

# **NormCAD**



## **программирование**

ООО ЦРСАП "САПРОТОН"

г. Реутов, 2020 г.

УДК 681.3

ББК 32.972

NormCAD: программирование / Стебаков Е.И. – Реутов, ООО ЦРСАП "САПРОТОН", 2020. – 70 с.: ил.

Книга знакомит с программированием в NormCAD и на его основе. В ней содержится описание языка и среды программирования в NormCAD, примеры написания кода и его использования. Приводится пример комплексной автоматизации расчета строительной конструкции на основных стадиях проектирования.

*Корректура Екатерины Федоровой*

ООО ЦРСАП "САПРОТОН"

г. Реутов, ул. Победы, 9

т. (495) 510-94-97

Отпечатано в ООО ЦРСАП "САПРОТОН"  
143966, Московская область, г. Реутов, ул. Победы, 9  
[www.normcad.ru](http://www.normcad.ru)

(C) ООО ЦРСАП "САПРОТОН", 2020

(C) Стебаков Е.И., 2020

## **Предисловие**

Среда отладки NormCAD может быть использована не только разработчиками, но и рядовыми пользователями для настройки текста комментариев в отчетах, для поиска и срочного исправления ошибок в алгоритме, для проверки алгоритма на соответствие нормам. Возможны и другие цели ее применения, например, для уяснения особенностей расчета по нормам.

Встроенный язык программирования NormCAD является узкоспециализированным интерпретируемым языком. Использование интерпретации обуславливает невысокую скорость его выполнения, но, как правило, она не критична, учитывая компактность кода написанных на нем алгоритмов, возможности современных компьютеров и то, что этот язык предусматривает постоянное взаимодействие с пользователем в процессе исполнения, т.к. скорость реакции пользователя значительно меньше скорости интерпретации кода.

Новые версии NormCAD (начиная с 9.0) позволяют использовать средства компиляции, что значительно ускоряет расчет при обработке больших таблиц исходных данных, что позволяет помимо подробного расчета отдельных элементов, который дает возможность детальных проверок, использовать тот же алгоритм для проведения расчета сооружений в целом или их фрагментов. Имеется также возможность экспорта алгоритмов NormCAD в Visual Basic для создания на основе этих алгоритмов приложений, не зависящих от среды NormCAD, в других средствах разработки.

Средства программирования в NormCAD нацелены, прежде всего, на задачу автоматизации инженерных расчетов с подробным документированием выполненных вычислений. По умолчанию все строки алгоритма, задействованные в выполненном расчете, автоматически отражаются в тексте отчета. Это составляет основную особенность средств программирования NormCAD, в отличие от других инструментов, в которых для документирования расчета необходимо выполнение дополнительных действий, т.к. эта задача не является в них основной. Помимо автоматического документирования в NormCAD имеются и другие средства, направленные на автоматизацию расчетов по нормам: учет размерности переменных; извлечение данных из таблиц с интерполяцией промежуточных значений.

Язык программирования в NormCAD не претендует на универсальность и поддерживает только минимально необходимый набор функций и управляющих конструкций, что не исключает подключение внешних функций, созданных на основе других языков, которое может быть использовано также для ускорения больших по объему вычислений, не требующих подробного документирования.

Другой особенностью средств программирования в NormCAD является автоматическое создание диалоговых окон по ходу выполнения расчета с вопросами к пользователю и запросом недостающих данных. Отсутствие необходимости создания их вручную, с одной стороны, значительно ускоряет процесс разработки, с другой стороны, возможности автоматических средств диалога с пользователем также ограничены небольшим набором наиболее часто используемых диалоговых

окон. Эти ограничения компенсируются возможностью подключения диалоговых окон, созданных на основе других языков программирования.

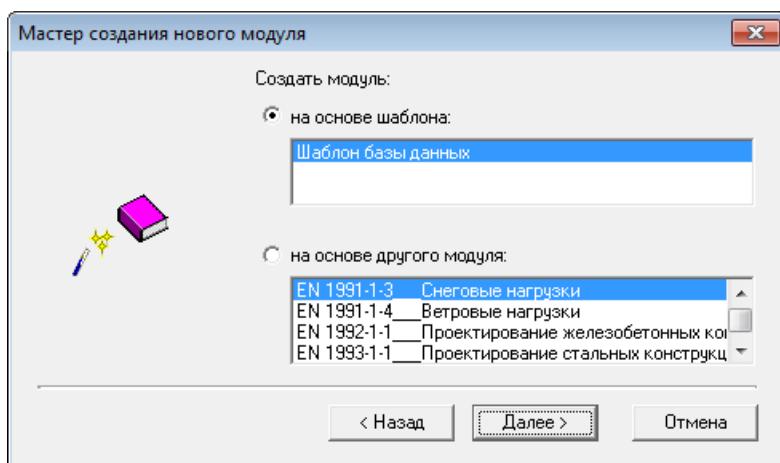
Средства программирования в NormCAD ориентированы на доступность рядовым инженерам, не требуют большого опыта в программировании и достаточно просты для освоения, т.к. элементы и конструкции языка программирования сведены к минимуму. Процесс разработки значительно упрощается благодаря возможностям среды отладки. В то же время требуется определенный навык для освоения некоторых специфических приемов программирования в NormCAD.

# 1. Основы программирования в NormCAD

## 1.1 Модули NormCAD

В модулях NormCAD хранятся алгоритмы расчета и таблицы с данными. Для каждого нормативного документа (СНиП, Пособия к СНиП) или его части создаются отдельные модули, основу которых составляют файлы книг электронных таблиц (с расширением nrb).

Для создания нового модуля воспользуйтесь Мастером создания модулей (команда **Создать модуль** в меню **Сервис** основной программы).



В первом окне Мастера вводится название создаваемого модуля. При этом запрашивается сокращенное, несклоняемое имя, например, СНиП II-23-81 (оно будет использоваться в ссылках отчета) и развернутое описательное имя, например, "Стальные конструкции". Эти элементы должны соответствовать номеру и названию нормативного документа, на основе которого создается модуль, и составляют полное имя модуля: "СНиП II-23-81\_\_Стальные конструкции".

Новый модуль создается на основе шаблона или другого существующего модуля.

После создания модуль открывается в **Отладчике** для редактирования.

Редактирование модуля осуществляется в **Отладчике**.

Вызов **Отладчика** из программы NormCAD выполняется кнопкой **Алгоритм**. В появившемся диалоговом окне следует указать, что модуль открыт для редактирования.

При этом открывается модуль, на основе которого выполняется текущий расчет в основной программе, т.е. соответствующий активному документу. Для открытия модулей в **Отладчике** служит также команда **Открыть модуль** в меню **Сервис** основной программы.

Дополнительную информацию о редактировании модулей см. также в меню **Справка Отладчика**.

Открытие модулей в **Отладчике** для просмотра выполняется кнопкой **Алгоритм**. Далее в диалоговом окне следует указать, что модуль открыт для

просмотра. При этом открывается модуль, на основе которого выполняется текущий расчет в основной программе, т.е. соответствующий активному документу.

В процессе редактирования модулей в Отладчике записываются их резервные копии. Резервные копии создаются в момент сохранения изменений в пунктах, таблицах или во всем модуле в целом. Всего записываются до 8 резервных копий.

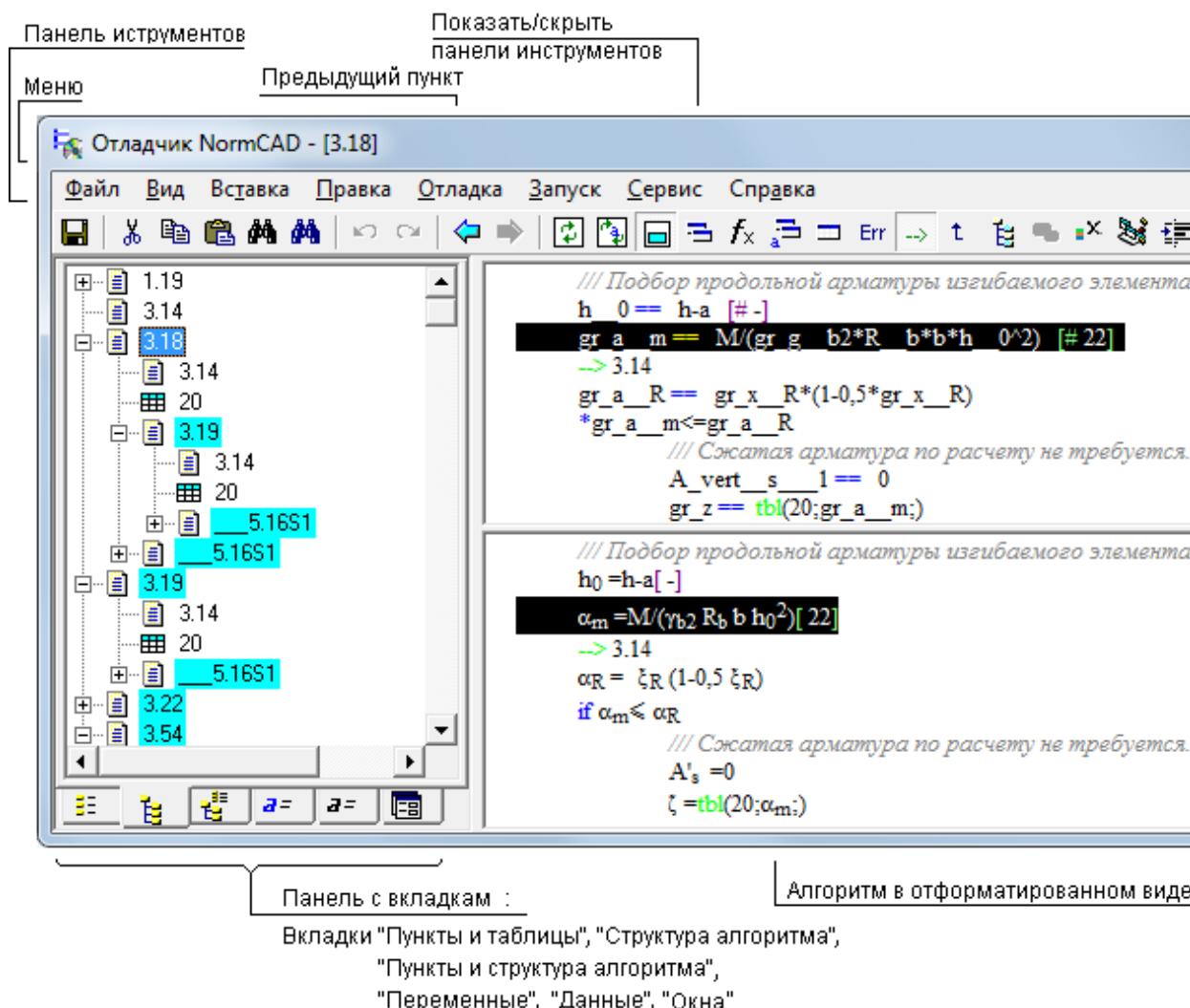
Для восстановления до предыдущих редакций служит команда **Восстановить модуль** в меню **Сервис** основной программы, с помощью которой можно выбрать вариант модуля из сохраненных файлов резервных копий и заменить им текущую редакцию модуля.

## 1.2 Работа в отладчике

Отладчик NormCAD предназначен для просмотра, редактирования и отладки алгоритма и таблиц с данными модулей нормативных документов.

Просмотр алгоритма и таблиц может использоваться для контроля правильности выполненных расчетов.

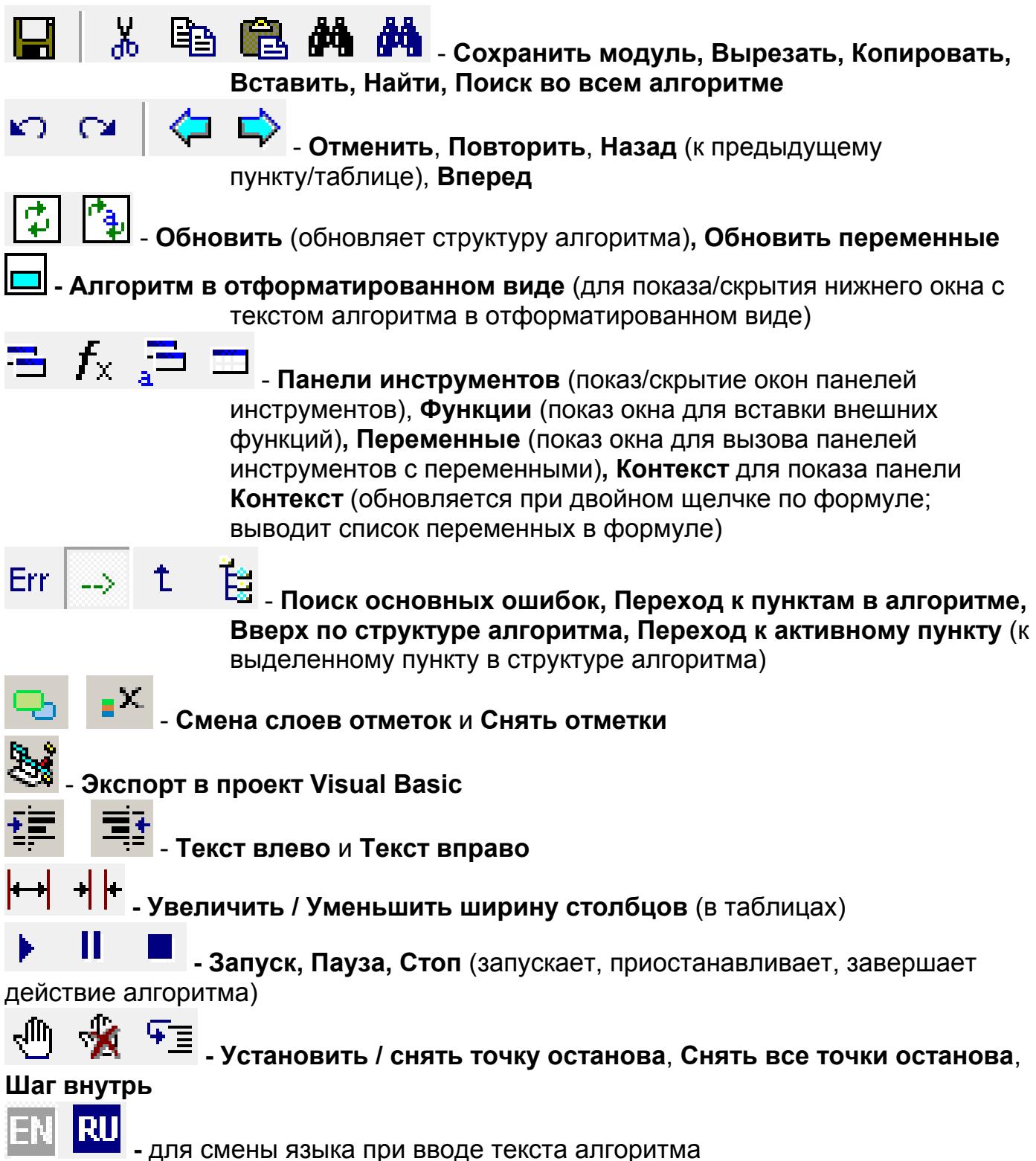
Основные элементы среды отладчика приведены на схеме:



Средства редактирования Отладчика предназначены:

- для внесения изменений и исправлений в существующий алгоритм модулей нормативных документов
- для ввода новых пунктов в существующие модули
- для ввода в программу новых нормативных документов

Кнопки панели инструментов:



Вызов **Отладчика** из программы NormCAD выполняется кнопкой **Алгоритм**



При вводе текста алгоритма можно использовать панели инструментов:

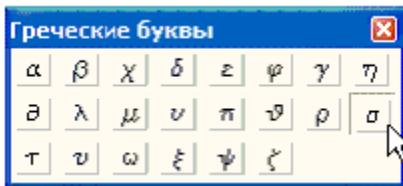
- для ввода обычного текста кнопку с символом x:



- для ввода индексов кнопку x\_2



- для символа **S** панель Греческие буквы



Для того, чтобы скрыть или показать основные плавающие панели



инструментов, служат кнопки

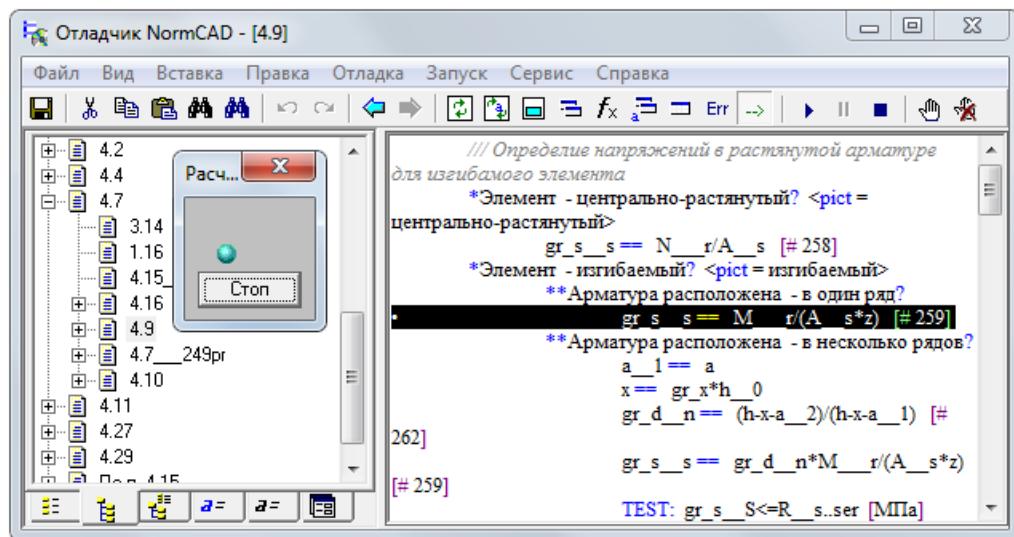
Отладчик NormCAD предоставляет следующие специальные средства для упрощения процесса разработки:

- автоматически строится древовидная структура алгоритма, позволяющая установить связи различных его элементов (в отличие от традиционных сред разработки, где алгоритм представлен в виде разрозненных подпрограмм); выявление конечного количества ветвей этой структуры упрощает процедуру планирования процесса отладки;
- алгоритм автодокументируется: дополняется ссылками и комментариями, и отображается как обычные математические формулы, т.е. так, как они должны выглядеть в тексте отчета;
- система поиска ошибок;
- средства разработки визуального интерфейса;
- система поиска по всему тексту алгоритма в различных его элементах (например, только в условиях расчета);
- алгоритм можно автоматически конвертировать в проект Visual Basic.

## Отладка и запуск алгоритма в отладчике

В тексте алгоритма можно установить точки останова, где выполнение текста алгоритма будет приостановлено.

Установка и удаление точек останова выполняется с помощью кнопок **Установить/снять точки останова** и **Снять все точки останова**



Продолжение выполнения алгоритма выполняется командой **Запуск** (клавиша **F5** или кнопка ).

Пошаговая отладка выполняется командой **Шаг внутрь** .

В момент приостановки выполнения алгоритма можно ознакомиться с текущими значениями переменных в соответствующих вкладках в левой части окна отладчика.

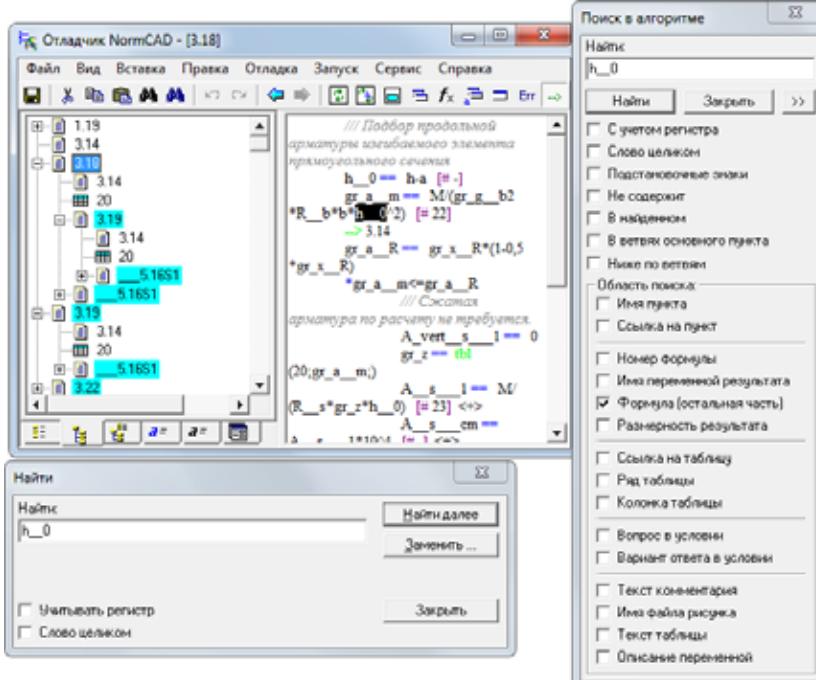
Приостановка выполнения алгоритма в любой момент времени осуществляется командой **Приостановить** (кнопка ).

Прекращение выполнения алгоритма – командой **Стоп** (кнопка ).

### Поиск и замена в тексте алгоритма

Поиск в тексте алгоритма всех пунктов модуля выполняется командой **Сервис\Поиск во всем алгоритме ...** или кнопкой .

Предварительно выбираются опции поиска в окне **Поиск в алгоритме**. Найденные пункты выделяются цветом в структуре пунктов и таблиц.



