Задание 4. Вывести содержимое каталога Windows по указанному в табл. 5 формату на экран и в текстовый файл.

3, 8	Только подкаталоги	Именам	Последняя буква имени S или T

```
PS C:\Users\Sofie> Get-ChildItem -Path "C:\Windows" -Directory | Where-Object { $_.Name[-1] -match '[sStT]' } | Sort-Object Name | Se
lect-Object Name, FullName | Tee-Object "$env:USERPROFILE\Desktop\subdirs_ST.txt" >> C:\Users\Sofie\Desktop\Уник\2OCCT\Pr3\files.txt
PS C:\Users\Sofie> cd C:\Users\Sofie\Desktop\Уник\2OCCT\Pr3
PS C:\Users\Sofie\Desktop\Уник\2OCCT\Pr3> type files.txt

Name                    FullName
----                    --------
addins                  C:\Windows\addins
appcompat               C:\Windows\appcompat
AppReadiness            C:\Windows\AppReadiness
Boot                    C:\Windows\Boot
Containers              C:\Windows\Containers
Cursors                 C:\Windows\Cursors
diagnostics             C:\Windows\diagnostics
Downloaded Program Files C:\Windows\Downloaded Program Files
en-US                   C:\Windows\en-US
Fonts                   C:\Windows\Fonts
InboxApps               C:\Windows\InboxApps
L2Schemas               C:\Windows\L2Schemas
LiveKernelReports       C:\Windows\LiveKernelReports
Logs                    C:\Windows\Logs
Microsoft.NET           C:\Windows\Microsoft.NET
ModemLogs               C:\Windows\ModemLogs
Offline Web Pages       C:\Windows\Offline Web Pages
PolicyDefinitions       C:\Windows\PolicyDefinitions
RemotePackages          C:\Windows\RemotePackages
Resources               C:\Windows\Resources
schemas                 C:\Windows\schemas
ServiceProfiles         C:\Windows\ServiceProfiles
ShellComponents         C:\Windows\ShellComponents
ShellExperiences        C:\Windows\ShellExperiences
symbols                 C:\Windows\symbols
SystemApps              C:\Windows\SystemApps
SystemResources         C:\Windows\SystemResources
Tasks                   C:\Windows\Tasks
Vss                     C:\Windows\Vss
WaaS                    C:\Windows\WaaS
WinSxS                  C:\Windows\WinSxS
```

Команда получает список только подкаталогов (-Directory) из каталога C:\Windows.
Фильтр Where-Object { $_.Name[-1] -match '[sStT]' } выбирает только те, у которых последняя буква имени — S или T.
Результат сортируется по имени (Sort-Object Name), выводятся столбцы *Name* и *FullName*, и всё сохраняется в файл files.txt, проверка с помощью type.

