# Контрольные задания по теме

# *«Тема 4.10. Текстовые файлы»*

## 

## 4.10.1. Задание

1. **Получить вариант задания.**
2. **Формализовать заданный вариант задания из п. 4.10.2 (с учетом используемых элементов управления на форме).**
3. **Написать программный код /процедур для решения заданной задачи, при этом необходимо использовать процедуры (процедуры-функции или процедуры-подпрограммы) с соответствующими входными и выходными формальными параметрами и в этих процедурах не должно быть ввода исходных данных и вывода результатов.**
4. **Написать программный код вызывающей процедуры, в которой описываются и инициализируются исходные данные, задаются конкретные значения переменным, которые являются фактическими параметрами вызывающей процедуры, и вызывается процедура основной задачи, а затем вызывается процедура вывода результатов.**

## Варианты контрольной работы по теме «Текстовые файлы»

***Вариант № 1***

1. Дан текстовый файл. Написать программный код трех процедур.
2. Составить **процедуру-функцию** с необходимыми формальными параметрами, которая ***построчно*** считывает текст из файла с помощью ***потокового доступа*** и находит количество абзацев в этом тексте, если первая строка каждого абзаца начинается с 5 пробелов («красная строка»). Пустые строки между абзацами не учитывать.
3. Составить **процедуру** с необходимыми формальными параметрами, которая полученный в п.1) результат добавляет в этот же файл.
4. Ввод имени файла, вызов составленных процедур и вывод результата должны осуществляться в **событийной процедуре.**
5. Ввести из текстового файла строку, в которой записано *полное имя файла*, т.е. имя диска, список каталогов (путь), собственно имя и расширение. Записать в новую строку только имя файла (без расширения). Полученную строку записать в новый текстовый файл.

***Вариант № 2***

1. Дан текстовый файл. Написать программный код трех процедур.
2. Составить **процедуру-функцию** с необходимыми формальными параметрами, которая ***построчно*** считывает текст из файла с помощью ***потокового доступа*** и находит самую длинную строку в этом тексте.
3. Составить **процедуру** с необходимыми формальными параметрами, которая полученный в п.1) результат записывает в новый файл.
4. Ввод имен файлов, вызов составленных процедур и вывод результата должны осуществляться в **событийной процедуре.**
5. Ввести фразу из текстового файла, и удалить все знаки «+» перед символами, являющимися цифрами. Полученную строку записать в существующий текстовый файл на место исходной строки.

***Вариант № 3***

1. Дан текстовый файл. Написать программный код трех процедур.
2. Составить **процедуру-функцию** с необходимыми формальными параметрами, которая считывает текст из файла с помощью ***объекта* My.Computer.FileSystem**и находит строку, в которой чаще всего встречается буква «а».
3. Составить **процедуру** с необходимыми формальными параметрами, которая полученный в п.1) результат дописывает в этот же файл.
4. Ввод имени файла, вызов составленных процедур и вывод результата должны осуществляться в **событийной процедуре.**
5. Ввести символьную строку из текстового файла и получить новую строку из символов, расположенных между первой и второй запятой. Если в исходной строке нет двух запятых, то записать в нее сообщение «**Строка пуста**». Полученный результат записать в этот же файл на место исходной строки.

***Вариант № 4***

1. Дан текстовый файл. Написать программный код трех процедур.
2. Составить **процедуру-подпрограмму** с необходимыми формальными параметрами, которая ***построчно*** считывает из файла 2 строки с помощью ***потокового доступа*** и определяет, сколько раз каждый символ первой строки входит во вторую строку.
3. Составить **процедуру** с необходимыми формальными параметрами, которая полученный в п.1) результат записывает в новый файл.
4. Ввод имен файлов, вызов составленных процедур и вывод результата должны осуществляться в **событийной процедуре.**
5. Ввести символьную строку из текстового файла и подсчитать, сколько в данной строке цифр, и составить из этих цифр число. Полученный результат добавить в тот же текстовый файл.

***Вариант № 5***

1. Дан текстовый файл. Написать программный код трех процедур.
2. Составить **процедуру-функцию** с необходимыми формальными параметрами, которая считывает текст из файла с помощью ***объекта* My.Computer.FileSystem** и находит самую короткую строку в этом тексте.
3. Составить **процедуру** с необходимыми формальными параметрами, которая полученный в п.1) результат добавляет в этот же файл.
4. Ввод имени файла, вызов составленных процедур и вывод результата должны осуществляться в **событийной процедуре.**
5. Ввести фразу из текстового файла и определить, есть ли в заданной строке одинаковые символы, и если такие символы есть, то составить из них новую строку. Если в исходной строке нет одинаковых символов, то записать в нее сообщение «**Строка пуста**». Полученную строку записать в существующий текстовый файл на место исходной строки.

