Отчет по Лабораторной работе №3

«Оболочка командной строки Windows PowerShell»

Группа 2-МВ-4

Ярошевская Д.А.

1-3. Ознакомление с теоретическими сведениями и оболочкой

4. Вывести содержимое каталога Windows по указанному в таблице формату на экран и в текстовый файл

Что выводить (имена, размер, дата создания, атрибуты)	Сортировать по	Условие отбора
Только файлы	По размеру	Размер > 10000

Собираем последовательность команд с помощью знака | , в которой параметр **-File** показывает только файлы,
команда **Where-Object {$_.Length -gt 10000}** фильтрует их по заданному условию, **Sort-Object** сортирует, **Select-Object** оставляет только необходимые сведения (имя, размер, дата создания, атрибуты), **Tee-Object** выводит результат на экран и сохраняет в файл.

```
PS C:\education\lab 3> Get-ChildItem "C:\Windows" -File | Where-Object { $_.Length -gt 10000} |
>> Sort-Object Length | Select-Object Name, Length, CreationTime, Attributes |
>> Tee-Object -FilePath "C:\education\lab 3\task4.txt"

Name                      Length CreationTime            Attributes
----                      ------ ------------            ----------
winhlp32.exe               12288 24.06.2025 22:19:31         Archive
PLDDATA.XML                19868 18.11.2020 19:01:18         Archive
Core.xml                   23885 01.04.2024 19:33:07         Archive
CoreSingleLanguage.xml     29857 21.04.2020 0:02:42          Archive
WinREDism.log              39909 18.11.2020 7:53:40          Archive
hh.exe                     40960 23.06.2025 14:08:06         Archive
Nord.Setup.dll             41024 25.08.2023 21:56:29         Archive
mib.bin                    43131 01.04.2024 10:22:33         Archive
pyshellext.amd64.dll       52968 04.02.2025 16:38:26         Archive
bootstat.dat               67584 23.06.2025 14:26:03 System, Archive
setupact.log               73288 23.06.2025 16:17:17         Archive
twain_32.dll               78848 09.09.2025 23:58:37         Archive
bfsvc.exe                 126976 15.07.2025 13:57:44         Archive
splwow64.exe              229376 15.10.2025 16:38:56         Archive
WMSysPr9.prx              316640 01.04.2024 10:22:19         Archive
notepad.exe               360448 09.09.2025 23:58:22         Archive
regedit.exe               585728 15.10.2025 16:38:58         Archive
pyw.exe                   770552 04.02.2025 16:37:54         Archive
py.exe                    772840 04.02.2025 16:37:52         Archive
HelpPane.exe             1081344 09.09.2025 23:58:14         Archive
explorer.exe             3129376 15.10.2025 16:38:46         Archive
ru-RU.log               11541411 18.11.2020 18:20:53         Archive
PFRO.log                24115158 18.11.2020 7:26:49          Archive
```

Просмотрим содержимое файла с помощью **Get-Content.**

```
PS C:\education\lab 3> Get-Content "C:\education\lab 3\task4.txt"

Name                      Length CreationTime            Attributes
----                      ------ ------------            ----------
winhlp32.exe               12288 24.06.2025 22:19:31         Archive
PLDDATA.XML                19868 18.11.2020 19:01:18         Archive
Core.xml                   23885 01.04.2024 19:33:07         Archive
CoreSingleLanguage.xml     29857 21.04.2020 0:02:42          Archive
WinREDism.log              39909 18.11.2020 7:53:40          Archive
hh.exe                     40960 23.06.2025 14:08:06         Archive
Nord.Setup.dll             41024 25.08.2023 21:56:29         Archive
mib.bin                    43131 01.04.2024 10:22:33         Archive
pyshellext.amd64.dll       52968 04.02.2025 16:38:26         Archive
bootstat.dat               67584 23.06.2025 14:26:03 System, Archive
setupact.log               73288 23.06.2025 16:17:17         Archive
twain_32.dll               78848 09.09.2025 23:58:37         Archive
bfsvc.exe                 126976 15.07.2025 13:57:44         Archive
splwow64.exe              229376 15.10.2025 16:38:56         Archive
WMSysPr9.prx              316640 01.04.2024 10:22:19         Archive
notepad.exe               360448 09.09.2025 23:58:22         Archive
regedit.exe               585728 15.10.2025 16:38:58         Archive
pyw.exe                   770552 04.02.2025 16:37:54         Archive
py.exe                    772840 04.02.2025 16:37:52         Archive
HelpPane.exe             1081344 09.09.2025 23:58:14         Archive
explorer.exe             3129376 15.10.2025 16:38:46         Archive
ru-RU.log               11541411 18.11.2020 18:20:53         Archive
PFRO.log                24115158 18.11.2020 7:26:49          Archive
```

5. Вывести в текстовый файл список свойств процесса, возвращаемый командлетом Get-process и на экран – их общее количество.

