

КАЗАНСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ
ИНСТИТУТ ЭКОЛОГИИ И ПРИРОДОПОЛЬЗОВАНИЯ
Кафедра метеорологии, климатологии и экологии атмосферы

Т.Р. АУХАДЕЕВ

**СОВРЕМЕННЫЕ МЕТОДЫ РАБОТЫ С ДАННЫМИ
ЦИФРОВЫХ МЕТЕОРОЛОГИЧЕСКИХ АРХИВОВ**

Учебно-методическое пособие

Казань, 2022

УДК 681.518.3
ББК 26.23

Принято на заседании учебно-методической комиссии Института экологии и природопользования КФУ Протокол № 3 от 11 мая 2022 года

Рецензенты:

кандидат географических наук,
старший преподаватель кафедры метеорологии, климатологии и экологии атмосферы КФУ **М.В. Сабирова**
кандидат географических наук,
доцент кафедры ландшафтной экологии КФУ **А.Г. Шарифуллин**

Аухадеев Т.Р.

Современные методы работы с данными цифровых метеорологических архивов / Т.Р. Аухадеев– Казань: Казан. ун-т, 2022. – 31 с.

Учебно-методическое пособие содержит в себе методологические основы работы с данными для подготовки курсовых и выпускных квалификационных работ студентов, обучающихся по направлению 05.03.04 «Гидрометеорология», может быть использовано также для проведения занятий по практическим курсам «Климаты России и сопредельных территорий», «Климатология».

Пособие содержит сведения по методам численного решения задач, некоторые начальные сведения по программированию на языке Visual Basic for Applications и Python и практические задания для самостоятельной работы студентов. Предполагается, что пользователь имеет опыт работы в операционной системе Windows и знаком с основами работы в табличном процессоре MS Excel, знает основы среды VBA (Visual Basic for Applications) или Python.

© Аухадеев Т.Р., 2022

© Казанский университет, 2022

СОДЕРЖАНИЕ

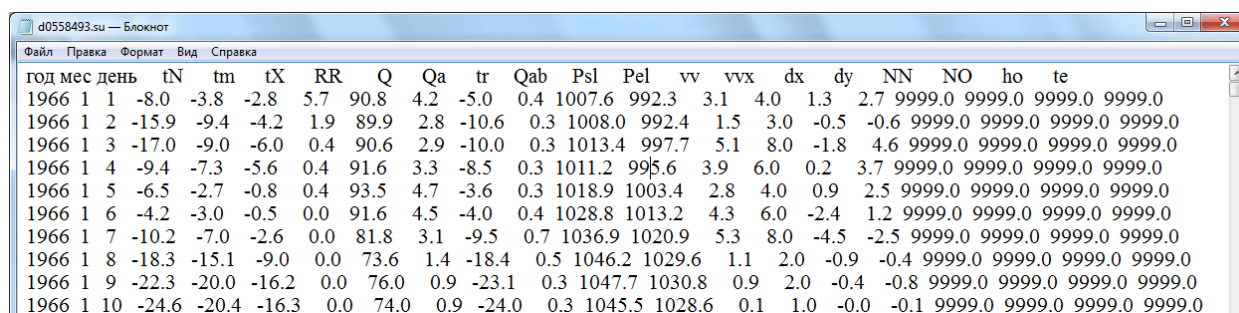
1. ОБЗОР ИСХОДНЫХ ДАННЫХ	4
2. ПОСЛЕДОВАТЕЛЬНОСТЬ ИМПОРТА ДАННЫХ ИЗ АРХИВА	5
2.1. Этап 1. Подготовительный	5
2.2. Этап 2. Вариант импорта данных одной из станций в табличный процессор Microsoft Excel с применением встроенных функций	5
2.3. Вариант импорта данных в табличный процессор Microsoft Excel с помощью написания авторских макросов	7
2.4. Пример выполнения импорта суточных метеорологических данных	15
2.5. Задания для выполнения	18
3. ПОСЛЕДОВАТЕЛЬНОСТЬ ИМПОРТА ДАННЫХ ИЗ АРХИВА СРЕДСТВАМИ СРЕДЫ ПРОГРАММИРОВАНИЯ PYTHON	19
3.1. Данные при работе с которыми рекомендуется использовать Python	20
3.2. Этап 1. Подготовительный. Установка и настройка ПО	21
3.3. Этап 2. Работа с данными архива средствами Python	22
4. ПРИВЕДЕНИЕ ДАННЫХ К УДОБНОМУ ДЛЯ ДАЛЬНЕЙШЕЙ ОБРАБОТКИ ВИДУ СРЕДСТВАМИ VBA	26
5. ИТОГОВЫЕ ЗАДАНИЯ	30

1. ОБЗОР ИСХОДНЫХ ДАННЫХ

Данные фонда ВНИИГМИ-МЦД, как правило, представляют из себя именованные файлы с специфическим расширением «.su», «.v2» или «.bak». Имя файла информативно и представляют собой координаты места метеорологической станции, так файл с суточными данными по станции Казань ЦГМС имеющегося архива назван «d0558493.su» (55,8° с.ш., 49,3° в.д.).

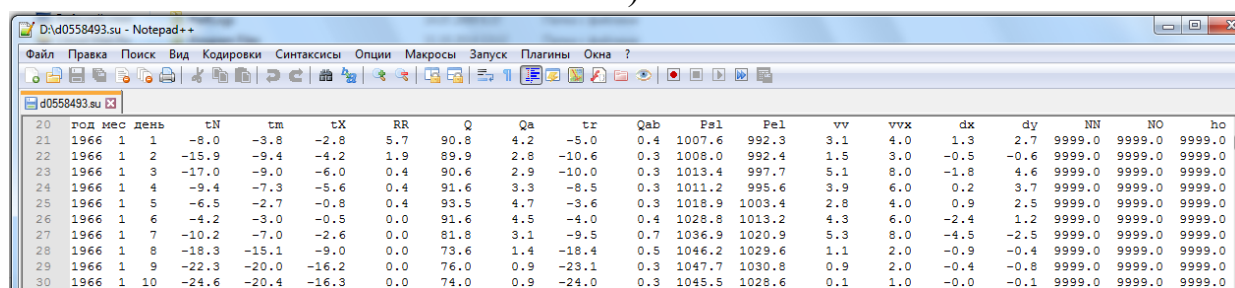
Вне зависимости от расширения файлов с данными, файлы данных являются последовательными (содержат внутри себя только текст в виде последовательности символов и не содержат конструкций, управляющих выводом этого текста на экран).

Для обзора таких файлов можно использовать простейшие текстовые редакторы, предназначенные для создания и редактирования текстовых документов (Блокнот, Notepad++ и т.д.). Общий вид представления данных приведен на рис. 1.