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

```
PS C:\Users\Sofie\Desktop\Уник\2ОССТ\Pr3> $pro = Get-Process
PS C:\Users\Sofie\Desktop\Уник\2ОССТ\Pr3> $pro > list.txt
PS C:\Users\Sofie\Desktop\Уник\2ОССТ\Pr3> $pro.count
161
PS C:\Users\Sofie\Desktop\Уник\2ОССТ\Pr3> type list.txt

Handles  NPM(K)    PM(K)      WS(K)     CPU(s)      Id  SI ProcessName
-------  ------    -----      -----     ------      --  -- -----------
    492      21    28184      23288      14,70   12020   1 Adobe Crash Processor
    182      12     2848       8384       0,28   10268   1 AdobeIPCBroker
    308      14    22076      37528       0,41     676   1 ai
    308      14    47704      57840       0,38   14996   1 ai
    307      18    43356      27876       0,17   16308   1 aimgr
    668      56   149112     106444       2,88    2032   1 AmneziaVPN
    392      12     2572      12568                2904   0 AmneziaVPN-service
    343     152    47236      44760              11512   0 AmneziaVPN-service
   3771     488   405556     146016               2920   0 avp
   1094     180   145824       8152      63,95    5544   1 avpui
     50       4      608       2408       0,00   11268   1 CCXProcess
    380      29    14544      24376       0,13    8756   1 CefSharp.BrowserSubprocess
    406      32    17972      30172       0,06   10992   1 CefSharp.BrowserSubprocess
   1579      60    82372     107232     111,66   11076   1 CefSharp.BrowserSubprocess
    463      33    17024      34872       0,63   11472   1 CefSharp.BrowserSubprocess
    458      32    14960      26024      43,19   11572   1 CefSharp.BrowserSubprocess
    413      27    79748      98344       2,78     680   1 chrome
    288      23    32436      64408       0,61    1052   1 chrome
    349      26    58284     122216       1,34    3000   1 chrome
    359      37    71652     123984       9,55    3904   1 chrome
    254      15    12356      18004       0,19    5368   1 chrome
    312      22    25532      55896       0,64    6084   1 chrome
    570      52   109688     158988      52,52    6252   1 chrome
    621      26    65064      40864       0,52    7480   1 chrome
    398      34   134916     171680      43,30    7540   1 chrome
    340      27   112144     155120      23,73    7988   1 chrome
    557      32    91180     122852       5,36    9304   1 chrome
    340      26    97280     144248       8,39   10116   1 chrome
    223      19    14080      30688       0,03   10200   1 chrome
    249      16    11872      21488       1,22   11332   1 chrome
```

В переменную $pro сохранила список всех процессов, которые выводит команда Get-Process.
Записала этот список в текстовый файл list.txt.
Команда $pro.count показала, сколько всего процессов запущено.
Затем открыла файл и убедилась, что туда сохранились данные обо всех процессах.

Задание 6. Создать текстовый файл, содержащий список выполняемых процессов, упорядоченный по возрастанию указанного в табл.6 параметра. Имена параметров процессов указаны в (табл. 6)

Список выводимых параметров процессов	Сортировать по значению параметра	Вывести процессы, у которых
Id, Имя процесса,	Время старта	Id > 40

```
PS C:\Users\Sofie\Desktop\Уник\2ОССТ\Pr3> Get-Process | Where-Object { $_.Id -gt 100 } |
>> Select-Object ProcessName, Id, PriorityClass, UserProcessorTime, TotalProcessorTime |
>> Sort-Object TotalProcessorTime |
>> Out-File "time.txt"
PS C:\Users\Sofie\Desktop\Уник\2ОССТ\Pr3> type time.txt


ProcessName        : RzBTLEManager
Id                 : 3968
PriorityClass      :
UserProcessorTime  :
TotalProcessorTime :

ProcessName        : RzAppManager
Id                 : 3956
PriorityClass      :
UserProcessorTime  :
TotalProcessorTime :

ProcessName        : RzChromaConnectServer
Id                 : 4016
PriorityClass      :
UserProcessorTime  :
TotalProcessorTime :

ProcessName        : RzChromaConnectManager
Id                 : 3992
PriorityClass      :
UserProcessorTime  :
TotalProcessorTime :

ProcessName        : Razer Synapse Service
Id                 : 6604
PriorityClass      :
UserProcessorTime  :
TotalProcessorTime :

ProcessName        : OutlineService
Id                 : 2348
PriorityClass      :
UserProcessorTime  :
TotalProcessorTime :
```

Команда Get-Process получает сведения обо всех запущенных процессах.
Фильтр Where-Object { $_.Id -gt 100 } выбирает только те процессы, у которых идентификатор больше 100.
С помощью Select-Object отображаются только нужные поля: *ProcessName*, *Id*, *PriorityClass*, *UserProcessorTime* и *TotalProcessorTime*.
Sort-Object TotalProcessorTime сортирует их по общему времени использования процессора.

