

Lotus LearningSpace
Release 5

Мои сервисыКаталог курсовМои курсы

tki studentos

Список курсов

ОПД-08 - Операционные системы

Введение

Особенности архитектуры современных операционных систем

Структура операционных систем

Тест 1

Загрузка программ

Многозадачность и параллельное выполнение

Управление памятью

Тест 2

Средства межпроцессных взаимодействий

Управление вводом - выводом

Безопасность

Особенности платформы Microsoft .NET

Тест 3

Лабораторные работы

Лабораторная работа № 1

Лабораторная работа № 2

Лабораторная работа № 3

Лабораторная работа № 4

Лабораторная работа № 5

Лабораторная работа № 6, 7

Лабораторная работа № 8

Самостоятельная работа

Курсовая работа

Лабораторная работа № 6,7

Работа с файловой системой

Содержание

Автор: Serv Fromru.com
Опубликовано: 16.05.2008

Цель работы

1) Изучение объектов WSH работы с файловой системой.
2) Изучение приемов программирования с использованием объектов работы с файловой системой.
3) Получение навыков разработки сценариев.

Содержание работы

1) Изучение основных свойств и методов объектов FileSystemObject, Drive, Folder, File и TextStream .
2) Выработка навыков использования объектов FileSystemObject, Drive, Folder, File, TextStream.
3) Изучение и выполнение всех рассмотренных в руководстве примеров.
4) Разработка процедуры решения поставленной задачи.
5) Отладка и выполнение процедуры.
6) Составление отчета.

Содержание отчета

1) Краткие теоретические сведения.
2) Постановка задачи.
3) Описание процедуры.
4) Руководство пользователю процедуры.
5) Листинг процедуры.
6) Результаты выполнения процедуры.

Краткие теоретические сведения

Объект **FileSystemObject** нужен для дисковых операций. Он позволяет читать, писать, удалять файлы и создавать папки. Этот компонент – **Scripting.FileSystemObject**. **FileSystemObject** имеет методы, описанные в таб. 6.1.

Специальные обозначения:

[...] – необязательный параметр

src, dst – источник и приёмник. Если особо не оговаривается, то это строки с путями

path – путь к файлу или каталогу

name – имя файла или каталога

spec – идентификатор системного каталога. (0 – Windows, 1 – System, 2 – Temp)

unicode – определяет режим текста (TristateTrue – Unicode, TristateFalse – ASCII, TristateUseDefault – по умолчанию)

ioMode – определяет режим открытия (1 – для чтения, 2 – для записи, 8 – для добавления в конец файла)

Таблица 6.1.

Свойство	Описание
Drives	Список дисков, доступных на данной машине, включая сетевые
Метод	Описание
BuildPath(path,name)	Возвращает путь, состоящий из двух аргументов (просто вставляет разделитель)

http://10.115.0.250/LearningSpace5/Program/UI/Main/Themes/Kendall/Main.asp[12.09.2019 13:46:33]

CopyFile(src,dst[,ovw])	Копирует файл src в dst. Если ovw=true, то заменяет dst
CopyFolder(src,dst[,ovw])	Копирует каталог src со всеми подкаталогами в dst.
CreateFolder(path)	Создаёт каталог name
CreateTextFile(FileName[,ovw[,Unicode]])	Создаёт текстовый файл FileName и связывает с ним текстовый пот unicode=true, то создаётся файл Unicode.
DeleteFile(name[,force])	Удаляет файл name. Если force==true, то не обращает внимания на атрибуты файла
DeleteFolder(path[,force])	Удаляет каталог src со всеми подкаталогами name
DriveExists(path)	Возвращает true, если диск path доступен
FileExists(path)	Возвращает true, если файл path существует
FolderExists(path)	Возвращает true, если каталог path существует
GetAbsolutePathName(path)	Нормализует путь path
GetBaseName(path)	Возвращает имя файла (без расширения)
GetDrive(drive)	Возвращает объект Drive от указанного диска
GetDriveName(path)	Возвращает имя диска
GetExtensionName(path)	Возвращает расширение файла
GetFile(path)	Возвращает объект File, расположенный по указанному пути.
GetFileName(path)	Возвращает путь к файлу (без расширения)
GetFolder(path)	Возвращает объект Folder, расположенный по указанному пути.
GetParentFolderName(path)	Возвращает имя последнего каталога в пути
GetSpecialFolder(spec)	Возвращает путь к системным каталогам
GetTempName()	Возвращает случайно сгенерированное имя файла (для временных файлов)
MoveFile(src,dst)	Перемещает файл src в dst
MoveFolder(src,dst)	Перемещает каталог src в dst
OpenTextFile(path[,iomode[,create[,unicode]]])	Открывает текстовый файл для чтения/записи (определяется iomode)