год	мес	день	tN	tm	tX	RR	Q	Qa	tr	Qab	Psl	Pel	vv	vvx	dx	dy	NN	NO	ho	te
1966	1	1	-8.0	-3.8	-2.8	5.7	90.8	4.2	-5.0	0.4	1007.6	992.3	3.1	4.0	1.3	2.7	9999.0	9999.0	9999.0	9999.0
1966	1	2	-15.9	-9.4	-4.2	1.9	89.9	2.8	-10.6	0.3	1008.0	992.4	1.5	3.0	-0.5	-0.6	9999.0	9999.0	9999.0	9999.0
1966	1	3	-17.0	-9.0	-6.0	0.4	90.6	2.9	-10.0	0.3	1013.4	997.7	5.1	8.0	-1.8	4.6	9999.0	9999.0	9999.0	9999.0
1966	1	4	-9.4	-7.3	-5.6	0.4	91.6	3.3	-8.5	0.3	1011.2	995.6	3.9	6.0	0.2	3.7	9999.0	9999.0	9999.0	9999.0
1966	1	5	-6.5	-2.7	-0.8	0.4	93.5	4.7	-3.6	0.3	1018.9	1003.4	2.8	4.0	0.9	2.5	9999.0	9999.0	9999.0	9999.0
1966	1	6	-4.2	-3.0	-0.5	0.0	91.6	4.5	-4.0	0.4	1028.8	1013.2	4.3	6.0	-2.4	1.2	9999.0	9999.0	9999.0	9999.0
1966	1	7	-10.2	-7.0	-2.6	0.0	81.8	3.1	-9.5	0.7	1036.9	1020.9	5.3	8.0	-4.5	-2.5	9999.0	9999.0	9999.0	9999.0
1966	1	8	-18.3	-15.1	-9.0	0.0	73.6	1.4	-18.4	0.5	1046.2	1029.6	1.1	2.0	-0.9	-0.4	9999.0	9999.0	9999.0	9999.0
1966	1	9	-22.3	-20.0	-16.2	0.0	76.0	0.9	-23.1	0.3	1047.7	1030.8	0.9	2.0	-0.4	-0.8	9999.0	9999.0	9999.0	9999.0
1966	1	10	-24.6	-20.4	-16.3	0.0	74.0	0.9	-24.0	0.3	1045.5	1028.6	0.1	1.0	-0.0	-0.1	9999.0	9999.0	9999.0	9999.0

а)



20	год	мес	день	tN	tm	tX	RR	Q	Qa	tr	Qab	Psl	Pel	vv	vvx	dx	dy	NN	NO	ho	te
21	1966	1	1	-8.0	-3.8	-2.8	5.7	90.8	4.2	-5.0	0.4	1007.6	992.3	3.1	4.0	1.3	2.7	9999.0	9999.0	9999.0	9999.0
22	1966	1	2	-15.9	-9.4	-4.2	1.9	89.9	2.8	-10.6	0.3	1008.0	992.4	1.5	3.0	-0.5	-0.6	9999.0	9999.0	9999.0	9999.0
23	1966	1	3	-17.0	-9.0	-6.0	0.4	90.6	2.9	-10.0	0.3	1013.4	997.7	5.1	8.0	-1.8	4.6	9999.0	9999.0	9999.0	9999.0
24	1966	1	4	-9.4	-7.3	-5.6	0.4	91.6	3.3	-8.5	0.3	1011.2	995.6	3.9	6.0	0.2	3.7	9999.0	9999.0	9999.0	9999.0
25	1966	1	5	-6.5	-2.7	-0.8	0.4	93.5	4.7	-3.6	0.3	1018.9	1003.4	2.8	4.0	0.9	2.5	9999.0	9999.0	9999.0	9999.0
26	1966	1	6	-4.2	-3.0	-0.5	0.0	91.6	4.5	-4.0	0.4	1028.8	1013.2	4.3	6.0	-2.4	1.2	9999.0	9999.0	9999.0	9999.0
27	1966	1	7	-10.2	-7.0	-2.6	0.0	81.8	3.1	-9.5	0.7	1036.9	1020.9	5.3	8.0	-4.5	-2.5	9999.0	9999.0	9999.0	9999.0
28	1966	1	8	-18.3	-15.1	-9.0	0.0	73.6	1.4	-18.4	0.5	1046.2	1029.6	1.1	2.0	-0.9	-0.4	9999.0	9999.0	9999.0	9999.0
29	1966	1	9	-22.3	-20.0	-16.2	0.0	76.0	0.9	-23.1	0.3	1047.7	1030.8	0.9	2.0	-0.4	-0.8	9999.0	9999.0	9999.0	9999.0
30	1966	1	10	-24.6	-20.4	-16.3	0.0	74.0	0.9	-24.0	0.3	1045.5	1028.6	0.1	1.0	-0.0	-0.1	9999.0	9999.0	9999.0	9999.0

б)

Рис. 1. Вид представления данных в окне текстового редактора «Блокнот» (а); «Notepad++» (б)

Приоритетным является работа с данными в редакторе Notepad++, благодаря более удобному представлению в нем структуры массивов. Данные в файлах структурированы следующим образом:

- По оси «X» в заголовочной строке представлены обозначения метеопараметров;
- По оси «Y» представлена шкала времени с шагом в 1 сутки (для суточных данных) , в 3 часа (для срочных);
- Под каждым из заголовков представлены среднесуточные (или срочные) значения метеовеличин;

В качестве десятичного делителя в исходном файле используется точка. D В качестве константы пропуска данных зачастую используется значение «9999.0».

2. ПОСЛЕДОВАТЕЛЬНОСТЬ ИМПОРТА ДАННЫХ ИЗ АРХИВА

2.1. Этап 1. Подготовительный

Для исключения возможности неверного толкования исходных данных средой MS Excel (часто, значения с десятичными долями автоматически понимаются в формате дат, так, например значение «12.1» может быть интерпретировано Excel как «12 января»), можно произвести замену десятичного делителя «.» на «,» до импорта данных. Для этого, в окне текстового редактора Блокнот или Notepad++ используется команда «Найти/Заменить», вызываемая сочетанием горячих клавиш «Ctrl+N» или из меню «Правка» в окне программы.

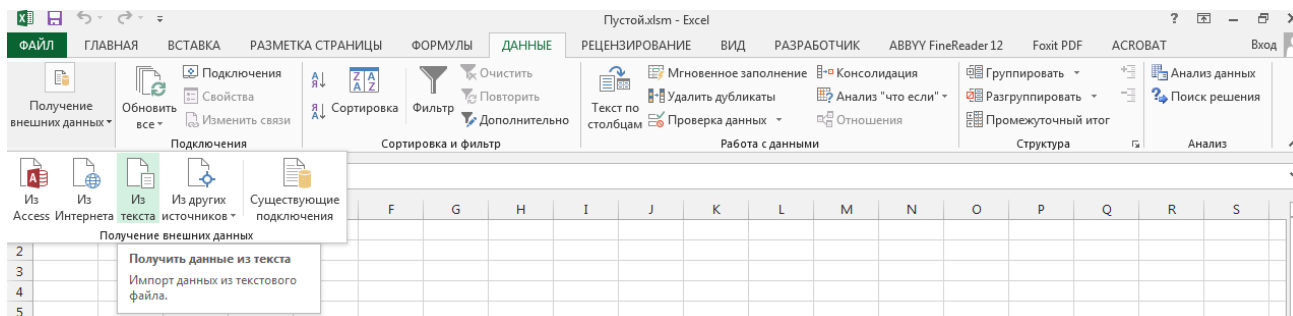
Избежать этой процедуры можно, задав в качестве десятичного делителя точку в подменю «Подробнее» меню «Мастер текстов (импорт)» (рассматривается чуть ниже).