***Вариант № 6***

1. Дан текстовый файл. Написать программный код трех процедур.
2. Составить **процедуру-подпрограмму** с необходимыми формальными параметрами, которая ***построчно*** считывает текст из файла с помощью ***потокового доступа,*** находит в файле последнюю «красную строку» («красная строка» начинается с 5 пробелов) и ее номер относительно всех строк файла.
3. Составить **процедуру** с необходимыми формальными параметрами, которая полученный в п.1) результат записывает в этот же файл вместо исходного текста.
4. Ввод имени файла, вызов составленных процедур и вывод результата должны осуществляться в **событийной процедуре.**
5. Ввести из текстового файла строку, состоящую из двух чисел и одного из знаков арифметических операций: **+,** -, \*или **/** между ними. Вывести в другой текстовый файл результат данного арифметического выражения.

***Вариант № 7***

1. Дан текстовый файл. Написать программный код трех процедур.
2. Составить **процедуру-подпрограмму** с необходимыми формальными параметрами, которая с помощью ***объекта* My.Computer.FileSystem** считывает из файла текст, выровненный по левому краю, и выравнивает этот текст по правому краю, добавив в начало каждой непустой строки нужное количество пробелов (ширину текста считать равной 50 символов).
3. Составить **процедуру** с необходимыми формальными параметрами, которая полученный в п.1) результат записывает в новый файл.
4. Ввод имен файлов, вызов составленных процедур и вывод результата должн9ы осуществляться в **событийной процедуре.**
5. Ввести из текстового файла строку и заменить каждое третье слово «всегда» на слово «часто». Полученную строку записать в существующий текстовый файл на место исходной строки.

***Вариант № 8***

1. Дан текстовый файл. Написать программный код трех процедур.
2. Составить **процедуру-функцию** с необходимыми формальными параметрами, которая подсчитывает количество цифр в строке.
3. Составить **процедуру-подпрограмму** с необходимыми формальными параметрами, которая ***построчно*** считывает текст из исходного файла с помощью ***потокового доступа,***  для каждой прочитанной строки, используя процедуру-функцию п.1), определяет количество цифр и ***построчно*** с помощью ***потокового доступа*** записывает в новый файл только те строки исходного файла, которые содержат ровно 3 цифры.
4. Ввод имен файлов и вызов составленной процедуры п.2) должны осуществляться в **событийной процедуре.**
5. Ввести символьную строку из текстового файла и удалить из нее символы, расположенные между первой и второй запятой. Если в исходной строке нет двух запятых, то записать в нее сообщение «**Строка пуста**». Полученный результат добавить этот же файл.

***Вариант № 9***

1. Дан текстовый файл. Написать программный код трех процедур.
2. Составить **процедуру-подпрограмму** с необходимыми формальными параметрами, которая считывает текст из файла с помощью ***объекта* My.Computer.FileSystem** и находит в файле первую «красную строку» («красная строка» начинается с 5 пробелов) и ее номер относительно всех строк файла.
3. Составить **процедуру** с необходимыми формальными параметрами, которая полученный в п.1) результат записывает в новый файл.
4. Ввод имен файлов, вызов составленных процедур и вывод результата должны осуществляться в **событийной процедуре.**
5. Ввести строку из текстового файла, содержащую фразу, лишенную знаков препинания. Вставить перед словами «**а**», «**но**» и «**чтобы**» – з**апятые**, а в конце фразы – **точку**. Полученную строку записать в существующий текстовый файл на место исходной строки.

***Вариант № 10***

1. Дан текстовый файл. Написать программный код трех процедур.
2. Составить **процедуру-подпрограмму** с необходимыми формальными параметрами, которая считывает текст из файла с помощью ***объекта* My.Computer.FileSystem,**находит в файле количество пустых строк и удаляет их из файла (т.е. возвращает по ссылке содержимое исходного файла без пустых строк).
3. Составить **процедуру** с необходимыми формальными параметрами, которая полученный в п.1) результат записывает в этот же файл вместо исходного текста.
4. Ввод имени файла, вызов составленных процедур и вывод результатов (измененный файл и количество удаленных строк) должны осуществляться в **событийной процедуре.**
5. Ввести из текстового файла строку, состоящую из слов, разделенных одним пробелом, и определить, сколько слов начинается и заканчивается одной и той же буквой. Полученный результат дописать в тот же текстовый файл.