Имеется возможность многократного поиска в алгоритме с наложением результатов поиска с помощью использования маркеров разного цвета. Для этого можно использовать:

- панель выбора цвета отметок (для раскрытия панели нажмите кнопку >> в окне **Поиск в алгоритме**)



- кнопки на панели инструментов **Смена слоев отметок** (при наложении отметок) и **Снять отметки**

Поиск и замена в тексте алгоритма отдельных пунктов модуля выполняется командами **Найти** и **Заменить** меню **Правка**.

Поиск ссылок на активный пункт осуществляется командой **Показать ссылки на активный пункт** меню **Сервис** (кнопка ). Найденные пункты выделяются цветом в структуре пунктов и таблиц.

## Поиск ошибок

Следующими командами в меню Сервис осуществляется поиск ошибочных ссылок: **Показать ошибочные ссылки**, **Показать зацикливания**, **Показать ссылки на несуществующие пункты**.

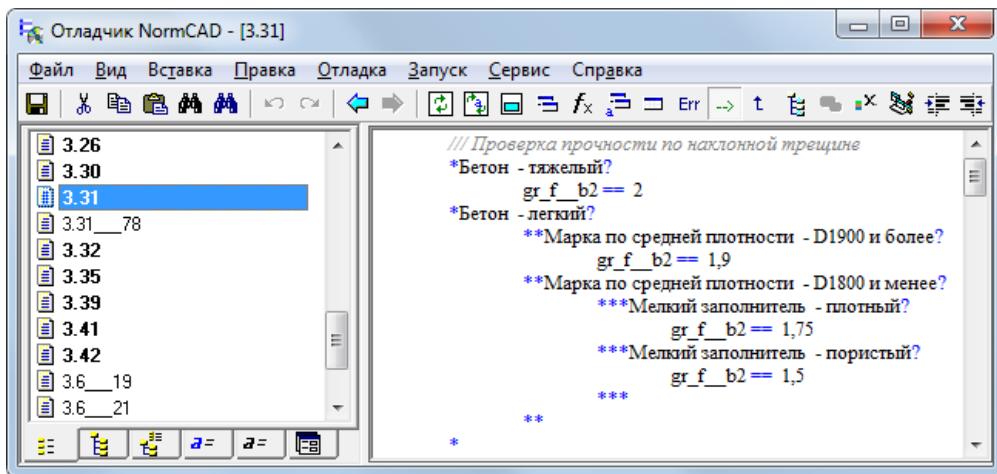
Поиск ошибок в тексте алгоритма осуществляется с помощью кнопки на панели инструментов **Поиск основных ошибок** или командами меню **Сервис**. В меню **Сервис** выбирается:

1. область поиска:
  - во всем модуле
  - в активном пункте
2. вид ошибок:
  - ошибки определенного вида
  - все ошибки

Ошибканые пункты выделяются цветом в структуре пунктов и таблиц. В нижней части экрана появляется окно с описанием ошибок. Двойной щелчок по строке описания приводит к переходу к соответствующей ошибке в тексте алгоритма.

## 1.3 Ввод пунктов и таблиц

В качестве основной единицы алгоритма принят так называемый "пункт". Он соответствует процедуре или подпрограмме по общепринятой терминологии и содержит информацию о пункте и части пункта нормативного документа. Так алгоритм пункта 3.31 выглядит в окне отладчика:



## Ввод данных из таблиц

Имеется несколько способов ввода таблиц из нормативных документов. Выбор способа зависит от структуры таблицы.

Можно выделить два основных способа ввода табличных данных:

1. с помощью стандартных таблиц, которые можно создавать и редактировать в среде **Отладчика NormCAD**
2. с помощью **Справочников** - баз данных Access, обычно хранящихся в папке ... \Program Files\NormCAD\Dat\Tables (для доступа к ним служит команда Сервис\Справочники основной программы NormCAD; для вставки – команда Отладчика Сервис\Диалоговые окна\Вставить Справочник).

Использование стандартных таблиц NormCAD позволяет автоматически извлекать из них промежуточные значения по интерполяции, использовать в ячейках формулы и условия, организовывать обращения с вопросами к пользователю.

Так выглядит таблица 26\_1 в отладчике NormCAD:

	A	B	C	D
1	N_l/N	_0/h		
2		6	8	10
3	0	0,93	0,92	0,91
4	0,5	0,92	0,91	0,9
5	1	0,92	0,91	0,89
6				
7				
8				

Для ввода данных из стандартных таблиц NormCAD используется функция `tbl`:

<имя переменной>==tbl(имя таблицы;[значение для ряда в таблице];[значение в колонке таблицы]).

Приведем несколько примеров:

`y==tbl(26__1;0,5;8)` присваивает переменной `y` значение 0,91

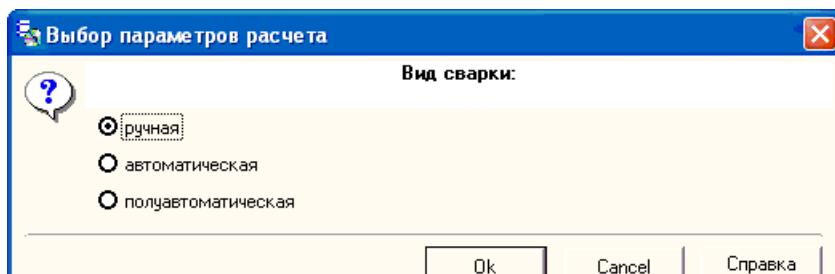
Для промежуточных значений для ряда или в колонке таблицы используется линейная интерполяция:

`y==tbl(26__1;0;9)` присваивает переменной `y` значение 0,915

При вводе данных из таблиц в диалоге с пользователем может производиться выбор условий расчета. Например, при обращении к следующей таблице

	A	B	C	D
1	Вид сварки	$t_{max}$		
2		0,4	0,5	0,6
3	ручная	5	5	6
4	автоматическая	4	4	5
5	полуавтоматическая	4	4	5

при запуске на расчет появляется стандартное диалоговое окно для выбора условий расчета



При этом значение для ряда таблицы не указывается. Например, в табл. 38 СНиП "Стальные конструкции"

`y==tbl(38__3;;0,4)` присваивает переменной `y` значение 5.

Приведем пример таблицы, содержащей проверку условий и формулы (табл. 77 СНиП "Стальные конструкции"):

	A	B	C
1	Вид нагрузки	$gr_a$	
2		$=gr_a < 40$	$=gr_a > 40$
3	Сосредоточенная к верхнему поясу	$=1,75+0,09*gr_a$	$=3,3+0,053*gr_a-4,5*10^{-5}*gr_a^2$
4	Сосредоточенная к нижнему поясу	$=5,05+0,09*gr_a$	$=6,6+0,053*gr_a-4,5*10^{-5}*gr_a^2$
5	Равномерно распределенная к верхнему поясу	$=1,6+0,08*gr_a$	$=3,15+0,04*gr_a-2,7*10^{-5}*gr_a^2$
6	Равномерно распределенная к нижнему поясу	$=3,8+0,08*gr_a$	$=5,35+0,04*gr_a-2,7*10^{-5}*gr_a^2$

Обращение к таблице: `gr_y == tbl(77__1;;gr_a)`. При `gr_a=1` и равномерно-распределенной нагрузке, приложенной к верхнему поясу, получим `gr_y=1,68`, т.к. выполняется условие `gr_a <= 40`.

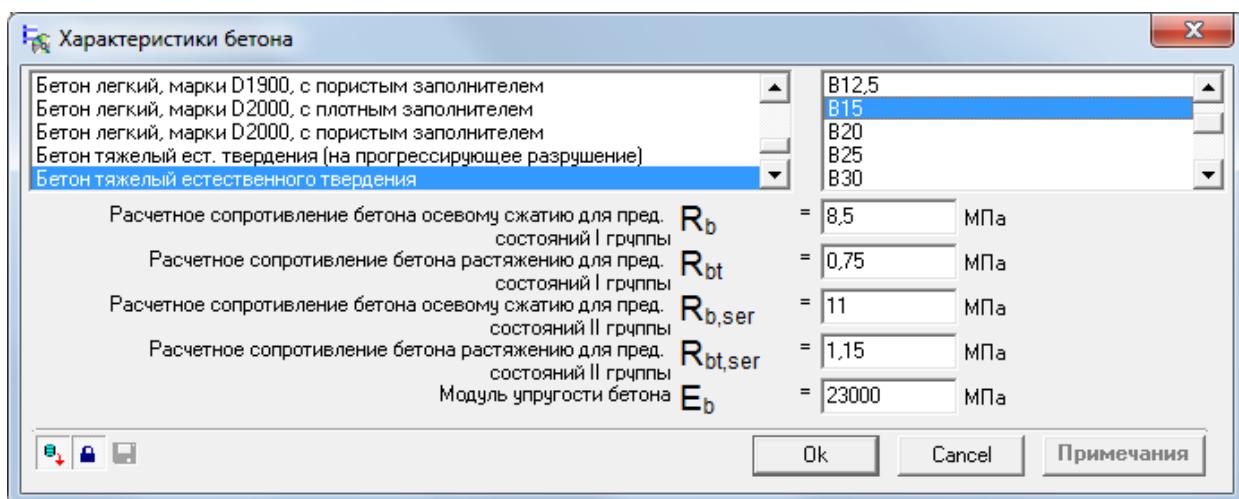
**Справочники** не позволяют извлекать из них промежуточные значения по интерполяции и использовать другие перечисленные выше возможности стандартных таблиц, а ограничивают ввод фиксированными значениями. Они предназначены для ввода данных из таблиц, имеющих простейший структурный вид – разделение на группы и подгруппы (например, данные из сортаментов, где группами записей являются типы профилей, а подгруппами – марки профиля).

Физически справочники хранятся в файле баз данных Access. В таких файлах имеется по две таблицы. В таблице Main описываются группы записей (например, перечисляются различные типы профилей). В таблице Data – записи для каждой подгруппы данных (например, для каждой марки профиля).

Каждому справочнику соответствует одноименная группа данных в модуле NormCAD. Для обращения к **Справочникам** функции алгоритмов не используются. Диалоговые окна **Справочников** вызываются при обращении пользователя к соответствующей группе данных или показываются автоматически при запуске задачи на расчет.

Для вставки **Справочника** необходимо предварительно создать группу данных, включающую переменные, для которых вводятся значения.

Пример окна для ввода данных из **Справочника**:



При более сложной структуре данных в таблицах используются диалоговые окна произвольного вида. При этом можно создать несколько **Справочников** и обращаться к ним через элементы управления DataGroupBox (см. далее в разделе 11.8 "Разработка интерфейса взаимодействия с пользователем").

Основное преимущество использования **Справочников** – быстрый способ организации обращений к данным структурного вида. Но они отличаются меньшей гибкостью по сравнению со стандартными таблицами NormCAD.

По своей структуре таблицы нормативных документов могут быть простыми и вложенными, когда ячейка таблицы включает в себя вложенную таблицу.

Сложные таблицы, включающие вложенные таблицы, разбиваются на ряд простых стандартных таблиц NormCAD, к которым обращаются, используя условия в алгоритме и функции tbl. В качестве примера можно привести табл. 77 СНиП "Стальные конструкции".

## Отсутствие значения в ячейке таблицы

Некоторые таблицы в нормативных документах содержат прочерки в ячейках. При отсутствии значения в ячейке таблицы NormCAD ставится символ "-" (минус).

## Ввод пунктов

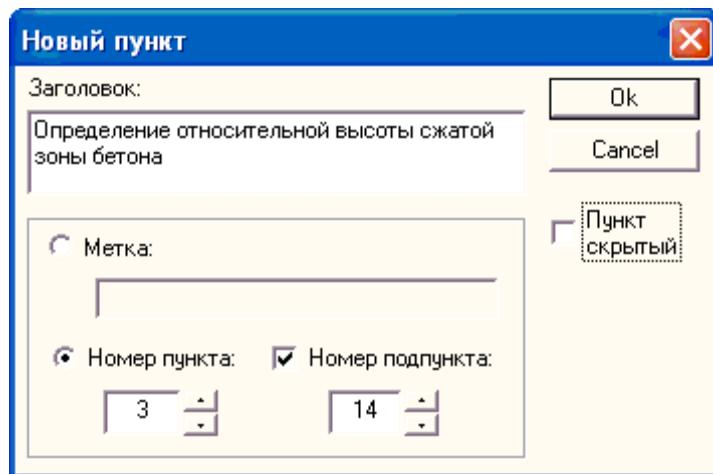
При вводе нового пункта (команда **Пункт** меню **Вставка**) в диалоговом окне новый пункт вводится следующая информация:

1) **Заголовок** - развернутый описательный заголовок пункта.

2) **Сокращенное имя пункта**, которое используется для обращения к пункту в алгоритме. Оно может совпадать с номером пункта в нормативном документе, например 3.14, или иметь имя произвольного вида, которое вводится в поле **Метка**.

3) Может быть установлено, что пункт является скрытым.

Для пунктов, которые не являются скрытыми, можно ввести также дополнительную информацию в окне **Основные пункты** меню **Вид**.



## Скрытая часть в именах переменных и пунктов

В нормативных документах одна и та же переменная может использоваться для обозначения разных величин. В этом случае используются переменные со скрытой частью (той, которая не отражается в отчете). Например, в алгоритме можно использовать переменные a, a\_1, a\_xxx. Во всех случаях в отчете переменная будет выглядеть как a. Скрытая часть отделяется тремя знаками подчеркивания.

Этот же прием используется в обозначениях пунктов и таблиц, что позволяет разбивать сложные пункты и таблицы на несколько простых. Например, пункты алгоритма 1.1\_1, 1.1, 1.1\_xxx будут упоминаться в отчете как п.1.1.

## Скрытые пункты

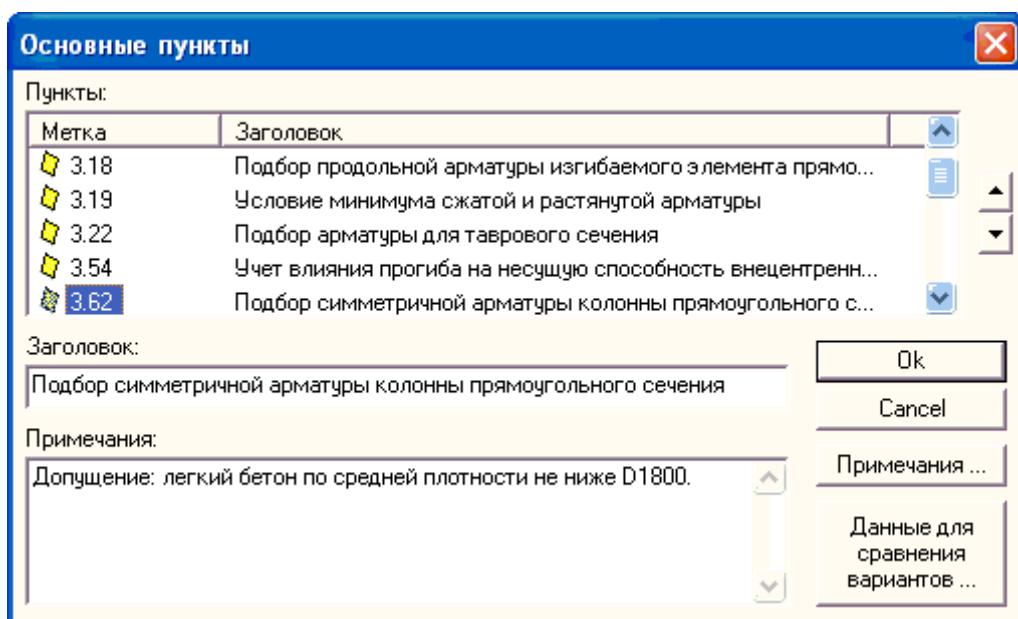
В нормативных документах встречаются пункты, которые невозможно использовать без предварительного обращения к другим пунктам. Такие пункты отмечаются в модуле как скрытые, т.е. невидимые из пользовательского интерфейса основной программы NormCAD. Для скрытия пункта служит переключатель в форме для создания пункта или контекстное меню вкладки **Элементы** со списком пунктов и таблиц (при выделении соответствующего пункта).

Отменить установку того, что пункт является скрытым (невидимым из пользовательского интерфейса основной программы), можно из контекстного меню вкладки **Элементы**.

## Заголовки пунктов

Сокращенное имя пункта (которое используется для обращения к пункту в алгоритме) не должно содержать пробелов. Имена пунктов отображаются в отчете в соответствии с правилами, принятыми для переменных. Т.е. символы после трех символов подчеркивания ("\_\_") в отчете не отображаются. Это позволяет разбивать на несколько пунктов один пункт в нормативном документе. Например, пункт 15.5 можно разбить на пункты 15.5\_\_Common, 15.5\_\_1 и 15.5\_\_2. При расчете по любому из этих пунктов в отчете будет ссылка на пункт 15.5.

Заголовок пункта можно изменить командой **Переименовать** из контекстного меню вкладки **Элементы** (со списком пунктов и таблиц), предварительно выделив значение пункта или таблицы, которые требуется переименовать. При этом редактируется сокращенное имя пункта (которое используется для обращения к пункту в алгоритме). Разворнутый описательный заголовок пункта можно отредактировать с помощью команды **Основные пункты** меню **Вид**.



## **Порядок пунктов**

Порядок пунктов в древовидной структуре основной программы можно изменить в диалоговом окне **Основные пункты** меню **Вид** Отладчика. Для этого служат кнопки со стрелками.

## **Примечание к расчету по пункту**

К основным пунктам можно добавить примечание в диалоговом окне **Основные пункты** меню **Вид** отладчика (кнопка **Примечания...**).

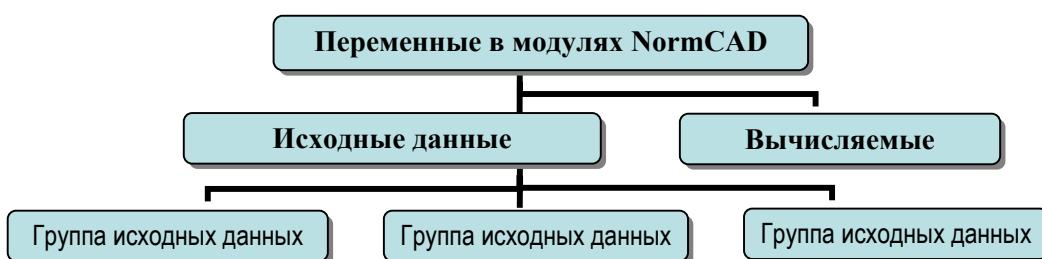
## 2. Язык программирования в NormCAD

### 2.1 Переменные в NormCAD

В NormCAD все переменные делятся на два вида по возможности их ввода пользователем:

1. Переменные, значения которых доступны для ввода и редактирования пользователем, т.е. являются исходными данными. Они, в свою очередь, подразделяются на группы исходных данных. Для каждой группы данных автоматически создается диалоговое окно для их ввода и редактирования пользователем. При необходимости возможно создание специальных диалоговых окон, которые заменяют автоматические.

2. Прочие переменные, которые не относятся к исходным данным. Они не вводятся пользователем, а принимают значения в ходе вычисления по алгоритму расчета.

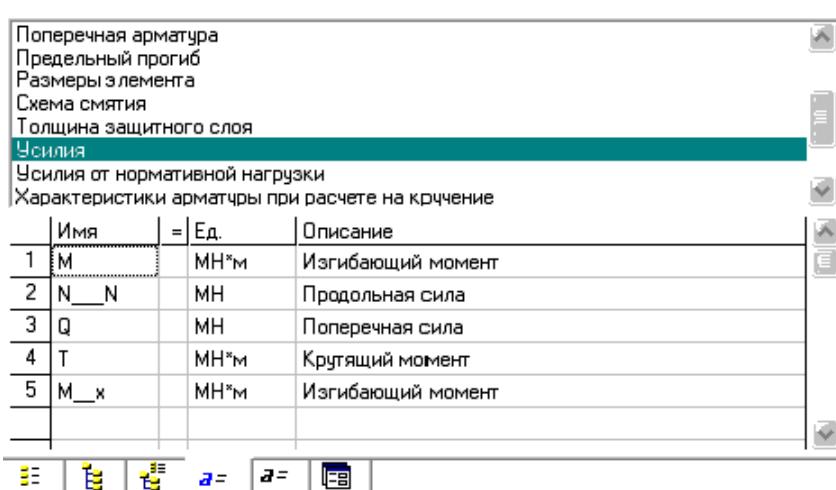


Разделение переменных на группы обусловлено в основном тем, что в NormCAD имеется несколько универсальных средств для работы с группами переменных: автоматическое создание диалоговых окон, операции по удалению значений и др.

Для создания группы данных выберите в Отладчике команду **Группа данных** меню **Вставка**. И далее введите имя группы данных, а затем опишите хотя бы одну переменную из этой группы в таблице на вкладке **Исходные данные**.

Описание переменных осуществляется в таблицах на вкладках **Исходные данные** и **Переменные**.

Для каждой переменной указывается ее имя, значение по умолчанию, размерность и описание.



Если переменные не вводятся пользователем, а принимают значения в ходе вычисления по алгоритму расчета, их не требуется относить к какой-либо группе данных. Их следует описать в таблице вкладки **Переменные**.

	Имя	=	Ед.	Описание
107	gr_y_s			Коэффициент, учитывающий неравномерное распределение
108	gr_y_sw			Коэффициент, учитывающий шаг и диаметр стержня
109	h_o	м		Рабочая высота сечения

Если для переменных еще не введено описание, то при запуске на расчет оно выполняется программой автоматически по следующим правилам:

- Если переменная не принимает значений в результате вычислений в алгоритме, она записывается в группу данных **Разные данные**.
- В противном случае переменная записывается в таблицу на вкладке **Переменные**, т.е. считается, что она не относится к исходным данным.

