



# Руководство для написания скриптов в NormCAD/NormFEM (стиль «вики»)

## Введение

**NormCAD** — программный комплекс для расчёта строительных конструкций по нормам. Он имеет встроенный интерпретируемый язык и поддерживает экспорт алгоритмов в Visual Basic. Для автоматизации расчётов предусмотрены библиотеки **API NormCAD** и **API NormFEM**, которые позволяют управлять модулями NormCAD из внешних приложений (например, VBA или Python) и получать результаты расчётов.

Данная вики собирает ключевую информацию из официальной документации и примеров, необходимую для написания скриптов, связанных с проведением проверок по нормам. В качестве примера рассматривается общая проверка несущей способности элемента, но описанные шаги можно адаптировать под любую нормативную проверку.

## 1. Среда программирования NormCAD

### 1.1 Язык и отладчик

- Встроенный язык NormCAD ориентирован на инженерные расчёты; он интерпретируется и автоматически документирует выполняемые строки <sup>1</sup>.
- Алгоритмы хранятся в файлах модулей (расширение `.nrb`). Для их создания используется «Мастер создания модулей».
- Отладчик позволяет редактировать код, выполнять пошаговое выполнение и просматривать значения переменных.

### 1.2 Экспорт кода в Visual Basic

- Для ускорения расчётов или интеграции с другими системами алгоритмы NormCAD можно экспортировать в Visual Basic (VBA). В менеджере NormCAD доступна кнопка «Создать **VB проект (EXE) на API**», которая формирует проект на основе активного документа, использующий API NormCAD <sup>2</sup>.
- В NormFEM имеется команда «**Экспорт в VB программу**» на кнопке «Сохранить» для автоматического создания кода с использованием API NormFEM <sup>3</sup>.

## 2. Общий порядок выполнения проверки

Ниже представлен типовой порядок действий при автоматизации проверки несущей способности элемента:

1. **Формирование расчётной схемы.** Создайте проект в NormFEM (если требуется конечно-элементный анализ) либо используйте существующие файлы нагрузки и сечений.

2. **Экспорт автоматического кода.** Используйте команды «Создать VB модуль» (в NormCAD) или «Экспорт в VB программу» (в NormFEM) для получения шаблона кода на VBA. Шаблон содержит вызовы API и массивы исходных данных.
3. **Подготовка скрипта.** В редакторе VBA (например, в Excel) создайте новый модуль, подключите библиотеки API (`NCAPI.dll` для NormCAD и `ncfemapi.dll` для NormFEM) и вставьте экспортированный код.
4. **Передача данных.** Заполните массивы исходных данных (условия, переменные, таблицы узлов и элементов) и передайте их через функции API (см. раздел 3).
5. **Выполнение расчёта.** Запустите расчёт через `ClcCalc` (NormCAD) или `Calc` (NormFEM).
6. **Анализ результатов.** Считайте коэффициенты использования (`MaxResult`), массивы усилий (`GetArrNM`, `GetArrQ`) и перемещений (`GetArrZ`), сравните с нормативными значениями и сформируйте отчёт.
7. **Формирование отчёта.** Сохраните отчёт через `MakeReport` и, при необходимости, оформите его в формате Word (`SendToWord`) <sup>4</sup>.

## 3. API NormCAD

API NormCAD — библиотека `NCAPI.dll`, предоставляющая доступ к модулям NormCAD из внешнего приложения. Ниже приведены ключевые методы и переменные, необходимых для выполнения проверки.

### 3.1 Создание объектов

```
Dim ncApiR As ncApi.Report
Dim Vars As Object

' Создаём экземпляр объекта отчёта
Set ncApiR = New ncApi.Report

' Создаём объект переменных расчётного модуля. Имя библиотеки модуля
' зависит от конкретного нормативного расчёта (пример имени показан в
' документации).
Set Vars = CreateObject("NC_XXXXXXXXXXXXXX_VARS")
```

### 3.2 Настройка переменных задания

Перед передачей данных необходимо задать переменные:

- `Norm` – название нормативного документа (например, "СП 16.13330.2017") <sup>5</sup>.
- `TaskName` – название задания/расчёта (отображается в отчёте) <sup>5</sup>.
- `Unit` – перечень пунктов задания (строка, например "п.5.2, п.5.3") <sup>6</sup>.