Способ включения данного шага в макрос импорта данных будет рассмотрен ниже. Также будет рассмотрена процедура замены пропусков (в файлах данных они представлены «9999.0»).

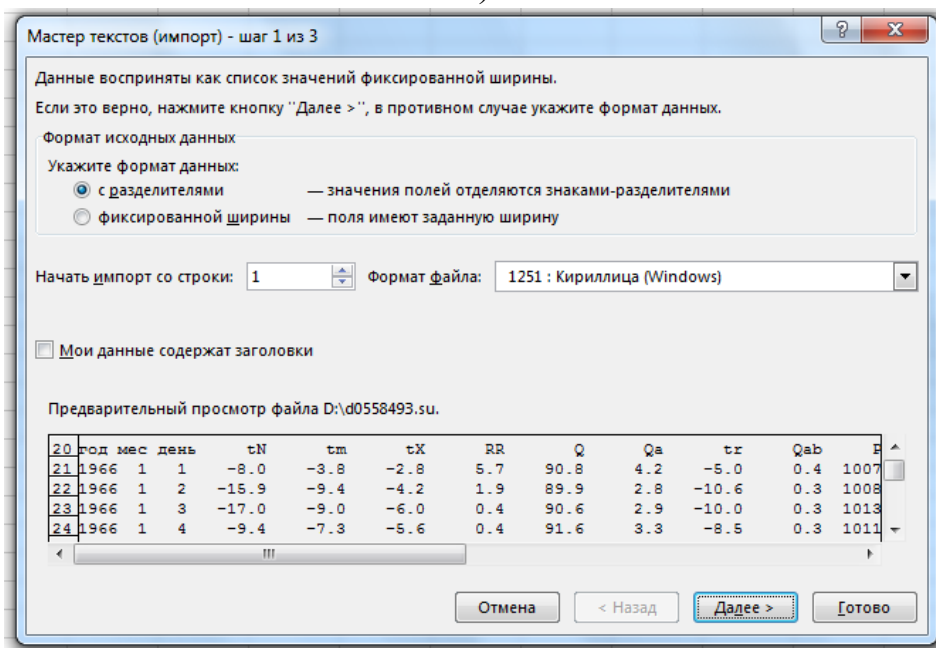
2.2 Этап 2. Вариант импорта данных одной из станций в табличный процессор Microsoft Excel с применением встроенных функций

Для импорта данных из файлов, созданных в сторонних приложениях в среде Excel существует инструментарий получения внешних данных, вызываемый командами «Данные»-«Получение внешних данных»-«Из текста» (рис. 2а). Данный инструментарий именуется как меню «Мастер текстов (импорт)». Далее в проводнике Windows следует выбрать конкретный файл с данными. Для корректного импорта необходимо в окне (рис. 2б) задать в качестве разделителей знак пробела.

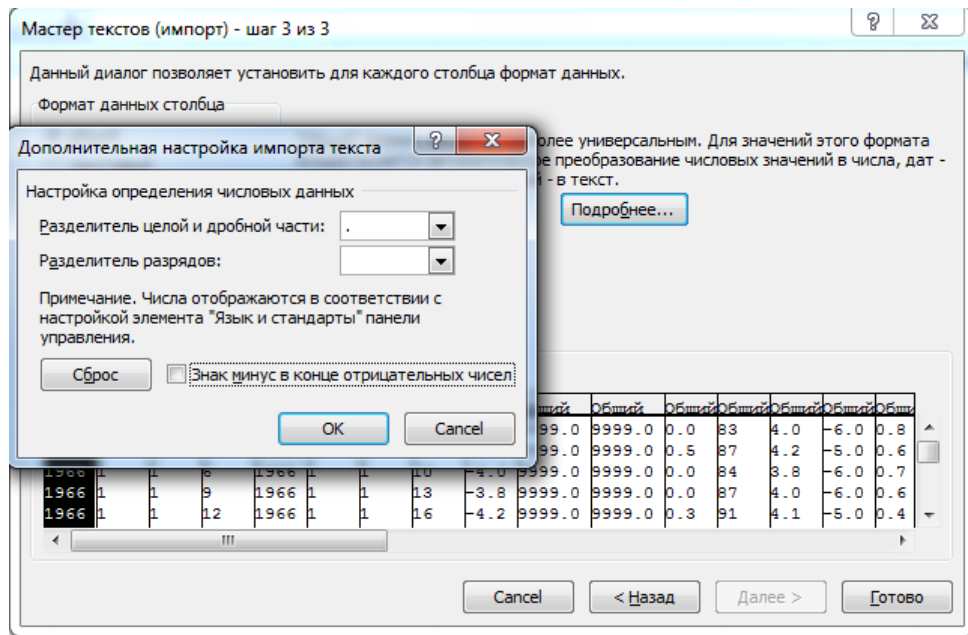
На третьем шаге импорта возможно указать в качестве разделителя целой и дробной части точку (Рис.2в), в этом случае «Шаг первый подготовительный» можно не выполнять».



а)



б)



в)

Рис. 2. Вид окна импорта внешних данных

Данный способ, несомненно, удобен в случае импорта всех данных (всех строк) хранящихся в файле сравнительно небольшого архива данных. Импорт таким образом данных суточного и более высокого (часового, минутного и т.д.) разрешения может вызвать затруднения в связи с переполнением оперативной памяти компьютера.

2.3. Вариант импорта данных в табличный процессор Microsoft Excel с помощью написания авторских макросов

Написание макросов, несмотря на кажущуюся сложность, имеет ряд преимуществ. К которым можно отнести:

- возможность импорта или экспорта не всего массива данных из исходного файла, что значительно экономит время выполнения и не загружает файл ненужными данными;
- возможность параллельного выполнения сопутствующих вычислений в процессе выполнения макроса, импортирующего данные;
- Возможность импорта данных из большого количества файлов-данных в одну или несколько книг Excel.

- Возможность обработки большого массива данных без помещения данных в таблицу MS Excel (данные считываются из файла данных, обрабатываются и записываются в другой txt файл).

Для понимания процесса считывания данных из последовательного файла необходимо привлечение строковых функций VBA. Выборочное считывание данных из файла также удобно производить с использованием строковых функций, некоторые из которых приведены в таблице 1.

Таблица 1

Некоторые строковые функции языка программирования VBA

Функция	Описание
InStr	Возвращает целое число, указывающее начальную позицию первого вхождения одной строки в другую.
InStrRev	Возвращает позицию первого вхождения одной строки в другую, начиная с правого конца строки.
Left	Возвращает строку, содержащую указанное число знаков с левой стороны строки.
Len	Возвращает целое число, показывающее число знаков в строке.
LTrim	Возвращает строку, содержащую копию указанной строки без пробелов в начале.
Mid	Возвращает строку, содержащую указанное число знаков строки.
Right	Возвращает строку, содержащую указанное число знаков с правой стороны строки.
RTrim	Возвращает строку, содержащую копию указанной строки без пробелов в конце.
Space	Возвращает строку, состоящую из указанного числа пробелов.
Trim	Возвращает строку, содержащую копию указанной строки без пробелов в начале и конце.