Имена переменным присваиваются по правилам, определяющим как переменные будут выглядеть в отформатированном тексте отчета:

1. Символы после трех символов подчеркивания ("\_\_\_\_") в отчете не отображаются.  
Например: A\_\_\_\_1 - A.
2. Символы после двух символов подчеркивания ("\_\_") отображаются в виде нижних индексов.  
Например: A\_\_1 – A<sub>1</sub>.
3. Символы после двух символов подчеркивания ("\_up\_") отображаются в виде верхних индексов.  
Например: A\_up\_1 – A<sup>1</sup>.
4. Если перед символами стоит сочетание "gr\_" - они отображаются буквами греческого алфавита (т.е. с использованием соответствующего шрифта).  
Например: gr\_g - γ
5. Если перед символами стоит сочетание "grl\_" - они отображаются буквами греческого алфавита с надчеркиванием.  
Например: grl\_l - λ
6. Если перед символами стоит сочетание "III\_" - они отображаются с надчеркиванием.  
Например: III\_K - K̄.
7. Две точки отображаются запятой.  
Например: gr\_e\_b..ult - ε<sub>b,ult</sub>
8. Для вертикальной черты используется сочетание "\_vert".  
Например: a\_vert - a'.
9. Символ подчеркивания ("\_"), встречающийся в других случаях, отменяет предшествующий формат символов.  
Например: gr\_S\_y\_i – S<sub>y<sub>i</sub></sub>

В нормативных документах одна и та же переменная может использоваться для обозначения разных величин. В этом случае используются переменные со скрытой частью (той, которая не отражается в отчете). Например, в алгоритме можно

использовать переменные  $a$ ,  $a_1$ ,  $a_{xxx}$ . Во всех случаях в отчете переменная будет выглядеть как  $a$ . Скрытая часть отделяется тремя знаками подчеркивания.

Если в тексте отчета переменная должна быть заключена в квадратные скобки, например  $[F]$ , создается переменная, в имени которой квадратные скобки заменены на фигурные:  $\{F\}$ .

## Советы по разделению переменных на группы

- В группе, как правило, не должно быть более 20 переменных.
- В одной группе не должно быть смешения характеристик, которые не имеют смысловой связи.
- Переменные должны объединяться в группы с учетом возможности использования выбора значений из **Справочника**, например, все характеристики из сортамента объединяются в одну группу.

## Переименование группы переменных

Группы переменных можно переименовать, выполнив команду **Переименовать** из контекстного меню, предварительно выделив позицию в списке **Групп данных** на вкладке **Исходные данные** Отладчика.

## Изменение принадлежности переменной к группе

Ряды и ячейки в таблицах описания переменных можно перемещать вырезанием и копированием. Соответственно, при этом будет меняться их принадлежность к группам переменных.

## Числовые и текстовые переменные

Переменные, используемые в алгоритмах модулей NormCAD, могут принимать как числовые, так и текстовые значения. При присвоении переменной текстового значения текст заключается в кавычки.

## Использование массивов

Для массивов используются переменные с индексами. Например, опишем массив  $a_{ij}$  при  $i=1\div 3; j=3\div 4$ . Требуется ввести переменные:  $a_{13}, a_{14}, a_{23}, a_{24}, a_{33}, a_{34}$ .

К членам массива можно обращаться, используя выражения, заключенные символами процента, например:  $a_{\%i\%\%j\%}$  или  $a_{\%i+1\%\%j-1\%}$ .

При этом в индексах допускается использовать только следующие переменные:  $i, j, k, l, m$  и  $n$ , а также следующие арифметические выражения:  $i+1, i+2, i-1, i-2$  и подобные с переменными  $j, k, l, m$  и  $n$ .

Возможны массивы переменного размера. Для работы с ними используются функции **Array**, **SetValue**, **GetValue** и др. – для их вставки и просмотра описания см. в

окне **Функции**, вызываемом кнопкой . Массивы могут включать и текстовые значения, а также как числовые, так и текстовые значения в одном массиве. Работа с текстовыми массивами не отличается от работы с числовыми массивами.

## Размерности переменных

При задании размерности следует придерживаться выбранной системы единиц измерения. Обычно за основу принимаются системы СИ (Н, м, Па) или СГС (кгс, см, кгс/см<sup>2</sup>). Придерживаться выбранной системы требуется для того, чтобы не использовать в формулах дополнительные коэффициенты перевода из одной системы измерений в другую. При выборе системы измерений, как правило, принимается система измерений, используемая в нормативном документе.

Размерность переменных устанавливается в Отладчике во вкладках **Исходные данные** и **Переменные** в столбце “Ед.”.

Следует напомнить, что при вводе данных возможен выбор единиц измерения, в которых будут вводиться данные.

Данные при вводе преобразуются в соответствии с правилами, устанавливаемыми в диалоговом окне **Правила преобразования единиц измерения**, вызов которого осуществляется командой меню **Сервис \ Преобразование единиц измерения**.

## Задание переменным значения по умолчанию

Для того чтобы максимально уменьшить количество информации, вводимой пользователем, используются значения по умолчанию. Например, при расчете стальных конструкций в большинстве случаев принято, что  $E=2,06 \cdot 10^5$  МПа, а коэффициент Пуассона равен 0,3. Для задания таких значений создается документ для расчета, вводятся данные, которые будут служить значениями по умолчанию, выбирается команда **Сохранить данные по умолчанию** меню **Данные**.

При работе в Отладчике запоминаются введенные значения для всех переменных, которые они могут получать в процессе отладки. Для того чтобы оставить только те из них, которые будут приняты по умолчанию перед сохранением модуля, выполните команду **Удалить все значения, кроме данных по умолчанию** в меню **Данные** основной программы.

## 2.2 Выражения в NormCAD

### Ввод формул

При вводе формул для присваивания значений переменной используется удвоенный символ "равно".

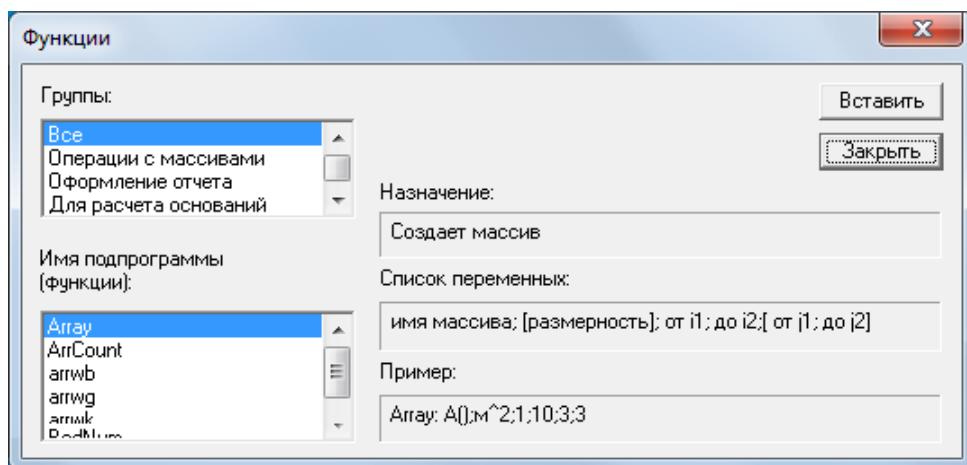
Например,  $y == x + 1$ .

В формулах используются арифметические операторы (+ - \* / ^) и наиболее распространенные функции:

SIN; COS; TAN; SINH; COSH; TANH; ASIN; ACOS; ATAN (основные тригонометрические и гиперболические функции); LN (натуральный логарифм); LOG10 (десятичный логарифм); EXP (экспоненциальная функция); ABS (модуль числа); SQRT (корень квадратный); MAX; MIN (максимальное и минимальное из списка чисел); INT (целая часть числа)

А также константы: pi= 3,14159265; TRUE - истина; FALSE - ложь.

Если требуемой функции нет среди встроенных, нужно использовать внешнюю функцию. Можно воспользоваться существующей внешней функцией из списка в окне **Функции**, вызываемом кнопкой .



Имеется возможность подключения функций, написанных в Visual Basic или другой среде разработки. Для этого создается проект ActiveX DLL, содержащий класс со свойством Instancing, имеющим значение MultiUse, и свойство, имеющее тип Public. Например, введем функцию *Summ3*, суммирующую значения трех переменных.

В проекте Visual Basic ТС создается общедоступный класс TestClass с методом *Summ3*:

```
Public Property Get Summ3(Condition As String, _
    Report As String, x1 As Single, x2 As Single, x3 As Single) As Single
    Summ3= x1+x2+x3
End Property
```

где Condition - список условий, разделенный уточненным символом ~ (используется как для передачи значения в функцию, так возврата значения); Report - возвращаемый текст отчета.

В NormCAD будет записано:

X == ImportFunction(TC.TestClass; **Summ3**; q;2,5;3)

		до 10 переменных или чисел
		общедоступный метод
		_____ библиотека.общедоступный класс

## **Нумерация формул**

Формуле можно присвоить ее номер в нормативном документе, если в конце строки алгоритма указать номер в квадратных скобках. Перед номером ставится символ #. Например: "[#1]". Этот номер будет использован при описании формулы в отчете.

## **Текстовые выражения**

Возможно объединение нескольких текстовых и числовых выражений с помощью символа &, например T == "Текст1 "&X&" Текст2 "&Y.

В текстовых выражениях употребляются следующие сочетания, используемые для форматирования текста отчета:

- \_tab\_ - символ табуляции,
- \_wrap\_ - перевод строки,
- \_hand\_ - ставится в начале заголовка.

Текстовые выражения могут использоваться:

- В формулах при присваивании значений переменным;
- В комментариях.

## **Комментарии в алгоритме**

Текст алгоритма состоит из комментариев и исполняемой части. Перед комментариями ставятся символы "///". Все комментарии автоматически включаются в отчет. Комментарии могут включать формулы. Если комментарии включают формулы, то текстовые части комментария должны заключаться в кавычки, а в начале комментария ставится "///=". Для объединения частей комментария используется символ &. Например, если N имеет значение 5, то комментарий

///="Расчет "&N&"-го слоя."

будет выглядеть в отчете, как

Расчет 5-го слоя.

## **2.3 Условия и другие управляющие конструкции**

### **Виды условий**

В NormCAD различаются текстовые и вычисляемые условия.

Например:

"Бетон – тяжелый." – текстовое условие;

A>=B+1 – вычисляемое.

## **Переход в зависимости от выполнения вычисляемого условия в алгоритме**

Переход в зависимости от выполнения вычисляемого условия аналогичен оператору if в Visual Basic. Если условие истинно - выполняются все операторы до конца условия. В конце условия ставятся символы \*. Например:

```
*q_sw<0,25*R_bt*b  
...  
*
```

### **Вложенные условия**

Условия могут быть вложенными. На уровень вложенности указывает количество символов \* в начале условия. Их количество должно соответствовать уровню вложенности. Например:

```
*q_sw<0,25*R_bt*b [MH/m]  
**Q>Q_b [MH]  
...  
**  
**Q<=Q_b [MH]  
Q_sw == 0  
**  
*
```

## **Переход в зависимости от выполнения текстового условия в алгоритме**

Переход в зависимости от выполнения условия в виде вопроса к пользователю выполняется аналогично предыдущему. Такие условия включают вопрос и предлагаемый вариант ответа через символ "-" (минус). В тексте алгоритма условия должны заканчиваться вопросительным знаком.

Текстовые условия также могут быть вложенными. Например:

```
*Бетон - тяжелый?  
...  
*Бетон - легкий?  
**Марка по средней плотности-D1900 и более?  
...  
**Марка по средней плотности-D1800 и менее?  
...  
**  
*
```

В текстовых условиях допускается использовать формулы. При этом текстовые части комментария должны заключаться в кавычки, а в начале комментария ставится "`=`". Для объединения частей комментария используется символ `&`. Пример: `=/*Грунты слоя "&i - песчаные?`

## Обработка сложных условий

В алгоритме могут сочетаться текстовые и вычисляемые условия. Например:

```
*Тип стержня - двуветвевой стержень?  
**Тип составного стержня - составной стержень, ветви  
которого соединены планками?  
***1/n<5  
...  
***  
***1/n>=5  
...  
***  
**Тип составного стержня - составной стержень, ветви которого  
соединены решетками?  
...
```

Несколько условий, использующих формулы, можно объединить логическими операторами `AND(<условие1>; <условие2>; ...)` и `NOT(AND(...))`.

## Циклы

Циклы организуются с помощью оператора `cicle`, обеспечивающего возврат на начало условия.

Например, требуется выполнить некоторые действия при переменной `i`, принимающей значения от 1 до 10:

```
i==1  
*i<=10  
... <тело цикла>  
i==i+1  
  
*cicle
```

## Проверка условий

Проверка условия выполняется с помощью оператора `"TEST:"`. После оператора `"TEST:"` должно следовать вычисляемое условие. При невыполнении условия расчет прекращается, при выполнении продолжается. В отчете это отражается соответствующей записью. Например: `TEST: k > 0.`

## **Переход к пунктам**

Оператор "`-->`" выполняет переход к алгоритму пункта. Например, для перехода к пункту 3.2 в алгоритме требуется строка "`-->3.2`".

Этот оператор является аналогом оператора вызова подпрограммы. После перехода и выполнения алгоритма вызываемого пункта происходит возврат к месту обращения.

## **Об указании единиц измерения в алгоритме**

В формульных условиях (за исключением случая, когда результатом выражения является безразмерная величина) требуется указать размерность, которая ставится после условия в квадратных скобках, например:

`*Q>Q_b [MH]`

Размерность в условиях указывается, так как в сложных условиях программа не всегда может определить ее автоматически.

В обычных формулах (когда некоторой переменной присваивается значение) указывать размерность не требуется, так как она устанавливается программой автоматически по размерности переменной, которой присваивается результат.

## **2.4 Оформление отчета, вывод результатов**

### **Заголовки в отчете**

Обычно в отчете заголовки создаются и нумеруются автоматически. Текст заголовка берется по первой строке пункта (если первая строка пункта не содержит текста, заголовок создается автоматически). Перед заголовком ставится номер, отделяющийся скобкой, т.е. 1), 2) и т.д.

Альтернативой является создание заголовков вручную, причем если хотя бы один заголовок создан вручную, другие заголовки автоматически не создаются и не нумеруются.

Для того чтобы создать заголовок вручную, добавляется строка комментария, в начале которой ставится префикс `_hand_`, например:

`///_hand_ мой заголовок;`  
`или ///_hand_1) мой заголовок,`

при этом заголовок может включать и выражения, например:

`///=_hand_ "&N&") Расчет слоя"&N,`

где N – переменная, принимающая значения номера слоя в цикле. Символ & используется для объединения нескольких строк или выражений.

В большом отчете (если он содержит больше 10-20 страниц) целесообразно снабжать его разделы заголовками, созданными вручную.

## **Вставка рисунков в отчет**

На данный момент имеется следующие способы вставки рисунков в отчет при его генерации:

1. в текстовых условиях, например:

\*Тип сечения – 1 <pict=1>

где <pict=1> указывает на рисунок 1.bmp в соответствующей папке из каталога Drawings

2. из нестандартных диалоговых окон для ввода данных. Область рисунка ограничивается скрытым элементом управления HidePicture.

## **Вопросы к пользователю с использованием картинок или схем, иллюстрирующие варианты ответа**

Если для ответа на вопрос пользователю желательно дать некоторую графическую информацию (иллюстрации для вариантов ответа), используется выражение <pict=...>, которое ставится после вопроса в условии, и вместо многоточия указывается имя файла картинки, расположенного в папке ... Program Files\NormCAD\Drawings.

## **Способы настройки отражения формул в отчете**

Метод программирования, используемый в NormCAD, отличается от других языков программирования тем, что все строки алгоритма автоматически получают отражение в отчете. В некоторых случаях это не требуется, например, при увеличении индекса в цикле и других случаях, когда выполняемые действия не столь существенны для их описания в отчете. Для того чтобы строка алгоритма не отражалась в отчете, следует в конце строки написать <Hide>.

В NormCAD при создании отчета имеется режим подготовки сокращенного отчета (устанавливается кнопкой **Ускоренный отчет**). В этом режиме в текст отчета вставляются только результаты проверок (т.е. когда используется оператор TEST:), а также отчет по строкам алгоритма, отмеченные в конце строки отметкой <+>. Такой отметкой можно выделять наиболее важные места алгоритма, например, строки, в которых вычисляется площадь требуемой арматуры.

## **Вывод результатов в виде таблиц**

Необходимость в выводе результатов в табличном виде возникает, когда в одном отчете описаны несколько небольших однотипных расчетов. Тогда в таблицу сводятся основные результаты по каждому из них.

В документ можно добавить одну и более таблиц. Таблицы будут располагаться во вкладках “Таблица 1”, “Таблица 2” и т.д. В качестве примера можно использовать модуль по СНиП “Нагрузки и воздействия”.

Для вставки таблиц используются внешние функции:

- Caption для описания заголовка таблицы, например:

```
i == ImportFunction(NCTable.Functions;Caption;"Снеговая нагрузка,  
кПа";1).
```

где 1 – номер таблицы для данного расчета (вкладка “Таблица 1”),

- Cols для описания заголовков колонок, например:

```
i == ImportFunction(NCTable.Functions;Cols; "NN|Вариант  
нагрузки|Зона|gr_m|Нормативная нагрузка|Расчетная нагрузка";1)
```

- ToTable для описания текста таблицы (вводятся значения ячеек, разделенные знаками табуляции \_tab\_ и перевода на другую строку \_wrap\_), например:

```
i == ImportFunction(NCTable.Functions;ToTable;TextTable;1) .
```

### **3. Разработка интерфейса взаимодействия с пользователем**

#### **Разработка диалоговых окон для ввода данных**

На заключительной стадии автоматизации нормативного документа (т.е. после ввода алгоритма и распределения переменных по группам) можно упростить взаимодействие пользователя с программой разработкой специальных диалоговых окон для ввода исходных данных, хотя это не является обязательным, и при небольшом количестве групп данных допустимо ограничиться стандартными диалоговыми окнами, создаваемыми автоматически.

Для вставки диалогового окна выберите группу данных на вкладке **Окна отладчика** и, после нажатия правой клавиши мыши, выберите команду **Вставить окно**. После этого в Visual Basic (если он установлен) будет создан и открыт новый проект с единственной формой, в которую нужно вставить элементы управления NCTextBox и NCLtTextBox для каждой переменной из группы данных. Заполните в них параметры: Var – имя переменной, Measure – ее размерность.

Так же можно вставить в форму и другие элементы управления, в т.ч. и входящие в комплект NormCAD: DataGroupBox – для вызова диалогового окна справочника; DataGroupButton – для вызова стандартного диалогового окна. После вставки элементов управления DataGroupBox или DataGroupButton необходимо заполнить свойства ResourceCaption (оно должно принимать значение свойства Caption элементов управления DataGroupBox или DataGroupButton) и ResourceVar (имя соответствующей переменной в вызываемом окне) для элементов управления NCTextBox и NCLtTextBox.

Для вставки в отчет рисунков из нестандартных диалоговых окон область рисунка ограничивается скрытым элементом управления HidePicture.

С многочисленными примерами нестандартных диалоговых окон можно ознакомиться, открыв в Отладчике имеющиеся модули нормативных документов и воспользовавшись командой **Редактировать в Visual Basic** контекстного меню вкладки **Окна**.

#### **Вставка окна справочника**

Для любой группы данных может быть автоматически создан справочник и соответствующее окно для выбора значений из этого справочника. Справочники используют для ввода данных ниспадающие списки, из которых выбираются значения (например, для характеристик сечения - марка или наименование и номер ТУ или ГОСТ).

В справочниках хранится информация для различных групп исходных данных, имеющая простейший структурный вид – разделение на группы и подгруппы. В таком виде удобно хранить данные из сортаментов и других подобных таблиц.

Для вставки **Справочника** выберите группу данных на вкладке **Окна отладчика** и, после нажатия правой клавиши мыши, выберите команду **Добавить Справочник**.

Физически справочники хранятся в файле баз данных Access в папке ...\\NormCAD\\Dat\\Tables для добавления справочника. В таких файлах имеется по две таблицы. В таблице Main описываются группы записей (например, перечисляются различные типы профилей). В таблице Data – записи для каждой подгруппы данных (например, для каждой марки профиля).

Для редактирования справочника выберите команду **Справочники** основной программы.

### **Примечание к позиции справочника**

Для того чтобы добавить примечание к позиции **Справочника**, служит столбец “**Note**”. Его можно редактировать в редакторе **Справочников** NormCAD (команда **Справочники** меню **Сервис** основной программы) или в MS Access.

### **Иллюстрирование справочника рисунками**

Группы записей в справочнике можно проиллюстрировать рисунком в формате bmp. Расположение рисунка указывается в столбце “Picture” справочника. Например, это используется при выборе типа профиля, когда пользователем задаются характеристики сечения (группа данных “Основные характеристики сечений” при расчете по СНиП “Стальные конструкции”).

### **Вопросы к пользователю с использованием картинок или схем, иллюстрирующие варианты ответа**

Если для ответа на вопрос пользователю желательно дать некоторую графическую информацию (иллюстрации для вариантов ответа), используется выражение <pict=...>, которое ставится после вопроса в условии, и вместо многоточия указывается имя файла картинки, расположенного в папке ...Program Files\\NormCAD\\Drawings.