Командлет **Get-Process** выводит список всех процессов, запущенных в системе, **Get-Member** позволяет просмотреть полный список элементов объекта, а параметр **–MemberType Property** – только те элементы, которые являются свойствами.

С помощью **Out-File** записываю в файл, затем из него же подсчитываю количество свойств с помощью **.Count**.

```
PS C:\education\lab 3> Get-Process | Get-Member -MemberType Property | Out-File "C:\education\lab 3\task5.txt"
>> (Get-Content "C:\education\lab 3\task5.txt").Count
60
```

Проверяем содержимое файла:

```
PS C:\education\lab 3> Get-Content "C:\education\lab 3\task5.txt"


   TypeName: System.Diagnostics.Process

Name                       MemberType Definition
----                       ---------- ----------
BasePriority               Property   int BasePriority {get;}
Container                  Property   System.ComponentModel.IContainer Container {get;}
EnableRaisingEvents        Property   bool EnableRaisingEvents {get;set;}
ExitCode                   Property   int ExitCode {get;}
ExitTime                   Property   datetime ExitTime {get;}
Handle                     Property   System.IntPtr Handle {get;}
HandleCount                Property   int HandleCount {get;}
HasExited                  Property   bool HasExited {get;}
Id                         Property   int Id {get;}
MachineName                Property   string MachineName {get;}
MainModule                 Property   System.Diagnostics.ProcessModule MainModule {get;}
MainWindowHandle           Property   System.IntPtr MainWindowHandle {get;}
MainWindowTitle            Property   string MainWindowTitle {get;}
MaxWorkingSet              Property   System.IntPtr MaxWorkingSet {get;set;}
MinWorkingSet              Property   System.IntPtr MinWorkingSet {get;set;}
Modules                    Property   System.Diagnostics.ProcessModuleCollection Modules {get;}
NonpagedSystemMemorySize   Property   int NonpagedSystemMemorySize {get;}
NonpagedSystemMemorySize64 Property   long NonpagedSystemMemorySize64 {get;}
PagedMemorySize            Property   int PagedMemorySize {get;}
```

6. Создать текстовый файл, содержащий список выполняемых процессов, упорядоченный по возрастанию указанного в таблице параметра. Имена параметров процессов указаны в таблице.

Список выводимых параметров процессов	Сортировать по значению параметра	Вывести процессы, у которых
Имя процесса, PriorityClass, ProductVersion, Id	Имя процесса	Id > 100

Аналогично 4 заданию используем **Where-Object** для фильтрации, **Sort-Object** для сортировки, **Select-Object** для выбора выводимых параметров и записываем в файл, затем проверяем.

```
PS C:\education\lab 3> Get-Process | Where-Object { $_.Id -gt 100 } | Sort-Object Name |
>> Select-Object Name, PriorityClass, ProductVersion, Id | Out-File "C:\education\lab 3\task6.txt"
PS C:\education\lab 3> Get-Content "C:\education\lab 3\task6.txt"

Name                                PriorityClass ProductVersion                                Id
----                                ------------- --------------                                --
AdobeIPCBroker                             Normal 7.2.1.38                                   17624
AggregatorHost                                                                                9900
AltServer                                  Normal 1.7.2.0                                    25588
AppleMobileDeviceService                                                                      4372
ApplePhotoStreams                          Normal 7.21.0.23                                  13536
ApplicationFrameHost                       Normal 10.0.26100.3912                            25028
APSDaemon                                  Normal                                            19252
audiodg                                                                                      27268
AutoModeDetect                                                                               15432
backgroundTaskHost                         Normal 10.0.26100.1                               28780
backgroundTaskHost                         Normal 10.0.26100.1                               28908
backgroundTaskHost                         Normal 10.0.26100.1                               13192
backgroundTaskHost                         Normal 10.0.26100.1                               23272
browserhost                                Normal 4,1,1,0                                    27596
```

7. Создать HTML-файл, содержащий список выполняемых процессов, упорядоченный по возрастанию указанного в таблице параметра. Имена параметров процессов указаны в таблице.

