

ГУАП

КАФЕДРА № 42

ОТЧЕТ
ЗАЩИЩЕН С ОЦЕНКОЙ
ПРЕПОДАВАТЕЛЬ

доцент, канд. техн. наук
должность, уч. степень, звание

подпись, дата

А.В. Аграновский
инициалы, фамилия

ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ №5

Разработка командлетов PowerShell

по курсу: Инструментальные средства
информационных систем

РАБОТУ ВЫПОЛНИЛ

СТУДЕНТ гр. №

4321

подпись, дата

Г.В. Буренков
инициалы, фамилия

Санкт-Петербург 2025

СОДЕРЖАНИЕ

1 Цель работы	2
2 Выполненные упражнения со скриншотами.....	3
3 Справочная информация о командах из индивидуального задания.....	11
4 Выводы о проделанной работе	16
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	17

1 Цель работы

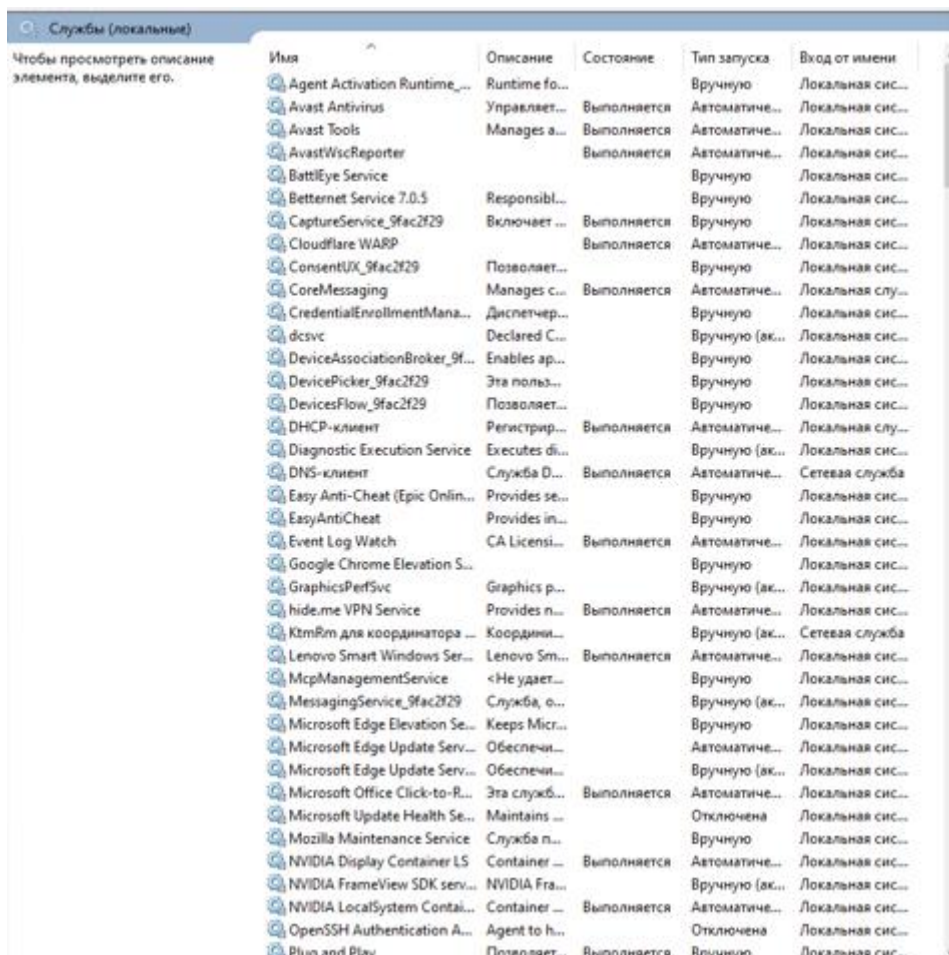
Целью данной лабораторной работы является изучение командлетов PowerShell, усовершенствование навыков администрирования в ОС Windows, а также выполнение индивидуального варианта. На рисунке 1 изображены варианты для выполнения данной работы.