Пример использования:

```
// fsoDirs.js - получение системных каталогов
var fso=WScript.CreateObject("Scripting.FileSystemObject");
var s="Специальные каталоги:";
s+="\nWindows "+fso.GetSpecialFolder(0);
s+="\nSystem "+fso.GetSpecialFolder(1);
s+="\nTemp "+fso.GetSpecialFolder(2);
WScript.Echo(s);
```

Наряду с **FileSystemObject** используются так же объекты **Drive**, **Folder**, **File** и **TextStream**. Свойства и методы перечислены в таблицах 6.2- 6.5 соответственно.

Объект **Drive** (таблица 6.2.) представляет собой логический или сетевой диск операционной системы. Получить объект можно с помощью метода **FileSystemObject.GetDrive**.

Таблица 6.2.

Свойство	Описание
AvailableSpace	Доступное пространство (с учётом квоты) на диске, в байтах
DriveLetter	Буква диска
DriveType	Тип диска (см. ниже)
FileSystem	Тип файловой системы (строка)
FreeSpace	Доступное пространство (без учёта квоты) на диске, в байтах
IsReady	true, если устройство готово
Path	Путь к диску (буква диска и двоеточие)
RootFolder	Объект Folder, представляющий корневой каталог
SerialNumber	Строка, содержащая серийный номер диска
ShareName	Имя сетевого ресурса, «шары»
TotalSize	Общий объем диска, в байтах
VolumeName	Метка тома

Тип диска может быть следующим:

- 0 – неизвестен
- 1 – отключаемый
- 2 – жёсткий
- 3 – сетевой
- 4 – CD-ROM
- 5 – RAM диск

Пример использования:

```
// fsoEnumDrv.js - вывод списка подключенных дисков
var fso=WScript.CreateObject("Scripting.FileSystemObject");
var e=new Enumerator(fso.Drives);
var s="";
var n="";
for(;!e.atEnd();e.moveNext())
{
var      x=e.item();
          s=s+x.DriveLetter;
          s+=" - ";
          switch(x.DriveType)
          {
            case 1: n="Отключаемый";break;
            case 2: n="Жёсткий";break;
            case 3: n="Сетевой";break;
            case 4: n="CD-ROM";break;
            case 5: n="RAM диск";break;
            default: n="Unknown";
          }
          s+=n+" ";
          if(x.DriveType==3)
            n=x.ShareName;
          else if (x.IsReady)
            n=x.VolumeName;
          else
            n="[Drive not ready]";
          s+="\""+n+"\"\\n";
        }
WScript.Echo(s);
```

Объект **Folder** (таблица 6.3.) представляет собой каталог операционной системы.

Таблица 6.3.

Свойство	Описание
Attributes	Атрибуты каталога
DateCreated	Дата создания каталога
DateLastAccessed	Дата последнего доступа к каталогу
DateLastModified	Дата последней модификации каталога
Drive	Буква диска, на котором находится каталог
Files	Список файлов, находящихся в каталоге
IsRootFolder	Если это корневой каталог диска, то равно true
Name	Имя файла
ParentFolder	Объект Folder родительского каталога
Path	Путь к каталогу
ShortName	Короткое (MS-DOS - совместимое) имя каталога
ShortPath	Короткий (MS-DOS - совместимый) путь к каталогу
Size	Размер каталога
SubFolders	Список подкаталогов
Type	Строка с информацией о типе каталога (из реестра)
Метод	Описание
Copy(dst[, ovw])	Копирует каталог в dst. Если ovw=true, то заменяет dst
Delete(force)	Удаляет каталог. Если force==true, то не обращает внимания на атрибут ReadOnly
Move(dest)	Перемещает каталог в dst