Для организации импорта данных из текстовых файлов необходимо также знать основы организации работы с такими файлами средствами VBA.

Для начала работы с текстовым файлом используется оператор открытия файла `open`, имеющей следующий синтаксис:

`Open имя For назначение As номер`

Здесь имя — полное имя файла (строка, содержащая имя файла с путем и расширением), номер — номер файла, назначение — это ключевое слово `Input`, `Output` или `Append`.

- `Input` означает, что файл открывается для считывания из него информации.
- `Output` — файл открывается для записи в него информации.
- `Append` — файл открывается для добавления информации.

В качестве номера файла номер рекомендуется использовать переменную типа `Integer`.

Поставленной в данной методичке задаче соответствует назначение `Input`

Последовательность выполнения импорта данных

- 1) Создать макрос. Для чего клавишами «`Alt+F8`» необходимо вызвать окно управления макросами, в строку названия ввести имя макроса, нажать кнопку «Создать».
- 2) В открывшемся редакторе VBA внутри созданной процедуры, ограниченной ключевыми словами `Sub` и `End Sub` необходимо задать набор команд для выполнения.
- 3) Сохранить результат работы.

Рассмотрим, для начала, импорт данных одной станции (на примере станции Казань). Предположим, из объемного массива срочных данных необходимо импортировать данные лишь температуры воздуха. Согласно рис.1б данные среднесуточной температуры воздуха (с учетом знака) записаны с 37 по 41 символ каждой строки. Подсказкой может служить указатель позиции курсора в нижней части редактора «`Notepad++`» как указано на рис. 3.

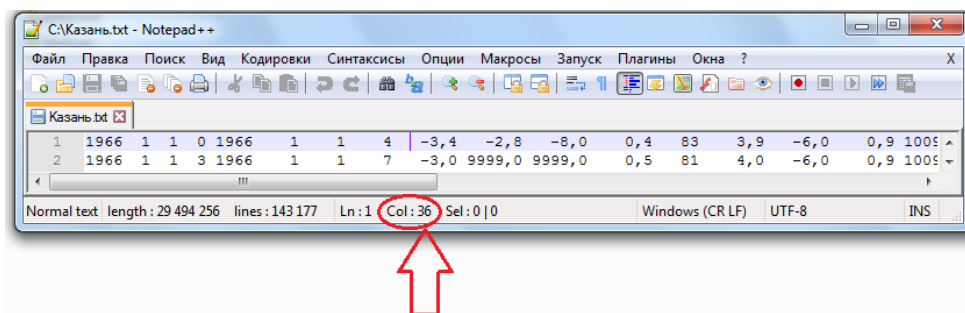


Рис. 3. Определение местоположения курсора в редакторе редактора «Notepad++»

Текст макроса считывания интересующих нас данных в среду MS Excel может быть представлен следующим образом:

```
Sub import1()
Open "C:\...\Казань.txt" For Input As #1 'открываем файл исходных данных
i = 1 'начальное значение строки текущего листа Excel
Do While Not EOF(1) 'выполнение цикла до конца файла (EOF- "End Of File")
Line Input #1, s 'считывание строки из исходного файла и запись ее в переменную s
Cells(i, 1) = Trim(Mid(s, 1, 4)) 'считывание и запись в Excel обозначения ГОДА
Cells(i, 2) = Trim(Mid(s, 6, 2)) 'считывание и запись в Excel обозначения МЕСЯЦА
Cells(i, 3) = Trim(Mid(s, 9, 2)) 'считывание и запись в Excel обозначения ДНЯ
Cells(i, 4) = Trim(Mid(s, 12, 2)) 'считывание и запись в Excel обозначения СРОКА
Cells(i, 5) = Trim(Mid(s, 36, 5)) 'считывание и запись в Excel значения ТЕМПЕРАТУРЫ
i = i + 1 'опреатор инкремента
Loop
Close #1 'закрываем файл исходных данных
End Sub
```

В приведенном макросе используются функции Mid и Trim. Синтаксис функции Mid: Mid(строковая_переменная, начальная_позиция, количество_символов). Функция Trim используется для удаления лишних пробелов в случае если значение имеют длину меньшую заданной в функции Mid. Изменяя параметры начальная_позиция и количество_символов можно добиться получения любой характеристики из указанной строковой переменной.

Такой подход удобен, но его использование для импорта данных из одного единственного файла не всегда оправдано с точки зрения экономии времени. Однако он незаменим при импорте из большого количества файлов. Так, например, на территории ПФО расположено более 200 метеостанций. Импорт данных из такого количества файлов затруднителен с помощью.

Для организации импорта данных из файлов в некоторой папке и лучшего восприятия необходимо выполнить подготовительные операции.

Так, необходимо на отдельный лист вывести названия файлов и путь к файлам в папке. Для чего рекомендуется воспользоваться приведенным ниже макросом (рис. 4).

```

Sub FileList()
Dim V As String
Dim BrowseFolder As String
With Application.FileDialog(msoFileDialogFolderPicker) 'открываем диалоговое окно выбора папки
    .Title = "Выберите папку или диск": .Show 'заголовок диалогового окна
    On Error Resume Next: Err.Clear
    V = .SelectedItems(1) ' переменной V присваивается путь к папке с файлами
End With
BrowseFolder = CStr(V) 'преобразуем переменную V к строковому типу данных
ActiveWorkbook.Sheets.Add 'добавляем лист и выводим на него шапку таблицы
ActiveSheet.Name = "Files" 'задаем имя добавленного листа
Range("A1").Value = "Имя файла"
Range("B1").Value = "Путь"
'вызываем процедуру вывода списка файлов , измените True на False, если не нужно выводить файлы из вложенных папок
ListFilesInFolder BrowseFolder, True
End Sub

```

```

Private Sub ListFilesInFolder(ByVal SourceFolderName As String, ByVal IncludeSubfolders As Boolean)
'Dim FSO As Object: Dim SourceFolder As Object
'Dim SubFolder As Object: Dim FileItem As Object
Set FSO = CreateObject("Scripting.FileSystemObject")
Set SourceFolder = FSO.getfolder(SourceFolderName)
r = 2 'задаем начальную строку для вывода
'далее выводим данные по каждому файлу
For Each FileItem In SourceFolder.Files 'для каждого файла в папке выполняются следующие действия
    Cells(r, 1).Formula = FileItem.Name 'вывод имени файла
    Cells(r, 2).Formula = FileItem.Path 'вывод пути к файлу
    r = r + 1
    X = SourceFolder.Path
Next FileItem 'переход к следующему файлу
'вызываем процедуру повторно для каждой вложенной папки
If IncludeSubfolders Then
    For Each SubFolder In SourceFolder.SubFolders
        ListFilesInFolder SubFolder.Path, True
    Next SubFolder
End If
Columns("A:E").AutoFit 'автоматическое установление ширины столбца
End Sub

```

Рис. 4. Макрос считывания и вывода списка файлов в директории

Результат работы такого макроса на примере папки с данными по станциям Республики Татарстан приведен на рис. 5.

