 32
	"A-VI"	10, 12, 14, 16, 18, 20, 22
	"Bp-I"	3, 4, 5
	"A400C"	10, 12, 14, 16, 18, 20, 22, 25, 28, 32, 36, 40
	"A500C"	3, 4, 5, 6, 8, 10, 12, 14, 16, 18, 20, 22, 25, 28, 32, 36, 40
СП 52-101-03	"A240"	6, 7, 8, 9, 10, 12, 14, 16, 18, 20, 22, 25, 28, 32, 36, 40
	"A300"	10, 12, 14, 16, 18, 20, 22, 25, 28, 32, 36, 40, 45, 50, 55, 60, 70, 80
	"A400"	6, 7, 8, 9, 10, 12, 14, 16, 18, 20, 22, 25, 28, 32, 36, 40
	"A500"	10, 12, 14, 16, 18, 20, 22, 25, 28, 32, 36, 40
	"B500"	3, 4, 5, 6, 7, 8, 10, 12
СП 63.13330	"A240"	6, 7, 8, 9, 10, 12, 14, 16, 18, 20, 22, 25, 28, 32, 36, 40
	"A400"	6, 7, 8, 9, 10, 12, 14, 16, 18, 20, 22, 25, 28, 32, 36, 40
	"A500"	10, 12, 14, 16, 18, 20, 22, 25, 28, 32, 36, 40
	"A600"	10, 12, 14, 16, 18, 20, 22, 25, 28, 32, 36, 40
	"A600C"	10, 12, 14, 16, 18, 20, 22, 25, 28, 32, 36, 40
	"B500"	3, 4, 5, 6, 8, 9, 10, 12, 14
	"Bp500"	3, 4, 5
Следует использовать символы "A","B","C" и "p" латинского алфавита.		

Для учета ограничений по прогибам и перемещениям следует задавать следующие данные

	Признак учета ограничений	Относительное ограничение (в долях длины)	Абсолютное ограничение
При расчетах по Eurocode			
Горизонтальный прогиб от характеристических комбинаций	DeflectECCheck[0]	DeflectECLimit_L[0]	DeflectECLimit_Abs[0]
Горизонтальный прогиб от квазипостоянных комбинаций	DeflectECCheck[1]	DeflectECLimit_L[1]	DeflectECLimit_Abs[1]
Горизонтальный прогиб от частых комбинаций	DeflectECCheck[2]	DeflectECLimit_L[2]	DeflectECLimit_Abs[2]
Вертикальный прогиб от характеристических комбинаций	DeflectECCheck[3]	DeflectECLimit_L[3]	DeflectECLimit_Abs[3]
Вертикальный прогиб от квазипостоянных комбинаций	DeflectECCheck[4]	DeflectECLimit_L[4]	DeflectECLimit_Abs[4]
Вертикальный прогиб от частых комбинаций	DeflectECCheck[5]	DeflectECLimit_L[5]	DeflectECLimit_Abs[5]
Горизонтальные перемещения от характеристических комбинаций	DisplacementCheck[0]	DisplacementLimit_L[0]	DisplacementLimit_Abs[0]
Горизонтальные перемещения от квазипостоянных комбинаций	DisplacementCheck[1]	DisplacementLimit_L[1]	DisplacementLimit_Abs[1]
Горизонтальные перемещения от частых комбинаций	DisplacementCheck[2]	DisplacementLimit_L[2]	DisplacementLimit_Abs[2]
Вертикальные перемещения от характеристических комбинаций	DisplacementCheck[3]	DisplacementLimit_L[3]	DisplacementLimit_Abs[3]
Вертикальные перемещения от квазипостоянных комбинаций	DisplacementCheck[4]	DisplacementLimit_L[4]	DisplacementLimit_Abs[4]
Вертикальные перемещения от частых комбинаций	DisplacementCheck[5]	DisplacementLimit_L[5]	DisplacementLimit_Abs[5]
При расчетах по СниП, СП, ДБН			
Горизонтальный прогиб от всех	DeflectSNIPCheck[0]	DeflectSNIPLimit_L[0]	DeflectSNIPLimit_Abs[0]