***Вариант № 11***

1. Дан текстовый файл, каждая строка которого изображает целое число, дополненное слева и справа несколькими пробелами. Написать программный код трех процедур.
2. Составить **процедуру-функцию** с необходимыми формальными параметрами, которая ***построчно*** считывает текст из файла с помощью ***потокового доступа*** и находит сумму этих чисел.
3. Составить **процедуру** с необходимыми формальными параметрами, которая полученный в п.1) результат записывает в новый файл.
4. Ввод имен файлов, вызов составленных процедур и вывод результата должны осуществляться в **событийной процедуре.**
5. Ввести символьную строку из текстового файла и сформировать новую строку, записывая в нее те символы, которые встречаются в исходной строке ровно один раз. Полученную строку записать в существующий текстовый файл на место исходной строки.

***Вариант №12***

1. Дан текстовый файл, каждая строка которого изображает вещественное число, дополненное слева и справа несколькими пробелами. Написать программный код трех процедур.
2. Составить **процедуру-подпрограмму** с необходимыми формальными параметрами, которая считывает текст из файла с помощью ***объекта* My.Computer.FileSystem**инаходит в файле количество чисел с ненулевой дробной частью.
3. Составить **процедуру** с необходимыми формальными параметрами, которая полученный в п.1) результат добавляет в этот же файл.
4. Ввод имени файла, вызов составленных процедур и вывод результата должны осуществляться в **событийной процедуре.**
5. Ввести строку из текстового файла и определить длину самой длинной подстроки, состоящей из подряд стоящих букв «**а**». Полученный результат записать в новый текстовый файл.

***Вариант № 13***

1. Дан текстовый файл, каждая строка которого изображает вещественное число, дополненное слева и справа несколькими пробелами. Написать программный код трех процедур.
2. Составить **процедуру-функцию** с необходимыми формальными параметрами, которая получает строку и определяет, является ли эта строка записью вещественного числа с нулевой дробной частью.
3. Составить **процедуру-подпрограмму** с необходимыми формальными параметрами, которая ***построчно*** считывает текст из исходного файла с помощью ***потокового доступа*** и ***построчно*** с помощью ***потокового доступа*** записывает в новый файл только те строки исходного файла, которые содержат вещественные числа с нулевой дробной частью, используя процедуру-функцию п.1).
4. Ввод имен файлов и вызов составленной процедуры п.2) должны осуществляться в **событийной процедуре.**
5. Ввести символьную строку из текстового файла и удалить из нее все слова, имеющие нечетную длину, причем все слова разделены одним пробелом. Полученную строку записать в существующий текстовый файл на место исходной строки.

***Вариант № 14***

1. Дан текстовый файл. Написать программный код трех процедур.
2. Составить **процедуру-функцию** с необходимыми формальными параметрами, которая ***построчно*** считывает текст из файла с помощью ***потокового доступа*** и находит количество строк, в которых нет ни одной точки.
3. Составить **процедуру** с необходимыми формальными параметрами, которая полученный в п.1) результат записывает в новый файл.
4. Ввод имен файлов, вызов составленных процедур и вывод результата должны осуществляться в **событийной процедуре.**
5. Ввести строку, которая находится в текстовом файле и содержит несколько предложений. Определить, какое предложение длиннее, с учетом того, что предложения разделены точкой. Полученный результат добавить в тот же текстовый файл.

***Вариант № 15***

1. Дан текстовый файл, каждая строка которого изображает целое или вещественное число, дополненное слева и справа несколькими пробелами. Написать программный код трех процедур.
2. Составить **процедуру-функцию** с необходимыми формальными параметрами, которая получает строку и определяет, является ли эта строка записью целого числа (т.е. не содержит точку).
3. Составить **процедуру-подпрограмму** с необходимыми формальными параметрами, которая ***построчно*** считывает текст из исходного файла с помощью ***потокового доступа*** и ***построчно*** с помощью ***потокового доступа*** записывает в новый файл только те строки исходного файла, которые являются записью целого числа, используя процедуру-функцию п.1).
4. Ввод имен файлов и вызов составленной процедуры п.2) должны осуществляться в **событийной процедуре.**
5. Ввести символьную строку из текстового файла и преобразовать строку по правилу: если в строке четное число символов, то вставить в середину строки букву «О», а если нечетное, то удалить символ, расположенный в середине строки. Полученную строку записать в существующий текстовый файл на место исходной строки.

***Вариант № 16***

1. Дан текстовый файл, каждая строка которого изображает целое число, дополненное слева и справа несколькими пробелами. Написать программный код трех процедур.
2. Составить **процедуру-функцию** с необходимыми формальными параметрами, которая считывает текст из файла с помощью ***объекта* My.Computer.FileSystem** и находит в этом тексте строку с максимальным числом.
3. Составить **процедуру** с необходимыми формальными параметрами, которая полученный в п.1) результат добавляет в этот же файл.
4. Ввод имени файла, вызов составленных процедур и вывод результата должны осуществляться в **событийной процедуре.**
5. Ввести символьную строку из текстового файла и подсчитать, сколько раз в ней встречается подстрока «**АВВА**». Полученный результат записать в новый текстовый файл.