Задание 7. Создать HTML-файл, содержащий список выполняемых процессов, упорядоченный по возрастанию указанного в табл.5 параметра. Имена параметров процессов указаны в табл. 5.

```
PS C:\Users\Sofie\Desktop\Уник\2ОССТ\Pr3> Get-Process | ? { $_.Id -gt 100 }
| select ProcessName,Id,PriorityClass,UserProcessorTime,TotalProcessorTime |
 sort TotalProcessorTime | ConvertTo-Html | Out-File "task7.html"
>>
```

ProcessName	Id	PriorityClass	UserProcessorTime	TotalProcessorTime
CCXProcess	11268	Normal	00:00:00	00:00:00
cmd	16920	Normal	00:00:00.0156250	00:00:00.0156250
conhost	11352	Normal	00:00:00.0312500	00:00:00.0312500
conhost	16948	Normal	00:00:00.0156250	00:00:00.0312500
jusched	11300	Normal	00:00:00.0156250	00:00:00.0468750
chrome	10372	Idle	00:00:00.0468750	00:00:00.0625000
CefSharp.BrowserSubprocess	10992	Idle	00:00:00.0468750	00:00:00.0625000
CompPkgSrv	17708	Normal	00:00:00.0468750	00:00:00.0625000
CefSharp.BrowserSubprocess	8756	Normal	00:00:00.1093750	00:00:00.1250000
aimgr	16308	Normal	00:00:00.0937500	00:00:00.1718750
chrome	5368	Normal	00:00:00.1562500	00:00:00.1875000
UserOOBEBroker	12320	Normal	00:00:00.1093750	00:00:00.1875000
taskhostw	6484	Normal	00:00:00.0781250	00:00:00.2031250
chrome	15924	Normal	00:00:00.1406250	00:00:00.2187500
RuntimeBroker	8972	Normal	00:00:00.0781250	00:00:00.2500000
dllhost	8464	Normal	00:00:00.1406250	00:00:00.2500000
AdobeIPCBroker	10268	Normal	00:00:00.2656250	00:00:00.2812500
ai	676	Normal	00:00:00.2187500	00:00:00.4062500
ai	14996	Normal	00:00:00.4375000	00:00:00.4687500

Команда формирует список процессов по заданию 5 и сохраняет результат в HTML-файл. У некоторых системных процессов значения времени не отображаются, так как эти данные недоступны без повышенных прав.

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

```
PS C:\Users\Sofie\Desktop\Уник\2ОССТ\Pr3> (Get-ChildItem 'C:\Windows' -Recurse -File -Include
*.bmp,*.jpg -ErrorAction SilentlyContinue | Measure-Object Length -Sum).Sum
53814246
```

Команда перебирает файлы bmp и jpg в каталоге C:\Windows и всех его подкаталогах (-Recurse), суммирует их размеры (Measure-Object Length -Sum) и выводит общий объём в байтах. Ключ -ErrorAction SilentlyContinue скрывает сообщения о недоступных папках.

Задание 9. Вывести на экран сведения о ЦП компьютера.

```
PS C:\Users\Sofie> Get-WmiObject Win32_Processor | Format-List
>>


Caption            : Intel64 Family 6 Model 158 Stepping 11
DeviceID           : CPU0
Manufacturer       : GenuineIntel
MaxClockSpeed      : 3600
Name               : Intel(R) Core(TM) i3-8100 CPU @ 3.60GHz
SocketDesignation  : CPUSocket
```