Пример использования:

```
// fsoCreateText.js - создание файла с деревом каталогов
var of;
function ProcessDirectory(dir,prefix)
{
var      foldPrefix=prefix+"-";
          prefix+="|";
var      newPrefix=prefix+" ";
}
```

```
var    fc = new Enumerator(dir.SubFolders);
      while(!fc.atEnd())
      {
          of.WriteLine(foldPrefix+fc.item().Name);
          ProcessDirectory(fc.item(),newPrefix);
          fc.moveNext();
      }
      fc = new Enumerator(dir.Files);
      while(!fc.atEnd())
      {
          of.WriteLine(prefix+fc.item().Name);
          fc.moveNext();
      }
  }

if(WScript.Arguments.Count()==2)
{
var    fso=WScript.CreateObject("Scripting.FileSystemObject");
var    of=fso.CreateTextFile(WScript.Arguments(0),true);
      ProcessDirectory(fso.GetFolder(WScript.Arguments(1)),"")
      of.Close()
}else
      WScript.Echo("Использование: cscript //nologo "+WScript.ScriptFullName+ " tree.txt c:\\windows\\t
```

Объект **File** (таблица 6.4.) представляет собой файл операционной системы.

Таблица 6.4.

Свойство	Описание
Attributes	Атрибуты файла
DateCreated	Дата создания файла
DateLastAccessed	Дата последнего доступа к файлу
DateLastModified	Дата последней модификации файла
Drive	Буква диска, на котором находится файл
Name	Имя файла
ParentFolder	Объект Folder родительского каталога
Path	Путь к файлу
ShortName	Короткое (MS-DOS - совместимое) имя файла
ShortPath	Короткий (MS-DOS - совместимый) путь к файлу
Size	Размер файла
Type	Строка с информацией о типе файла документа (из реестра)
Метод	Описание
Copy(dst[, ovw])	Копирует файл в dst. Если ovw=true, то заменяет dst
Delete(force)	Удаляет файл. Если force==true, то не обращает внимания на атриб
Move(dest)	Перемещает файл в dst
OpenAsTextStream([iomode[, unicode]])	Открывает файл как текстовый для чтения/записи (определяется iom

Объект **TextStream** (таблица 6.5.) позволяет читать и записывать текстовую информацию. Он может быть и только для файлов, но и для любых объектов, предоставляющих интерфейс Stream. Внутри его хранится фай или текущая позиция чтения/записи.

Таблица 6.5.

Свойство	Описание
AtEndOfLine	равно true, если указатель находится в конце строки
AtEndOfStream	равно true, если указатель находится в конце потока
Column	Номер колонки (столбца) указателя
Line	Номер строки указателя
Метод	Описание
Close()	Закрывает поток
Read(count)	Читает count символов из потока и возвращает в виде строки. Передвигает указатель
ReadAll()	Возвращает весь текстовый поток как строку
ReadLine()	Считывает одну строку текста из потока и возвращает строку. Передвигает указатель
Skip(count)	Передвигает указатель на count символов
SkipLine	Передвигает указатель на начало следующей строки текста
Write(str)	Записывает строку str в поток в позиции указателя

- WriteBlankLines(count) Записывает count пустых текстовых строк в поток
- WriteLine(str) Записывает строку str в поток в позиции указателя и добавляет символ конца строки

Если методам Write и WriteLine объекта TextStream не-Unicode-файла передать строку с Unicode-символами, то ошибку.

```
// fsoCreateText.js - создание файла с таблицей умножения
if(WScript.Arguments.Count()==1)
{
    var fso=WScript.CreateObject("Scripting.FileSystemObject");
    var tf=fso.CreateTextFile(WScript.Arguments(0),true);
    tf.WriteLine("Таблица Пифагора:");
    var i;
    tf.Write(" |");
    for(i=1;i<10;i++)
    {
        tf.Write(" "+i+"|");
    }
    tf.WriteLine("");
    tf.WriteLine("-+---+---+---+---+---+---+---+---+---+");
    var j;
    for(i=1;i<10;i++)
    {
        tf.Write(i);
        for(j=1;j<10;j++)
        {
            tf.Write(" |");
            if(i*j<10)
                tf.Write(" ");
            tf.Write(i*j);
        }
        tf.WriteLine(" |");
    }
    tf.WriteLine("-+---+---+---+---+---+---+---+---+---+");
    tf.Close();
}
else
    WScript.Echo("Использование: cscript //nologo "+WScript.ScriptFullName+ " pifagor.txt");
```