№ варианта	1	2	3	4	5	6	7	8	9	10
Номер первого командлета	1	2	3	4	5	6	7	8	9	10
Номер второго командлета	21	4	17	15	23	16	13	19	12	5
Номер третьего командлета	13	10	12	11	17	24	22	27	21	26
№ варианта	11	12	13	14	15	16	17	18	19	20
Номер первого командлета	11	12	13	14	15	16	17	18	19	20
Номер второго командлета	24	22	5	23	6	11	22	15	18	14
Номер третьего командлета	3	1	25	8	24	7	4	25	4	9
№ варианта	21	22	23	24	25	26	27	28	29	30
Номер первого командлета	21	22	23	24	25	26	27	28	4	5
Номер второго командлета	27	8	2	12	7	15	5	17	16	11
Номер третьего командлета	8	14	13	5	17	9	12	8	28	21

Рисунок 1 – Табличка вариантов

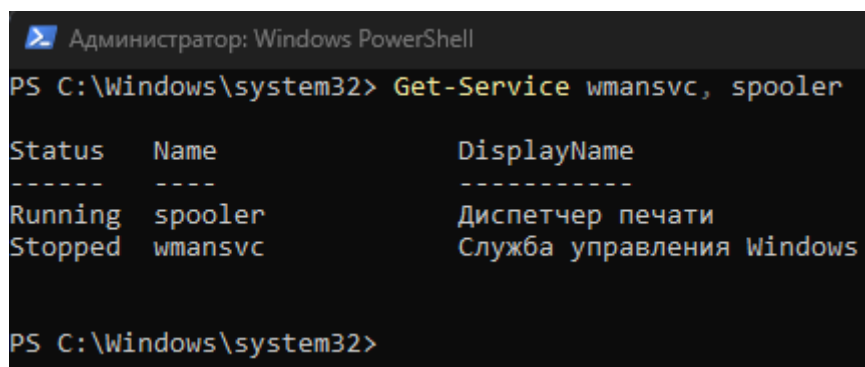
2 Выполненные упражнения со скриншотами

В процессе выполнения упражнений 2.5 - 2.11 я разобрался различных командлетах PowerShell. В первую очередь я разобрался с командлентами Get-Service, Sort-Object, Get-Process. На рисунках 2-17 изображены различные результаты командлет.



Имя	Описание	Состояние	Тип запуска	Вход от имени
Agent Activation Runtime...	Runtime fo...		Вручную	Локальная сис...
Avast Antivirus	Управляет...	Выполняется	Автоматиче...	Локальная сис...
Avast Tools	Manages a...	Выполняется	Автоматиче...	Локальная сис...
AvastWscReporter		Выполняется	Автоматиче...	Локальная сис...
BattlEye Service			Вручную	Локальная сис...
Betternet Service 7.0.5	Responsibl...		Вручную	Локальная сис...
CaptureService_9fac2f29	Включает ...	Выполняется	Вручную	Локальная сис...
Cloudflare WARP		Выполняется	Автоматиче...	Локальная сис...
ConsentUX_9fac2f29	Позволяет...		Вручную	Локальная сис...
CoreMessaging	Manages c...	Выполняется	Автоматиче...	Локальная слу...
CredentialEnrollmentMana...	Диспетчер...		Вручную	Локальная сис...
dcsvcs	Declared C...		Вручную (ак...	Локальная сис...
DeviceAssociationBroker_9f...	Enables ap...		Вручную	Локальная сис...
DevicePicker_9fac2f29	Эта польз...		Вручную	Локальная сис...
DevicesFlow_9fac2f29	Позволяет...		Вручную	Локальная сис...
DHCP-клиент	Регистрир...	Выполняется	Автоматиче...	Локальная слу...
Diagnostic Execution Service	Executes di...		Вручную (ак...	Локальная сис...
DNS-клиент	Служба D...	Выполняется	Автоматиче...	Сетевая служба
Easy Anti-Cheat (Epic Onlin...	Provides se...		Вручную	Локальная сис...
EasyAntiCheat	Provides in...		Вручную	Локальная сис...
Event Log Watch	CA Licensi...	Выполняется	Автоматиче...	Локальная сис...
Google Chrome Elevation S...			Вручную	Локальная сис...
GraphicsPerfSvc	Graphics p...		Вручную (ак...	Локальная сис...
hide.me VPN Service	Provides n...	Выполняется	Автоматиче...	Локальная сис...
KtmRm для координатора ...	Координи...		Вручную (ак...	Сетевая служба
Lenovo Smart Windows Ser...	Lenovo Sm...	Выполняется	Автоматиче...	Локальная сис...
McpManagementService	< Не удае...		Вручную	Локальная сис...
MessagingService_9fac2f29	Служба, о...		Вручную (ак...	Локальная сис...
Microsoft Edge Elevation Se...	Keeps Micr...		Вручную	Локальная сис...
Microsoft Edge Update Serv...	Обеспечи...		Автоматиче...	Локальная сис...
Microsoft Edge Update Serv...	Обеспечи...		Вручную (ак...	Локальная сис...
Microsoft Office Click-to-R...	Эта служб...	Выполняется	Автоматиче...	Локальная сис...
Microsoft Update Health Se...	Maintains ...		Отключена	Локальная сис...
Mozilla Maintenance Service	Служба п...		Вручную	Локальная сис...
NVIDIA Display Container LS	Container ...	Выполняется	Автоматиче...	Локальная сис...
NVIDIA FrameView SDK serv...	NVIDIA Fra...		Вручную (ак...	Локальная сис...
NVIDIA LocalSystem Contai...	Container ...	Выполняется	Автоматиче...	Локальная сис...
OpenSSH Authentication A...	Agent to h...		Отключена	Локальная сис...
Plug and Play	Позволяет...	Выполняется	Вручную	Локальная сис...