нагрузок			
Горизонтальный прогиб от постоянных и длительных нагрузок	DeflectSNIPCheck[4]	DeflectSNIPLimit_L[4]	DeflectSNIPLimit_Abs[4]
Горизонтальный прогиб от временных нагрузок	DeflectSNIPCheck[1]	DeflectSNIPLimit_L[1]	DeflectSNIPLimit_Abs[1]
Вертикальный прогиб от всех нагрузок	DeflectSNIPCheck[2]	DeflectSNIPLimit_L[2]	DeflectSNIPLimit_Abs[2]
Вертикальный прогиб от постоянных и длительных нагрузок	DeflectSNIPCheck[5]	DeflectSNIPLimit_L[5]	DeflectSNIPLimit_Abs[5]
Вертикальный прогиб от временных нагрузок	DeflectSNIPCheck[3]	DeflectSNIPLimit_L[3]	DeflectSNIPLimit_Abs[3]
Вертикальные перемещения от всех нагрузок	DisplacementCheck[8]	DisplacementLimit_L[8]	DisplacementLimit_Abs[8]
Вертикальные перемещения от постоянных и длительных нагрузок	DisplacementCheck[7]	DisplacementLimit_L[7]	DisplacementLimit_Abs[7]
Вертикальные перемещения от временных нагрузок	DisplacementCheck[9]	DisplacementLimit_L[9]	DisplacementLimit_Abs[9]
Горизонтальные перемещения от всех нагрузок	DisplacementCheck[6]	DisplacementLimit_L[6]	DisplacementLimit_Abs[6]
Горизонтальные перемещения от постоянных и длительных нагрузок	DisplacementSNIPCheckEx[0]	DisplacementExSNIPLimit_L[0]	DisplacementExSNIPLimit_Abs[0]
Горизонтальные перемещения от временных нагрузок	DisplacementSNIPCheckEx[1]	DisplacementExSNIPLimit_L[1]	DisplacementExSNIPLimit_Abs[1]

*UINT ApiSetConcrete(ScadAPI lpAPI, LPCSTR Text, const ApiConcreteElem * Concrete, UINT QntElem, const UINT * ListElem);*

Назначение

Задание группы армирования.

Параметры

lpAPI – указатель на объект API;
Text – имя группы;
Concrete – указатель на адрес структуры ***ApiConcreteElem;***
QntElem – число элементов;
ListElem – список элементов.

Возвращаемое значение

Номер группы армирования стержней.

*APICode ApiChangeConcrete(ScadAPI lpAPI, UINT Num, LPCSTR Text, const ApiConcreteElem * Concrete, UINT QntElem, const UINT * ListElem);*

Назначение

Корректировка группы армирования.

Параметры

lpAPI – указатель на объект API;
Num – номер группы;
Text – имя группы;
Concrete – указатель на адрес структуры ***ApiConcreteElem;***
QntElem – число элементов. Если *QntElem*=0, то список элементов не корректируется;
ListElem – список элементов.

Возвращаемое значение

Код возврата.

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

Назначение

Задание имени группы армирования.

Параметры

lpAPI – указатель на объект API;
Name – имя группы.

Возвращаемое значение

Код возврата.

LPCSTR ApiGetNameConcrete(ScadAPI lpAPI, UINT Num);

Назначение

Чтение имени группы армирования.

Параметры

lpAPI – указатель на объект API.

Возвращаемое значение

Указатель на строку с именем группы армирования.

APICode ApiDeleteConcrete(ScadAPI lpAPI, UINT Num);

Назначение

Удаление группы армирования.

Параметры

lpAPI – указатель на объект API;
Num – номер группы армирования;

Возвращаемое значение

Код возврата.

APICode ApiClearConcrete(ScadAPI lpAPI);

Назначение

Удаление всех групп армирования.

Параметры

lpAPI – указатель на объект API;

Возвращаемое значение

Код возврата.

2.18.2. Заданное армирование стержней

UINT ApiGetQuantityArmElemRod(ScadAPI lpAPI);

Назначение

Получение числа групп заданного армирования стержней.

Параметры

lpAPI – указатель на объект API.

Возвращаемое значение

Число групп заданного армирования стержней.

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

Назначение

Получение данных о группе заданного армирования стержней.

Параметры

lpAPI – указатель на объект API.

NumStr – номер группы;

ArmRod – указатель на адрес структуры ***ApiArmRod*** с информацией о группе заданного 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==SCTRUE)
        double   DeltaS2; // расстояние между рядами S2 (IsS2L2==SCTRUE)
        UINT     dS1L2;   // диаметр S1 второго ряда (IsS1L2==SCTRUE) (в мм)
        UINT     nS1L2;   // количество стержней S1 второго ряда (IsS1L2==SCTRUE)
        UINT     dS2L2;   // диаметр S2 второго ряда (IsS2L2==SCTRUE) (в мм)
        UINT     nS2L2;   // количество стержней S2 второго ряда (IsS2L2==SCTRUE)
```