	A	B	
1	Имя файла	Путь	
2	Азнакаево.v2	C:\Татарстан-срочные\Азнакаево.v2	
3	Акташ.v2	C:\Татарстан-срочные\Акташ.v2	
4	Арск.v2	C:\Татарстан-срочные\Арск.v2	
5	Арск.v2	C:\Татарстан-срочные\Арск.v2	
6	Бугульма.v2	C:\Татарстан-срочные\Бугульма.v2	
7	Вязовые.v2	C:\Татарстан-срочные\Вязовые.v2	
8	Дрожжаное.v2	C:\Татарстан-срочные\Дрожжаное.v2	
9	Елабуга.v2	C:\Татарстан-срочные\Елабуга.v2	
10	Елабуга.v2	C:\Татарстан-срочные\Елабуга.v2	
11	Казань.v2	C:\Татарстан-срочные\Казань.v2	
12	Кайбицы.v2	C:\Татарстан-срочные\Кайбицы.v2	
13	Мензелинск.v2	C:\Татарстан-срочные\Мензелинск.v2	
14	Муслюмово.v2	C:\Татарстан-срочные\Муслюмово.v2	
15	Тетюши.v2	C:\Татарстан-срочные\Тетюши.v2	
16	Чулпаново.v2	C:\Татарстан-срочные\Чулпаново.v2	
17			

Рис.5. Результат работы макроса FileList

Полученная таблица позволяет в качестве пути к файлу в макросе *Import1* использовать данные полученной таблицы.

Для импорта данных всех станций заданной папки возможно видоизменить макрос *Import1* к виду в котором автоматизировано перебор и считывание всех текстовых файлов, создание соответствующих станциям листов в конечной книге Excel и запись данных и попутным приведением типов данных к числовому, пригодному для дальнейшей обработки (рис. 6).

```

Sub import2()
Dim StancName() As String
Dim DataWay() As String
iEndRow = Sheets("Files").Cells(Rows.Count, 1).End(xlUp).Row 'Определение количества станций по данным листа "Files"
N = iEndRow - 1
ReDim StancName(N) As String 'массив содержащий названия станций
ReDim DataWay(N) As String 'массив содержащий путь к файлам с данными для каждой станций
For k = 1 To N
Filename = Sheets("Files").Cells(k + 1, 1)
StancName(k) = Left(Sheets("Files").Cells(k + 1, 1), Len(Filename) - 3) 'заполнение массива с названиями станций
' функции Left и Len здесь использованы для того чтобы убрать из названия файла обозначения расширения файлов,
' например из названия "Казань.v2" будут убраны 3 последних символа, в результате чего именем станции станет "Казань"
DataWay(k) = Sheets("Files").Cells(k + 1, 2) 'заполнение массива путь к файлам с данными для каждой станций
ActiveWorkbook.Sheets.Add 'добавляем лист в книгу Excel
ActiveSheet.Name = StancName(k) 'присваиваем имя листу (имя совпадает с названием файла)
Open DataWay(k) For Input As #1 'открываем файл исходных данных
i = 1 ' начальное значение строки текущего листа Excel
Do While Not EOF(1) 'выполнение цикла до конца файла (EOF- "End Of File")
Line Input #1, s 'считывание строки из исходного файла и запись ее в переменную s
Sheets(StancName(k)).Cells(i, 1) = Val(Trim(Mid(s, 1, 4))) 'считывание и запись в Excel обозначения ГОДА
Sheets(StancName(k)).Cells(i, 2) = Val(Trim(Mid(s, 6, 2))) 'считывание и запись в Excel обозначения МЕСЯЦА
Sheets(StancName(k)).Cells(i, 3) = Val(Trim(Mid(s, 9, 2))) 'считывание и запись в Excel обозначения ДНЯ
Sheets(StancName(k)).Cells(i, 4) = Val(Trim(Mid(s, 12, 2))) 'считывание и запись в Excel обозначения СРОКА
Sheets(StancName(k)).Cells(i, 5) = CDbl(Trim(Mid(s, 36, 5))) 'считывание и запись в Excel значения ТЕМПЕРАТУРЫ
i = i + 1 ' оператор инкремента
Loop
Close #1 'закрываем файл исходных данных
Next k
End Sub

```

Рис. 6. Макрос считывания данных из последовательных файлов

Результатом выполнения макроса import2 является книга Excel содержащая количество листов равное количеству файлов с данными в выбранной папке. Причем каждый лист в книге имеет имя, совпадающее с оригинальным названием станции (именем исходного файла).

Импорт данных выполнен!

2.4. Пример выполнения импорта суточных метеорологических данных

В качестве примера рассмотрим импорт исходных суточных данных температуры воздуха из архива, содержащего данные с 19 станций Приволжского федерального округа (1966-2011гг.).

Нередко исходные файлы содержат в себе описание структуры записанных в них данных. Выглядеть такие файлы могут так как представлено на рис. 7.