Список выводимых параметров процессов	Сортировать по значению параметра	Вывести процессы, у которых
Имя процесса, PriorityClass, ProductVersion, Id	Имя процесса	Id > 100

Для формирования списка необходимых процессов используем команды из 6 задания, с помощью командлета **ConvertTo-Html** и параметра **–Property** преобразовываем данные в html, сохраняем в файл с расширением **.html**. Для просмотра используем Invoke Item.

```
PS C:\education\lab 3> Get-Process | Where-Object { $_.Id -gt 100 } | Sort-Object Name |
>> ConvertTo-Html -Property Name, PriorityClass, ProductVersion, Id | Out-File "C:\education\lab 3\task7.html"
PS C:\education\lab 3> Invoke-Item "C:\education\lab 3\task7.html"
```

Name	PriorityClass	ProductVersion	Id
AdobeIPCBroker	Normal	7.2.1.38	17624
AggregatorHost			9900
AltServer	Normal	1.7.2.0	25588
AppleMobileDeviceService			4372
ApplePhotoStreams	Normal	7.21.0.23	13536
ApplicationFrameHost	Normal	10.0.26100.3912	25028
APSDaemon	Normal		19252
audiodg			27268
AutoModeDetect			15432
backgroundTaskHost	Normal	10.0.26100.1	23272
backgroundTaskHost	Normal	10.0.26100.1	13192
browserhost	Normal	4,1,1,0	27596
browserhost	Normal	4,1,1,0	22084

8. Найти суммарный объем всех графических файлов (bmp, jpg), находящихся в каталоге Windows и всех его подкаталогах.

Используем параметр **–Recurse** для прохода по всем подкатологам, **-Include *.jpg *.bmp** для фильтрации по нужным расширениям. **-ErrorAction SilentlyContinue** игнорирует ошибки доступа. **Measure-Object Length –Sum** указывает, какое свойство измерять. Оборачиваем в скобки и добавляем **.Sum**, чтобы извлечь только числовое значение суммы.

```
PS C:\education\lab 3> (Get-ChildItem "C:\Windows" -Recurse -Include *.bmp, *.jpg -File
-ErrorAction SilentlyContinue | Measure-Object Length -Sum).Sum
105919066
```

9. Вывести на экран сведения о ЦП компьютера.

```
PS C:\education\lab 3> Get-WmiObject -Class Win32_Processor | Format-list *


PSComputerName                          : LAPTOP-FLLF3U0T
Availability                            : 3
CpuStatus                               : 1
CurrentVoltage                          : 8
DeviceID                                : CPU0
ErrorCleared                            :
ErrorDescription                        :
LastErrorCode                           :
LoadPercentage                          : 11
Status                                  : OK
StatusInfo                              : 3
AddressWidth                            : 64
DataWidth                               : 64
ExtClock                                : 100
L2CacheSize                             : 5120
```

10. Найти максимальное, минимальное и среднее значение времени выполнения командлетов dir и ps.

Инициализируем массив **$dirTimes**, в который записываем 5 значений времени выполнения командлета **dir**, измеряем их с помощью **Measure-Command**. Просматриваем получившиеся значения.

```
PS C:\education\lab 3> $dirTimes = 1..5 | ForEach-Object { (Measure-Command { dir }).TotalSeconds }
PS C:\education\lab 3> $dirTimes
0,0143759
0,0004256
0,0003443
0,0003339
0,0003191
```

К массиву применяем командлет **Measure-Object** и требуемые атрибуты.

```
PS C:\education\lab 3> $dirTimes | Measure-Object -maximum -minimum -average


Count    : 5
Average  : 0,00315976
Sum      :
Maximum  : 0,0143759
Minimum  : 0,0003191
Property :
```

То же самое делаем с командлетом ps.

```
PS C:\education\lab 3> $psTimes = 1..5 | ForEach-Object { (Measure-Command { ps }).TotalSeconds }
PS C:\education\lab 3> $psTimes
0,0167307
0,0042922
0,0069671
0,0038146
0,0055276
PS C:\education\lab 3> $psTimes | Measure-Object -maximum -minimum -average


Count    : 5
Average  : 0,00746644
Sum      :
Maximum  : 0,0167307
Minimum  : 0,0038146
Property :
```

11. Нахождение в заданном каталоге файла наибольшего размера и трех файлов наименьшего размера

Sort-Object Length –Descending сортирует список по размеру от большего к меньшему, **Select-Object –First 1** выбирает первый в этом списке файл.

```
PS C:\education\lab 3> Get-ChildItem "C:\Windows" -File | Sort-Object Length -Descending |
>> Select-Object -First 1


    Каталог: C:\Windows


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-a----        25.10.2025      9:31       24115158 PFRO.log
```

Аналогично сортируем от меньшего к большему и берем первые три файла.

```
PS C:\education\lab 3> Get-ChildItem "C:\Windows" -File | Sort-Object Length |
>> Select-Object -First 3


    Каталог: C:\Windows


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-a----        23.06.2025     16:17              0 setuperr.log
-a----        18.11.2020     18:20              6 core.ver
-a----        18.11.2020      1:59             12 csup.txt
```