***Вариант № 17***

1. Дан текстовый файл, каждая строка которого изображает целое число, дополненное слева и справа несколькими пробелами. Написать программный код трех процедур.
2. Составить **процедуру-функцию** с необходимыми формальными параметрами, которая ***построчно*** считывает текст из файла с помощью ***потокового доступа*** и находит в этом тексте строку с минимальным числом.
3. Составить **процедуру** с необходимыми формальными параметрами, которая полученный в п.1) результат записывает в новый файл.
4. Ввод имен файлов, вызов составленных процедур и вывод результата должны осуществляться в **событийной процедуре.**
5. Ввести строку из текстового файла и определить, расположены ли символы строки в алфавитном порядке. Записать в новую строку ответ «Да» или номер позиции первого символа, нарушающего алфавитный порядок. Полученную строку добавить в существующий текстовый файл.

***Вариант № 18***

1. Дан текстовый файл. Написать программный код трех процедур.
2. Составить **процедуру-подпрограмму** с необходимыми формальными параметрами, которая ***построчно*** считывает текст из файла с помощью ***потокового доступа,*** находит в файле все «красные строки» («красная строка» начинается с 5 пробелов) и вставляет перед каждой «красной строкой» пустую строку. Таким образом. процедура возвращает по ссылке содержимое исходного файла с пустыми строками перед каждой «красной строкой».
3. Составить **процедуру** с необходимыми формальными параметрами, которая полученный в п.1) результат записывает в этот же файл вместо исходного текста.
4. Ввод имени файла, вызов составленных процедур и вывод результата должны осуществляться в **событийной процедуре.**
5. Ввести символьную строку из текстового файла и преобразовать ее, поместив вначале все символы, расположенные на четных местах, а затем, в обратном порядке, все символы, расположенные на нечетных местах. Например, строка "**Программа**" превратится в "**ргамамроП**". Полученный результат записать в новый текстовый файл.

***Вариант № 19***

1. Дан текстовый файл. Написать программный код трех процедур.
2. Составить **процедуру-функцию** с необходимыми формальными параметрами, которая считывает текст из файла с помощью ***объекта* My.Computer.FileSystem**инаходит в файле все строки, в которых есть хотя бы один символ «!».
3. Составить **процедуру** с необходимыми формальными параметрами, которая полученный в п.1) результат записывает в новый файл.
4. Ввод имен файлов, вызов составленных процедур и вывод результата должны осуществляться в **событийной процедуре.**
5. Ввести символьную строку-слово из текстового файла и преобразовать ее по следующему правилу: если строка четной длины, то удалить первую половину слова, а если нечетной, то напечатать строку в обратном порядке. Полученную строку записать в существующий текстовый файл на место исходной строки.

***Вариант № 20***

1. Дан текстовый файл. Написать программный код трех процедур.
2. Составить **процедуру-функцию** с необходимыми формальными параметрами, которая подсчитывает в строке количество гласных букв.
3. Составить **процедуру-подпрограмму** с необходимыми формальными параметрами, которая ***построчно*** с помощью ***потокового доступа*** считывает текст из исходного файла для каждой прочитанной строки, используя процедуру-функцию п.1), определяет количество гласных букв и ***построчно*** с помощью ***потокового доступа*** записывает в новый файл только те строки исходного файла, которые содержат не менее 5 гласных букв.
4. Ввод имен файлов и вызов составленной процедуры п.2) должны осуществляться в **событийной процедуре.**
5. Ввести символьную строку из текстового файла и преобразовать ее в строку длиной 20 символов следующим образом: если длина строки больше 20, то отбросить первые символы, если длина строки меньше 20, то в ее начало добавить символы точки «.».Полученный результат добавит в тот же текстовый файл.

**Вариант № 21**

1. Дан текстовый файл. Написать программный код трех процедур.
2. Составить **процедуру-функцию** с необходимыми формальными параметрами, которая считывает текст из файла с помощью ***объекта* My.Computer.FileSystem**инаходит в файле все строки, в которых нет ни одной запятой.
3. Составить **процедуру** с необходимыми формальными параметрами, которая полученный в п.1) результат добавляет в этот же файл.
4. Ввод имени файла, вызов составленных процедур и вывод результата должны осуществляться в **событийной процедуре.**
5. Ввести **построчно** из текстового файла три строки S, S1 и S2 и заменить в строке S последнее вхождение строки S1 на строку S2. Если в исходной строке S нет совпадающих с S1 подстрок, то оставить ее без изменений. Полученную строку записать в существующий текстовый файл на место исходной строки.