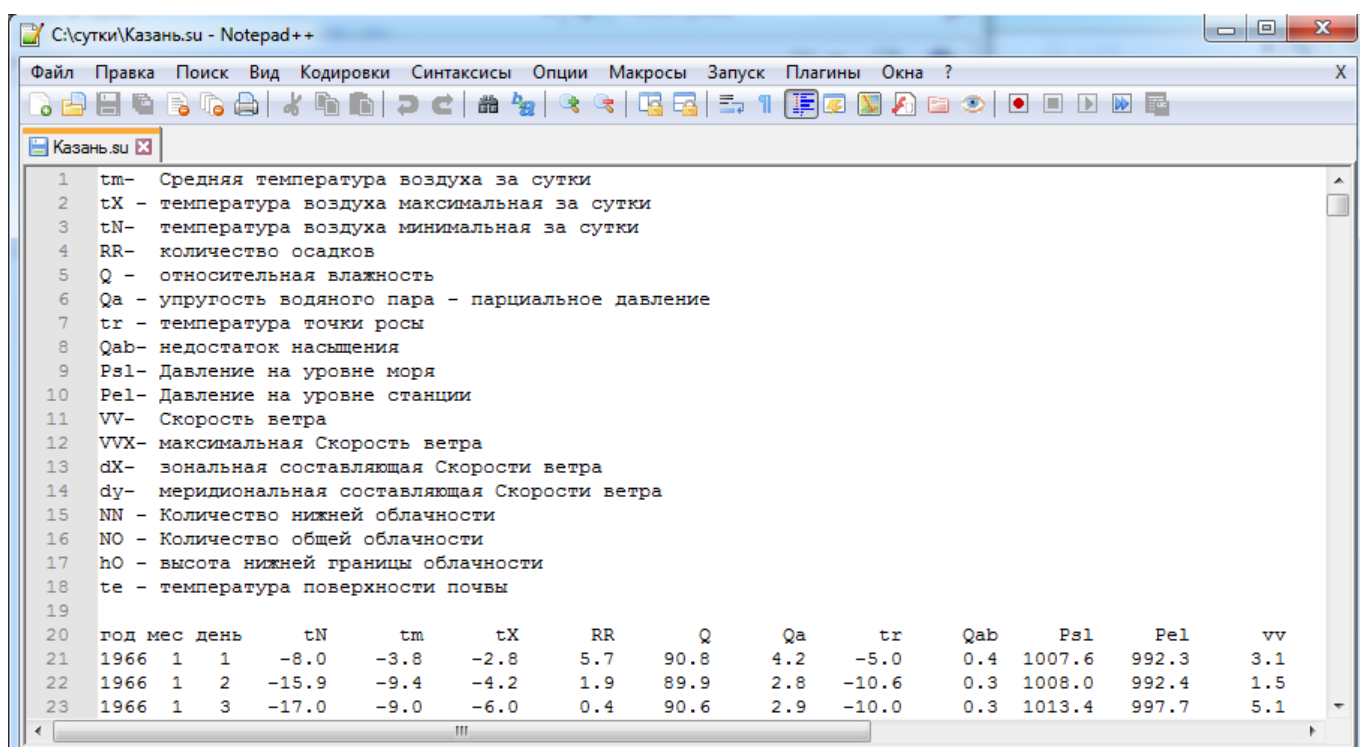


Рис. 7. Вариант представления исходных данных

Отличительной особенностью работы с такими файлами является то, что считывание данных, заносимых в конечную книгу MS Excel, должно начинаться не с первой строки файла, а с заданной (в приведенном на примере случае- с 21).

Существует масса методов добиться считывания данных с произвольной строки. В настоящем методическом указании воспользуемся самым простым- пропуском строк с 1 до 20 и начало занесения данных в таблицу с 21 строки (строки реализации данного метода выделены в приведенном ниже коде (рис. 8).


```

Sub import3()
Dim StancName() As String
Dim DataWay() As String
iEndRow = Sheets("Files").Cells(Rows.Count, 1).End(xlUp).Row 'Определение количества станций по данным листа "Files"
N = iEndRow - 1
ReDim StancName(N) As String 'массив содержащий названия станций
ReDim DataWay(N) As String 'массив содержащий путь к файлам с данными для каждой станций
For k = 1 To N
Filename = Sheets("Files").Cells(k + 1, 1)
StancName(k) = Left(Sheets("Files").Cells(k + 1, 1), Len(Filename) - 3) 'заполнение массива с названиями станций
' функции Left и Len здесь использованы для того чтобы убрать из названия файла обозначения расширения файлов,
' например из названия "Казань.su" будут убраны 3 последних символа, в результате чего именем станции станет "Казань"
DataWay(k) = Sheets("Files").Cells(k + 1, 2) 'заполнение массива путь к файлам с данными для каждой станций
ActiveWorkbook.Sheets.Add 'добавляем лист в книгу Excel
ActiveSheet.Name = StancName(k) 'присваиваем имя листу (имя совпадает с названием файла)
Open DataWay(k) For Input As #1 'открываем файл исходных данных
Sheets(StancName(k)).Cells(1, 1) = "Год": Sheets(StancName(k)).Cells(1, 2) = "Месяц"
Sheets(StancName(k)).Cells(1, 3) = "День": Sheets(StancName(k)).Cells(1, 4) = "Температура"
i = 1 ' начальное значение строки текущего листа Excel
Do While Not EOF(1)
Line Input #1, s 'считывание строки из исходного файла и запись ее в переменную s
If i < 21 Then GoTo 1 ' оператор условного перехода на операцию инкремента в случае не достижения 21 строки

Sheets(StancName(k)).Cells(i - 19, 1) = Trim(Mid(s, 1, 4)) 'считывание и запись в Excel обозначения ГОДА
Sheets(StancName(k)).Cells(i - 19, 2) = Trim(Mid(s, 6, 2)) 'считывание и запись в Excel обозначения МЕСЯЦА
Sheets(StancName(k)).Cells(i - 19, 3) = Trim(Mid(s, 10, 2)) 'считывание и запись в Excel обозначения ДНЯ
Sheets(StancName(k)).Cells(i - 19, 4) = Trim(Mid(s, 22, 6)) 'считывание и запись в Excel значения ТЕМПЕРАТУРЫ

1: i = i + 1 ' оператор инкремента
Loop 'циклический переход к оператору Do
Close #1 'закрываем файл исходных данных
Next k
End Sub

```

Рис. 8. Макрос считывания данных с произвольной строки

2.5. Задания для выполнения

1. Выполните задание приведенной выше в качестве примера.
2. Выполните импорт исходных суточных данных давления на уровне моря из архива, указанного преподавателем.
3. Выполните импорт исходных данных скорости ветра на уровне моря из архива данных, указанного преподавателем.
4. Аналогично выполните импорт других метеопараметров по указанию преподавателя.

3. ПОСЛЕДОВАТЕЛЬНОСТЬ ИМПОРТА ДАННЫХ ИЗ АРХИВА СРЕДСТВАМИ СРЕДЫ ПРОГРАММИРОВАНИЯ PYTHON

Язык программирования Python – это высокоуровневый язык программирования, представленный Гвидо Ван Россумом (Guido Van Rossum) в 1991 году. Python относится к свободно распространяемому программному обеспечению и распространяется под Python Software Foundation License. Это позволяет максимально широко и свободно использовать данный язык программирования в различных проектах, в том числе и научных. Python является мультиплатформенным (Windows/Linux/MacOS) языком программирования, что способствует интеграции как можно большего числа пользователей в единое сообщество. За время своего существования (более 20 лет) в ходе исторического и логического развития в языке python сформировалось две ветви, называемые python2 и python3 соответственно. Последняя является будущим языка, а python2 когда-нибудь исчезнет. Для начинающих пользователей рекомендуется сразу использовать ветвь python3.

Python в целом относится к интерпретируемым языкам программирования с динамической типизацией данных, что дает возможность не определять типы данных в момент описания или при первом обращении к ним, а работать более свободно, на ходу меняя тип данных переменных.

Стандартная библиотека языка python богата возможностями и включает, в частности, модули для работы с сетевыми протоколами, операционной системой и графические библиотеки. Обогащение языка специализированными модулями происходит за счет создания новых модулей и пакетов (объединений модулей). Стандартная библиотека python, дополненная свободно распространяемыми пакетами NumPy, SciPy, Matplotlib и Basemap, позволяет создать универсальную научно-вычислительную среду,

Популярность python обеспечивается простым и прозрачным синтаксисом языка, его стабильностью и широкими возможностями. Python одинаково

подходит как для обучения студентов программированию, так и для анализа данных.

3.1. Данные при работе с которыми рекомендуется использовать

Python

Архивы абсолютного большинства зарубежных климатических центров используют для хранения данных формат данных NetCDF.

NetCDF – это машинно-независимый, ориентированный на массив, многомерный, самоописываемый и переносимый формат данных, используемый различными научными сообществами. У него есть расширение файла. *NCили.cdf* (хотя считается, что между ними есть тонкие различия). В отличие от файлов *v.csvили.xlsx*, Формат NetCDF не может быть доступен и просмотрен напрямую с помощью программного обеспечения для работы с электронными таблицами.

Для несложных манипуляций над небольшими файлами в формате NetCDF3 (ныне устаревший формат) возможно использование надстройки на Microsoft Excel представляющая из себя набор макросов для отбора данных из файла и представления их в двумерном табличном виде. Надстройка снабжена понятным пользовательским меню.