### **Задание переменным значения по умолчанию**

Для того чтобы максимально уменьшить количество информации, вводимой пользователем, используются значения по умолчанию.

Например, при расчете стальных конструкций в большинстве случаев принято, что модуль упругости стали Е равен  $2,06 \cdot 10^5$  МПа, а коэффициент Пуассона равен 0,3.

Для задания таких значений создается документ для расчета, вводятся данные, которые будут служить значениями по умолчанию, выбирается команда **Сохранить данные по умолчанию** меню **Данные**.

## 4. Примеры программирования в среде NormCAD

### 4.1 Пример ввода алгоритмов в NormCAD

Работу в Отладчике удобно продемонстрировать на примере автоматизации в NormCAD реального нормативного документа с достаточно простой расчетной частью - СН 497-77 "Временная инструкция по проектированию, монтажу и эксплуатации воздухоопорных пневматических сооружений". Приведем текст основных расчетных пунктов СН 497-77 (с незначительными сокращениями):

2.1. Материалы для оболочек ВПС и их основные характеристики приведены в приложении 1.

2.2. Расчетные сопротивления  $R$  материалов оболочки приведены в табл.1.

Таблица 1

Шифр материала	Расчетное сопротивление $R$ , кгс/м		Шифр материала	Расчетное сопротивление $R$ , кгс/м	
	по основе $R_{oc}$	по утку $R_{yt}$		по основе $R_{oc}$	по утку $R_{yt}$
51-019	2350	1050	23-М	2350	1050
У-92	2350	1050	А-01	2250	2250
У-93	1280	830	Тентовый материал	1690	1250

3.1. Оболочки ВПС проектируются сферическими и цилиндрическими. Цилиндрические оболочки могут быть со сферическими или цилиндрическими торцами (рис.1).

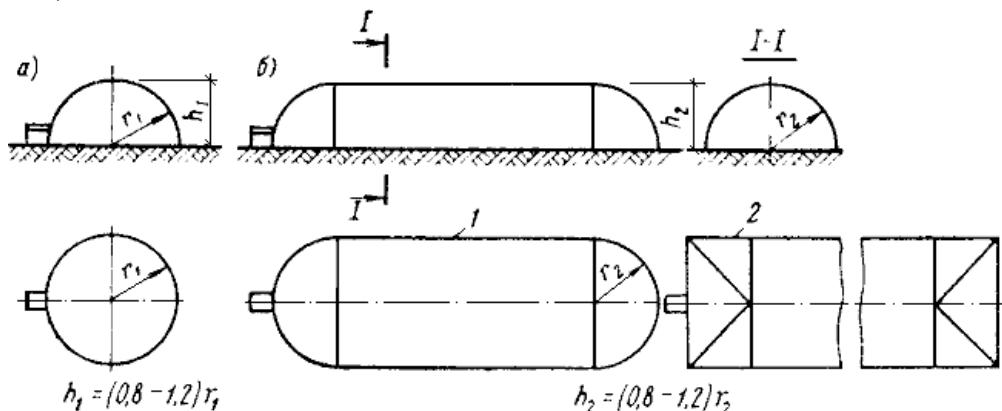


Рис. 1. Схемы оболочек ВПС

а - сферическая; б - цилиндрическая; 1 - план оболочки со сферическими торцами; 2 - то же, с цилиндрическими торцами.

3.2. Прочность оболочек проверяется по формулам:

$$T_{1,2} \leq R_{oc} \text{ или } T_{1,2} \leq R_{yt} \quad (1)$$

где  $T_1, T_2$  - усилия растяжения в оболочке (рис.2), кгс/м;

$R_{oc}$ ,  $R_{yt}$  - расчетные сопротивления материала оболочки по основе или утку, кгс/м.

Деформации оболочек не проверяются.

3.3. Усилия  $T_{1,2}$  определяются от давления воздуха внутри оболочки и внешних нагрузок от снега и ветра. Собственный вес оболочки не учитывается.

3.4. Усилия  $T_1^c$  и  $T_2^c$  (рис.2,а) в сферических оболочках определяются по формулам:

$$T_1^c = (0,5P + q_0nk)r_1; \quad (2)$$

$$T_2^c = (0,5P + q_0nk + 0,33P_c)r_1; \quad (3)$$

Усилия  $T_1^n$  и  $T_2^n$  (рис.2,б) в цилиндрических оболочках определяются по формулам:

$$T_1^n = (P + 1,15 q_0nk)r_2; \quad (4)$$

$$T_2^n = (0,5P + q_0nk + 0,33P_c)r_2; \quad (5)$$

где  $P$  - величина давления воздуха внутри оболочки, определяемая по формуле

$$P = 0,8q_0, \text{ но не менее } 40 \text{ кгс/м}^2;$$

$q_0$  - скоростной напор ветра, определяемый по главе СНиП по нагрузкам и воздействиям, кгс/м<sup>2</sup>;

$r_1$  - радиус сферической оболочки, м;

$r_2$  - радиус цилиндрической оболочки, м;

$n$  - коэффициент перегрузки 1,3;

$k$  - коэффициент, учитывающий изменение скоростного напора в зависимости от высоты и типа местности, принимаемый по главе СНиП по нагрузкам и воздействиям;

$P_c$  - расчетная величина снеговой нагрузки, определяемая по табл. 2.

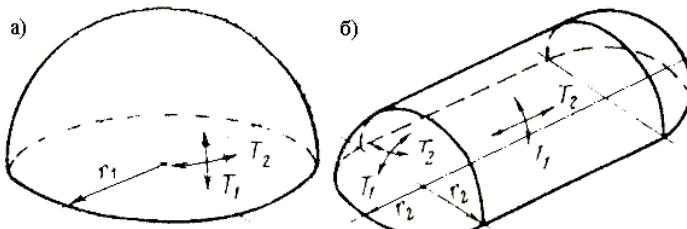


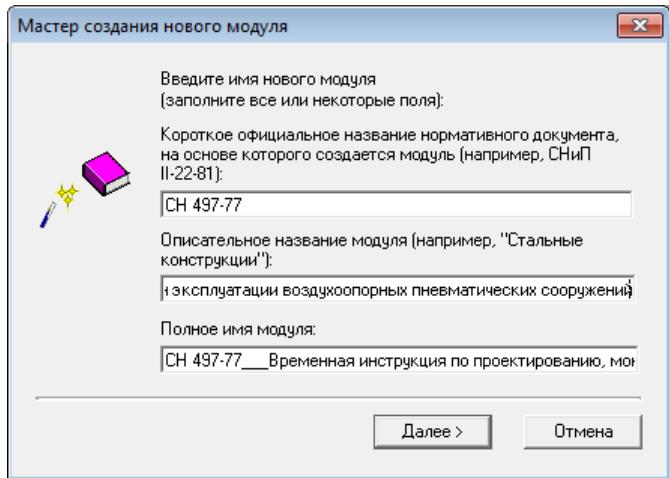
Рис. 2. Схемы действия усилий в оболочке  
а - сферической; б - цилиндрической

Таблица 2

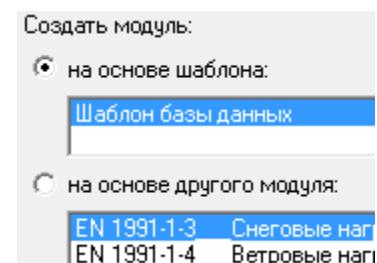
Районы СССР (принимаемые по главе СНиП по нагрузкам и воздействиям)	Расчетная величина снеговой нагрузки $P_c$ , кгс/м <sup>2</sup>
I	7
II	11
III	14
IV	22
V	25
VI	28

1. Для создания нового модуля воспользуемся Мастером создания модулей (команда **Создать модуль** в меню **Сервис** основной программы или кнопка **Мастер создания модулей** в окне Менеджера).

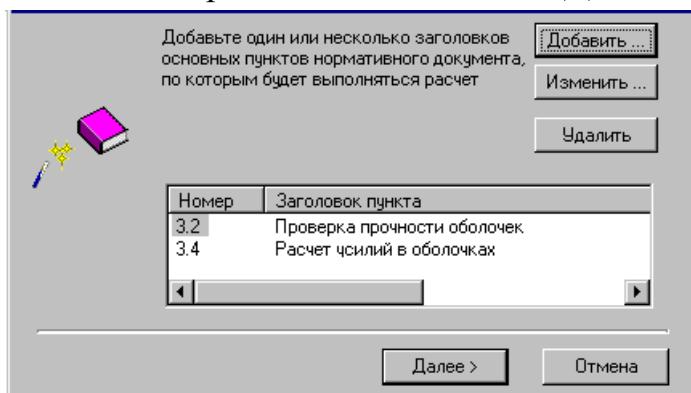
2. В первом окне Мастера вводится название создаваемого модуля:



3. В следующем окне Мастера выбираем: создается ли модуль на основе шаблона или на основе другого существующего модуля:



4. В следующем окне Мастера с помощью кнопки **Добавить**



вставим в модуль два новых пункта 3.2 и 3.4 (заметим, что добавлять и удалять пункты можно и в процессе редактирования модуля в Отладчике).

5. В последнем окне Мастера согласимся с тем, что после создания модуль будет открыт для редактирования, и нажмем кнопку **Создать**. После этого созданный модуль будет открыт в Отладчике.

6. Выберем на панели в левой части окна пункт 3.2 и введем его алгоритм.

```
/// Проверка прочности оболочек
R_oc == tbt(1;;"R_oc")
R_ut == tbt(1;;"R_ut")
TEST: T_1<=R_oc [кгс/м] [# 1]
TEST: T_1<=R_ut [кгс/м] [# 1]
TEST: T_2<=R_oc [кгс/м] [# 1]
TEST: T_2<=R_ut [кгс/м] [# 1]

/// Проверка прочности оболочек
R_oc =tbt(1;;"R_oc")
R_ut =tbt(1;;"R_ut")
TEST:T1≤Roc[kgs/m][ 1]
TEST:T1≤Rut[kgs/m][ 1]
TEST:T2≤Roc[kgs/m][ 1]
TEST:T2≤Rut[kgs/m][ 1]
```

Текст алгоритма вводится в верхней правой части окна. Ниже автоматически выводятся соответствующие формулы в отформатированном виде с индексами. Иногда удобно скрыть нижнюю правую часть окна. Для этого нажмите кнопку  . При вводе текста алгоритма можно использовать специальные панели инструментов (см. п. 11.2). Для того чтобы скрыть или показать основные плавающие панели инструментов, служат кнопки .

Прокомментируем введенный алгоритм:

```
// Проверка прочности оболочек
```

```
R_oc == tbl(1;"R_oc")
R_ut == tbl(1;"R_ut")
TEST: T_1<=R_oc [кгс/м] [# 1]
TEST: T_1<=R_ut [кгс/м] [# 1]
TEST: T_2<=R_oc [кгс/м] [# 1]
TEST: T_2<=R_ut [кгс/м] [# 1]
```

В первой строке приводится закомментированный текст, который будет использоваться программой при автоматической расстановке заголовков в тексте отчета.

В данном пункте выполняются проверки материала оболочки по прочности:

```
TEST: T_1<=R_oc [кгс/м] [# 1]
```

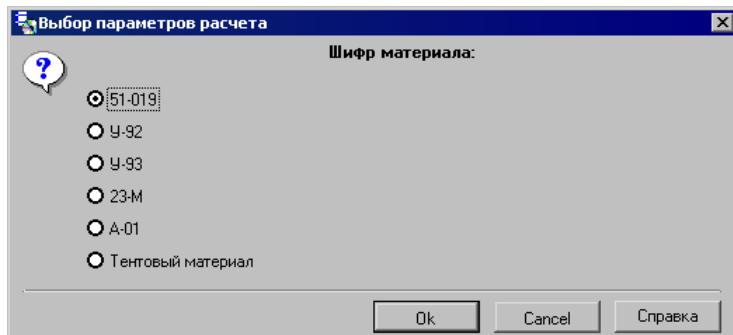
Проверка условия выполняется с помощью оператора "TEST:". При выполнении условия расчет продолжается, при невыполнении прекращается, и в конце отчета приводится сообщение о невыполнении проверки.

В конце строки алгоритма приведена размерность сравниваемых величин [кгс/м] и номер формулы [# 1].

Перед строками проверок из табл. 1 извлекаются значения прочностных характеристик материала:

```
R_oc == tbl(1;"R_oc")
R_ut == tbl(1;"R_ut")
```

Для извлечения табличных данных используется функция `tbl`. В скобках приводятся аргументы функции через точку с запятой. Первый из них: 1 – номер таблицы. Второй аргумент пропущен, следовательно, номер ряда в таблице уточняется вопросом к пользователю:



В третьем аргументе уточняется, из какой колонки таблицы берется значение.

В данном случае табл. 1 небольшая и не имеет разделения материалов на группы и подгруппы. Поэтому достаточно применить функцию `tbl`. Как правило, данные о характеристиках материалов имеют более сложную структуру, и для их ввода применяются справочники или нестандартные диалоговые окна.

7. Затем введем алгоритм п. 3.4:

```
/// Расчет усилий в оболочках
n == 1,3
q_o == tbl(5 СНиП 2.01.07;;)
k == tbl(6 СНиП 2.01.07;h;)
```

С помощью функции `tbl` для промежуточных значений  $h$  значения извлекаются из табл. 6 с линейной интерполяцией.

```
P == 0,8*q_o
*P<40 [кгс/м2]
P == 40
*
P_c == tbl(2;;)
*Tип оболочки - сферическая? <pict = сферическая>
/// Проверка условий по рис.1:
TEST:  $h \geq 0,8 * r_1$  [м]
TEST:  $h \leq 1,2 * r_1$  [м]
T_1 ==  $(0,5 * P + q_o * n * k) * r_1$  [# 2]
T_2 ==  $(0,5 * P + q_o * n * k + 0,33 * P_c) * r_1$  [# 3]
*Tип оболочки - цилиндрическая? <pict = цилиндрическая>
/// Проверка условий по рис.1:
TEST:  $h \geq 0,8 * r_2$  [м]
TEST:  $h \leq 1,2 * r_2$  [м]
T_1 ==  $(P + 1,15 * q_o * n * k) * r_2$  [# 4]
T_2 ==  $(0,5 * P + q_o * n * k + 0,33 * P_c) * r_2$  [# 5]
```

\*

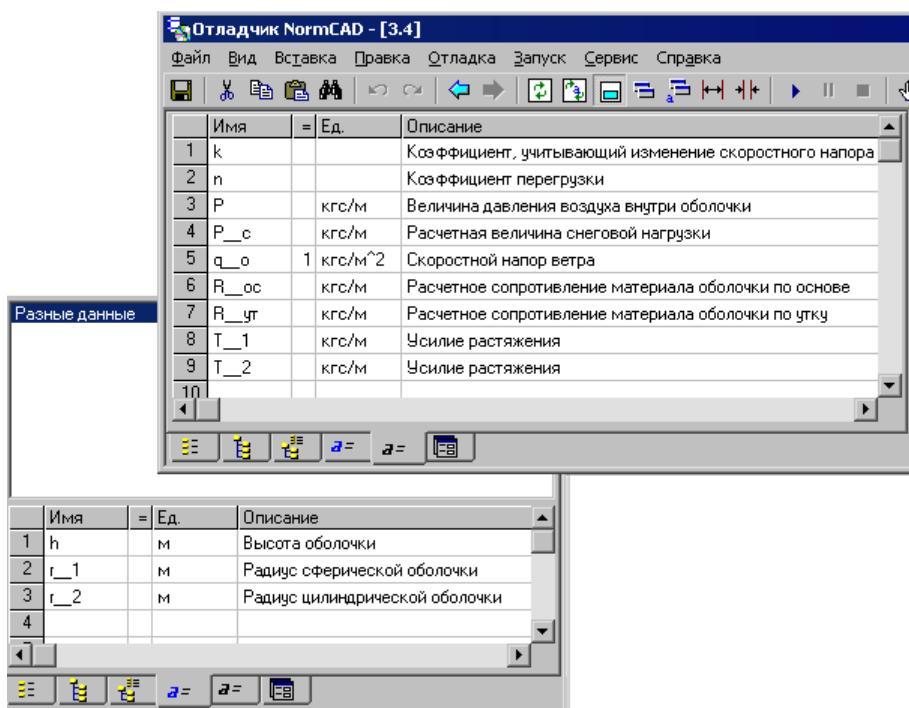
--> 3.2

В последней строке выполняется переход к п. 3.2, алгоритм которого приведен выше.

Для иллюстрации выбора типа оболочки потребуются файлы рисунков сферическая.bmp и цилиндрическая.bmp. Они должны располагаться в папке ...\\Program Files\\NormCAD\\Drawings\\CH 497-77 Временная инструкция по проектированию ... воздухоопорных пневматических сооружений\\3.4.

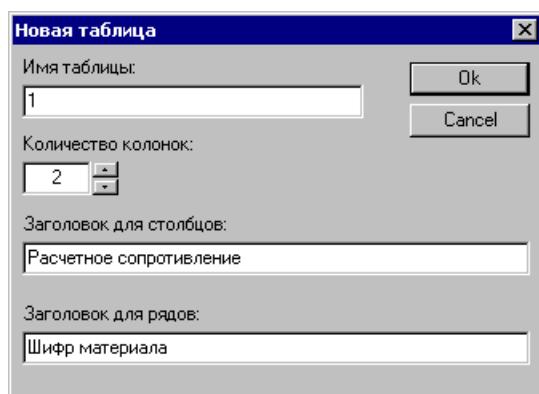
8. Нажмем кнопку **Автоописание переменных** 

9. Затем введем описание и размерности переменных во вкладках **Переменные** и **Исходные данные**.



10. В процессе выполнения команды **Автоописание переменных** была создана единственная группа данных **Разные данные**. Можно дать ей более подходящее название: "Геометрия оболочки". Для этого выделим название группы данных и воспользуемся командой контекстного меню **Переименовать**.

11. Вставим таблицу 1 (команда **Таблица** в меню **Вставка**).



12. Перейдем к табл. 1 и заполним ее клетки значениями:

	A	B	C
1	Шифр материала	Расчетное сопротивление	
2		R_ос	R_ут
3	51-019	2350	1050
4	У-92	2350	1050
5	У-93	1280	830
6	23-М	2350	1050
7	А-01	2250	2250
8	Тентовый	1690	1250

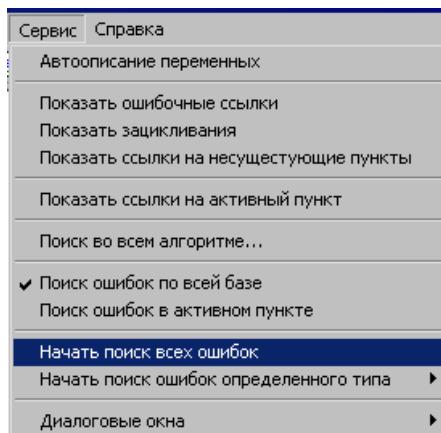
13. Вставим и заполним таблицу 2:

	A	B
1	Снеговой район	
2		=TRUE
3	I	7
4	II	11
5	III	14
6	IV	22
7	V	25
8	VI	28

14. Затем введем табл. 5 и 6 из СНиП 2.01.07 "Нагрузки и воздействия"

	A	B	C	D	
1			Тип местности		
2	z		A	B	C
3	0	0,75	0,5	0,4	
4	5	0,75	0,5	0,4	
5	10	1	0,65	0,4	
6	20	1,25	0,85	0,55	
7	40	1,5	1,1	0,8	
8	60	1,7	1,3	1	
9	80	1,85	1,45	1,15	
10	100	2	1,6	1,25	
11	150	2,25	1,9	1,55	
12	200	2,45	2,1	1,8	
13	250	2,65	2,3	2	
14	300	2,75	2,5	2,2	
15	350	2,75	2,75	2,35	

15. Выполним поиск ошибок в модуле введенного документа.



Предварительно выберем пункт **Поиск ошибок везде** в меню **Сервис** Отладчика. Затем выполним команду **Начать поиск всех ошибок**. Получаем сообщение "Ошибки не найдены".

На этом можно и закончить автоматизацию документа СН 497-77. Модуль нормативного документа практически готов к использованию. Для этого необходимо только перезагрузить NormCAD.

