

Чернівецький національний університет імені Юрія Федьковича  
Інститут фізико-технічних та комп'ютерних наук  
Відділ комп'ютерних технологій  
Кафедра математичних проблем управління і кібернетики

**Звіт**  
про виконання лабораторної роботи №1 «Робота з  
неформатованим текстом у PowerShell»  
з дисципліни  
**“Адміністрування операційних систем”**

Виконав: студент 441 групи  
Бужак Андрій

Перевірів: .....асист. Коцур М.П.

Оцінка:  
Дата захисту:

Чернівці  
2019

Завдання: Використовуючи редактор notepad створити у робочій папці файл із текстом. В якості розділових знаків використовувати пропуски, коми, крапки, знак оклику та знак питання (при необхідності, знаки можна комбінувати). Файл повинен містити не менше 1000 символів.

Використовуючи командлети PowerShell, розробити сценарій для підрахунку кількості слів у тексті, кількості використаних у тексті розділових знаків із урахуванням типу знаку та вивести слова по одному на рядок на екран. Оформити звіт про виконання роботи, у якому відобразити детальну інформацію про використані командлети та їх параметри, роздрук тексту сценарію, результати тестування сценарію. Роздрукований звіт та працездатність сценарію перевіряє та оцінює викладач.

Максимальна кількість балів за роботу – 10.

Термін здачі роботи – до 27.09.19.

### **Хід виконання**

Мною було розроблено сценарій для розв'язування поставленої задачі.

```
"  
$filePath = Read-Host -Prompt 'Input file path';  
"  
  
if (!(Test-Path $filePath))  
{  
    'File not found.';  
    "  
    pause;  
    exit;  
}  
  
$fileContent = [string](Get-Content $filePath);  
$separatorToCount = @ {[char]' ' = 0; [char]',' = 0; [char]'. = 0; [char]!' = 0; [char]'?' =  
0};  
  
foreach ($c in $fileContent.ToCharArray())  
{  
    if ($separatorToCount.ContainsKey($c))  
    {  
        $separatorToCount[$c]++;  
    }  
}  
  
$words = $fileContent.Split($separatorToCount.Keys).Where{ $_ -ne "" };  
  
'Number of words: ' + $words.Count;  
"  
  
"Separator to it's number:";  
foreach ($key in $separatorToCount.Keys)  
{  
    [string]::Format("{0}' = {1}", $key, $separatorToCount[$key]);  
}
```

```
",";
```

```
'Words:';
```

```
$words;
```

```
",";
```

```
pause;
```

### Пояснення до сценарію

1. Прошу користувача ввести шлях до файлу; перевіряю чи такий файл існує - якщо ні, то виводжу відповідне повідомлення і завершую виконання сценарію. “” використовую для відображення порожнього рядка.

```
",";
```

```
$filePath = Read-Host -Prompt 'Input file path';
```

```
",";
```

```
if (!(Test-Path $filePath))
```

```
{
```

```
    'File not found.';
```

```
    ";
```

```
    pause;
```

```
    exit;
```

```
}
```

2. Читаю вміст файлу. Оголошую хеш-таблицю формату [роздільник]=>[кількість].

```
$fileContent = [string](Get-Content $filePath);
```

```
$separatorToCount = @ {[char]' ' = 0; [char]',' = 0; [char]'. ' = 0; [char]!' = 0; [char]?' = 0};
```

3. Перевіряю всі символи вмісту файлу та наповнюю хеш-таблицю даними про кількість кожного роздільника.

```
foreach ($c in $fileContent.ToCharArray())
```

```
{
```

```
    if ($separatorToCount.ContainsKey($c))
```

```
    {
```

```
        $separatorToCount[$c]++;
```

```
    }
```

```
}
```

4. Методом Split .NET об'єкта System.String отримую масив підрядків розділених вказаними мною роздільниками, та відбираю тільки непорожні рядки методом Where. Таким чином отримую всі слова із вмісту файлу.

```
$words = $fileContent.Split($separatorToCount.Keys).Where{$_ -ne ""};
```

5. Виводжу кількість слів, дані про кількість кожного з роздільників та список всіх слів по одному в рядок.

```
'Number of words: ' + $words.Count;  
";
```

```
"Separator to it's number:";  
foreach ($key in $separatorToCount.Keys)  
{  
    [string]::Format("{0}' = {1}", $key, $separatorToCount[$key]);  
}  
";
```

```
'Words:';  
$words;  
";
```

```
pause;
```

## Результат виконання

The screenshot displays the results of a text analysis script. It consists of three overlapping windows:

- Text Editor (text.txt — Блокнот):** Contains the standard Lorem Ipsum placeholder text.
- PowerShell Window 1:** Shows the execution of a script named `Lab_1.ps1`. It prompts for an input file path and reports that the file was not found. It then prompts for a separator character, which is entered as a comma (`,`). The script then counts the words in the text, resulting in 531 words.
- PowerShell Window 2:** Displays the output of the script, which is a list of words from the text, sorted alphabetically. The words include: `amet, consectetur, adipiscing, elit, sed, do, eiusmod, tempor, incididunt, ut, labore, et, dolore, magna, aliqua, Rhoncus, urna, neque, viverra, justo, nec, ultrices, dui, sapien, eget, Non, quam, lacus, suspendisse, faucibus, interdum, posuere, lorem, Senectus, Elementum, pulvinar, etiam, non, quam, lacus, vestibulum, mattis, ullamcorper, velit, sed, ullamcorper, Lacinia, quis, vel, eros, donec, ac, odio, tempor, orci, dapibus, Tempus, quam, pellentesque, nec, nam, aliquam, sem, et`.

## Висновки

Отже, аналіз результатів виконання роботи дозволяє стверджувати, що завдання виконане в повному обсязі – розроблений код працює згідно поставлених вимог.