***Вариант № 22***

1. Дан текстовый файл. Написать программный код трех процедур.
2. Составить **процедуру-функцию** с необходимыми формальными параметрами, которая дублирует (удваивает) все гласные буквы в строке.
3. Составить **процедуру-подпрограмму** с необходимыми формальными параметрами, которая ***построчно*** считывает текст из исходного файла с помощью ***потокового доступа,***  для каждой прочитанной строки, используя процедуру-функцию п.1), дублирует (удваивает) все гласные буквы и ***построчно*** с помощью ***потокового доступа*** записывает в новый файл измененные строки.
4. Ввод имен файлов и вызов составленной процедуры п.2) должны осуществляться в **событийной процедуре.**
5. Ввести строку из текстового файла, в которой записано *полное имя файла*, т.е. имя диска, список каталогов (путь), собственно имя и расширение. Записать в новую строку только название последнего каталога (без символов «\»). Если файл находится в корневом каталоге, то записать в новую строку символ «\». Полученную строку добавить в существующий текстовый файл.

***Вариант № 23***

1. Дан текстовый файл. Написать программный код трех процедур.
2. Составить **процедуру-функцию** с необходимыми формальными параметрами, которая определяет, является ли строка симметричной (т.е. читается одинаково справа налево и слева направо).
3. Составить **процедуру-подпрограмму** с необходимыми формальными параметрами, которая ***построчно*** считывает текст из исходного файла с помощью ***потокового доступа*** и, используя процедуру-функцию п.1), ***построчно*** с помощью ***потокового доступа*** записывает в новый файл только симметричные строки.
4. Ввод имен файлов и вызов составленной процедуры п.2) должны осуществляться в **событийной процедуре.**
5. Ввести символьную строку из текстового файла. Если в строке четное число символов, то удалить из нее все последующие вхождения первого символа, а если нечетное, то удалить символ, расположенный в середине строки. Полученную строку записать в существующий текстовый файл на место исходной строки.

***Вариант № 24***

1. Дан текстовый файл, каждая строка которого изображает целое или вещественное число, дополненное слева и справа несколькими пробелами. Написать программный код трех процедур.
2. Составить **процедуру-функцию** с необходимыми формальными параметрами, которая получает строку и определяет, является ли эта строка записью вещественного числа (т.е. содержит точку).
3. Составить **процедуру-подпрограмму** с необходимыми формальными параметрами, которая ***построчно*** считывает текст из исходного файла с помощью ***потокового доступа*** и ***построчно*** с помощью ***потокового доступа*** записывает в новый файл только те строки исходного файла, которые являются записью вещественного числа, используя процедуру-функцию п.1).
4. Ввод имен файлов и вызов составленной процедуры п.2) должны осуществляться в **событийной процедуре.**
5. Ввести символьную строку из текстового файла, состоящую из слов, разделенных одним пробелом. Переставить последнее слово строки в начало текста, отделив его от последующего слова пробелом. Полученную строку записать в этот же текстовый файл вместо исходной строки.

**Вариант № 25**

1. Дан текстовый файл. Написать программный код трех процедур.
2. Составить **процедуру-функцию** с необходимыми формальными параметрами, которая считывает текст из файла с помощью ***объекта* My.Computer.FileSystem**идля каждой строки заменяет последнюю букву «а» на символ «\*».
3. Составить **процедуру** с необходимыми формальными параметрами, которая полученный в п.1) результат записывает в этот же файл вместо исходного текста.
4. Ввод имени файла, вызов составленных процедур и вывод результата должны осуществляться в **событийной процедуре.**
5. Ввести символьную строку из текстового файла, состоящую из слов, разделенных одним пробелом. Если первое и последнее слова совпадают, а количество слов больше **1,** то удалить первое и последнее слова. Полученную строку добавить в этот же текстовый файл.

***Вариант № 26***

1. Дан текстовый файл. Написать программный код трех процедур.
2. Составить **процедуру-функцию** с необходимыми формальными параметрами, которая ***построчно*** считывает текст из файла с помощью ***потокового доступа*** и находит количество строк, начинающихся цифрой.
3. Составить **процедуру** с необходимыми формальными параметрами, которая полученный в п.1) результат добавляет в этот же файл.
4. Ввод имени файла, вызов составленных процедур и вывод результата должны осуществляться в **событийной процедуре.**
5. Ввести последовательно из текстового файла три строки S, S1 и S2. Заменить в строке S первое вхождение строки S1 на строку S2. Если в исходной строке S нет совпадающих с S1 подстрок, то оставить ее без изменений. Полученную строку записать в новый текстовый файл.