После перезагрузки можно создать задания для расчета:

1. Расчет сферических оболочек
2. Расчет цилиндрических оболочек

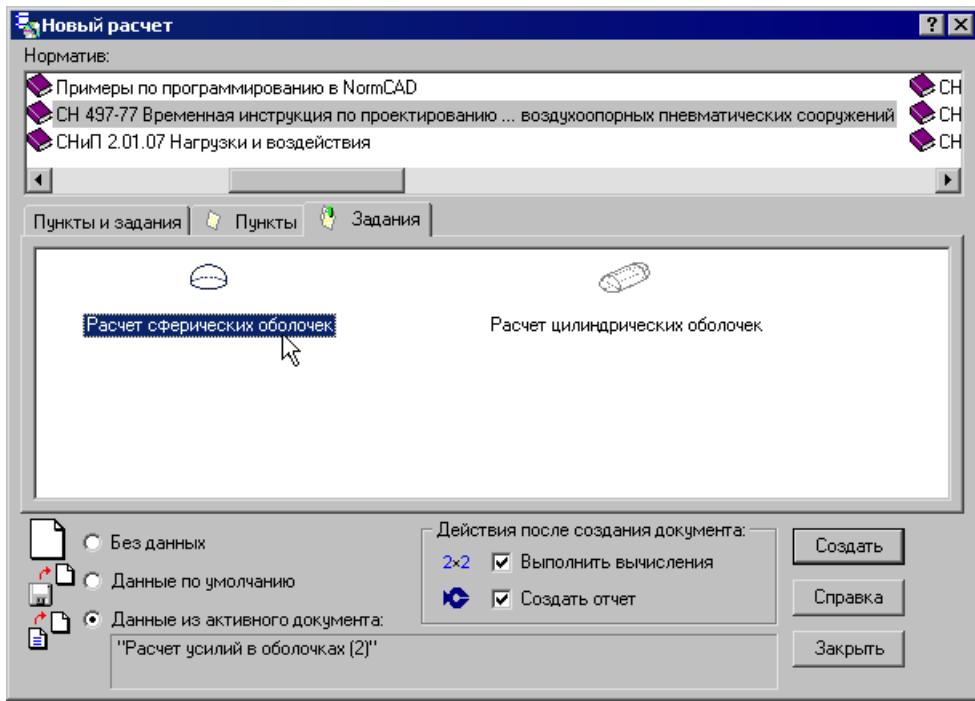
Для этого выполним следующее:

1. Командой **Новый расчет** создадим два документа и выполним расчет по п. 3.4 СН 497-77. В первом из них выберем условие "Тип оболочки - сферическая", во втором – "Тип оболочки - цилиндрическая".
2. Выберите команду **Создать** в подменю **Задания** меню **Сервис**.
3. В появившемся диалоговом окне во вкладке **Сведения** введите имя задания.
4. Во вкладке **Пункты** выберем требуемый норматив СН 497-77 ...; отметим требуемые пункты. В данном случае это один пункт – 3.4.
5. Во вкладке **Условия** выберем требуемые условия (предварительно в списке в верхней части окна выбирается документ, из которого извлекаются условия).
6. Во вкладке **Значок** указываем на файл с рисунком значка задания. Эти файлы должны иметь расширение bmp и размер 32x32 точек.

Подобные значки создаются и для групп данных. Они должны располагаться в папке ...\\Program Files\\NormCAD\\Dat\\Images\\Groups.

7. Нажмите кнопки **Сохранить** и **Закрыть**.

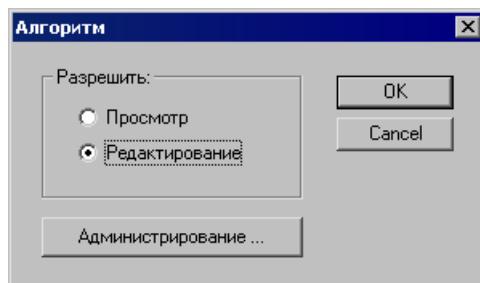
Задания запускаются на расчет двойным щелчком в окне **Новый расчет**.



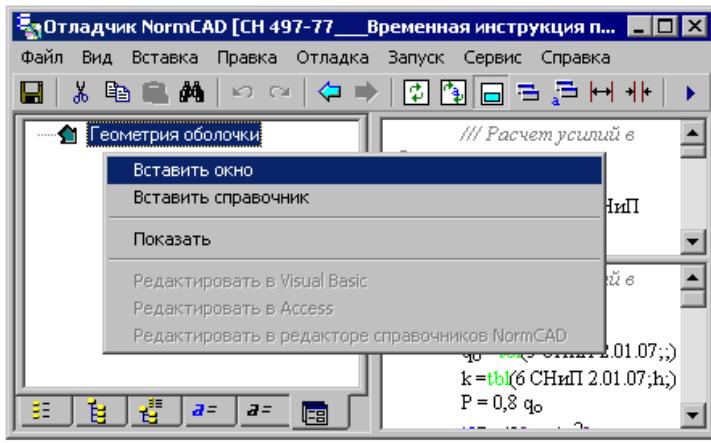
## 4.2 Пример разработки визуального интерфейса в NormCAD. Вставка диалогового окна.

Далее для ознакомления с возможностями разработки визуального интерфейса приложения создадим специальное диалоговое окно для группы данных "Геометрия оболочки".

Нажмем на кнопку Алгоритм и откроем его для редактирования:



Затем в окне Отладчика на вкладке **Окна** выделим значок группы данных "Геометрия оболочки" и выполним команду контекстного меню **Вставить окно**.

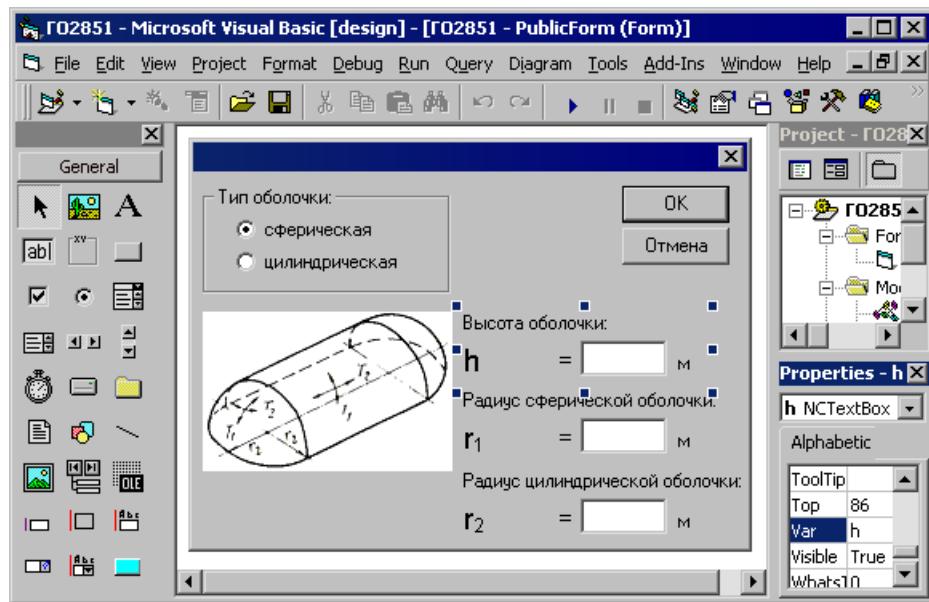


В результате в Visual Basic (если он установлен) будет создан и открыт новый проект ActiveX DLL с единственной формой, в которую нужно вставить элементы управления NCTextBox и NCLtTextBox для каждой переменной из группы данных. Заполните в них параметры: Var – имя переменной ( $h$ ,  $r_1$  и  $r_2$ ), Measure – ее размерность (м). Заметим, что переменные, получающие значения в форме, необязательно должны принадлежать к одной группе данных. Добавим в форму все необходимые компоненты и скомпилируем проект. Для привязки окна к модулю вносится соответствующая запись в файл Form.ini в рабочем каталоге NormCAD.

Структура записи в файле Form.ini:

[<имя файла модуля без расширения>]

<имя группы данных>=<имя проекта ActiveX DLL>



### 4.3 Пример вставки справочника

СН 497-77 разработан около 40 лет назад. За это время появились новые материалы, данные о которых потребуется добавить в наше приложение. Т.к. они не описаны в СН 497-77, назовем их пользовательскими. Таким образом,

введенные материалы разделяются на две группы. При такой структуре данных удобно создать для них справочник.

Вырежем две строки с описанием переменных R\_oc и R\_ut из таблицы на вкладке **Переменные**.

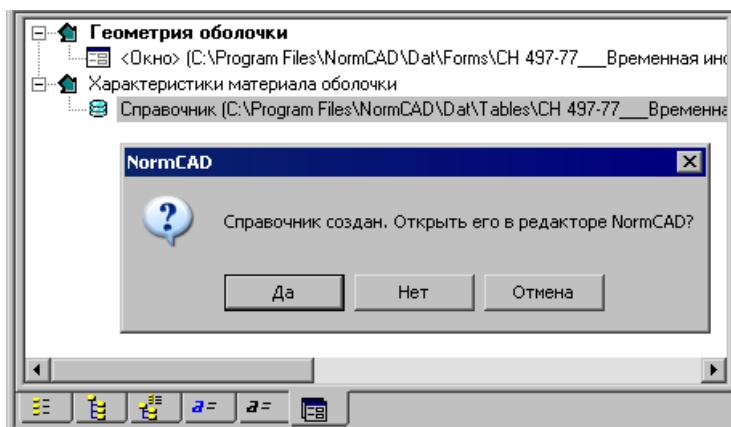
В меню **Вставка** выберем пункт **Группа данных**.

В ответ на запрос присвоим ей имя "Характеристики материала оболочки".

Вставим вырезанные строки с описанием переменных.

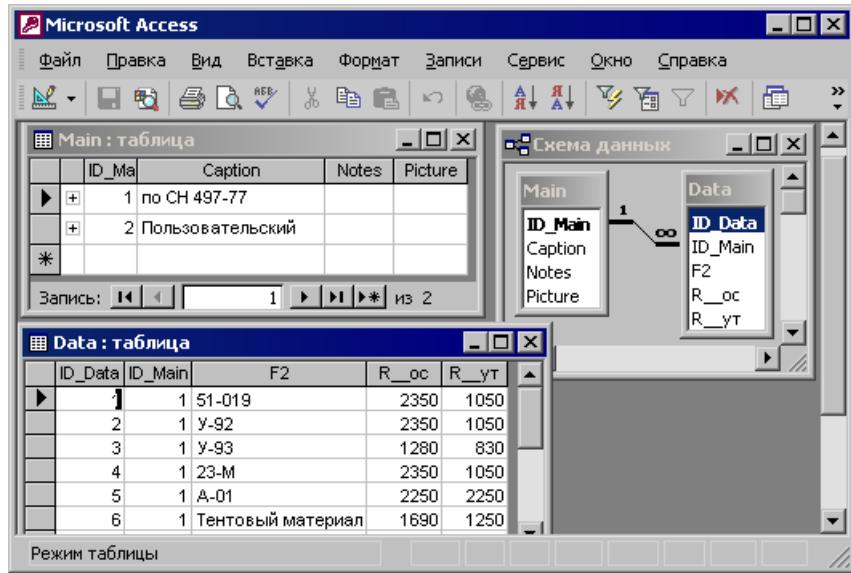
Геометрия оболочки		
Характеристики материала оболочки		
Имя	=	Ед.
1	R_oc	кгс/м
2	R_ut	кгс/м
3		

После выполнения команды **Обновить структуру элементов** меню **Вид** в окне Отладчика на вкладке **Окна** выделим значок группы данных "Характеристики материала оболочки" и выполним команду контекстного меню **Вставить справочник**.

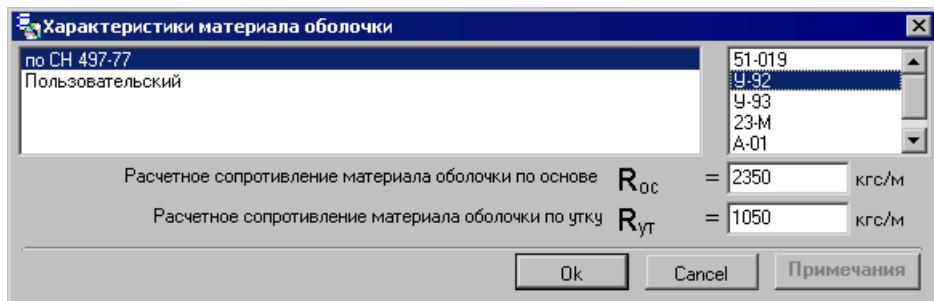


Ответим на предложение открыть справочник в редакторе NormCAD отрицательно и согласимся со следующим предложением открыть его в Access.

Справочник является файлом базы данных Access и включает две таблицы. В таблице Main описываются группы записей (например, перечисляются различные типы оболочек). В таблице Data – записи для каждой подгруппы данных (например, для каждой марки материала).



Так будет выглядеть окно Справочника. Оно автоматически создается при обращении к группе данных "Характеристики материала оболочки".



Т.к. данные о прочностных характеристиках материала теперь не требуется извлекать из табл. 1 в алгоритме, не забудем удалить из него соответствующие строки из п. 3.4:

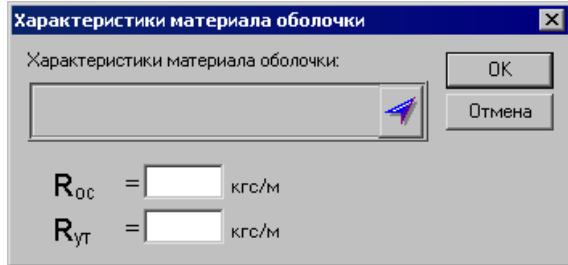
```

R_oc == tbl(1;"R_oc")
R_ut == tbl(1;"R_ut")
  
```

#### 4.4 Пример вставки диалогового окна, связанного со справочником

В двух предыдущих пунктах мы рассмотрели примеры вставки диалоговых окон произвольного вида и справочников для извлечения сгруппированных данных из баз данных Access. Покажем, как извлекать информацию из справочников в диалоговых окнах произвольного вида.

В нашем примере создавать такое окно не целесообразно, т.к. пользовательский интерфейс при этом не получает никаких улучшений. Добавим его в учебных целях. Для этого в окне Отладчика на вкладке **Окна** выделим значок группы данных "Характеристики материала оболочки" и выполним команду контекстного меню **Вставить окно**.



В верхний левый угол формы вставим элемент управления DataGroupBox и два элемента управления NCLtTextBox для переменных R\_oc и R\_ut. Присвоим следующие значения свойств элементов управления:

Элемент управления		Свойство	Значение свойства	Примечание
Имя (Name)	Тип			
R_oc	NCLtTextB ox	Var	R_oc	
		ResourceVar	R_oc	
		ResourceCaption	Характеристики материала оболочки	= свойству Caption DataGroupBox
		Measure	кгс/м	
		SendData	True	
R_yc	NCLtTextB ox	Var	R_yc	
		ResourceVar	R_yc	
		ResourceCaption	Характеристики материала оболочки	= свойству Caption DataGroupBox
		Measure	кгс/м	
		SendData	True	
DataGroup -Box1	DataGroup -Box	Base	Характеристики материала оболочки	имя файла базы данных Access
		Book	СН 497- 77 Временная инструкция по проектированию ... воздухоопорных пневматических сооружений	имя файла модуля нормативного документа
		Caption	Характеристики материала оболочки	текст надписи
		DataGroup	Характеристики материала оболочки	имя группы данных

## 5. Использование алгоритмов NormCAD

### 5.1 Экспорт алгоритмов модулей NormCAD

Одной из наиболее интересных возможностей NormCAD является получение автоматически создаваемого кода на Visual Basic или VBA (VBA поддерживают программы входящие в состав MS Office: MS Word, MS Excel и др., а также программы для подготовки чертежей AutoCAD, ZwCAD и др.), который может быть напрямую использован в различных независимых приложениях. Для использования автоматически создаваемого кода достаточно его минимальной переработки. Он может быть использован для автоматизации большого количества задач, включая сложные и комплексные задачи: обработка таблиц усилий, подбор сечений элементов, подбор армирования, поиск оптимальных решений, подготовка альбомов чертежей типовых конструкций, для задач с одновременным использованием нескольких нормативных документов (например, включающих сбор нагрузок, расчет усилий, проверки или подбор сечений).

Примеры использования кода см. в папке ... Program Files \ NormCAD \ Results \ Примеры \ VB-код на основе NormCAD.

### 5.2 Пример расчета на основе NormCAD с использованием VBA

Для любого расчета в NormCAD можно автоматически создать код на Visual Basic или VBA. Рассмотрим пример расчета на основе NormCAD с использованием VBA.

1. В программе NormCAD создаем новый документ и запускаем его на расчет. Для этого выбираем в окне **Новый расчет** норматив СП 52-101 "Бетонные и железобетонные конструкции", задание "Расчет плит по прочности".



2. При выполнении расчета вводим все запрашиваемые данные. Автоматически создается отчет.
3. В программе NormCAD нажимаем кнопку **Создать VB модуль**  и сохраняем файл модуля.
4. Открываем программу MS Excel, при этом создается новая электронная таблица.
5. В MS Excel входим в редактор VBA. Для этого можно нажать клавиши Alt+F11.
6. В редакторе VBA добавляем сохраненный в п. 3 модуль командой меню File\Import File.
7. Основу модуля составляет функция NCResult:

```

(General) ▾ (Declarations)

Option Explicit
Public Function NCResult() As Single
    On Error GoTo 1
    Dim Vars As Object
    Dim Conds As Object
    Set Vars = CreateObject("NC_167258518177598E02.Vars")
    Set Conds = Vars.Conds

    Vars(VN("gr_g_b1")).Value = 1
    Vars(VN("m_kp")).Value = 1
    Vars(VN("M_x")).Value = 9.80665E-03
    Vars(VN("M_y")).Value = 9.80665E-03
    Vars(VN("M_xy")).Value = 9.80665E-04
    Vars(VN("Q_x")).Value = 9.80665E-03
    Vars(VN("Q_y")).Value = 9.80665E-03
    Vars(VN("d_shx")).Value = 10
    Vars(VN("s_nx")).Value = 0.1

```

Функция включает:

- создание необходимых объектов
- передачу значений переменных
- передачу условий (передаются значения и условия из расчета в NormCAD)

- расчет по пунктам нормативного документа с получением максимального коэффициента использования (его значение становится значением функции)

8. Выполним следующие изменения в функции NCResult:

- 1) в описание функции добавим переменные усилий (изгибающих моментов и поперечных сил) M\_x, M\_y, M\_xy, Q\_x, Q\_y
- 2) заменяем текущие значения переменных их именами

```

Public Function NCResult(M_x As Single, M_y As Single, _
M_xy As Single, _
Q_x As Single, Q_y As Single) As Single

On Error GoTo 1
Dim Vars As Object
Dim Conds As Object
Set Vars = CreateObject("NC_167258518177598E02.Vars")
Set Conds = Vars.Conds

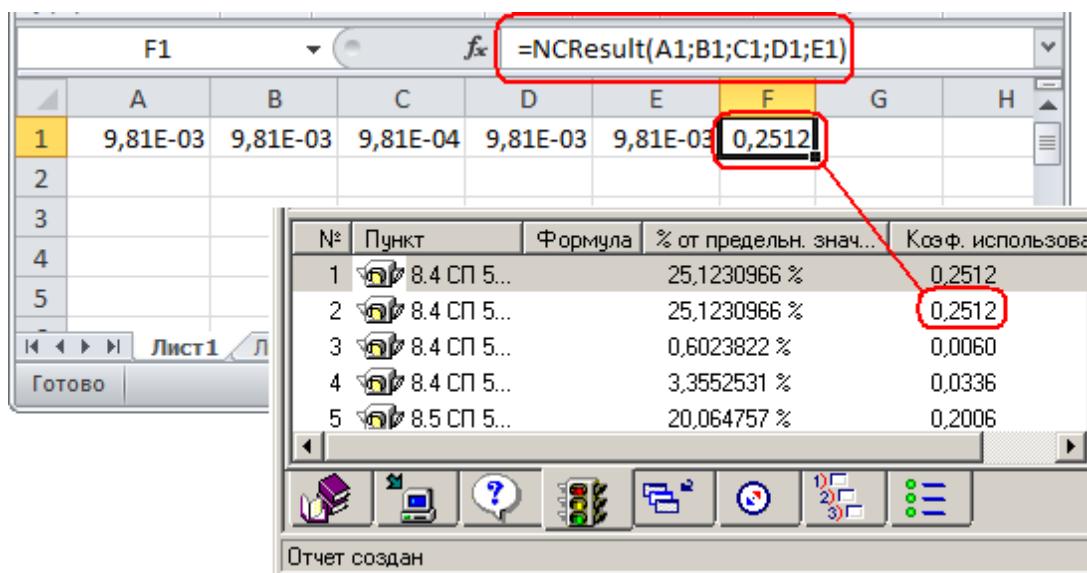
Vars(VN("gr_g_b1")).Value = 1
Vars(VN("m_kp")).Value = 1
Vars(VN("M_x")).Value = M_x
Vars(VN("M_y")).Value = M_y
Vars(VN("M_xy")).Value = M_xy
Vars(VN("Q_x")).Value = Q_x
Vars(VN("Q_y")).Value = Q_y

```

9. В ячейку F1 вставим формулу "=NCResult(A1;B1;C1;D1;E1)"

10. После ввода в ячейки A1, B1, C1, D1, E1 значений усилий получаем в ячейке F1 значение коэффициента использования, которое не отличается от полученного при расчете в NormCAD

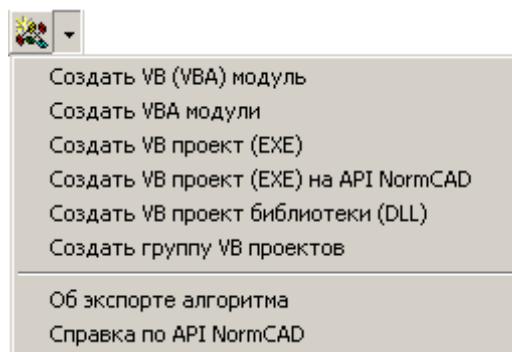
11.



### 5.3 Команды для экспорта алгоритмов модулей NormCAD в Visual Basic

Для упрощения использования кода модулей NormCAD в VBA и Visual Basic можно использовать различные средства автоматического создания программного кода на основе данных из активного документа NormCAD.