Рисунок 2 – Список служб



```
Администратор: Windows PowerShell
PS C:\Windows\system32> Get-Service wmansvc, spooler

Status      Name      DisplayName
-----
Running     spooler   Диспетчер печати
Stopped     wmansvc   Служба управления Windows

PS C:\Windows\system32>
```

Рисунок 3 – Изменение шрифта в cmd.exe

```
PS C:\Windows\system32> Get-Service | Sort-Object -property Status
```

Status	Name	DisplayName
Stopped	RemoteRegistry	Удаленный реестр
Stopped	RemoteAccess	Маршрутизация и удаленный доступ
Stopped	refsdedupsvc	Служба Dedup ReFS
Stopped	RpcLocator	Локатор удаленного вызова процедур ...
Stopped	Rockstar Service	Rockstar Game Library Service
Stopped	RetailDemo	Служба демонстрации магазина
Stopped	RasAuto	Диспетчер автоматических подключени...
Stopped	PrintNotify	Расширения и уведомления для принтеров
Stopped	PrintDeviceConf...	Служба настройки принтеров
Stopped	pla	Журналы и оповещения производитель...
Stopped	PushToInstall	Служба PushToInstall Windows
Stopped	PrintWorkflowUs...	PrintWorkflow_a600da64
Stopped	PrintScanBroker...	PrintScanBrokerService
Stopped	SCardSvr	Смарт-карта
Stopped	SharedAccess	Общий доступ к подключению к Интерн...
Stopped	SgrmBroker	Брокер мониторинга среды выполнения...
Stopped	SessionEnv	Настройка сервера удаленных рабочих...
Stopped	SmsRouter	Служба маршрутизатора SMS Microsoft...
Stopped	smphost	SMP дисковых пространств (Майкрософт)
Stopped	shipamsvc	Shared PC Account Manager
Stopped	SensrSvc	Служба наблюдения за датчиками
Stopped	SDRSVC	Программа архивации данных
Stopped	SCPolicySvc	Политика удаления смарт-карт
Stopped	ScDeviceEnum	Служба перечисления устройств чтени...
Stopped	SensorDataService	Служба данных датчиков
Stopped	Sense	Служба Advanced Threat Protection в...
Stopped	SEMgrSvc	Диспетчер платежей и NFC/защищенных...
Stopped	NaturalAuthenti...	Проверка подлинности на основе физи...
Stopped	MsKeyboardFilter	Фильтрация клавиатуры Microsoft
Stopped	msiserver	Установщик Windows
Stopped	Netlogon	Сетевой вход в систему
Stopped	NcdAutoSetup	Автоматическая настройка сетевых ус...
Stopped	NcaSvc	Помощник по подключению к сети
Stopped	MSISCSI	Служба инициатора Майкрософт iSCSI
Stopped	McpManagementSe...	McpManagementService
Stopped	MapsBroker	Диспетчер скачанных карт
Stopped	LxpSvc	Служба Language Experience Service
Stopped	MSDTC	Координатор распределенных транзакций
Stopped	MicrosoftEdgeEl...	Microsoft Edge Elevation Service (M...
Stopped	MessagingServic...	MessagingService_a600da64
Stopped	Netman	Сетевые подключения
Stopped	perceptionsimul...	Служба имитации восприятия Windows
Stopped	PenService_a600...	PenService_a600da64
Stopped	PeerDistSvc	BranchCache
Stopped	PimIndexMainten...	Служба контактных данных_a600da64
Stopped	PhoneSvc	Телефонная связь
Stopped	PerfHost	Хост библиотеки счетчика производит...
Stopped	P9RdrService_a6...	P9RdrService_a600da64
Stopped	NgcSvc	Служба Microsoft Passport
Stopped	NgcCtnrSvc	Контейнер службы Microsoft Passport
Stopped	NetTcpPortSharing	Служба общего доступа к портам Net.Tcp
Stopped	ose64	Office 64 Source Engine
Stopped	OCButtonService	OCButtonService
Stopped	NlaSvc	Сведения о расположении в сети