```
        char    reserved[sizeof(double)*14-sizeof(BYTE)*2-sizeof(UINT)*4];
};
```

Возвращаемое значение

Код возврата.

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

Назначение

Задание арматуры для стержней.

Параметры

lpAPI – указатель на объект API;
Text – имя задаваемой группы;
QntStr – 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 заданного армирования стержней.

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

Назначение

Корректировка группы заданного армирования стержней.

Параметры

lpAPI – указатель на объект API;
Num – номер корректируемой группы заданного армирования стержней;
Text – имя группы задаваемой арматуры;
QntStr – 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.

2.18.3. Заданное армирование пластин

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 * List);*

Назначение

Задание арматуры в 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 список номеров элементов, для которых задается арматура.

Возвращаемое значение

Номер группы заданного армирования пластин.

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

Назначение

Корректировка группы заданного армирования пластин.

Параметры

lpAPI – указатель на объект API;
Num – номер корректируемой группы заданного армирования пластин;
Text – имя группы;
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.

2.19. Конструктивные стальные элементы

UINT ApiGetQuantitySteel(ScadAPI lpAPI);

Назначение

Получение числа групп конструктивных стальных элементов.

Параметры

lpAPI – указатель на объект API.

Возвращаемое значение

Число групп конструктивных стальных элементов.

*APICode ApiGetSteel(ScadAPI lpAPI, UINT Num, const ApiSteelElem ** Steel, UINT *Qnt, const UINT **ListElem);*

Назначение

Получение данных о группе конструктивных стальных элементов.

Параметры

lpAPI – указатель на объект API.
Num – номер группы;
Steel – указатель на адрес структуры ***ApiSteelElem;***
QntElem – указатель на число элементов;
ListElem – указатель на список элементов

Возвращаемое значение

Код возврата.

Структура ***ApiSteelElem*** содержит информацию о группе:

```
struct ApiSteelElem
{
        const static UINT MAX_STRING_STEEL_NAME = 80;

        char   SteelMark[MAX_STRING_STEEL_NAME];        // марка стали
        BYTE   What_is;       // 0 - конструктивный элемент 1 - группа элементов
        WORD  ContructionType;  // тип конструкции: 0 - элемент общего вида, 1 - стойка, 2 - балка,
```

// 3 - элемент пояса фермы, 4 - элемент решетки фермы, 5 - опорный раскос фермы, 6 - опорная стойка фермы
WORD IndexSchema; // номер схемы вариации от 0 (not used)
BYTE bSnip; // TRUE - коэффициенты расчетной длины взять по СНиП
BYTE bRatio_deprecated; // не используется
BYTE bNoPlastic; // сечение работает только упруго

double Ry; // расчетное сопротивление Ry, если не задано марка стали
double m_GammaN; // коэффициент надежности по ответственности (первое предельное состояние)
double Koef_usl_rab; // коэффициент условий работы
double Res1; // резерв

double Koef_RasLen_XoZ; // коэффициент расчетной длины в плоскости XoZ
double Koef_RasLen_YoZ; // коэффициент расчетной длины в плоскости YoZ
double Lim_gibkA; // предельная гибкость при сжатия
double Lim_gibkB; // предельная гибкость при растяжения
double StepOutPlane_linear; // шаг раскрепления из плоскости (линейные размеры)

double Lim_gibkA_Angle; // предельная гибкость для сжатия
double Lim_gibkB_Angle; // предельная гибкость для растяжения

double CalcLength_X0Z; // расчетная длина в плоскости XoZ (линейные размеры)
double CalcLength_Y0Z; // расчетная длина плоскости YoZ (линейные размеры)

double m_GammaN2; // коэффициент надежности по ответственности (второе предельное состояние)- пока для стали не нужно
double m_GammaN_A; // коэффициент надежности по ответственности (аварийное состояние)
BYTE DisplacementCheck[10]; // признак использования ограничения по перемещениям
float DisplacementLimit_L[10]; // ограничение по перемещениям относительно длины элемента
BYTE IsCorrosion; // наличие коррозии
BYTE HasStiff; // наличие ребер жесткости
BYTE TrussElem_FullEffortsMode; // режим полного набора усилий для элемента фермы
BYTE IsTrueBeamMode; // признак разрезной балки — обычное (не холодногнутое сечение СП)
BYTE SteelDesignType; // 0 - undefined, 1 - стальное сечение, 2 - холодногнутое сечение
BYTE reserved2[1]; // резерв
double SteelSeismicCoef[2]; // сейсмические коэффициенты (SteelSeismicCoef[0] - Расчет на прочность при сейсмике; SteelSeismicCoef[1] - Расчет на устойчивость при сейсмике)
BYTE SlaveGroup; // дополнительная группа
double Corrosion; // толщина слоя коррозии
float DisplacementLimit_Abs[10]; // абсолютное ограничение по перемещениям (линейные размеры)
char SteelMarkUser[MAX_STRING_STEEL_NAME]; // имя стали (для заданного пользователем R_y)
double SpecialSteelCoef; // Коэффициент при особых (не сейсмических) воздействиях
double SpecialLargeSpanCoef; // Коэффициент понижающий расчетное сопротивление

double StepOutPlane_ratio; // коэффициент расстояния между раскреплениями к геом. длине (устойчивость плоской формы изгиба)
double EC_CoeffTorsionBuckling; // коэффициент (устойчивость крутильной формы)
double EC_LengthTorsionBuckling; // расчетная длина (устойчивость крутильной формы)
BYTE DeflectSNIPCheck[6]; // признак использования ограничения по прогибу
float DeflectSNIPLimit_L[6]; // ограничение по прогибу относительно длины элемента
float DeflectSNIPLimit_Abs[6]; // абсолютное ограничение по прогибу (линейные размеры)
BYTE DisplacementSNIPCheckEx[2]; // признак использования ограничения по перемещениям
float DisplacementExSNIPLimit_L[2]; // ограничение по перемещениям относительно длины элемента
float DisplacementExSNIPLimit_Abs[2]; // абсолютное ограничение по перемещениям (линейные размеры)
BYTE DeflectECCheck[6]; // признак использования ограничения по прогибу (EN)
float DeflectECLimit_L[6]; // ограничение по прогибу относительно длины элемента (EN)
float DeflectECLimit_Abs[6]; // абсолютное ограничение по прогибу (линейные размеры) (EN)
BYTE bPostbuckling; // 1 – (работа с гибкой стенкой не допускается)
BYTE CoefFibMode; // тип эпюры для $\varphi_б$ — см. таблицу ниже