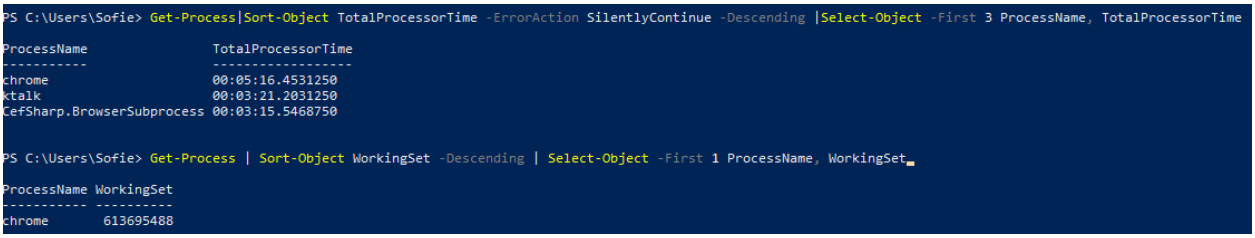
Команда Get-WmiObject Win32_Processor получает информацию о процессоре. Format-List выводит свойства в виде списка, чтобы каждое значение отображалось на отдельной строке.

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

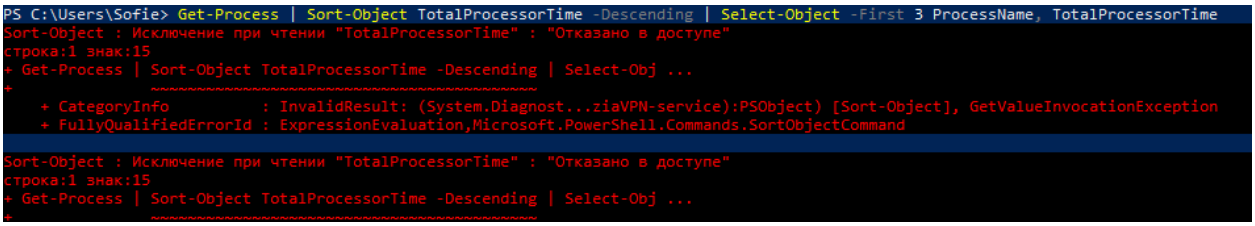
```
PS C:\Users\Sofie> $dirs = 1..5 | ForEach-Object {(Measure-Command{Get-ChildItem}).TotalMilliseconds}
PS C:\Users\Sofie> $dirs
5,7838
0,9317
0,8556
0,924
0,8566
PS C:\Users\Sofie> $dirs | Measure-Object -Minimum -Maximum -Average


Count    : 5
Average  : 1,87034
Sum      :
Maximum  : 5,7838
Minimum  : 0,8556
Property :



PS C:\Users\Sofie>  $pss = 1..5 | ForEach-Object {(Measure-Command{Get-Process}).TotalMilliseconds}
PS C:\Users\Sofie> $pss
7,2054
1,7851
1,8565
1,7829
4,0438
PS C:\Users\Sofie>  $pss | Measure-Object -Minimum -Maximum -Average


Count    : 5
Average  : 3,33474
Sum      :
Maximum  : 7,2054
Minimum  : 1,7829
Property :
```

1. И для dir, и для ps (get-childitem, get-process) создаю по массиву из пяти элементов, записываю в миллисекундах.
2. Вывожу массив для наглядности
3. С помощью командлета Measure-Object и необходимых мне параметров вычисляю среднее, минимальное и максимальное значения.

Задание 11. Разработать командлет для нахождения среди выполняющихся процессов имен трех процессов, использовавших более всего процессорного времени нахождения и среди выполняющихся процессов имени процесса с наибольшим размером рабочего множества страниц.



Командлет Get-Process используется для получения списка всех выполняющихся процессов в системе. Результаты передаются следующему командлету — Sort-Object, который сортирует объекты по значению свойства TotalProcessorTime. Этот параметр отражает суммарное время, затраченное процессом на использование процессора. С помощью параметра -Descending сортировка выполняется в порядке убывания. Командлет Select-Object с параметром -First 3 выбирает только первые три объекта из отсортированного списка и выводит значения свойств ProcessName и TotalProcessorTime.



Поскольку у меня до выдачи результата вылезала огромная куча ошибок и было неудобно скринить, добавила параметр -ErrorAction SilentlyContinue, который позволяет игнорировать процессы, к которым нет разрешения.