Контрольные вопросы

- Объекты работы с файловой системой .
- Свойства объекта FileSystemObject.
- Методы объекта FileSystemObject.
- Свойства объекта Drive.
- Методы объекта Drive.
- Свойства объекта Folder.
- Методы объекта Folder.
- Свойства объекта File.
- Методы объекта File.
- Свойства объекта TextStream.
- Методы объекта TextStream.
- Копирование файлов.
- Поиск файлов.
- Поиск каталогов.
- Переименование файлов.
- Переименование каталогов.
- Создание файлов.
- Создание каталогов.
- Удаление файлов.
- Удаление каталогов.
- Обработка текстовых файлов.

Варианты заданий

1) Разработать процедуру, выполняющую заданную в индивидуальном задании последовательность операций (каталогами) с помощью различных объектов работы с файловой системой. Организовать ввод всех исходных параметров процедуры с помощью параметров командной строки при вызове процедуры. Предусмотреть возможность перенаправления результатов выполнения процедуры в файл.

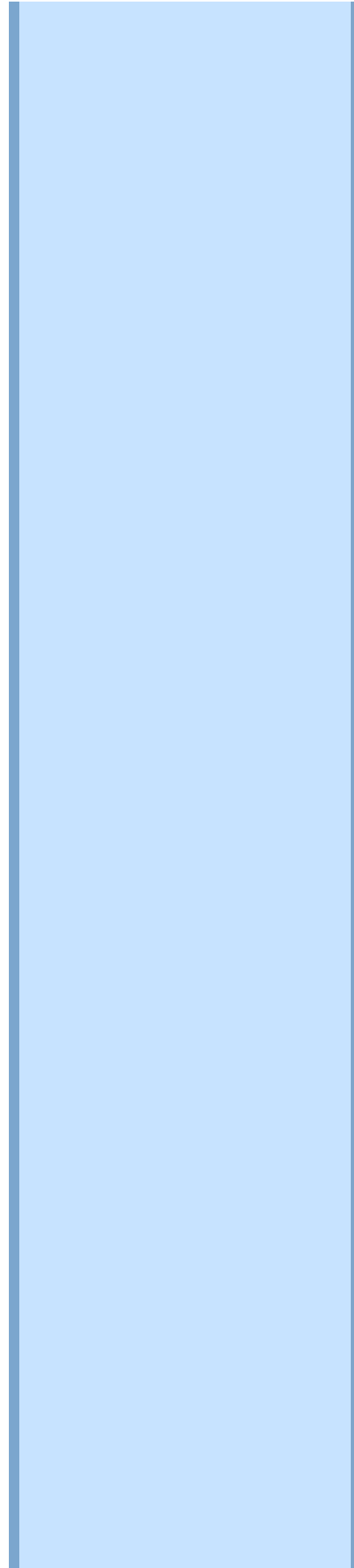
- 1. Копирование файлов:
 - 1.1 Создать исходный каталог;
 - 1.2 С помощью Блокнота создать исходный файл;
 - 1.3 Создать результирующий каталог;
 - 1.4 Скопировать исходный файл в результирующий каталог;
 - 1.5 Запустить Блокнот для редактирования файла в результирующем каталоге
 - 1.6 Сравнить файлы в исходном и результирующем каталогах;
- 2 Копирование каталогов:
 - 2.1 Создать исходный каталог;
 - 2.2 С помощью Блокнота создать исходный файл;
 - 2.3 Скопировать исходный каталог в результирующий каталог;
 - 2.4 Изменить атрибуты файла в результирующем каталоге;
 - 2.5 Вывести оглавление исходного и результирующего каталогов;
 - 2.6 Вывести содержимое файла результирующего каталога;
- 3 Переименование каталогов:
 - 3.1 Создать исходный каталог;
 - 3.2 С помощью Блокнота создать исходный файл;
 - 3.3 Переименовать исходный каталог в результирующий каталог;
 - 3.4 Изменить атрибуты файла в результирующем каталоге;
 - 3.5 Вывести оглавление исходного и результирующего каталогов;
 - 3.6 Вывести содержимое файла результирующего каталога;
- 4 Переименование файлов:
 - 4.1 Создать исходный каталог;
 - 4.2 С помощью Блокнота создать исходный файл;
 - 4.3 Создать результирующий каталог;
 - 4.4 Скопировать исходный файл в результирующий каталог;
 - 4.5 Переименовать файл в результирующем каталоге;
 - 4.6 Запустить Блокнот для редактирования файла в результирующем каталоге
 - 4.7 Сравнить файлы в исходном и результирующем каталогах;
- 5 Перемещение файлов:
 - 5.1 Создать исходный каталог;
 - 5.2 С помощью Блокнота создать исходный файл;
 - 5.3 Создать результирующий каталог;
 - 5.4 Переместить исходный файл в результирующий каталог;
 - 5.5 Запустить Блокнот для редактирования файла в результирующем каталоге
 - 5.6 Вывести содержимое файла результирующего каталога;
- 6 Перемещение каталогов:
 - 6.1 Создать исходный каталог;
 - 6.2 С помощью Блокнота создать исходный файл;
 - 6.3 Переместить исходный каталог в результирующий каталог;
 - 6.4 Переименовать файл в результирующем каталоге;
 - 6.5 Запустить Блокнот для редактирования файла в результирующем каталоге;