```
float  reserved_dbl32_1[2];      // резерв
float  reserved_dbl32_2[2];      // резерв
double StiffStep;                // шаг ребер

BYTE   check_Rod_SelectHeight_NotGreatThan; // 1- учитывать ограничения по высоте (не более чем)
float  value_Rod_SelectHeight_NotGreatThan; // ограничение по высоте (не более чем)
BYTE   check_Rod_SelectHeight_NotLessThan;  // 1- учитывать ограничения по высоте (не менее чем)
float  value_Rod_SelectHeight_NotLessThan;  // ограничение по высоте (не менее чем)
BYTE   check_Rod_SelectWidth_NotGreatThan;  // 1- учитывать ограничения по ширине (не более чем)
float  value_Rod_SelectWidth_NotGreatThan;  // ограничение по ширине (не более чем)
BYTE   check_Rod_SelectWidth_NotLessThan;   // 1- учитывать ограничения по ширине (не менее чем)
float  value_Rod_SelectWidth_NotLessThan;   // ограничение по ширине (не менее чем)
BYTE   check_Rod_SelectReducedThickness_NotLessThan; // подбирать сечения с приведенной толщиной не менее
float  value_Rod_SelectReducedThickness_NotLessThan; // ограничение сечения с приведенной толщиной не менее
BYTE   Use_bRatio_New; // всегда 1 (использовать новую схему задания коэффициентов)
int    EC_EffType_TorsionBuckling; // 0 - задан EC_CoeffTorsionBuckling, 1 - задан EC_LengthTorsionBuckling
BYTE   res1[2];                  // резерв