***Вариант № 27***

1. Дан текстовый файл. Написать программный код трех процедур.
2. Составить **процедуру-подпрограмму** с необходимыми формальными параметрами, которая считывает текст из файла с помощью ***объекта* My.Computer.FileSystem**и находит все строки, которые заканчиваются гласной буквой.
3. Составить **процедуру** с необходимыми формальными параметрами, которая полученный в п.1) результат добавляет в этот же файл.
4. Ввод имени файла, вызов составленных процедур и вывод результата должны осуществляться в **событийной процедуре.**
5. Ввести построчно из текстового файла две строки S и S0. Удалить из строки S первую подстроку, совпадающую с S0. Если в исходной строке S нет совпадающих с S0 подстрок, то оставить ее без изменений. Полученный результат записать в новый текстовый файл.

**Вариант № 28**

1. Дан текстовый файл. Написать программный код трех процедур.
2. Составить **процедуру-подпрограмму** с необходимыми формальными параметрами, которая ***построчно*** считывает текст из файла с помощью ***потокового доступа*** и находит все строки, в которых нет ни одного символа «:»
3. Составить **процедуру** с необходимыми формальными параметрами, которая полученный в п.1) результат записывает в этот же файл вместо исходного текста.
4. Ввод имени файла, вызов составленных процедур и вывод результата должны осуществляться в **событийной процедуре.**
5. Ввести символьную строку из текстового файла и определить, является ли она записью целого десятичного числа, кратного трем, а сообщение об этом добавить в существующий текстовый файл.

***Вариант № 29***

1. Дан текстовый файл, строки которого состоят из целых чисел, разделенных одним пробелом. Написать программный код трех процедур.
2. Составить **процедуру-подпрограмму** с необходимыми формальными параметрами, которая считывает текст из файла с помощью ***объекта* My.Computer.FileSystem** и находит две суммы: сумму положительных и сумму отрицательных чисел, записанных в этом файле.
3. Составить **процедуру** с необходимыми формальными параметрами, которая полученный в п.1) результат добавляет в этот же файл.
4. Ввод имени файла, вызов составленных процедур и вывод результата должны осуществляться в **событийной процедуре.**
5. Ввести строку из текстового файла, в которой записано *полное имя файла*, т.е. имя диска, список каталогов (путь), собственно имя и расширение. Записать в новую строку только название первого каталога (без символов «\»). Если файл находится в корневом каталоге, то записать в новую строку символ «\». Полученную строку записать в новый текстовый файл.

***Вариант № 30***

1. Дан текстовый файл, каждая строка которого содержит только русские буквы и в конце точку. Написать программный код трех процедур.
2. Составить **процедуру-функцию** с необходимыми формальными параметрами, которая получает строку и переставляет ее символы в алфавитном порядке (оставив точку последней).
3. Составить **процедуру-подпрограмму** с необходимыми формальными параметрами, которая ***построчно*** считывает текст из исходного файла с помощью ***потокового доступа*** и ***построчно*** с помощью ***потокового доступа*** записывает в новый файл преобразованные с использованием процедуры-функции п.1) строки исходного файла.
4. Ввод имен файлов и вызов составленной процедуры п.2) должны осуществляться в **событийной процедуре.**
5. Ввести из текстового файла символьную строку четной длины и определить, в какой половине этой строки буква «с» встречается чаще. Полученный результат записать в новую строку. Если исходная строка нечетной длины, то записать в новую строку сообщение «**Строка пуста**». Полученную строку добавить в существующий текстовый файл.

## 4.10.3. Пример выполнения контрольной работы по теме «Текстовые файлы»

1. Дан текстовый файл, каждая строка которого содержит только слова из русских букв, разделенных одним или несколькими пробелами. Написать программный код трех процедур.
2. Составить **процедуру-функцию** с необходимыми формальными параметрами, которая получает строку и переставляет ее символы в обратном порядке.
3. Составить **процедуру-подпрограмму** с необходимыми формальными параметрами, которая ***построчно*** считывает текст из исходного файла с помощью ***потокового доступа*** и ***построчно*** с помощью ***потокового доступа*** записывает в новый файл преобразованные строки исходного файла, причем в преобразованную строку записываются через один пробел по отдельности перевернутые наоборот с использованием процедуры-функции п.1) слова исходной строки файла.
4. Ввод имен файлов и вызов составленной процедуры п.2) должны осуществляться в **событийной процедуре.**

***Программный код решения задачи:***

Первое задание реализует процедура-функция **Perevorot()**, которая, получив через формальный параметр по значению строку s, с помощью регулярного цикла с отрицательным шагом записывает в новую строку temp все символы исходной строки s и возвращает созданную строку temp в качестве результата.