Рисунок 4 – Результат сортировки служб

PS C:\Windows\system32> Get-Process

Handles	NPM(K)	PM(K)	WS(K)	CPU(s)	Id	SI	ProcessName
279	18	9452	6200	153,88	4160	0	AdjustService
215	13	5636	6560	16,28	6240	0	AggregatorHost
786	52	75020	84248	101,84	26960	31	ApCent
162	9	1732	972	0,06	8988	0	AppVShNotify
833	19	28984	24164	5 015,97	8356	0	audiodg
602	29	8780	48888	0,16	27548	31	backgroundTaskHost
353	30	67028	131204	2,39	1276	31	chrome
277	27	80560	105564	84,98	3088	31	chrome
258	17	8196	23900	14,58	5296	31	chrome
288	24	29420	62744	0,67	8544	31	chrome
308	31	82212	105792	79,56	9004	31	chrome
386	30	103988	114964	14,20	11148	31	chrome
319	30	53432	106176	7,63	11348	31	chrome
429	21	11036	28412	0,30	12548	31	chrome
226	20	14548	34708	0,06	13200	31	chrome
181	14	7612	20300	0,11	15660	31	chrome
196	20	13580	25948	4,92	16848	31	chrome
266	25	47512	69132	7,95	17640	31	chrome
309	26	30376	67400	0,14	22076	31	chrome
518	30	72476	145168	2,83	22108	31	chrome
210	15	10428	21628	0,70	23416	31	chrome
241	11	2260	9720	0,05	24708	31	chrome
369	28	26776	60280	179,61	25588	31	chrome
2056	80	144320	243124	218,72	25736	31	chrome
473	41	62292	54592	5 691,14	25796	31	chrome
170	12	2080	13932	7,05	21164	31	conhost
269	15	7440	26600	3,17	25248	31	conhost
759	26	2688	3024	32,73	1000	0	csrss
792	115	27892	7212	142,20	26852	31	csrss
581	22	8504	36272	9,34	24848	31	ctfmon
775	44	270972	176500	639,50	10780	31	Discord
199	12	10812	33388	0,00	14440	31	Discord
1551	138	912060	658848	4 288,42	15544	31	Discord
1058	50	117404	127912	43,34	19436	31	Discord
279	16	11860	82936	1,77	20144	31	Discord
359	21	15880	55864	4,27	26476	31	Discord
323	31	7952	22420	0,30	6260	31	dllhost