int    EffType_XoZ;              // 0 - задан Koef_RasLen_XoZ, 1 - задан CalcLength_X0Z
int    EffType_YoZ;              // 0 - задан Koef_RasLen_YoZ, 1 - задан CalcLength_Y0Z
double CriticalMoment_za; // данные критического момента (EN или холодногнутые профили) положение точки
приложения нагрузки
double reserved_critical_moment;   // резерв
double CriticalMoment_MomentRatio; // данные критического момента (EN или холодногнутые профили)
соотношение концевых моментов
int    CriticalMoment_MomentType;  // данные критического момента (EN или холодногнутые профили) тип
эпюры моментов
WORD   CriticalMoment_LoadType; // данные критического момента (EN или холодногнутые профили)
расположение нагрузки
double reserved_New;               // резерв
int    StepOutPlane_type;          // 0 - задан StepOutPlane_ratio, 1 - задан StepOutPlane_linear
double CriticalMoment_k;   // Коэффициент расчетной длины, зависящие от условий закреплений опорных
сечений (поворот из плоскости изгиба)
double CriticalMoment_kw;  // Коэффициент расчетной длины, зависящие от условий закреплений опорных
сечений (депланация)
BYTE   SteelIsRollForming; // роликовое профилирование листового металла (только для холодногнутых)
BYTE   SteelThinNoTorsion; // 1 - закреплено от кручения (только для холодногнутых)
BYTE   SteelThinNoWarping; // 1 - не депланирует (только для холодногнутых)
BYTE   rest_reserved_CriticalMoment; // резерв
BYTE   Lattice_UseUserDefined; // решетка задана пользователем (сквозные сечения)
BYTE   Lattice_type; // тип решетки Default = 0, 1 - на планках (СНиП, EN), 2 -  раскосная решетка (СНиП, EN), 3
- треугольная решетка с распорками (СНиП, EN), 4 - крестовая решетка с распорками (СНиП), 5 -  крестовая решетка
(СНиП), 6- треугольная решетка (EN) — см. таблицу ниже
double Lattice_s;        // шаг решетки, применимо к типу решеток: 1,2,3,4,5,6
double Lattice_b;        // ширина планки, применимо к типу решетки: 1
double Lattice_t0;       // толщина планки, применимо к типу решетки: 1
double Lattice_Ad;       // площадь раскосов, применимо к типу решеток: 2,3,4,5,6
double Lattice_Av;       // площадь стоек, применимо к типу решеток: 2,3,4,6
BYTE Lattice_PostsProfileBaseName[8];        //имя  базы профиля стоек решетки
int Lattice_PostsProfileSectionIndex;        // номер раздела профиля стоек решетки
int Lattice_PostsStrIndex;                   // номер профиля стоек решетки
BYTE Lattice_StrutsProfileBaseName[8];       // имя базы профиля раскосов решетки
int Lattice_StrutsProfileSectionIndex;       // номер раздела профиля раскосов решетки
int Lattice_StrutsStrIndex;                  // номер профиля раскосов решетки
double Res3[41];         // резерв
};
```

Поле *SteelMark* зависит от выбранных норм проектирования. Допустимы следующие значения: "C235", "C245", "C255", "C275", "C285", "C345", "C345K", "C375", "C390", "C390K", "C440", "C590", "C590K", "20". При использовании

ДБН В.2.6-163:2010 и ДБН В.2.6-198:2014 дополнительно можно использовать "С355". Отметим, что следует использовать символы "С" и "К" латинского алфавита. Если строка *SteelMark* является пустой, то будет использовано значение Ry, заданное в описанной выше структуре *ApiSteelElem*.

Для учета ограничений по прогибам и перемещениям следует задавать следующие данные

	Признак учета ограничений	**Относительное ограничение (в долях длины)**	**Абсолютное ограничение**
При расчетах по Eurocode			