Второе задание реализует процедура-функция **Resh()**, которой через формальные параметры передаются две строки, содержащие имена исходного и нового файлов. В ней создаются два потоковых объекта: Rf типа StreamReader - для чтения исходного файла и wf типа StreamWriter - для записи в новый файл, причем, так как запись в новый файл должна производиться построчно, то поток создается со значением второго параметра, равным True, который означает, что при каждом использовании метода WriteLine будет производиться добавление строки в сущуствующий файл. Далее в итеративном цикле до конца потока для чтения считывается очередная строка исходного файла, из нее с помощью процедуры **DelProb()**удаляются лишние пробелы и создается массив слов Mas (см. пример в теме 4.9.3). Затем во внутреннем регулярном цикле каждое слово из массива Mas переворачивается наоборот с использованием процедуры-функции **Perevorot()**, и после завершения внутреннего цикла полученный массив перевернутых слов рассматриваемой строки с помощью функции Join склеивается через один пробел в одну новую строку line2, которая и записывается в новый файл.

|  |
| --- |
| **Option Strict On**  **Option Explicit On**  **Imports System.IO**  **Public Class Form1**  **'функция переворачивает строку наоборот**  **Function Perevorot(ByVal s As String) As String**  **Dim temp As String = ""**  **Dim L As Integer = s.Length**  **For i As Integer = L - 1 To 0 Step -1**  **temp = temp + s.Substring(i, 1)**  **Next**  **Return temp**  **End Function**  **'процедура удаления из строки лишних пробелов**  **'чтобы слова разделялись только одним пробелом**  **Sub DelProb(ByRef s As String)**  **s = s.Trim**  **Do While s.IndexOf(Space(2)) >= 0**  **s = s.Replace(Space(2), Space(1))**  **Loop**  **End Sub**  **Sub Resh(ByVal filename1 As String, ByVal filename2 As String)**  **'создание потока для чтения из исходного текстового файла**  **Dim Rf As New StreamReader(filename1)**  **'создание потока для записи в другой текстовый файл**  **'Значение True 2-го параметра означает, что будет**  **'производиться дозапись в конец файла**  **Dim wf As New StreamWriter(filename2, True)**  **Dim line1, line2, slovo As String**  **Dim Mas() As String**  **Try**  **Do While Not Rf.EndOfStream() 'пока не достигнут конец**  **'потока**  **line1 = Rf.ReadLine() 'чтение строки из 1 файла в**  **'line1**  **DelProb(line1) 'удаление из строки line1 лишних**  **' пробелов**  **Mas = Split(line1, " ") 'разбиваем строку line1 на**  **' массив слов**  **line2 = "" 'новая строка для записи в другой файл**  **For i As Integer = 0 To Mas.Length - 1**  **slovo = Mas(i) 'один элемент массива - это слово**  **'перевернули слово и снова записали в массив**  **Mas(i) = Perevorot(slovo)**  **Next**  **'склеили массив перевернутых слов в строку line2,**  **'разделив слова одним пробелом**  **line2 = Join(Mas, " ")**  **wf.WriteLine(line2) 'запись строки line2 во 2 файл**  **Loop**  **Catch ex As Exception**  **MsgBox("Невозможно прочитать данные из файла")**  **Finally**  **wf.Flush() 'очистка буфера потока**  **Rf.Close() 'закрытие потока**  **wf.Close()**  **End Try**  **End Sub**    **Private Sub Button1\_Click(...)**  **Dim filename1 As String = "", filename2 As String = ""**  **filename1 = TextBox1.Text**  **filename2 = TextBox2.Text**  **Resh(filename1, filename2)**  **End Sub**  **End Class** |

1. Дан текстовый файл, каждая строка которого содержит только слова из русских букв, разделенных одним или несколькими пробелами. Написать программный код четырех процедур.
2. Составить **процедуру-функцию** с необходимыми формальными параметрами, которая получает строку и переставляет ее символы в обратном порядке.
3. Составить **процедуру-подпрограмму** с необходимыми формальными параметрами, которая считывает текст из файла с помощью ***объекта* My.Computer.FileSystem,**и для каждой строки исходного файла переворачивает каждое слово наоборот с использованием процедуры-функции п.1), и записывает в новую строку через один пробел по отдельности перевернутые слова исходной строки.
4. Составить **процедуру** с необходимыми формальными параметрами, которая полученный в п.2) результат записывает в новый файл.
5. Ввод имен файлов, вызов составленных процедур и вывод результата должен осуществляться в **событийной процедуре.**

***Программный код решения задачи:***

Первое задание реализует процедура-функция **Perevorot( )**, которая получив через формальный параметр по значению строку s, с помощью регулярного цикла с отрицательным шагом записывает в новую строку temp все символы исходной строки s и возвращает созданную строку temp в качестве результата.