Рисунок 5 – Результат выполнения команды Get-Process

Администратор: Windows PowerShell

PS C:\Windows\system32> Get-Process | Get-Member

TypeName: System.Diagnostics.Process

Name	MemberType	Definition
Handles	AliasProperty	Handles = Handlecount
Name	AliasProperty	Name = ProcessName
NPM	AliasProperty	NPM = NonpagedSystemMemorySize64
PM	AliasProperty	PM = PagedMemorySize64
SI	AliasProperty	SI = SessionId
VM	AliasProperty	VM = VirtualMemorySize64
WS	AliasProperty	WS = WorkingSet64
Disposed	Event	System.EventHandler Disposed(System.Object, System....
ErrorDataReceived	Event	System.Diagnostics.DataReceivedEventHandler ErrorDa...
Exited	Event	System.EventHandler Exited(System.Object, System.Ev...
OutputDataReceived	Event	System.Diagnostics.DataReceivedEventHandler OutputD...
BeginErrorReadLine	Method	void BeginErrorReadLine()
BeginOutputReadLine	Method	void BeginOutputReadLine()
CancelErrorRead	Method	void CancelErrorRead()
CancelOutputRead	Method	void CancelOutputRead()
Close	Method	void Close()
CloseMainWindow	Method	bool CloseMainWindow()
CreateObjRef	Method	System.Runtime.Remoting.ObjRef CreateObjRef(type re...
Dispose	Method	void Dispose(), void IDisposable.Dispose()
Equals	Method	bool Equals(System.Object obj)
GetHashCode	Method	int GetHashCode()
GetLifetimeService	Method	System.Object GetLifetimeService()
GetType	Method	type GetType()
InitializeLifetimeService	Method	System.Object InitializeLifetimeService()
Kill	Method	void Kill()
Refresh	Method	void Refresh()
Start	Method	bool Start()
ToString	Method	string ToString()
WaitForExit	Method	bool WaitForExit(int milliseconds), void WaitForExit()
WaitForInputIdle	Method	bool WaitForInputIdle(int milliseconds), bool WaitF...
_NounName	NoteProperty	string _NounName=Process
BasePriority	Property	int BasePriority {get;}
Container	Property	System.ComponentModel.IContainer Container {get;}
EnableRaisingEvents	Property	bool EnableRaisingEvents {get;set;}
ExitCode	Property	int ExitCode {get;}

Рисунок 6 – Результат выполнения Get-Process в конвейере

```

Администратор: Windows PowerShell
PS C:\Windows\system32> Get-Process | Get-Member | Out-Host -Paging

TypeName: System.Diagnostics.Process

Name           MemberType      Definition
-----
Handles        AliasProperty   Handles = Handlecount
Name           AliasProperty   Name = ProcessName
NPM            AliasProperty   NPM = NonpagedSystemMemorySize64
PM             AliasProperty   PM = PagedMemorySize64
SI            AliasProperty   SI = SessionId
VM            AliasProperty   VM = VirtualMemorySize64
WS            AliasProperty   WS = WorkingSet64
Disposed       Event            System.EventHandler Disposed(System.Object, System....
ErrorDataReceived Event            System.Diagnostics.DataReceivedEventHandler ErrorDa...
Exited         Event            System.EventHandler Exited(System.Object, System.Ev...
OutputDataReceived Event            System.Diagnostics.DataReceivedEventHandler OutputD...
BeginErrorReadLine Method           void BeginErrorReadLine()
BeginOutputReadLine Method           void BeginOutputReadLine()
CancelErrorRead Method           void CancelErrorRead()
CancelOutputRead Method           void CancelOutputRead()
Close          Method           void Close()
CloseMainWindow Method           bool CloseMainWindow()
CreateObjRef   Method           System.Runtime.Remoting.ObjRef CreateObjRef(type re...
Dispose        Method           void Dispose(), void IDisposable.Dispose()
Equals         Method           bool Equals(System.Object obj)
GetHashCode    Method           int GetHashCode()
GetLifetimeService Method         System.Object GetLifetimeService()
GetType        Method           type GetType()
InitializeLifetimeService Method         System.Object InitializeLifetimeService()
Kill           Method           void Kill()
Refresh        Method           void Refresh()
Start          Method           bool Start()
ToString       Method           string ToString()
WaitForExit    Method           bool WaitForExit(int milliseconds), void WaitForExit()
WaitForInputIdle Method         bool WaitForInputIdle(int milliseconds), bool WaitF...
__NounName     NoteProperty    string __NounName=Process
BasePriority   Property        int BasePriority {get;}
Container      Property        System.ComponentModel.IContainer Container {get;}
EnableRaisingEvents Property        bool EnableRaisingEvents {get;set;}
ExitCode       Property        int ExitCode {get;}