Указанная надстройка свободна для распространения и доступна по ссылке <https://netcdf4excel.github.io>

Однако использование такой надстройки связано с рядом неудобств и ограничений.

К ограничениям относится:

- невозможность ее использования на продуктах Microsoft Office 2013 и более новых;
- невозможность работы с файлами в формате NetCDF4;
- невозможность работы с файлами большого объема.

Неудобство применения надстройки заключается в возможности лишь двумерного представления многомерных данных. Таким образом для расчета стандартного набора климатологических показателей пользователь вынужден неоднократно преобразовывать данных меняя структуру их представления. Совокупность этих манипуляций может стать источником ошибок.

Все указанные ограничения и неудобства не актуальны при использовании для обработки и визуализации данных языка программирования Python.

3.2. Этап 1. Подготовительный. Установка и настройка ПО

Для установки интерпретатора Python на ваш компьютер, первое, что нужно сделать – это скачать дистрибутив. Загрузить его можно с официального сайта, перейдя по ссылке <https://www.python.org/downloads/>

Для операционной системы Windows дистрибутив распространяется либо в виде исполняемого файла (с расширением exe), либо в виде архивного файла (с расширением zip).

Для удобства запуска примеров и изучения языка Python, советуем установить на свой ПК пакет Anaconda или PyCharm. Этот пакет включает в себя интерпретатор языка Python (есть версии 2 и 3), набор наиболее часто используемых библиотек и удобную среду разработки и исполнения, запускаемую в браузере.

Установка Anaconda

Для установки этого пакета, предварительно нужно скачать дистрибутив <https://www.continuum.io/downloads>.

Есть варианты под Windows, Linux и MacOS.

Для корректной работы программы при установке. Укажите дополнительные опции:

- Add Anaconda to the system PATH environment variable – добавить Anaconda в системную переменную PATH

- Register Anaconda as the system Python 3— использовать Anaconda, как интерпретатор Python 3 по умолчанию.

Установка PyCharm

Если в процессе разработки вам необходим отладчик и вообще вы привыкли работать в IDE, а не в текстовом редакторе, то тогда одним из лучших вариантов будет IDE PyCharm от JetBrains. Для скачивания данного продукта нужно перейти по ссылке <https://www.jetbrains.com/pycharm/download/>

IDE доступна для Windows, Linux и MacOS. Существуют два вида лицензии PyCharm – это Professional и Community. Мы будем использовать версию Community, так как она бесплатна и её функционала более чем достаточно для наших задач.

Для эффективной работы с данными помимо стандартных (уже встроенных) пакетов необходимы и дополнительные. Так, для манипуляций с файлами NetCDF необходим пакет NetCDF4.

Для установки дополнительных пакетов необходимо воспользоваться менеджером пакетов `pip`. Для чего, на компьютере с уже предустановленным Python необходимо вызвать командную строку (например сочетанием клавиш Win+R) и в открывшееся окно ввести команду:

```
pip install netCDF4
```

После ввода команды в окне командной строки отобразится информация о успешном завершении установки пакета и его расположении.

Приведенный алгоритм применим для установки любых пакетов совместимых с Python 3 и Windows.

3.3. Этап 2. Работа с данными архива средствами Python

В качестве примера рассмотрим архив данных по среднемесячной температуре воздуха Университета Восточной Англии Делавер (Delaware). Архив доступен для скачивания по ссылке:

```
https://psl.noaa.gov/data/gridded/data.UDel_AirT_Precip.html
```

По указанной ссылке скачаем файл данных. Например `air.mon.v501.nc`

Для начала работы с данными получим описание файла, для чего в редактора Python выполним следующие команды:

```
import netCDF4 as nc
fn="D:\\air.mon.mean.v501.nc"
ds=nc.Dataset(fn)
print(ds)
ds.close()
```

Результат работы приведенного скрипта на рис. 9.

```
1 import netCDF4 as nc           #импорт библиотеки netCDF4 и указание ее псевдонима "nc" для дальнейших вызовов
2 fn="D:\\air.mon.mean.v501.nc" #указание пути к файлу с данными
3 ds=nc.Dataset(fn)            #получение содержимого файла в переменную ds
4 print(ds)                    #вывод описания указанного файла
5 ds.close()                   #закрываем файл данных

<class 'netCDF4._netCDF4.Dataset'>
root group (NETCDF4_CLASSIC data model, file format HDF5):
  Conventions: CF-1.0
  history: created 2/2016 by CAS NOAA/ESRL PSD
  version: 5.01
  dataset_title: Terrestrial Air Temperature and Precipitation: 1900-2017 Gridded Monthly Time Series
  Source: http://climate.geog.udel.edu/~climate/html_pages/download.html
  References: https://www.psl.noaa.gov/data/gridded/data.UDel_AirT_Precip.html
  title: Terrestrial Air Temperature and Precipitation: V5.01
  dimensions(sizes): lat(360), lon(720), time(1416)
  variables(dimensions): float32 lat(lat), float32 lon(lon), float64 time(time), float32 air(time,lat,lon)
```

Рис. 9. Результат работы кода отображения содержимого файла.

Вывод содержимого файла содержит следующую информацию: формат, источник (ссылка или краткое описание), размерности, типы, названия переменных, хранящихся в файле.

Согласно выведенной информации в примере мы имеем дело с файлом в формате netCDF4, размерность файла -360 значений широты (переменная lat), 720 значений долготы (переменная lon), 1416 значений по времени (переменная time). Значения температуры хранятся в трехмерной матрице air(time,lat,lon).

Для просмотра значений параметров (широт, долгот, времени, температуры) необходимо обратиться к выбранной переменной. Например, одной из нижеприведенных строк:

```
lats=ds.variables['lat']
lons=ds.variables['lon']
t=ds.variables['air']
```

Вывод результата командой `print(lats[:])` выведет все значения широт содержащихся в переменной `lats`. Аналогичные манипуляции можно проводить и для других переменных.

Таким образом, обозревая значения `lats` и `lons` возможен выбор узла или территории (множества узлов) с данными для выборки временного ряда.

Для примера выберем данные для узла ближайшего к Казани (широта- 55° с.ш., долгота- 49° в.д.

Вывод содержимого переменной `lats` показал, что ближайшей к Казани является узел с широтой $55,25^{\circ}$ с.ш. и это элемент массива `lats` под номером 69. Аналогично поступаем с долготой, определяем, что ближайшим элементом по долготе является элемент массива `lons` с номером 98. (долгота $49,25^{\circ}$ в.д.).

На основании вышеизложенного выборка данных для узла с широтой $55,25^{\circ}$ с.ш. и долготой $49,25^{\circ}$ в.д. возможна так:

```
t=ds.variables['air'][:,69,98]
```

В переменную `t` помещены все 1416 значений среднемесячной температуры воздуха для узла с `lats[69]` и `lons[98]`. Переменная `t` представляет из себя одномерный массив с размером 1416 элементов.

Сохраним полученный результат в универсальном типе файлов- текстовом (`txt`). Для таких целей удобно использовать функцию `savetxt`, входящую в состав библиотеки `numpy`.