Второе задание реализует процедура **Reshenie( )**, которой через формальные параметры передается строка filename, содержащая имя исходного файла и по ссылке возвращаются строка s1с исходным файлом и строка s2 с перевернутыми словами, которую надо будет записать в новый файл. В ней сначала с помощью ***объекта* My.Computer.FileSystem** весь исходный файл считывается целиком в одну строку s1, затем полученная строка с помощью функции Split разбивается на массив строк Stroki по признаку конца строки (**vbNewLine**). Далее во внешнем цикле из каждой строки исходного файла с помощью процедуры **DelProb()**удаляются лишние пробелы и строка разбивается на массив слов slova по признаку одного пробела. Во внутреннем цикле каждое слово переворачивается наоборот с использованием процедуры-функции **Perevorot( )** и записывается в новую строку s2 через один пробел. После завершения внутреннего цикла в новую строку s2 добавляется признак конца строки (**vbNewLine**) для разделения строк в новом файле и далее повторение внешнего цикла для обработки очередной строки исходного файла.

Третье задание реализует процедура **vivodFile ()**, которой через формальные параметры передается строка filename, содержащая имя нового файла, и строка Sс перевернутыми словами для записи в новый файл.

|  |
| --- |
| **Option Strict On**  **Option Explicit On**  **Public Class Form1**  **'функция переворачивает строку наоборот**  **Function Perevorot(ByVal s As String) As String**  **Dim temp As String = ""**  **Dim L As Integer = s.Length**  **For i As Integer = L - 1 To 0 Step -1**  **temp = temp + s.Substring(i, 1)**  **Next**  **Return temp**  **End Function**  **'процедура удаления из строки лишних пробелов**  **'чтобы слова разделялись только одним пробелом**  **Sub DelProb(ByRef s As String)**  **s = s.Trim**  **Do While s.IndexOf(Space(2)) >= 0**  **s = s.Replace(Space(2), Space(1))**  **Loop**  **End Sub**    **'Процедура чтения данных из текстового файла целиком в одну**  **'строку s1 и формирование новой строки s2 для записи в новый файл**  **Sub Reshenie(ByVal filename As String, ByRef s1 As String, \_**  **ByRef s2 As String)**  **Dim line, OdnoSlovo As String**  **Dim Stroki() As String 'массив строк исходного файла**  **Dim slova() As String 'массив слов одной строки исходного файла**  **s2 = "" 'новая строка с перевернутым файлом для записи целиком**  **'Открываем существующий файл и загружаем его текстовое**  **'содержимое целиком в одну строку s1**  **Try**  **s1 = My.Computer.FileSystem.ReadAllText(filename)**  **Stroki = Split(s1, vbNewLine) 'массив строк исходного файла**  **For i As Integer = 0 To Stroki.Length - 1**  **line = Stroki(i) ' одна строка исходного файла**  **DelProb(line) 'удалили лишние пробелы**  **slova = Split(line, " ") 'разбили ее на массив слов**  **For j As Integer = 0 To slova.Length - 1**  **OdnoSlovo = Perevorot(slova(j))**  **s2 = s2 + OdnoSlovo + " "**  **Next**  **s2 = s2 + vbNewLine**  **Next**  **Catch ex As Exception**  **MsgBox("Файл не может быть открыт из-за " & \_**  **" следующей ошибки:" & vbNewLine & vbCrLf & ex.Message)**  **End Try**  **End Sub**    **'Процедура записи строки в текстовый файл**  **Sub vivodFile(ByVal filename As String, ByVal S As String)**  **'Запись строки-результата S в текстовый файл;значение False**  **'параметра задает,что если файл уже существует,то будет перезаписан.**  **Try**  **My.Computer.FileSystem.WriteAllText(filename, S, False)**  **Catch ex As Exception**  **MsgBox("ошибка при сохранении (создании) файла")**  **MsgBox("Данные в файл не могут быть переписаны из-за " & \_**  **" следующей ошибки:" & vbNewLine & ex.Message)**  **End Try**  **End Sub**  **Private Sub Button1\_Click(...)**  **Dim filename1, filename2 As String**  **Dim OldStr, NewStr As String**  **filename1 = TextBox1.Text**  **filename2 = TextBox2.Text**  **Reshenie(filename1, OldStr, NewStr)**  **vivodFile(filename2, NewStr)**  **TextBox3.Text = OldStr**  **TextBox4.Text = NewStr**  **End Sub**  **End Class** |