```

Рисунок 7 – Вывод процессов

```

Администратор: Windows PowerShell
PS C:\Windows\system32> Get-Process | Sort-object

Handles  NPM(K)  PM(K)  WS(K)  CPU(s)  Id  SI  ProcessName
-----
279      18      9452   6200   153,91  4160 0  AdjustService
217      13      5704   6612   16,28   6240 0  AggregatorHost
791      52      75020  84260  102,92  26960 31  ApCent
162      9       1732   972    0,06    8988 0  AppVShNotify
829      19      28004  24112  0,06    8356 0  audiodg
309      26      30376  67400  0,14    22076 31  chrome
518      30      72484  145172  2,84    22108 31  chrome
196      20      13564  25936  4,94    16848 31  chrome
266      25      47512  69136  8,19    17640 31  chrome
210      15      10420  21620  0,70    23416 31  chrome
2051     80      144712 243492 221,30  25736 31  chrome
473      45      62280  54300  5 691,19 25796 31  chrome
241      11      2260   9720   0,06    24708 31  chrome
409      35      26568  60172  179,88  25588 31  chrome
181      14      7612   20300  0,13    15660 31  chrome
258      17      8196   23900  14,59   5296 31  chrome
288      24      29444  62824  0,69    8544 31  chrome
353      30      67028  131204  2,39    1276 31  chrome
277      27      93036  112724  86,31   3088 31  chrome
308      31      82340  105924  79,70   9004 31  chrome
429      21      11036  28412  0,30    12548 31  chrome
226      20      14548  34712  0,06    13000 31  chrome
386      30      103988 114988  14,20   11148 31  chrome
320      33      53572  108316  7,75    11348 31  chrome
274      16      7640   27140  4,02    25248 31  conhost
170      12      2080   13932  7,22    21164 31  conhost
790      115     27892  7216   145,45  26852 31  csrss
752      26      2688   3016   32,75   1000 0  csrss
581      22      8276   36260  9,97    24848 31  ctfmon
1069     50      117488 128128  45,67   19436 31  Discord
281      16      11924  82968  1,80    20144 31  Discord
368      23      15920  55896  4,34    26476 31  Discord
772      44      261864 173220  651,89  10780 31  Discord
199      12      10812  33388  0,00    14440 31  Discord
1550     138     926148 680356  4 365,31 15544 31  Discord
319      29      7396   21940  0,31    6260 31  dllhost
1793     86      259936 97904  2 777,86 10116 31  dwm
527      32      23084  24208  45,38   8472 0  EABackgroundService

```

Рисунок 8 – Сортировка по времени использования ЦП

```

Администратор: Windows PowerShell
PS C:\Windows\system32> Get-Process | Sort-object CPU | Select-Object -First 5

Handles  NPM(K)  PM(K)  WS(K)  CPU(s)  Id  SI ProcessName
-----  -
0        0        60      8       0,00    0   0 Idle
199      12     10812   33388   0,00   14440 31 Discord
192      10     1876    12584   0,00   24448 0 SearchFilterHost
188      10     1916    11984   0,02   19196 0 SearchFilterHost
182      12     2960    19464   0,02   14676 31 svchost

```

Рисунок 9 – Вывод первых 5 отсортированных процессов ЦП

```

Администратор: Windows PowerShell
PS C:\Windows\system32> $a = Get-Process | Sort-object CPU | Select-Object -First 5
PS C:\Windows\system32> $a

Handles  NPM(K)  PM(K)  WS(K)  CPU(s)  Id  SI ProcessName
-----  -
0        0        60      8       0,00    0   0 Idle
192      10     1872    12600   0,00   24448 0 SearchFilterHost
199      12     10812   33388   0,00   14440 31 Discord
186      10     1916    12004   0,02   19196 0 SearchFilterHost
170      9      1788    9808    0,02    3688 0 svchost

```

Рисунок 10 – Ввод и вывод переменной

```

PS C:\Windows\system32> Get-Process | Measure-Object

Count      : 198
Average    :
Sum        :
Maximum    :
Minimum    :
Property   :