Таким образом выборка данных для одного пункта сведется к выполнению кода (рис. 10).

```
1 import netCDF4 as nc           #импорт библиотеки netCDF4
2 import numpy as np           #импорт библиотеки numpy
3 fn="D:\\air.mon.mean.v501.nc" #указание пути к файлу с исходными данными
4 OutF="D:\\T-Казань.txt"      #указание пути к файлу с результатом
5 ds=nc.Dataset(fn)           #получение содержимого файла в переменную ds
6 t=ds.variables['air'][:,69,98] #выборка данных для одного узла (Казань)
7 np.savetxt(OutF, t, fmt='%.2f', delimiter=' ') #сохранение отобранных данных
8 ds.close()
```

Рис. 10. Код выборки данных для одного узла

Параметр *fmt* внутри функции `savetxt` позволяет задать формат сохраняемых данных. Так, указывая *fmt* = '%.2f' мы задаем формат, предполагающий точность ограниченную двумя знаками после запятой.

Параметр *delimiter* = ' ' позволяет задать символ разделения между значениями.

4. ПРИВЕДЕНИЕ ДАННЫХ К УДОБНОМУ ДЛЯ ДАЛЬНЕЙШЕЙ ОБРАБОТКИ ВИДУ СРЕДСТВАМИ VBA

Предлагается выполнить приведение суточных данных по температуре воздуха к удобному для обработки виду. За основу примем результат выполнения примерного задания, приведенного выше. В данном пособии для примера рассмотрим архив суточных данных метеорологических наблюдений на 19 станциях ПФО.

Результатом выполнения задания 1 на приведенных в примере данных является книга MS Excel содержащая количество листов равное количеству станций (файлов с данными) в заданной папке. На каждом из листов данные представлены в виде таблицы вида (рис. 11):

	A	B	C	D	E
1	Год	Месяц	День	Температура	
2	1966	1	1	-3,8	
3	1966	1	2	-9,4	
4	1966	1	3	-9	
5	1966	1	4	-7,3	
6	1966	1	5	-2,7	
7	1966	1	6	-3	
8	1966	1	7	-7	
9	1966	1	8	-15,1	
10	1966	1	9	-20	
11	1966	1	10	-20,4	

Рис.11. Вид таблиц полученных после импорта данных из файлов

Полученные для каждой станции (с длинной ряда с1966 по 2011гг.) таблицы содержат 17112строк. Анализ данных в таком виде затруднителен, в результате чего возникает необходимость транспонирования данной таблицы. Одним из наиболее удобных видов представления метеорологических данных является вид при котором значения помещаются в матрицу, где в каждом из столбцов содержатся данные конкретного года (Рис. 12).

	A	B	C	D	E	F	G	H	I	J	K
1	Месяц	День	1966	1967	1968	1969	1970	1971	1972	1973	
2		1	1								
3		1	2								
4		1	3								
5		1	4								
6		1	5								
7		1	6								
8		1	7								
9		1	8								
10		1	9								
11		1	10								
12		1	11								
13		1	12								

Рис. 12. Одна из форм представления данных

Перед началом приведения данных к указанному виду, во избежание возникновения ошибок, необходимо выявить каким образом представлены ряды високосных и невисокосных лет. Для удобства дальнейшей обработки данных, часто в рядах отведены пустые строки для 29, 30, 31 февраля. (таким образом, все годы, независимо от их календарной продолжительности имеют одинаковую длину ряда- 366 (если пустые строки отведены только для 29 февраля), 367 (если для 29 и 30) или 368 (если для 29, 30, 31 февраля) дней. Кроме того, возможен вариант когда в исходных рядах на каждый месяц отведено по 31 строке, в таком случае в месяцы продолжительностью в 30 дней напротив 31 суток имеется пропуск в данных (в таком случае количество дней в ряду 1 года составляет $31 \cdot 12 = 372$ дня).

Одновременно с транспонированием матрицы данных проведем исключение кодов пропусков в данных и оставим на их месте пустые ячейки. Как указывалось ранее, пропуски в данных закодированы и имеют обозначение «9999».

Для примера, расположим новую транспонированную матрицу на тех же листах что и исходные, сместив ее на 7 столбцов вправо.

Пример получившегося кода VBA представлен на рис. 13.

```

Sub import4()
Dim StancName() As String
Dim DataWay() As String
iEndRow = Sheets("Files").Cells(Rows.Count, 1).End(xlUp).Row
N = iEndRow - 1
ReDim StancName(N) As String
For k = 1 To N 'цикл по листам
Filename = Sheets("Files").Cells(k + 1, 1)
StancName(k) = Left(Sheets("Files").Cells(k + 1, 1), Len(Filename) - 3)
Sheets(StancName(k)).Cells(1, 6) = "Месяц": Sheets(StancName(k)).Cells(1, 6) = "День" 'Заголовки

    For g = 1 To 46 ' длина ряда в нашем примере равна 46 годам (1966-2011гг.)
    i = 2 'начальная строка считывания
    Sheets(StancName(k)).Cells(1, g + 7) = g + 1965 'заполнение первой строки обозначениями года
        For d = 1 To 372 ' здесь задано количество дней в каждом году, организован перебор дней
        st = (g - 1) * 372 + 1 + d 'текущая строка считывания с текущего листа
        Sheets(StancName(k)).Cells(st, 6) = Sheets(StancName(k)).Cells(i, 2) 'заполнение обозначений дней
        Sheets(StancName(k)).Cells(st, 7) = Sheets(StancName(k)).Cells(i, 3) 'заполнение обозначений месяцев
        M = Sheets(StancName(k)).Cells(st, 4)
        If M = 9999 Then M = "" ' заменяем все коды пропусков нулевыми строками
        Sheets(StancName(k)).Cells(i, g + 7) = M ' запись значения метеопараметра в новую матрицу
        i = i + 1 ' оператор инкремента
        Next d
    Next g
Next k
End Sub

```

} (I)

} (II)

Рис. 13. Макрос приведения структуры представления данных к удобному для обработки виду

В приведенном макросе блок (I) заимствован из рассмотренных ранее макросов `import1`, `import2`. Блоком (II) реализована замена кодов пропусков на нулевые строки.

Приведение данных к удобному для дальнейшей обработки виду выполнено!

Аналогичным образом возможно организовать запись преобразованной матрицы из текущей книги MS Excel в стороннюю.

5. ИТОГОВЫЕ ЗАДАНИЯ

1. Выполните преобразование представления исходных данных как на рассмотренном примере на основе архива, указанного преподавателем.
2. Объедините макросы `import2`, `import3`, `import4` в единый макрос.
3. Выполните преобразование на примере макроса `import4` и средствами VBA поместите полученную матрицу в другую книгу Excel.
4. Выполните преобразование на примере макроса `import4` и средствами VBA поместите полученные матрицы в файлы с расширением “txt”.
5. Выполните выборку и преобразование данных к удобному для обработки виду средствами Python для пункта указанного преподавателем.

Учебное издание

АУХАДЕЕВ Тимур Ринатович

**СОВРЕМЕННЫЕ МЕТОДЫ РАБОТЫ С ДАННЫМИ ЦИФ-
РОВЫХ МЕТЕОРОЛОГИЧЕСКИХ АРХИВОВ**

Учебно-методическое пособие