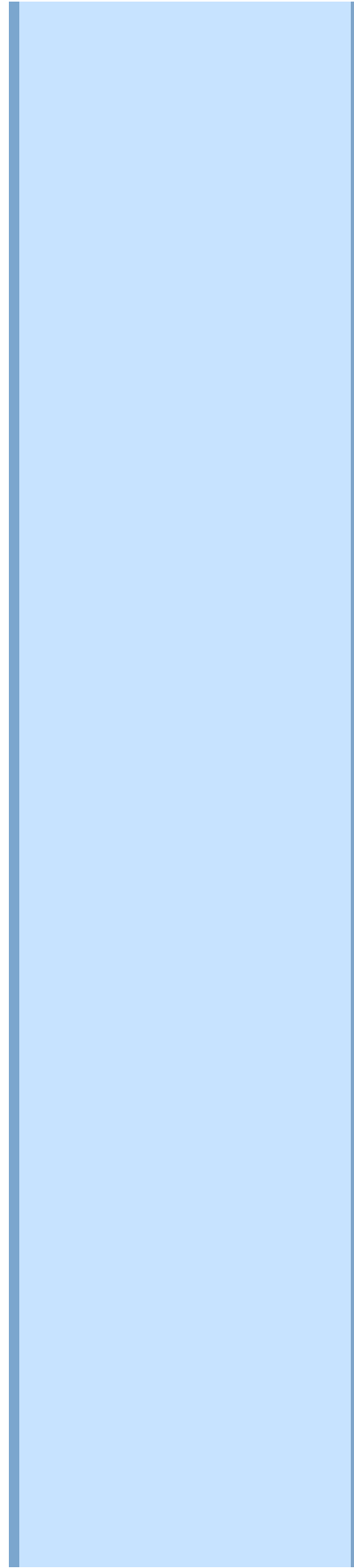
- 6.6 Вывести оглавление результирующего каталога;
- 7 Копирование дерева каталогов:
 - 7.1 Создать исходное дерево каталогов;
 - 7.2 С помощью Блокнота создать исходный файл;
 - 7.3 Скопировать исходное дерево каталогов в результирующий каталог;
 - 7.4 Изменить атрибуты файла в результирующем каталоге;
 - 7.5 Вывести оглавление исходного и результирующего каталогов;
 - 7.6 Вывести содержимое файла результирующего каталога;
- 8 Копирование файлов:
 - 8.1 Создать исходный каталог;
 - 8.2 С помощью MS Word создать исходный файл;
 - 8.3 Создать результирующий каталог;
 - 8.4 Скопировать исходный файл в результирующий каталог;
 - 8.5 Запустить MS Word для редактирования файла в результирующем каталоге;
 - 8.6 Вывести оглавление исходного и результирующего каталогов;
- 9 Переименование файлов:
 - 9.1 Создать исходный каталог;
 - 9.2 С помощью MS Excel создать исходный файл;
 - 9.3 Создать результирующий каталог;
 - 9.4 Скопировать исходный файл в результирующий каталог;
 - 9.5 Переименовать файл в результирующем каталоге;
 - 9.6 Запустить MS Excel для редактирования файла в результирующем каталоге;
 - 9.7 Сравнить файлы в исходном и результирующем каталогах;
- 10 Замена файлов:
 - 10.1 Создать исходный каталог;
 - 10.2 С помощью MS Excel создать исходный файл;
 - 10.3 Создать результирующий каталог;
 - 10.4 Скопировать исходный файл в результирующий каталог;
 - 10.5 Запустить MS Excel для редактирования файла в результирующем каталоге;
 - 10.6 Заменить файлы исходного каталога файлами результирующего каталога;
 - 10.7 Сравнить файлы в исходном и результирующем каталогах;