Команды для создания модулей, проектов или групп проектов на языке Visual Basic запускаются из меню кнопки **Создать VB(VBA) модуль** на панели инструментов **Основная 2** в основном окне программы:



Команда **Создать VB(VBA) модуль** создает для активного расчета программный модуль, который нужно подключать в редакторе VBA для использования в расчетах по таблицам MS Excel или подключать к проектам на Visual Basic (предполагает основное использование в среде VBA, которая поддерживается программами MS Excel, MS Word, MS Access, AutoCAD, ZwCAD и др.; содержит только ссылку на код библиотеки DLL с алгоритмами расчетов по нормативному документу; требует дополнительных действий для использования: подключения модуля к проекту Visual Basic или импорт модуля в редакторе VBA,

внесения изменения в код модуля – см. Пример расчета на основе NormCAD с использованием VBA)

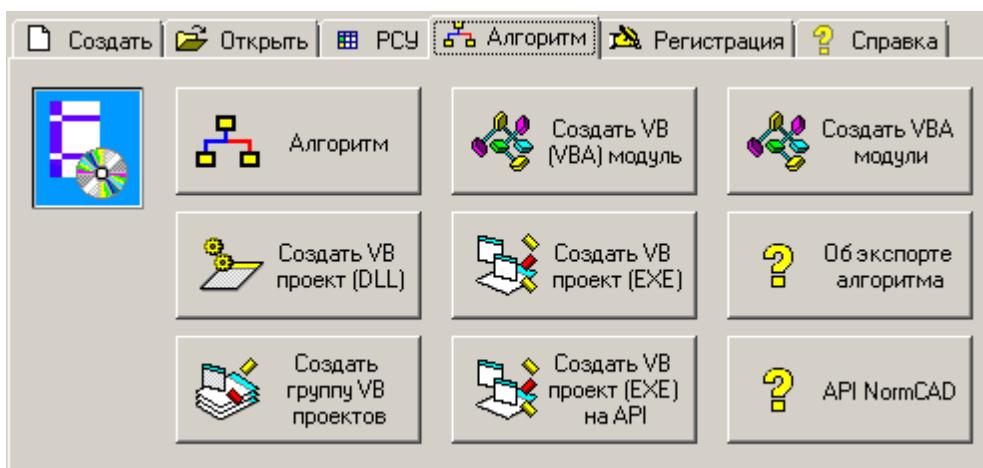
Команда **Создать VBA модули** создает для активного расчета программные модули со всеми используемыми алгоритмами, что позволяет выполнять отладку/просмотр всего алгоритма в среде VBA MS Excel и других программ (предполагает использование в среде VBA; содержит код алгоритмов расчетов по нормативному документу; требует дополнительных действий для использования: импорт модулей в редакторе VBA, необходимо установить ссылку на компонент NormCAD Table Values в меню Tools\References)

Команда **Создать VB проект (EXE)** создает для активного расчета готовый VB проект (содержит только ссылку на код библиотеки DLL с алгоритмами расчетов по нормативному документу; не требует дополнительных действий для использования: после компилирования в редакторе Visual Basic может запускаться как готовая программа)

Команда **Создать VB проект (DLL)** создает для активного модуля алгоритмов по нормативному документу VB проект библиотеки DLL (содержит код алгоритмов расчетов по нормативному документу; требует дополнительных действий для использования: подключения к библиотеке)

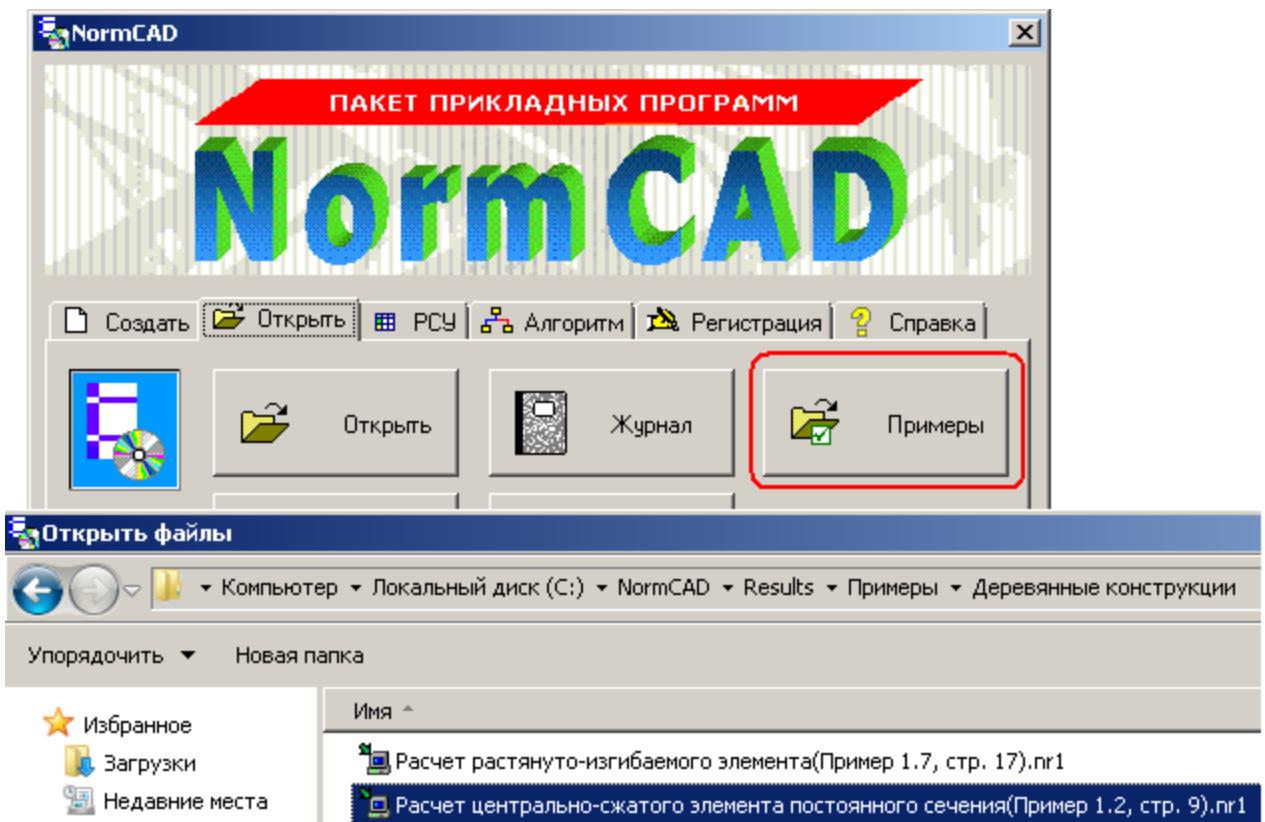
Команда **Создать группу VB проектов** создает для активного расчета группу VB проектов с подключенным проектом библиотеки DLL с алгоритмами по нормативному документу (содержит код алгоритмов расчетов по нормативному документу; не требует дополнительных действий для использования; может быть использована также для отладки в редакторе Visual Basic алгоритмов расчетов по нормативному документу)

Для вызова этих команд можно использовать так же соответствующие кнопки на вкладке **Алгоритм** в окне **Менеджера NormCAD** (быстрый вызов клавишей F12):



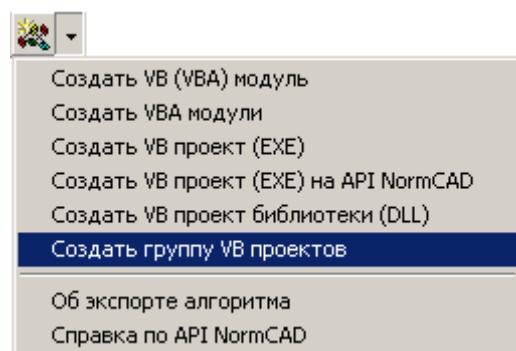
## 5.4 Пример автоматического создания кода готовой программы на языке Visual Basic на основе расчета в NormCAD

Откроем пример любого расчета:

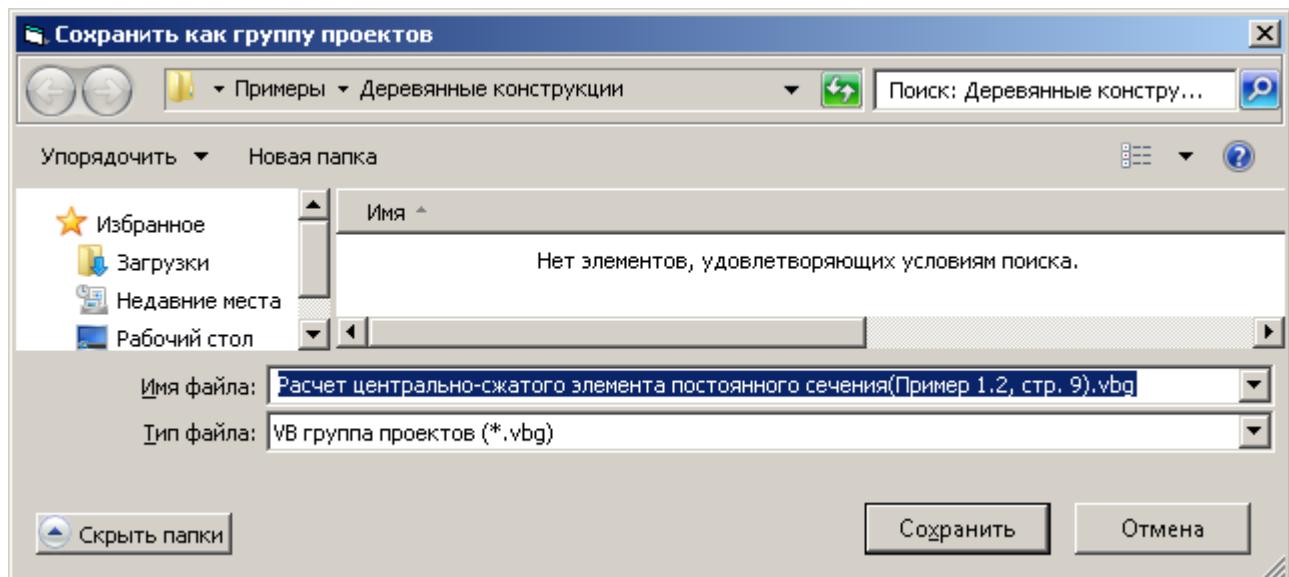


Примечание. Проект можно создать только на основе файла расчета, в который введены все необходимые данные и условия расчета. Перед созданием VB проекта нужно выполнить расчет и ввести данные и условия, если они не были заданы ранее.

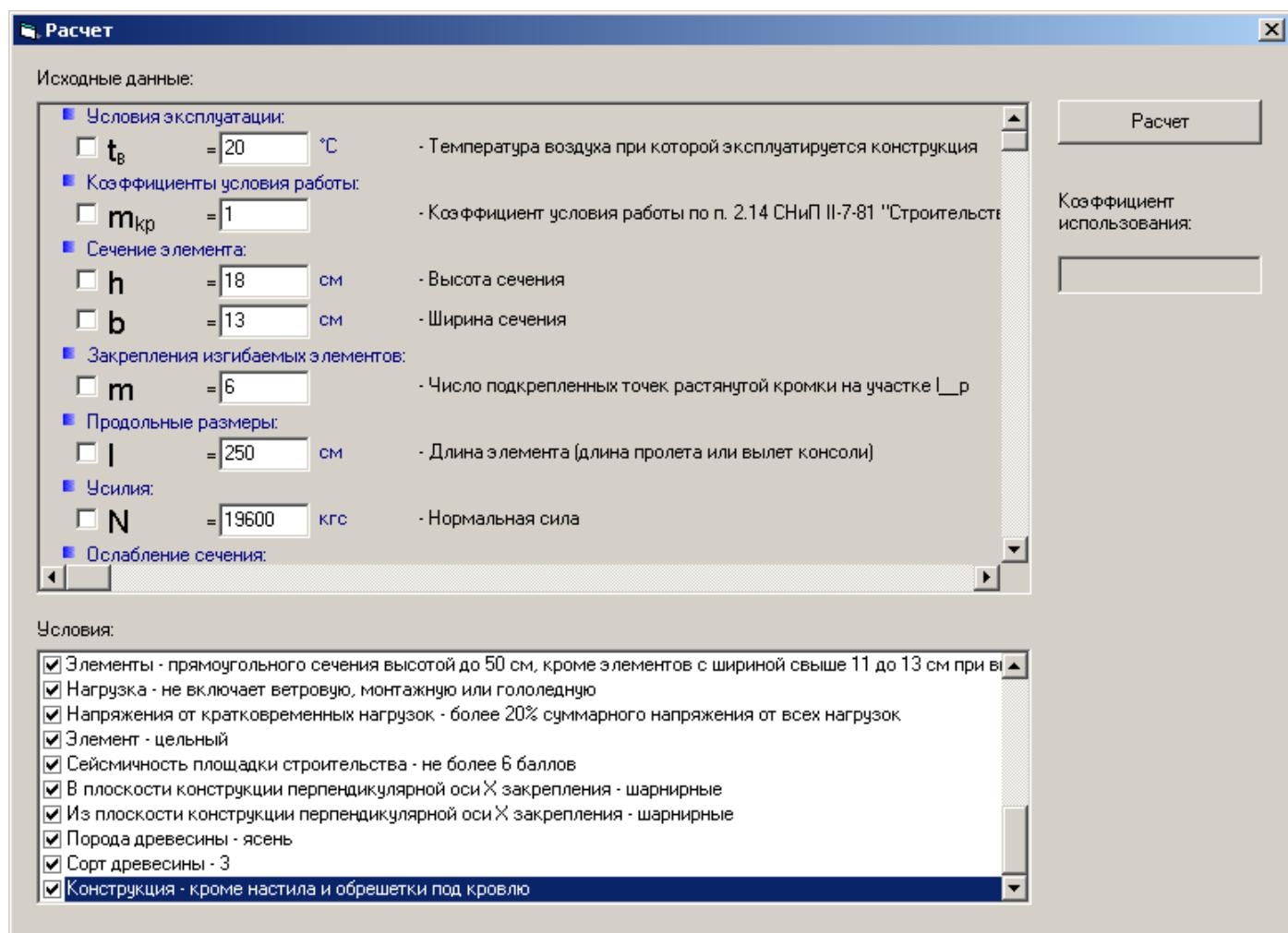
Выберем команду **Создать группу VB проектов** из меню кнопки **Алгоритм**:



Согласимся с предложенным путем для сохранения группы проектов (или выберем другой путь):



В результате автоматически создается группа проектов и открывается в среде Visual Basic. Запустим полученную программу на расчет клавишей **F5**. Будет показано диалоговое окно со всеми данными и условиями расчета (которые можно изменять; при снятии галочек с условий расчета будет предложен выбор из возможных вариантов при запуске на расчет):



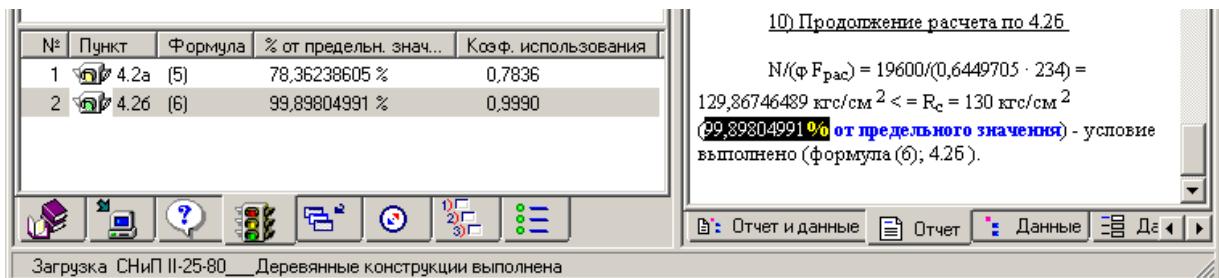
Нажмем в нем кнопку **Расчет** и получим значение коэффициента использования:

Расчет

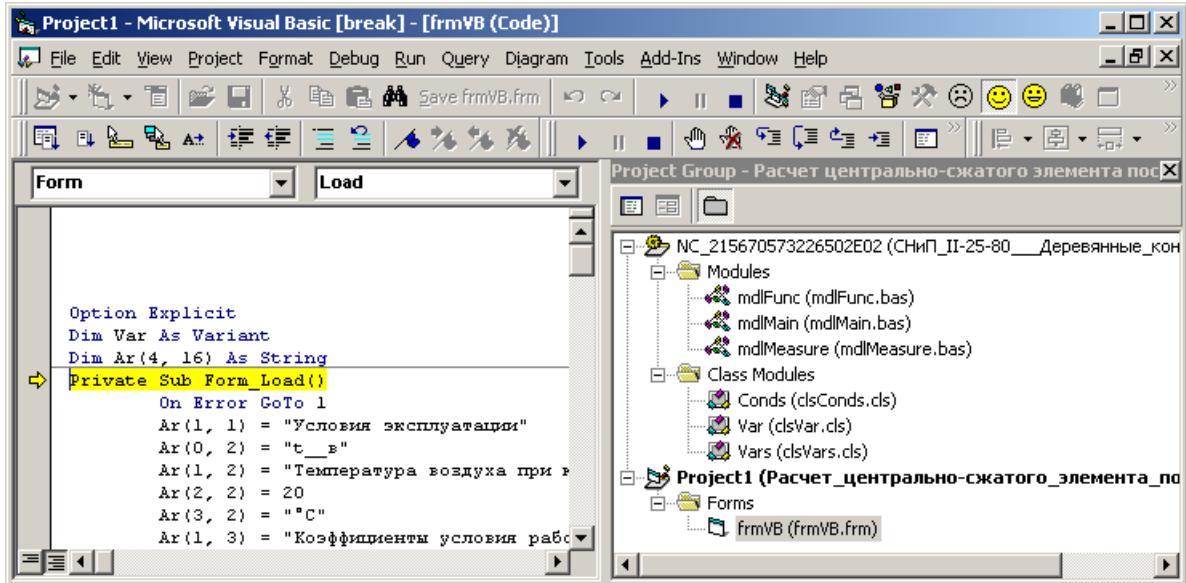
Коэффициент использования:

0,9989806

Это значение соответствует полученному при расчете в NormCAD:



Пройдемся по формулам алгоритма в режиме построчной отладки (многократно нажимая клавишу F8):



Перейдем к вызову основного пункта

```
Vars(VN("N")).Value = NCFList4.GetValue("N")
Vars(VN("F_o")).Value = NCFList4.GetValue("F_o")
For i = 1 To lstCond.ListCount
    If lstCond.Selected(i - 1) Then
        Conds.Add lstCond.List(i - 1)
    End If
Next
Vars.Result = 0
Vars.Ex ("S_" & VN("4.2"))
NCResult = Max(NCResult, Vars.Result)
txtR = NCResult
Exit Sub
1:
```

Далее можно перейти к алгоритму пункта

```
(General) S_4_pnt_2
If ((M_d_ / W_pac) / (N_ / F_pac) <= 0.1) Then
    ' Требуется проверка по формуле (6) без учета М.
    Call S_4_pnt_26_изг
End If
End Sub

' Пункт "4.2"

Public Sub S_4_pnt_2()
    ' Расчет центрально-сжатых элементов сплошного сечения.
```

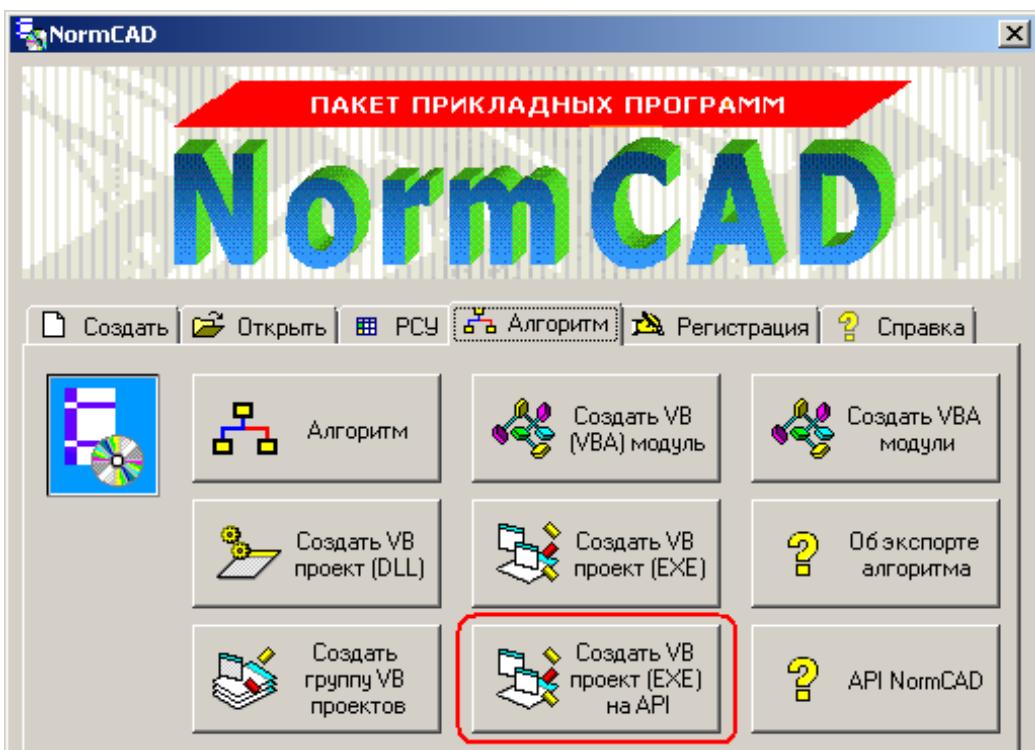
Примеры кода на Visual Basic на основе расчетов в NormCAD см. в папке Program Files\NormCAD\Results\Примеры\VB-код на основе NormCAD.