Горизонтальный прогиб от характеристических комбинаций	DeflectECCheck[0]	DeflectECLimit_L[0]	DeflectECLimit_Abs[0]
Горизонтальный прогиб от квазипостоянных комбинаций	DeflectECCheck[1]	DeflectECLimit_L[1]	DeflectECLimit_Abs[1]
Горизонтальный прогиб от частых комбинаций	DeflectECCheck[2]	DeflectECLimit_L[2]	DeflectECLimit_Abs[2]
Вертикальный прогиб от характеристических комбинаций	DeflectECCheck[3]	DeflectECLimit_L[3]	DeflectECLimit_Abs[3]
Вертикальный прогиб от квазипостоянных комбинаций	DeflectECCheck[4]	DeflectECLimit_L[4]	DeflectECLimit_Abs[4]
Вертикальный прогиб от частых комбинаций	DeflectECCheck[5]	DeflectECLimit_L[5]	DeflectECLimit_Abs[5]
Горизонтальные перемещения от характеристических комбинаций	DisplacementCheck[0]	DisplacementLimit_L[0]	DisplacementLimit_Abs[0]
Горизонтальные перемещения от квазипостоянных комбинаций	DisplacementCheck[1]	DisplacementLimit_L[1]	DisplacementLimit_Abs[1]
Горизонтальные перемещения от частых комбинаций	DisplacementCheck[2]	DisplacementLimit_L[2]	DisplacementLimit_Abs[2]
Вертикальные перемещения от характеристических комбинаций	DisplacementCheck[3]	DisplacementLimit_L[3]	DisplacementLimit_Abs[3]
Вертикальные перемещения от квазипостоянных комбинаций	DisplacementCheck[4]	DisplacementLimit_L[4]	DisplacementLimit_Abs[4]
Вертикальные перемещения от частых комбинаций	DisplacementCheck[5]	DisplacementLimit_L[5]	DisplacementLimit_Abs[5]
При расчетах по СниП, СП, ДБН			
Горизонтальный прогиб от всех нагрузок	DeflectSNIPCheck[0]	DeflectSNIPLimit_L[0]	DeflectSNIPLimit_Abs[0]
Горизонтальный прогиб от постоянных и длительных нагрузок	DeflectSNIPCheck[4]	DeflectSNIPLimit_L[4]	DeflectSNIPLimit_Abs[4]
Горизонтальный прогиб от временных нагрузок	DeflectSNIPCheck[1]	DeflectSNIPLimit_L[1]	DeflectSNIPLimit_Abs[1]
Вертикальный прогиб от всех нагрузок	DeflectSNIPCheck[2]	DeflectSNIPLimit_L[2]	DeflectSNIPLimit_Abs[2]
Вертикальный прогиб от постоянных и длительных нагрузок	DeflectSNIPCheck[5]	DeflectSNIPLimit_L[5]	DeflectSNIPLimit_Abs[5]
Вертикальный прогиб от временных нагрузок	DeflectSNIPCheck[3]	DeflectSNIPLimit_L[3]	DeflectSNIPLimit_Abs[3]
Вертикальные перемещения от всех нагрузок	DisplacementCheck[8]	DisplacementLimit_L[8]	DisplacementLimit_Abs[8]
Вертикальные перемещения от постоянных и длительных нагрузок	DisplacementCheck[7]	DisplacementLimit_L[7]	DisplacementLimit_Abs[7]
Вертикальные перемещения от временных нагрузок	DisplacementCheck[9]	DisplacementLimit_L[9]	DisplacementLimit_Abs[9]
Горизонтальные перемещения от всех нагрузок	DisplacementCheck[6]	DisplacementLimit_L[6]	DisplacementLimit_Abs[6]
Горизонтальные перемещения от постоянных и длительных нагрузок	DisplacementSNIPCheckEx[0]	DisplacementExSNIPLimit_L[0]	DisplacementExSNIPLimit_Abs[0]
Горизонтальные перемещения от временных нагрузок	DisplacementSNIPCheckEx[1]	DisplacementExSNIPLimit_L[1]	DisplacementExSNIPLimit_Abs[1]

Типы решеток

1	на планках	
2	раскосная решетка	
3	треугольная решетка с распорками	
4	крестовая решетка с распорками	
5	крестовая решетка	
6	треугольная решетка	

Типы эпюры для $\varphi_б$

Номер типа	**Количество закреплений сжатого пояса в пролете**	**Вид нагрузки в пролете**	**Эпюра М**	**Пояс, к которому приложена нагрузка**
1	Без закреплений	Сосредоточенная		Сжатый
2	Без закреплений	Сосредоточенная		Растянутый
3	Без закреплений	Равномерно распределенная		Сжатый
4	Без закреплений	Равномерно распределенная		Растянутый