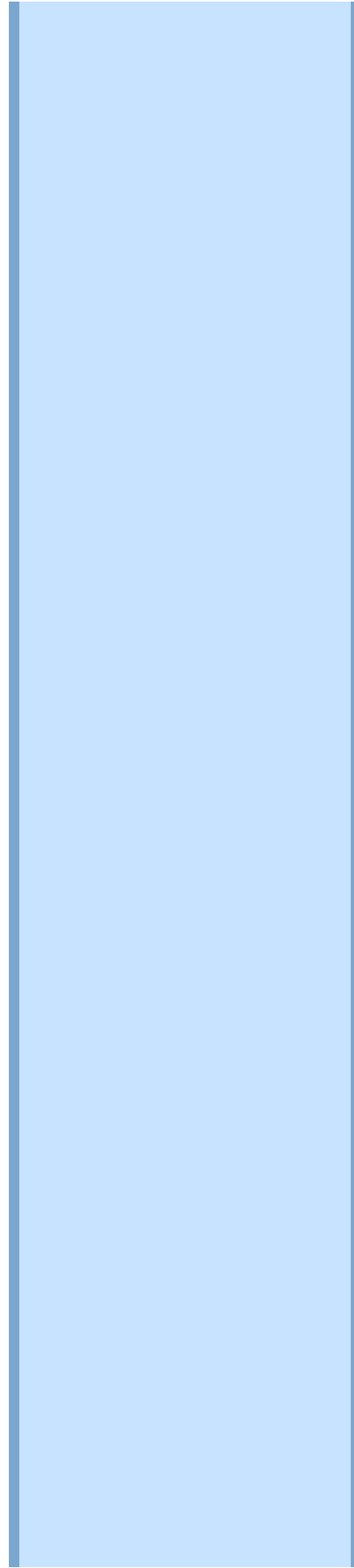
II) Разработать процедуру обработки текстового файла, в соответствии с индивидуальным заданием. Организовать исходных данных в процедуру с помощью параметров командной строки при вызове процедуры. Создание исходных данных в текстовом файле произвести в процедуре путем вызова Блокнота. Предусмотреть возможность перенаправления результатов выполнения процедуры в файл.