## 5.5 API NormCAD

API NormCAD – библиотека прикладного программирования в NormCAD. Библиотека API NormCAD содержит функции пользовательского программирования для выполнения основных команд и задач NormCAD, включая:

- открытие и сохранение файлов
- ввод данных и условий расчета
- выполнение расчетов
- подготовка отчетов
- передача отчетов в MS Word с оформлением расчетов рамкой и штампом

В окне Менеджера NormCAD имеется кнопка **Создать VB проект (EXE) на API** для создания на основе активного документа файлов программы с использованием API NormCAD:



Описание основных функций и команд API NormCAD:

#### **Функции/процедуры:**

**VarsCount (только чтение)** – Возвращает число переменных в модуле

**SetConds (ArConds() As String)** – Передает массив условий

**GetConds (ArConds() As String)** – Возвращает массив условий

**SetVars (Vars As Object)** – Передает объект переменных

**GetVars () As Object** – Возвращает объект переменных

**GetVarArray (Var As Variant)** – Возвращает массив переменных

**SaveDat (File As String)** – Сохраняет файл с расширением dat с данными

**LoadDat (File As String)** – Загружает файл с расширением dat с данными

**SaveNr1 (File As String)** – Сохраняет файл с расширением nr1 с условиями

**LoadNr1 (File As String)** – Загружает файл с расширением nr1 с условиями

(File – полное имя файла, включая его расположение)

**ClcLoadNorm** – Загружает модуль (предварительно должно быть задано имя модуля через переменную Norm)

**ClcLoadData** – Загружает данные в модуль (предварительно должны быть выполнены функции LoadDat или SetVars)

**ClcLoadConds** – Загружает условия в модуль (предварительно должны быть выполнены функции LoadNr1 или SetConds)

**LoadProp** – Загружает параметры расчета из реестра в модуль

**ClcCalc** – Запускается выполнение расчета (предварительно должны быть заданы переменные TaskName и Unit)

**MakeReport (File As String)** – Сохраняет отчет в файл

**SendToWord (File As String)** – Сохраняет отчет с рамкой и штампом в файл

**TestKey () As Boolean** – Выполняет проверку работы ключа защиты (требуется перед созданием отчета)

**MaxResult (только чтение)** – Возвращает максимальное значение коэффициента использования

### **Переменные:**

**Norm** – Название нормативного документа

**TaskName** – Название задания

**Unit** – Перечень пунктов в задании

Для начала работы создается экземпляр объекта библиотеки API NormCAD (NCAPIdll) и объект переменных расчетного модуля. Пример создания экземпляра объекта библиотеки API NormCAD:

```
Set ncApiR = New ncApi.Report
```

Пример создания объекта переменных:

```
Set Vars = CreateObject("NC_219532362921554E02.Vars")
```

где NC\_219532362921554E02 – пример имени библиотеки расчетного модуля.

Это имя можно получить автоматически – см. Пример расчета на основе NormCAD с использованием VBA.

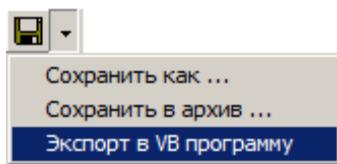
## **5.7 API NormFEM**

API NormFEM - библиотека прикладного программирования в NormFEM. NormFEM - модуль для расчета методом конечных элементов усилий в элементах конструкций в составе пакета NormCAD.

Библиотека API NormFEM содержит функции пользовательского программирования для выполнения основных команд и задач NormFEM, включая:

- ввод данных
- выполнение расчетов
- получение массивов результатов

В окне программы **NormFEM** в раскрываемом списке кнопки **Сохранить** имеется команда меню **Экспорт в VB программу** для автоматического получения из текущего проекта программы на Visual Basic с использованием NormFEM API



Описание основных функций и команд API NormFEM:

#### **Функции / процедуры:**

##### **1) Ввод данных**

**SetPath (AppPath As String, TempDir As String, Project As String, ParentForm)**

Процедурой **SetPath** передается:

- путь к папке NormFEM **AppPath**
- путь к папке для временных файлов **TempDir**
- имя задачи **Project**
- объект основной формы программы **ParentForm**

Основная форма программы должна включать следующие элементы:

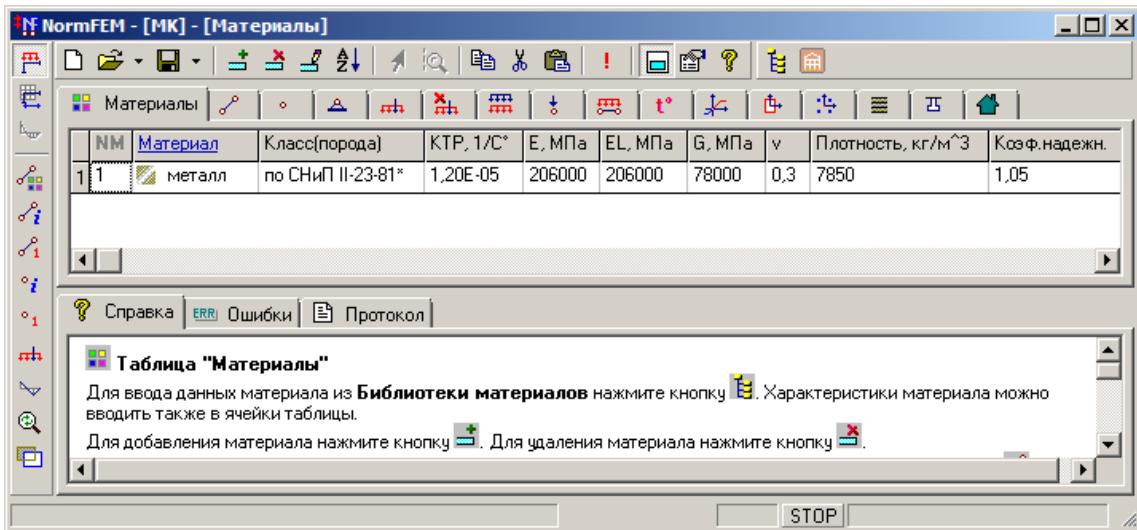
**txtProgress** - окно для вывода процента выполнения

**txtReport** - окно для вывода текста протокола

**SetArr(TblName As String, Arr() As String)** - Передает массив значений для таблицы проекта **TblName**

Пример использования **SetArr**:

Передаются значения из таблицы Материалы:



```

Public Sub AddA_d01()
    'Таблица "Материалы"
    'Колонки - NM : Материал : Класс(порода) : FormData : КТР : Е : EL : G : v : Плотность : Коэф.надежн.

    Dim A_d01(10, 0) As String

    A_d01(0, 0) = "1": A_d01(1, 0) = "    металл": A_d01(2, 0) = "по СНиП II-23-81*": A_d01(3, 0) = "металл\сталь\прокат и отливки": A_d01(4, 0) = "1,20E-05"

    A_d01(5, 0) = "206000": A_d01(6, 0) = "206000": A_d01(7, 0) = "78000": A_d01(8, 0) = "0,3": A_d01(9, 0) = "7850": A_d01(10, 0) = "1,05"

    Api.SetArr "d01", A_d01
End Sub

```

Имя таблицы TblName должно совпадать с расширением соответствующего файла таблицы в проекте. Описание таблиц имеется в файле Tables.ini и в справке по NormFEM.

Данные по координатам и типам узлов и элементов также передаются с использованием процедуры **SetArr** таблицами:

- g01 - Основные узлы (нагруженные сосредоточенной нагрузкой и опорные):

```

Public Sub AddA_g01()
    'Таблица "Основные узлы"
    'Колонки - Группа узлов : X : Y : Z

    Dim A_g01(3, 15) As String

    A_g01(0, 0) = "1": A_g01(1, 0) = "600": A_g01(2, 0) = "0": A_g01(3, 0) = "0"
    A_g01(0, 1) = "1": A_g01(1, 1) = "900": A_g01(2, 1) = "0": A_g01(3, 1) = "0"
    A_g01(0, 2) = "1": A_g01(1, 2) = "0": A_g01(2, 2) = "0": A_g01(3, 2) = "0"
    .....
    Api.SetArr "g01", A_g01
End Sub

```

где в строках массива передаются номера групп узлов; X, Y, Z - координаты узлов, см

- g02 - Все узлы:

```
Public Sub AddA_g02()  
  
    ' Таблица "Все узлы"  
  
    ' Колонки - X : Y : Z  
  
    Dim A_g02(2, 163) As String  
  
    A_g02(0, 0) = "0": A_g02(1, 0) = "0": A_g02(2, 0) = "0"  
    A_g02(0, 1) = "0": A_g02(1, 1) = "100": A_g02(2, 1) = "0"  
    A_g02(0, 2) = "0": A_g02(1, 2) = "0": A_g02(2, 2) = "300"  
    .....  
  
    Api.SetArr "g02", A_g02  
End Sub
```

где в строках массива передаются X, Y, Z - координаты узлов, см

- g03 - Элементы :

```
Public Sub AddA_g03()  
  
    ' Таблица "Элементы"  
  
    ' Колонки - Группа элементов : Узел 1 : Узел 2 : Метка  
  
    Dim A_g03(3, 189) As String  
  
    A_g03(0, 0) = "2": A_g03(1, 0) = "1": A_g03(2, 0) = "2": A_g03(3, 0) = "538"  
    A_g03(0, 1) = "2": A_g03(1, 1) = "3": A_g03(2, 1) = "4": A_g03(3, 1) = "539"  
    A_g03(0, 2) = "2": A_g03(1, 2) = "5": A_g03(2, 2) = "6": A_g03(3, 2) = "509"  
    .....  
  
    Api.SetArr "g03", A_g03  
End Sub
```

где в строках массива передаются номера групп элементов; номер узла 1; номер узла 2; индекс элемента (может хранить информацию об индексе объекта чертежа).

Номера узлов соответствуют номерам строк в таблице Все узлы.

**Prop()** - Вызывает окно **Параметры**

## 2) Расчет

**Calc** - Запускает выполнение расчета

**StopEx** - При необходимости останавливает выполнение расчета:

```
Private Sub cmdStop_Click()
    Api.StopEx
End Sub
```

### 3) Вывод результатов

**GetArrZ(ArrZ() As Double)** - Возвращает массив значений перемещений, где:

**ArrZ(i, j, k, m)** - массив значений перемещений

**i** - номер узла (от 1)

**j** - номер загружения (от 0)

**k** - номер направления перемещения:

- для плоских систем (Mode3D = False):

1 - перемещение вдоль оси X, м

2 - перемещение вдоль оси Y, м

3 - поворот в узле, радиан

- для пространственных систем (Mode3D = True):

1 - перемещение вдоль оси X, м

2 - перемещение вдоль оси Y, м

3 - перемещение вдоль оси Z, м

4 - поворот в узле вокруг оси X, радиан

5 - поворот в узле вокруг оси Y, радиан

6 - поворот в узле вокруг оси Z, радиан

**m** - в зависимости от длительности нагрузки:

0 - для всей нагрузки

1 - для длительной нагрузки

**GetArrNM(ArrNM() As Double)** - Возвращает массив значений нормальных сил и изгибающих моментов, где:

**ArrNM(i, j, k, m)** - массив значений перемещений

**i** - номер элемента (от 1)

**j** - номер загружения (от 0)

**k** - в зависимости от вида и расположения усилия:

- для плоских систем (Mode3D = False):

0 - нормальная сила в начале элемента, тс

1 - нормальная сила в конце элемента, тс

2 - изгибающий момент в начале элемента, тс<sup>\*</sup>м

**3** - изгибающий момент в конце элемента, тс<sup>\*м</sup>

- для пространственных систем (Mode3D = True):

0 - нормальная сила в начале элемента, тс

1 - нормальная сила в конце элемента, тс

2 - изгибающий момент вокруг оси X в начале элемента, тс<sup>\*м</sup>

3 - изгибающий момент вокруг оси X в конце элемента, тс<sup>\*м</sup>

4 - изгибающий момент вокруг оси Y в начале элемента, тс<sup>\*м</sup>

5 - изгибающий момент вокруг оси Y в конце элемента, тс<sup>\*м</sup>

6 - крутящий момент, тс<sup>\*м</sup>

**m** - в зависимости от длительности нагрузки:

0 - для всей нагрузки

1 - для длительной нагрузки

**GetArrQ(ArrQ0) As Double** - Возвращает массив значений поперечных сил,

где:

**ArrQ(i, j, k, m)** - массив значений перемещений

**i** - номер элемента (от 1)

**j** - номер загружения (от 0)

**k** - в зависимости от вида и расположения усилия:

- для плоских систем (Mode3D = False):

1 - поперечная сила в начале элемента, тс

2 - поперечная сила в конце элемента, тс

- для пространственных систем (Mode3D = True):

1 - поперечная сила вдоль оси X в начале элемента, тс

2 - поперечная сила вдоль оси X в конце элемента, тс

3 - поперечная сила вдоль оси Y в начале элемента, тс

4 - поперечная сила вдоль оси Y в конце элемента, тс

**m** - в зависимости от длительности нагрузки:

0 - для всей нагрузки

1 - для длительной нагрузки

**GetArr(TblName As String, Arr() As String)** - Возвращает массив значений для таблицы проекта **TblName**

**Переменные:**

**Steps As Long** - Число промежуточных точек для определения усилий между узлами

**Mode3D As Boolean (только чтение)** - Возвращает True, если задача трехмерная; False, если задача двумерная

Для начала работы создается экземпляр объекта библиотеки API NormFEM (ncfemapi.dll). Пример создания экземпляра объекта библиотеки API NormFEM:

```
Set Api = New ncfemapi.main
```

## 5.8 Пример комплексной автоматизации на основных стадиях проектирования конструкции

Основными стадиями проектирования конструкции являются:

- построение расчетной схемы
- определение нагрузок
- задание параметров сечений
- расчет усилий и перемещений
- получение усилий и перемещений от сочетаний нагрузок
- проверка сечений по нормам
- корректировка сечений
- документирование (чертежи, таблицы, отчеты)

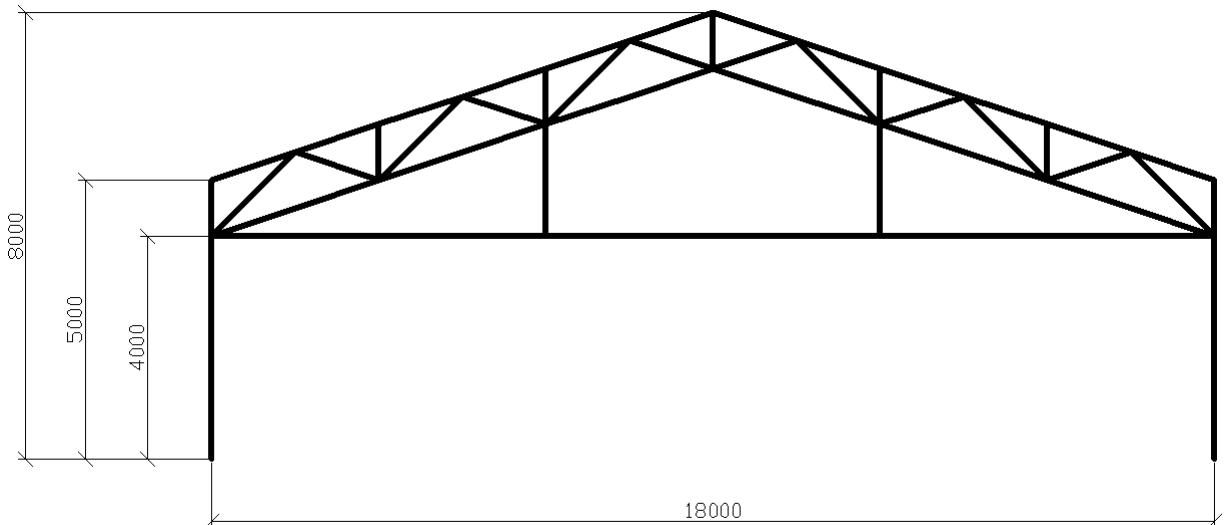
Весь комплекс работ по проектированию обычно выполняется с использованием разных программ, специализированных для отдельных стадий. Некоторые из этих этапов приходится выполнять многократно для выполнения всех требований норм. Возникает проблема переноса больших объемов данных между программами, который часто выполняется вручную.

С помощью различных средств программирования NormCAD возможно выполнить комплексную автоматизацию проектирования на всех основных стадиях в одной программе.

Далее будет рассмотрен пример такой комплексной автоматизации с помощью различных средств программирования NormCAD. Пример реализован с использованием VBA Excel, как наиболее доступного средства программирования.

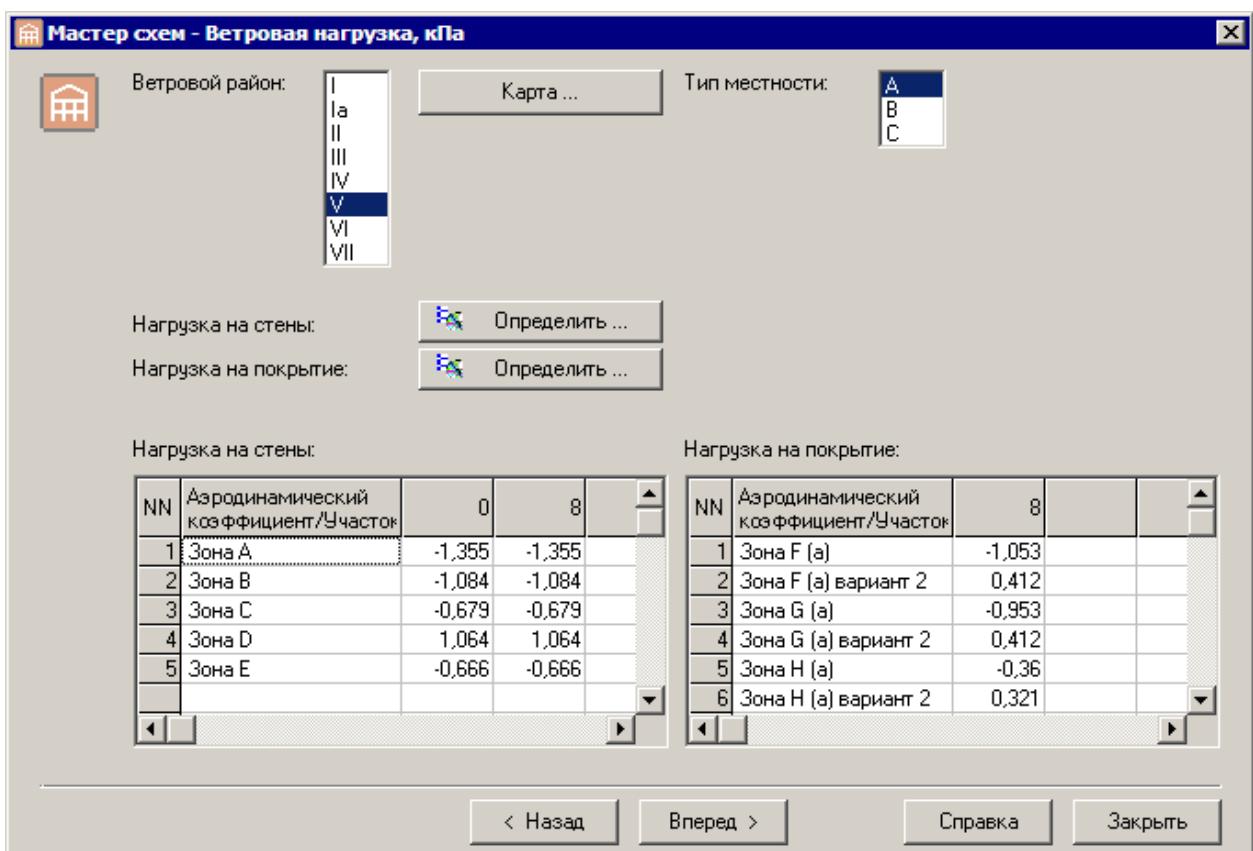
Автоматизируем проектирование рамы каркаса производственного здания из легких металлических конструкций. Требуется выполнить проверку такой рамы на действие сочетаний нагрузок в произвольных районах строительства.

Схема рамы и таблица сечений приводится ниже. Шаг рам – 6 м. Длина здания – 30 м.