5	Два и более, делящие пролет на равные части	Любая		Любой
6	Одно в середине	Сосредоточенная в середине		Любой
7	Одно в середине	Сосредоточенная в четверти		Сжатый
8	Одно в середине	Сосредоточенная в четверти		Растянутый

9	Одно в середине	Равномерно распределенная		Сжатый
10	Одно в середине	Равномерно распределенная		Растянутый
11	Без закреплений	Сосредоточенная на конце консоли		Сжатый
12	Без закреплений	Сосредоточенная на конце консоли		Растянутый
13	Без закреплений	Равномерно распределенная		Растянутый

*UINT ApiSetSteel(ScadAPI lpAPI, LPCSTR Text, const ApiSteelElem * Steel, UINT QntElem, const UINT * ListElem);*

Назначение

Задание группы конструктивных стальных элементов.

Параметры

lpAPI – указатель на объект API;
Text – имя группы;
Steel – указатель на адрес структуры ***ApiSteelElem;***
QntElem – число элементов;
ListElem – список элементов.

Возвращаемое значение

Номер группы конструктивных стальных элементов.

*APICode ApiChangeSteel(ScadAPI lpAPI, UINT Num, LPCSTR Text, const ApiSteelElem * Steel, UINT QntElem, const UINT * ListElem);*

Назначение

Корректировка группы конструктивных стальных элементов.

Параметры

lpAPI – указатель на объект API;
Num – номер группы;
Text – имя группы;
Steel – указатель на адрес структуры ***ApiSteelElem;***
QntElem – число элементов. Если *QntElem*=0, то список элементов не корректируется;
ListElem – список элементов.

Возвращаемое значение

Код возврата.

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

Назначение

Задание имени группы конструктивных стальных элементов.

Параметры

lpAPI – указатель на объект API;
Name – имя группы.

Возвращаемое значение

Код возврата.

LPCSTR ApiGetNameSteel(ScadAPI lpAPI, UINT Num);

Назначение

Чтение имени группы конструктивных стальных элементов.

Параметры

lpAPI – указатель на объект API.

Возвращаемое значение

Указатель на строку с группы конструктивных стальных элементов.

APICode ApiDeleteSteel(ScadAPI lpAPI, UINT Num);

Назначение

Удаление группы конструктивных стальных элементов.

Параметры

lpAPI – указатель на объект API;
Num – номер узла;

Возвращаемое значение

Код возврата.

APICode ApiClearSteel(ScadAPI lpAPI);

Назначение

Удаление всех групп конструктивных стальных элементов.

Параметры

lpAPI – указатель на объект API;

Возвращаемое значение

Код возврата.

*UINT SCAD_WINAPI ApiSetUnificationSteel(ScadAPI lpAPI, LPCSTR Text, UINT Qnt, const UINT * LstSteelGrps)*

Назначение

Создание группы унификации стальных элементов.

Параметры

lpAPI – указатель на объект API;
Text – имя группы;
Qnt – количество групп конструктивных элементов, которые следует включить в группу унификации;
LstSteelGrps – список номеров групп конструктивных элементов;

Возвращаемое значение

Номер созданной группы унификации.

UINT ApiGetQuantityUnificationSteel (ScadAPI lpAPI);

Назначение

Получение числа групп унификации конструктивных стальных элементов.

Параметры

lpAPI – указатель на объект API.

Возвращаемое значение

Число групп унификации конструктивных стальных элементов.

APICode SCAD_WINAPI ApiDeleteUnificationSteel(ScadAPI lpAPI, UINT Num)

Назначение

Удаление группы унификации стальных элементов.

Параметры

lpAPI – указатель на объект API;
Num – номер группы;

Возвращаемое значение

Код возврата.

2.20. Пример создания проекта

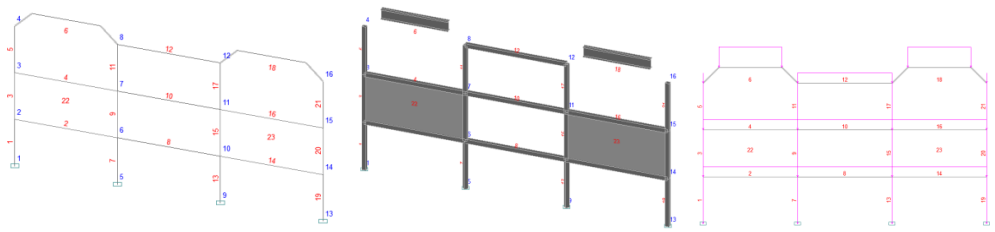

Рис 2-2. Тестовая расчетная схема со схемой загружения "Собственный вес"