### 3.3 Передача исходных данных и условий

Метод	Назначение
<code>SetVars(Vars As Object)</code>	Передаёт в расчётный модуль объект переменных (значения исходных параметров) <sup>7</sup> .
<code>SetConds(ArConds() As String)</code>	Передаёт массив условий расчёта <sup>7</sup> .
<code>LoadDat(File As String)</code> / <code>SaveDat(File As String)</code>	Загружает/сохраняет файлы исходных данных (*.dat) <sup>8</sup> .
<code>LoadNr1(File As String)</code> / <code>SaveNr1(File As String)</code>	Загружает/сохраняет файлы условий (*.nr1) <sup>8</sup> .
<code>ClcLoadNorm</code>	Загружает нормативный модуль (до вызова нужно присвоить <code>Norm</code> имя модуля) <sup>9</sup> .
<code>ClcLoadData</code>	Загружает данные в модуль (после <code>LoadDat</code> или <code>SetVars</code> ) <sup>10</sup> .
<code>ClcLoadConds</code>	Загружает условия расчёта (после <code>LoadNr1</code> или <code>SetConds</code> ) <sup>10</sup> .
<code>LoadProp</code>	Загружает параметры расчёта из реестра (использовать при необходимости) <sup>11</sup> .

### 3.4 Запуск расчёта и получение результатов

- Запуск вычислений: `ClcCalc` (предварительно должны быть заданные переменные `TaskName` и `Unit`) <sup>11</sup>.
- Получение отчёта: `MakeReport(File As String)` сохраняет отчёт в файл, `SendToWord(File As String)` сохраняет отчёт с рамкой и штампом в формате Word <sup>4</sup>.
- Проверка ключа: `TestKey()` возвращает `True`, если аппаратный ключ защиты подключен <sup>4</sup>.
- Максимальный коэффициент использования: `MaxResult` – `read-only` свойство, возвращает максимальное значение коэффициента использования по проверяемому элементу <sup>12</sup>.

Пример минимального вызова расчёта:

```
Sub RunCheck()
    ' Настройка имени норматива и задания
    ncApiR.Norm = "СП 16.13330.2017"
    ncApiR.TaskName = "Проверка балки"
    ncApiR.Unit = "п.5.2"

    ' Загрузка нормативного модуля
    ncApiR.ClcLoadNorm

    ' Передача данных и условий (пример с заранее заполненными массивами)
    ncApiR.SetVars Vars
```

```

ncApiR.SetConds ArConds
ncApiR.ClcLoadData
ncApiR.ClcLoadConds

' Запуск расчёта
ncApiR.ClcCalc

' Сохранение отчёта
ncApiR.MakeReport "C:\\Reports\\Balka_Report.txt"

MsgBox "Максимальный коэффициент использования: " & ncApiR.MaxResult
End Sub

```

## 4. API NormFEM

API NormFEM — библиотека `ncfemapi.dll` для взаимодействия с модулем конечных элементов **NormFEM**. Она используется, когда необходимо определить усилия и перемещения в элементах конструкции и затем передать их в NormCAD для нормативной проверки.

### 4.1 Создание и настройка объекта

```

Dim nfApi As ncfemapi.Main
Set nfApi = New ncfemapi.Main

```

Перед использованием необходимо задать путь к папкам, временным файлам и имени проекта, а также указать форму, содержащую элементы для отображения прогресса и протокола:

```

nfApi.SetPath AppPath:="C:\\Program Files\\NormFEM", _
               TempDir:="C:\\Temp", _
               Project:="FrameProject", _
               ParentForm:=Me ' форма с txtProgress и txtReport

```

### 4.2 Передача данных (таблицы)

Данные передаются таблицами, где имя таблицы (`TblName`) соответствует расширению файла в проекте. Основные таблицы:

Таблица	Назначение	Структура	Источник
g01	Основные узлы (нагруженные и опорные)	Строки: номер группы узлов; X, Y, Z – координаты <sup>13</sup> .	Руководство
g02	Все узлы	Строки: X, Y, Z – координаты <sup>14</sup> .	Руководство
g03	Элементы	Строки: номер группы элементов; номер узла 1; номер узла 2; индекс элемента <sup>15</sup> .	Руководство

Таблица	Назначение	Структура	Источник
...	Другие таблицы (материалы, нагрузки и т.д.)	Описание таблиц находится в файле <b>Tables.ini</b> и справке NormFEM <sup>16</sup> .	Руководство

Пример передачи массива для таблицы **Материалы**:

```
Dim Arr() As String
ReDim Arr(1 To 10) ' Пример размера
' Заполнить массив строками таблицы
nfApi.SetArr "m00", Arr
```

#### 4.3 Запуск расчёта и остановка

- **Calc** — запускает расчёт <sup>17</sup>.
- **StopEx** — останавливает выполнение (если нужно прервать) <sup>18</sup>.