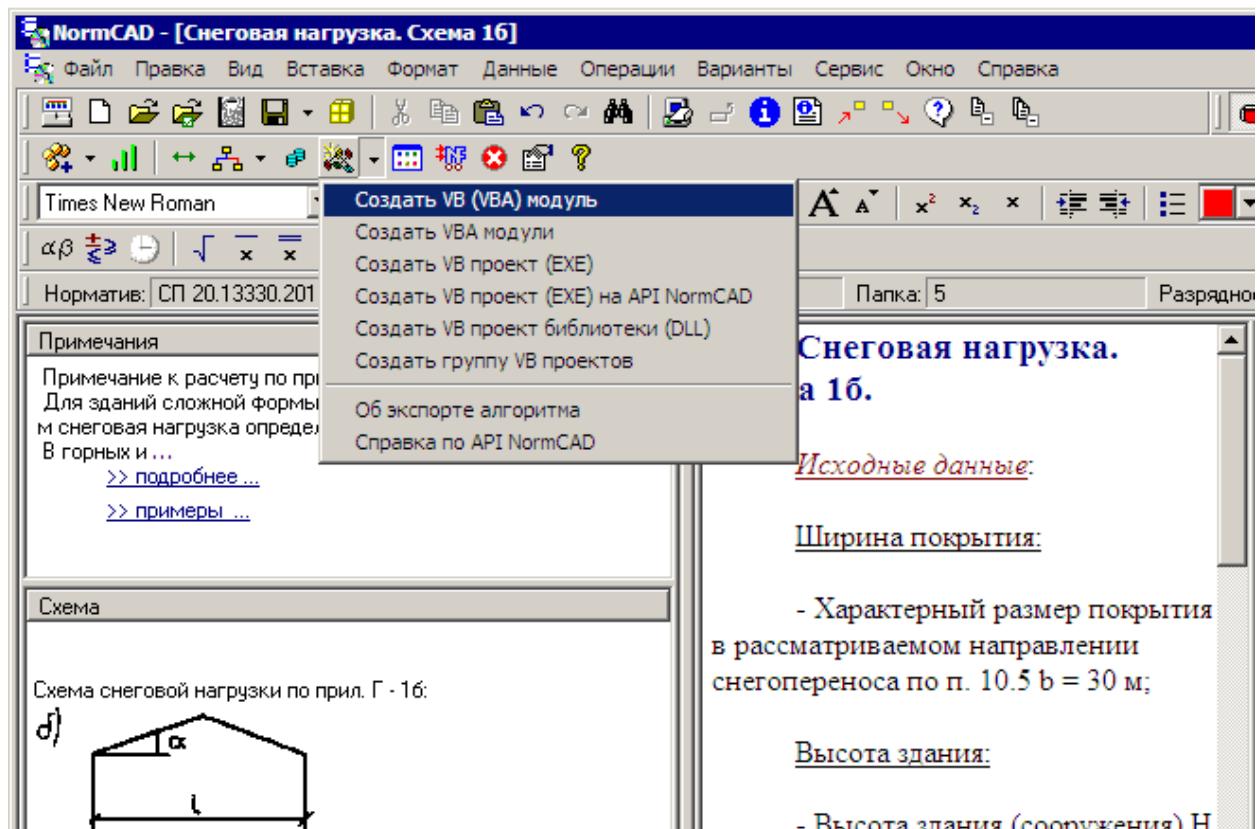
колонны	Гн. I 140x4
нижний пояс ферм	Гн. I 100x3
верхний пояс ферм	Гн. I 100x3
решетка ферм	Гн. I 60x3
опорные раскосы	Гн. I 80x3
стойки ферм	Гн. I 60x3
затяжка ферм	Гн. I 80x4

Шаг 1. Выполняется конечно-элементный расчет рамы в NormFEM с использованием Мастера схем. При этом мы задаемся определенными снеговыми и ветровыми районами. В Мастере схем создаются файлы расчетов в NormCAD по СП 20.13330.2011 "Нагрузки и воздействия" соответствующих снеговых и ветровых нагрузок на раму.



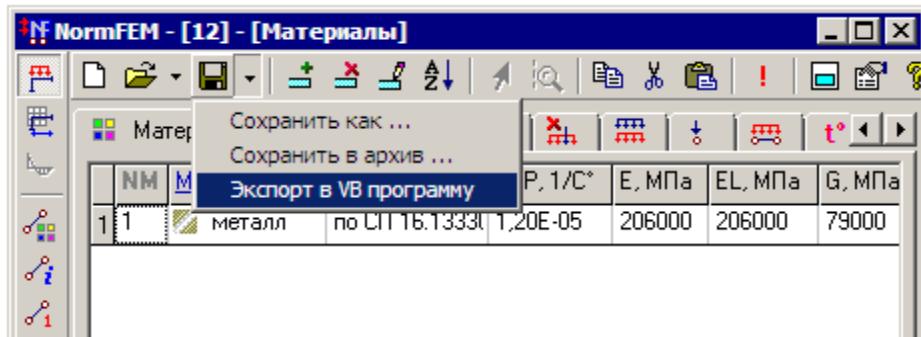
- Ветровая нагрузка. Профиль 2.nr1
- Ветровая нагрузка. Стены.nr1
- Снеговая нагрузка. Схема 16.nr1

Шаг 2. Файлы расчета нагрузок открываются в NormCAD, и выполняется получение для данных расчетов автоматического кода в Visual Basic командой **Создать VB (VBA) модуль**.



- Ветровая нагрузка. Профиль 2.bas
- Ветровая нагрузка. Стены.bas
- Снеговая нагрузка. Схема 16.bas

Шаг 3. На основе данных из проекта расчета в NormFEM получим программу с использованием API NormFEM на Visual Basic командой **Экспорт в VB программу**.



12.vbp  
 frmVB.frm  
 mdlFuncApi.bas

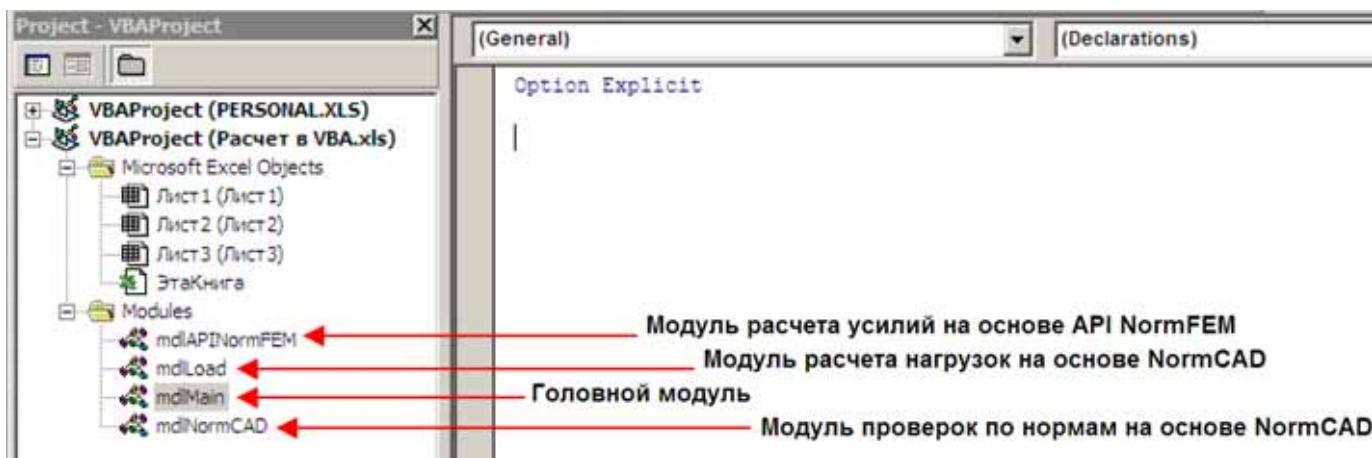
Шаг 4. Выполняем расчеты в NormCAD по СП 16.13330.2011 "Стальные конструкции" для всех видов сочетаний усилий в сечениях элементов при произвольных значениях усилий и размерах сечений.

- Расчет растянуто-изгибаемых элементов.nр1
- Расчет сжато-изогнутых элементов.nр1
- Расчет центрально-растянутых элементов без ослаблений.nр1
- Расчет центрально-сжатых элементов.nр1

Шаг 5. Файлы расчета сечений открываются в NormCAD, и выполняется получение для данных расчетов автоматического кода в Visual Basic командой **Создать VB (VBA) модуль**.

- Расчет растянуто-изгибаемых элементов.bas
- Расчет сжато-изогнутых элементов.bas
- Расчет центрально-растянутых элементов без ослаблений.bas
- Расчет центрально-сжатых элементов.bas

Шаг 6. Создается файл MS Excel, входим в редактор VBA (для этого в MS Excel можно использовать сочетание клавиш Alt+F11) и вставляем модули VBA:



## Шаг 7. Заполнение модулей VBA

1) Заполняем модуль mdlNormCAD – модуль проверок по нормам на основе NormCAD.

Используем автоматический код, полученный из расчетов сечений в NormCAD, полученных на шаге 5:

- Расчет растянуто-изгибаемых элементов.bas
- Расчет сжато-изогнутых элементов.bas
- Расчет центрально-растянутых элементов без ослаблений.bas
- Расчет центрально-сжатых элементов.bas

Вносим изменения в эти файлы в функции коэффициентов использования:

- меняем названия функций с отражением смысла расчетов
- вставляем в описание функций список переменных:
  - номер элемента iEL
  - переменные для усилий N, M, Q
- заменяем числовые значения усилий переменными N, M, Q
- числовые значения характеристик сечения меняем значениями элементов массива A\_d02() (это массив данных об элементах из программы с использованием API NormFEM, полученной на шаге 3; далее мы будем ее использовать для заполнения модуля mdlAPINormFEM)
  - числовые значения расчетных длин (в необходимых случаях и других данных) меняем значениями элементов массива ArEL() (это массив дополнительных данных об элементах, которые отсутствуют в конечно-элементном расчете в NormFEM; его требуется заполнить вручную)
  - добавляем основную функцию модуля для получения коэффициентов использования всей конструкции (с обращением к функциям

коэффициентов использования для отдельных элементов с расчетом на разные виды сочетаний усилий)

Пример фрагмента таких изменений в функции коэффициента использования растянуто-изгибающего элемента:

```
Public Function TestMNmax(iEL As Long, N As Double, M As Double, Q As Double) As Double
    Dim l_efx As Double, l_efy As Double

    Dim GrEL As Long
    GrEL = CLng(A_g03(0, iEL))

    l_efx = ArEL(0, GrEL)
    l_efy = ArEL(1, GrEL)

    Vars("l_efx").Value = CMtoMM * l_efx
    Vars("l_efy").Value = CMtoMM * l_efy

    Vars("N").Value = TCtoN * N

    Vars("M_x").Value = TCtoN * MtoMM * M
    Vars("M_y").Value = 0
    Vars("Q_y").Value = TCtoN * Q / 2 ' - деление на 2 - на одну стенку сечения

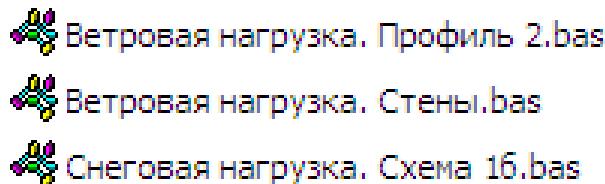
    Vars("gr_g_c").Value = 1

    Dim iGrEL As Long
    iGrEL = ArrIGrEL(GrEL)

    Vars("h").Value = CMtoMM * A_d02(7, iGrEL)
    Vars("b").Value = CMtoMM * A_d02(6, iGrEL)
    Vars("A").Value = CMtoMM2 * A_d02(8, iGrEL)
    Vars("J_x").Value = CMtoMM4 * A_d02(9, iGrEL)
    Vars("J_y").Value = CMtoMM4 * A_d02(10, iGrEL)
```

2) Заполняем модуль mdlLoad – модуль расчета нагрузок на основе NormCAD.

Используем автоматический код, полученный из расчетов сечений в NormCAD, полученных на шаге 2:



Вносим изменения в эти файлы в функции коэффициентов использования:

- преобразовываем функции в процедуры
- меняем названия процедур с отражением смысла расчетов
- вставляем в описание процедур список переменных:

- базовая ветровая и снеговая нагрузка (нормативное значение ветрового давления и вес снегового покрова на 1 м<sup>2</sup> горизонтальной поверхности земли)
- ветровой и снеговой район (WR, SR)
- тип местности TM (A, B, C)
- заменяем числовые и текстовые значения переменными
- извлекаем значения текста таблиц нагрузок в NormCAD
- переводим тестовые значения в соответствующие двумерные массивы распределения нагрузок

Пример таких изменений в функции, преобразованной в процедуру для заполнения массива значений ветровой нагрузки на стены:

```

Public Sub Wind(w_o As Double, WR As String, TM As String)

    Vars("Z_1").Value = 8
    Vars("h_rp").Value = 5
    Vars("b_rp").Value = 6
    Vars("a_rp").Value = 6
    Vars("H_H").Value = 8
    Vars("l_w").Value = 18
    Vars("b_w").Value = 30
    Vars("d_w").Value = 30
    Vars("h_1").Value = 5
    Vars("n_z").Value = 1

    Vars("w_o").Value = w_o

    Conds.Clear

    Conds.Add "Ветровой район - " & WR
    Conds.Add "Тип местности - " & TM

    Conds.Add "Размеры расчетных поверхностей - не совпадают с размерами здания"
    Conds.Add "Тип здания или сооружения - здание"
    Conds.Add "Тип конструкции - здание со стальным каркасом и ограждающими конструкциями"
    Conds.Add "Тип конструкции - одноэтажное здание"
    Conds.Add "Тип здания - производственное высотой до 36 м при отношении высоты к пролету менее 1,5"

    Conds.Add "Схема ветровой нагрузки по прил. Д - 1.2 Вертикальные стены прямоугольных в плане зданий"

    Vars.Ex ("S_" & VN("прил.Д"))
    Dim TextTable As String
    TextTable = Vars("TextTable").Value
    | - выполняем расчет по прил. Д СП 20.13330
    | - получаем значение текста таблицы с нагрузками

    ArWind = GetTblAr(TextTable)      - переводим текст таблицы в двумерный массив

End Sub

```

3) Заполняем модуль mdlAPINormFEM – модуль расчета усилий на основе API NormFEM.

Используем автоматический код, полученный из расчетов сечений в NormCAD, полученных на шаге 3:



При этом вносим следующие изменения в эти файлы:

- заменяем числовые значения нагрузок формулами с массивами распределения нагрузок, которые были получены при заполнении модуля mdlLoad:

```
A_d05(8, 8) = ArSnow(5, 1) * wk
A_d05(8, 9) = ArSnow(5, 1) * wk * 0.7
A_d05(8, 10) = ArSnow(5, 1) * wk
A_d05(8, 11) = ArSnow(5, 1) * wk * 0.7
A_d05(8, 12) = ArSnow(5, 1) * wk
A_d05(8, 13) = ArSnow(5, 1) * wk * 0.7
A_d05(8, 14) = ArSnow(5, 1) * wk
A_d05(8, 15) = ArSnow(5, 1) * wk * 0.7
A_d05(8, 16) = -ArWind(2, 4) * wk
A_d05(8, 17) = -ArWind(2, 4) * wk
A_d05(8, 18) = ArWind(2, 5) * wk
A_d05(8, 19) = ArWind(2, 5) * wk
A_d05(8, 20) = ArWind2(2, 1) * wk
A_d05(8, 21) = ArWind2(2, 2) * wk
A_d05(8, 22) = ArWind2(2, 5) * wk
A_d05(8, 23) = ArWind2(2, 6) * wk
A_d05(8, 24) = ArWind2(2, 7) * wk
A_d05(8, 25) = ArWind2(2, 8) * wk
```

- добавляем процедуры:
  - для получения массива усилий от сочетаний нагрузок
  - для получения массива максимальных и минимальных усилий для групп элементов ферм
- добавляем основную функцию модуля для получения массивов усилий:

```
Public Function Calc(Gp As Double, Gs As Double) As Boolean
'Функция расчета усилий
```

```
ReportAdd "Расчет усилий"
```

```
Gpkr = Gp
Gstn = Gs
```

```
'Заполнение массивов данных для расчета
AddAllArr
```

```
'Расчет в API NormFEM
```

```
nfApi.Calc
```

```
Calc = nfApi.Result
```

```
If Not Calc Then Exit Function
```

```
'Получение массивов усилий из API NormFEM
```

```
nfApi.GetArrZ ArrZ  
nfApi.GetArrNM ArrNM  
nfApi.GetArrQ ArrQ
```

```
'Расчет усилий от сочетаний нагрузок
```

```
GetComb
```

```
'Определение максимальных и минимальных усилий в элементах ферм  
MaxMinN
```

```
Set nfApi = Nothing  
End Function
```

4) Заполняем модуль mdlMain – головной модуль с одной функцией для получения коэффициентов использования всей конструкции (с предварительным получением массивов нагрузок и усилий):

```
Public Function KUs(SR As String, WR As String, TM As String, _  
    Gp As Double, Gs As Double) As Double
```

```
' SR - снеговой район (I - VIII)  
' WR - ветровой район (Ia - VII)  
' TM - тип местности (A, B, C)  
' Gp - расчетное значение веса покрытия, кПа  
' Gs - расчетное значение веса стен, кПа
```

```
'Расчет нагрузок  
Load SR, WR, TM
```

```
'Расчет усилий
```

```
If Not Calc(Gp, Gs) Then
```

```
    'Показ сообщения, если расчет конструкции выполнить не удалось
```

```
    MsgBox "Расчет конструкции выполнить не удалось", vbCritical  
    Exit Function
```

```
End If
```

```
'Проверка усилий по нормам
```

```
KUs = GetKUs ' - коэффициент использования
```

```
End Function
```

## Шаг 8. Вставка формулы с функцией на VBA в файле Excel:

OKРУГЛ	<input type="button" value="X"/>	<input checked="" type="button" value="√"/>	=	=KUs(C4;C14;C24;C31;C33)
	A	B	C	D

Файл примера можно увидеть в папке ... Program Files \ NormCAD \ Results \ Примеры \ VB-код на основе NormCAD \ Комплексная автоматизация.

В файле примера добавлены также элементы управления, в частности, списки для возможности выбора значений снегового и ветрового районов и типа местности, добавлены кнопки пересчета, окно с выводом протокола расчета:

	A	B	C	D	E	F
1						
2	<b>Расчет конструкций каркаса производственного здания</b>					
3						
4	Снеговой район:		I II III IV V VI VII VIII			
5						
6						
7						
8						
9						
10						
11						
12						
13						
14	Ветровой район:		Ia I II III IV V VI VII			
15						
16						
17						
18						
19						
20						
21						
22						
23						
24	Тип местности:		A B C			
25						
26						
27						
28						
29	Расчетное значение веса, кПа:					
30						
31	- покрытия:		0,6			
32						
33	- стен:		0,4			
34						
35						
36	Коэффициент использования:		0,8274		Пересчет	
37						
38	(при <= 1 требования норм выполняются)				Остановить	
39						

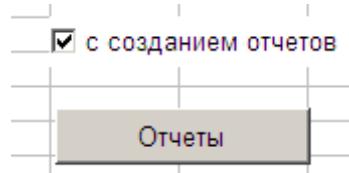
В примере реализована комплексная автоматизация проектирования конструкции сразу по нескольким стадиям:

- определение нагрузок
- расчет усилий и перемещений
- получение усилий и перемещений от сочетаний нагрузок
- проверка сечений по нормам

Далее покажем возможность добавления автоматизации и стадии документирования с получением отчетов по наиболее нагруженным элементам в NormCAD. При этом используем библиотеку API NormCAD.

Вставим в файл примера модуль mdlAPINormCAD и элементы управления:

- кнопку Отчеты
- флажок "с созданием отчетов"



С реализацией документирования с получением отчетов на основе API NormCAD можно ознакомиться в файле примера в модуле mdlAPINormCAD. Там имеются комментарии с разъяснениями.

При нажатии кнопки Отчеты должно появиться окно с файлами, которые можно открыть в NormCAD. Можно убедиться, что при пересчете в NormCAD получается тот же результат коэффициента использования для наиболее нагруженного элемента.

# **Содержание**

<b>ПРЕДИСЛОВИЕ.....</b>	<b>3</b>
<b>1. ОСНОВЫ ПРОГРАММИРОВАНИЯ В NORMCAD .....</b>	<b>5</b>
1.1 Модули NORMCAD .....	5
1.2 РАБОТА В ОТЛАДЧИКЕ .....	6
1.3 Ввод пунктов и таблиц.....	10
<b>2. ЯЗЫК ПРОГРАММИРОВАНИЯ В NORMCAD .....</b>	<b>17</b>
2.1 ПЕРЕМЕННЫЕ В NORMCAD .....	17
2.2 Выражения в NORMCAD.....	20
2.3 Условия и другие управляющие конструкции .....	22
2.4 Оформление отчета, вывод результатов.....	25
<b>3. РАЗРАБОТКА ИНТЕРФЕЙСА ВЗАИМОДЕЙСТВИЯ С ПОЛЬЗОВАТЕЛЕМ .....</b>	<b>28</b>
<b>4. ПРИМЕРЫ ПРОГРАММИРОВАНИЯ В СРЕДЕ NORMCAD .....</b>	<b>31</b>
4.1 ПРИМЕР ВВОДА АЛГОРИТМОВ В NORMCAD.....	31
4.2 ПРИМЕР РАЗРАБОТКИ ВИЗУАЛЬНОГО ИНТЕРФЕЙСА В NORMCAD. ВСТАВКА ДИАЛОГОВОГО ОКНА.....	39
4.3 ПРИМЕР ВСТАВКИ СПРАВОЧНИКА .....	40
4.4 ПРИМЕР ВСТАВКИ ДИАЛОГОВОГО ОКНА, СВЯЗАННОГО СО СПРАВОЧНИКОМ .....	42
<b>5. ИСПОЛЬЗОВАНИЕ АЛГОРИТМОВ NORMCAD .....</b>	<b>44</b>
5.1 ЭКСПОРТ АЛГОРИТМОВ МОДУЛЕЙ NORMCAD .....	44
5.2 ПРИМЕР РАСЧЕТА НА ОСНОВЕ NORMCAD с ИСПОЛЬЗОВАНИЕМ VBA .....	44
5.3 Команды для экспорта алгоритмов модулей NORMCAD в VISUAL BASIC.....	46
5.4 ПРИМЕР АВТОМАТИЧЕСКОГО СОЗДАНИЯ КОДА ГОТОВОЙ ПРОГРАММЫ НА ЯЗЫКЕ VISUAL BASIC НА ОСНОВЕ РАСЧЕТА В NORMCAD .....	48
5.5 API NORMCAD .....	51
5.7 API NORMFEM .....	53
5.8 ПРИМЕР КОМПЛЕКСНОЙ АВТОМАТИЗАЦИИ НА ОСНОВНЫХ СТАДИЯХ ПРОЕКТИРОВАНИЯ КОНСТРУКЦИИ.....	59