```
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,'I',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,ApiPlateAxecX,2,Size,2,Elem);

        // 4-е загружения
        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 0 5 18 1 0 0.3 1");
        ApiSetLoadName(handle,4,"Ветер");
        //  преобразование статических загружений 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);

}
```

2.21. 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 загружений 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 (РСУ) 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 NameWorkCat);*

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;

NameWorkCat – указатель на полное имя каталога рабочих файлов. Если данный параметр равени 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.

3.3. 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.

BOOL ApiYesRSR (ScadAPI lpAPI);

Назначение

Проверка наличия в результатах расчета вычисленных РСР(реакций).

Параметры

lpAPI – указатель на объект API.

Возвращаемое значение

TRUE – РСР вычислены, FALSE – нет.

BOOL ApiYesRSDeflection (ScadAPI lpAPI);

Назначение

Проверка наличия в результатах расчета вычисленных прогибов.

Параметры

lpAPI – указатель на объект API.

Возвращаемое значение

TRUE – прогибы вычислены, FALSE – нет.

BOOL ApiYesPunching (ScadAPI lpAPI);

Назначение

Проверка наличия в результатах расчета вычисленных продавливаний.

Параметры

lpAPI – указатель на объект API.

Возвращаемое значение

TRUE – прогибы вычислены, FALSE – нет.

3.4. 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 загружений;
- *API_RESULT_MODE* – формы колебаний;
- *API_RESULT_STABIL* – формы потери устойчивости;
- *API_RESULT_STABIL_COMB* – формы потери устойчивости от комбинаций загружений;
- *API_RESULT_MASS* – сформированные узловые массы для динамического загружения;
- *API_RESULT_FRAGMENT* – реакции от фрагмента схемы;
- *API_RESULT_FRAGMENT_COMB* – реакции от фрагмента схемы для комбинаций загружений;
- *API_RESULT_FRAGMENT_LINK* – реакции в связях,
- *API_RESULT_FRAGMENT_LINK_COMB* – реакции в связях от комбинаций загружений.

Каждое загружение 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 загружения;
    WORD     NumSchemUnite;    //  номер задачи вариации
    WORD     NumLoadSchemUnite; // номер загружения 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 загружения
    double * Comb;             //  указатель на на рассматриваемую комбинацию
                               //     загружений линейной задачи: метод ApiGetQuantityLoad
                               //     для определения числа загружений. В Comb[i] находится
                               //     коэффициент загружения 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 загружение 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 Y при шестикомпонентном воздействии
- *ApiDYN_FORMZ* – направление Z при шестикомпонентном воздействии
- *ApiDYN_FORMUX* – направление U_X при шестикомпонентном воздействии
- *ApiDYN_FORMUY* – направление U_Y при шестикомпонентном воздействии
- *ApiDYN_FORMUZ* – направление U_Z 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 загружение монтажа + динамическая огибающая
- *ApiDYN_LB_SI* – базовое загружение монтажа + динамическая огибающая гармонического воздействия;
- *ApiDYN_LN_TIME* – нелинейность+интегрирование по времени
- *ApiDYN_LB_TIME* – базовое загружение монтажа + интегрирование по времени

UINT ApiGetResultQuantityLoad (ScadAPI lpAPI);

Назначение

Число загружений.

Параметры

lpAPI – указатель на объект API.

Возвращаемое значение

число загружений.

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 загружений.

Параметры

lpAPI – указатель на объект API.

Возвращаемое значение

число динамических загружений.

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

Назначение

Число строк результатов для загружения/шага сохранения нелинейного процесса.

Параметры

lpAPI – указатель на объект API.
Type – тип анализируемых данных: API_RESULT_LOAD и т.д.;
NumLoad – номер загружения для линейного процесса и номер сохраненных результатов для нелинейного.

Возвращаемое значение

число строк результатов для загружения/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 загружения/шага нелинейного процесса.

Параметры

lpAPI – указатель на объект API.
Type – тип анализируемых данных: API_RESULT_LOAD и т.д.;
NumLoad – номер загружения для линейного процесса и номер сохраненных результатов для нелинейного;
NumStr – номер строки результатов загружения;
APD – указатель на структуру ApiLoadingData.

Возвращаемое значение

Код возврата.

3.5. Динамика

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

Назначение

Приведенные узловые массы.

Параметры

lpAPI – указатель на объект API.
NumLoad – номер загружения 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

Формы колебаний.

Параметры

lpAPI – указатель на объект API.

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

NumMode – номер формы колебаний;

NumNode – номер узла.

Возвращаемое значение

Указатель на значения формы колебаний; в i-й позиции которого находится значение смещения по направлению степени свободы i +1;

Если ошибки в данных, то указатель – NULL.

3.6. Перемещения

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

Назначение

Перемещения узла.

Параметры

lpAPI – указатель на объект API.

NumLoad – номер загружения 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 загружений.

Параметры

lpAPI – указатель на объект API.

NumLoad – номер загружения 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.

3.7. 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 загружения 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 загружений.

Параметры

lpAPI – указатель на объект API.

NumLoad – номер загружения 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 загружений: 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 загружения 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 загружений.

Параметры

lpAPI – указатель на объект API.

NumLoad – номер загружения 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 загружений: в i-й позиции которого находится значение реакции по направлению степени i +1.

Если ошибки в данных, то указатель – NULL.

3.8. 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.

3.9. РСУ, РСП, РСР(реакций в связях), РСПрогибов и РСПродавливания

Для чтения вычисленных усилий/напряжений/перемещений 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. Для перемещений равно 3 – x,y,z.
        UINT            LengthData;    // число строк РСУ/РСП
        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 нагрузок
        BYTE     YesFire;        // признак наличия пожара
        BYTE     YesShortTerm;   // входят кратковременные

        BYTE     Res[2];
        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 РСУ 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 NumNode, ApiElemRsu * rsu);*

Назначение

Информация о РСП в узле.

Параметры

lpAPI – указатель на объект API.
NumNode – номер узла;
rsu – указатель на структуру ApiElemRsu.

Возвращаемое значение

Код возврата.

*APICode ApiGetRsr(ScadAPI lpAPI, UINT NumNode, ApiElemRsu * rsu);*

Назначение

Информация о РСР в узле.

Параметры

lpAPI – указатель на объект API.
NumNode – номер epkf;
rsu – указатель на структуру ApiElemRsu.

Возвращаемое значение

Код возврата.

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

Назначение

Информация о РСПрогибов в узле.

Параметры

lpAPI – указатель на объект API.
NumElem – номер элемента;
rsu – указатель на структуру ApiElemRsu.

Возвращаемое значение

Код возврата.

3.10. 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 использовав методы, описанные в разделе 2.16.