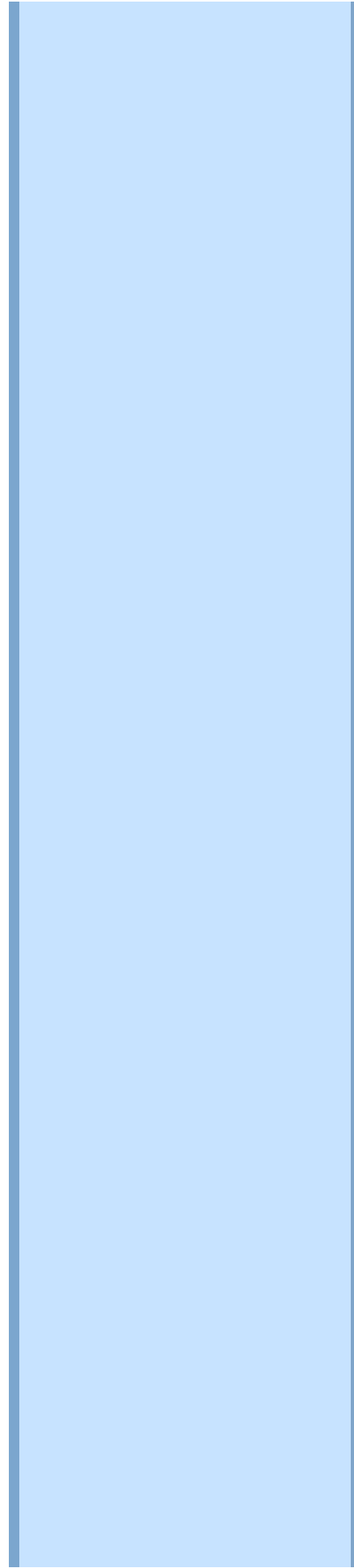
- 1) Задан исходный текст на русском языке. Длина текста - не более NL символов, длина строки - не более N\$ слова - не более NW символов. Исходный текст должен заканчиваться точкой. Составить процедуру, которая убирает лишние пробелы между словами, оставляя их по одному. В качестве результата вывести исходный и измененный тексты.
- 2) Задан исходный текст на русском языке. Длина текста - не более NL символов, длина строки - не более N\$ слова - не более NW символов. Исходный текст должен заканчиваться точкой. Составить процедуру, которая находит слово максимальной длины. В качестве результата вывести исходный текст, найденное слово и его длину.
- 3) Задан исходный текст на русском языке. Длина текста - не более NL символов, длина строки - не более N\$ слова - не более NW символов. Исходный текст должен заканчиваться точкой. Составить процедуру, которая находит слова, оканчивающиеся заданной буквой. В качестве результата вывести исходный текст, найденные слова и их количество.
- 4) Задан исходный текст на русском языке. Длина текста - не более NL символов, длина строки - не более N\$ слова - не более NW символов. Исходный текст должен заканчиваться точкой. Составить процедуру, которая находит слова заданной длины. В качестве результата вывести исходный текст, найденные слова и их количество.

- 5) Задан исходный текст на русском языке. Длина текста - не более NL символов, длина строки - не более N_с слова – не более NW символов. Исходный текст должен заканчиваться точкой. Составить процедуру, которая находит слова, начинающиеся с гласной буквы. В качестве результата вывести исходный текст, найденные с количество.
- 6) Задан исходный текст на русском языке. Длина текста - не более NL символов, длина строки - не более N_с слова – не более NW символов. Исходный текст должен заканчиваться точкой. Составить процедуру, которая находит слова, оканчивающиеся гласной буквой. В качестве результата вывести исходный текст, найденные с количество.
- 7) Задан исходный текст на русском языке. Длина текста - не более NL символов, длина строки - не более N_с слова – не более NW символов. Исходный текст должен заканчиваться точкой. Составить процедуру, которая находит слова, начинающиеся с заданной буквы. В качестве результата вывести исходный текст, найденные с количество.
- 8) Задан исходный текст на русском языке. Длина текста - не более NL символов, длина строки - не более N_с слова – не более NW символов. Исходный текст должен заканчиваться точкой. Составить процедуру, которая находит слова, в которые входит заданная буква, но она не является первой буквой слова. В качестве результата вывести исходный текст, найденные слова и их количество.
- 9) Задан исходный текст на русском языке. Длина текста - не более NL символов, длина строки - не более N_с слова – не более NW символов. Исходный текст должен заканчиваться точкой. Составить процедуру, которая находит слова, в которые входит заданная буква, но она не является последней буквой слова. В качестве результата вывести исходный текст, найденные слова и их количество.
- 10) Задан исходный текст на русском языке. Длина текста - не более NL символов, длина строки - не более N_с слова – не более NW символов. Исходный текст должен заканчиваться точкой. Составить процедуру, которая находит слова, в которые не входит заданная буква. В качестве результата вывести исходный текст, найденные с количество.









Course Tools

- Schedule List
- Progress Report