#### 4.4 Получение результатов

После расчёта можно получить массивы с результатами:

Функция	Описание	Особенности
<b>GetArrZ(ArrZ() As Double)</b>	Возвращает массив перемещений. Индекс <b>k</b> определяет направление (X, Y, Z и углы) и зависит от типа задачи (плоская/пространственная). Параметр <b>m</b> определяет тип нагрузки (вся или длительная) <sup>19</sup> .	
<b>GetArrNM(ArrNM() As Double)</b>	Возвращает массив нормальных сил и изгибающих моментов. Индекс <b>k</b> задаёт тип усилия (нормальная сила или момент в начале/конце элемента) <sup>20</sup> .	

Функция	Описание	Особенности
GetArrQ(ArrQ() As Double)	Возвращает массив поперечных сил; для плоских систем доступны силы в начале/конце элемента, для пространственных — вдоль осей X и Y <sup>21</sup> .	
GetArr(TblName As String, Arr() As String)	Возвращает массив данных для указанной таблицы (обратная операция к SetArr) <sup>22</sup> .	

#### 4.5 Дополнительные свойства

- **Steps** — количество промежуточных точек на элемент для детализации распределения усилий <sup>23</sup>.
- **Mode3D** (только чтение) — возвращает **True** для трёхмерной задачи и **False** для плоской <sup>24</sup>.

#### 4.6 Пример использования API NormFEM

```

Sub FEMAnalysis()
    Dim nfApi As ncfemapi.Main
    Set nfApi = New ncfemapi.Main

    ' Настройка путей и проекта
    nfApi.SetPath "C:\\NormFEM", "C:\\Temp", "FrameExample", Me

    ' Передача таблиц (пример заполнения массива материалов)
    Dim mArr() As String
    ReDim mArr(1 To 2)
    mArr(1) = "Steel; 2.55e5; 0.3"
    mArr(2) = "Concrete; 3e4; 0.2"
    nfApi.SetArr "m00", mArr

    ' Передача таблиц узлов и элементов (g01, g02, g03)
    ' ... (заполнение массивов ArG01(), ArG02(), ArG03())

    ' Запуск расчёта
    nfApi.Calc

    ' Получение перемещений
    Dim ArrZ() As Double
    nfApi.GetArrZ ArrZ

    ' Дальнейшая обработка результатов...

```

```
' Освобождение объекта  
Set nfApi = Nothing  
End Sub
```

## 5. Практические советы

- **Проверяйте версию NormCAD/NormFEM.** API может изменяться между версиями, поэтому убедитесь, что используете актуальную документацию.
- **Работайте в отдельном окружении.** Для интеграции с Excel или Python рекомендуется создать копию проектов и выполнять автоматизацию на копиях, чтобы не потерять исходные данные.
- **Логирование.** Функции `txtProgress` и `txtReport` в форме позволяют отслеживать ход расчёта и протокол; запишите их значения в файл для последующего анализа.
- **Комбинации нагрузок.** Для расчётов по сочетаниям усилий используйте функции NormCAD (например, модуль «РСУ») или пишите отдельные процедуры в VBA, суммирующие результаты, полученные из `GetArrNM` / `GetArrQ`.
- **Генерация отчётов.** После выполнения проверки полезно автоматизировать сохранение отчётов в Word с помощью `SendToWord` и добавить макросы Word для оформления (подписи, штампы).

## Заключение

Данная вики предоставляет базовую структуру для написания скриптов с использованием **API NormCAD** и **API NormFEM**. Следуя шагам подготовки данных, запуска расчёта и чтения результатов, вы можете автоматизировать любую нормативную проверку: расчёт прочности и устойчивости, огнестойкости, теплотехнический анализ и т.п. Используйте приведённые функции и примеры в качестве основы и адаптируйте их под конкретные нормы и требования вашей задачи.

---

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 NCBkP.pdf  
<https://normcad.ru/s/book/NCBkP.pdf>

21 22 23 24 normcad.ru  
[https://normcad.ru/s/API\\_NormFEM.htm](https://normcad.ru/s/API_NormFEM.htm)