```

Рисунок 11 – Подсчет количества процессов

```

Администратор: Windows PowerShell
PS C:\Windows\system32> Get-Process | Measure-Object | Get-Member

TypeName: Microsoft.PowerShell.Commands.GenericMeasureInfo

Name      MemberType Definition
-----
Equals    Method     bool Equals(System.Object obj)
GetHashCode Method    int GetHashCode()
GetType   Method     type GetType()
ToString  Method     string ToString()
Average    Property   System.Nullable[double] Average {get;set;}
Count     Property   int Count {get;set;}
Maximum    Property   System.Nullable[double] Maximum {get;set;}
Minimum    Property   System.Nullable[double] Minimum {get;set;}
Property   Property   string Property {get;set;}
Sum        Property   System.Nullable[double] Sum {get;set;}

```

Рисунок 12 – Вывод Children объекта Measure-Object


```
Администратор: Windows PowerShell
PS C:\Windows\system32> Get-Process | Measure-Object -property VM -average -minimum -maximum

Count      : 197
Average    : 1990939022601,1
Sum        :
Maximum    : 3764494774272
Minimum    : 8192
Property   : VM

PS C:\Windows\system32>
```

Рисунок 13 – Выполнение объекта с ключами

```
Администратор: Windows PowerShell
PS C:\Windows\system32> D:
PS D:\> Get-Process -id 0 | Get-Member -MemberType Property | Select-Object name | Tee-Object -File object.txt | Measure-Object

Count      : 52
Average    :
Sum        :
Maximum    :
Minimum    :
Property   :
```

object.txt

Файл | Изменить | Просмотр

```
Name
----
BasePriority
Container
EnableRaisingEvents
ExitCode
ExitTime
Handle
HandleCount
HasExited
Id
MachineName
MainModule
MainWindowHandle
MainWindowTitle
MaxWorkingSet
MinWorkingSet
Modules
NonpagedSystemMemorySize
NonpagedSystemMemorySize64
PagedMemorySize
PagedMemorySize64
PagedSystemMemorySize
PagedSystemMemorySize64
PeakPagedMemorySize
PeakPagedMemorySize64
PeakVirtualMemorySize
PeakVirtualMemorySize64
PeakWorkingSet
PeakWorkingSet64
PriorityBoostEnabled
PriorityClass
PrivateMemorySize
PrivateMemorySize64
PrivilegedProcessorTime
ProcessName
```

Строка 1, столбец 1 1461 символ 100% Windows (CRLF) UTF-16 LE

Рисунок 14 – Подсчет объектов и вывод результата в файл

```
Администратор: Windows PowerShell
PS D:\> dir

Каталог: D:\

Mode                LastWriteTime         Length Name
----                -
d-----         14.03.2025         19:14         AntiMicroX
d-----         15.03.2025         13:02         bloody
d-----         31.03.2025         21:00         BurenkovLAB2
da----         20.04.2025         17:01         Cosmos
d-----         26.02.2025         19:59         CrystalDiskMark8
d-----         30.04.2025         21:08         cursor
d-----         08.05.2025         16:30         Downloads_Chrome
d-----         13.04.2025         23:57         Games
d-----         24.04.2025         12:16         GitTest
d-----         26.02.2025         19:49         osulazer
d-----         08.05.2025         12:32         SteamLibrary
d-----         08.05.2025         12:32         клипы
-a----         01.05.2025         14:37        16462 command_help.txt
-a----         08.05.2025         17:33        3038 object.txt

PS D:\> Get-Content object.txt

Name
----
BasePriority
Container
EnableRaisingEvents
ExitCode
ExitTime
Handle
HandleCount
HasExited
Id
MachineName
MainModule
MainWindowHandle
MainWindowTitle
MaxWorkingSet
```

Рисунок 15 – Вывод файла

```
Администратор: Windows PowerShell
PS D:\> Get-Help Get-Content -Detailed

ИМЯ
    Get-Content

ОПИСАНИЕ
    Gets the content of the item at the specified location.

СИНТАКСИС
    Get-Content [-Credential <System.Management.Automation.PSCredential>] [-Delimiter <System.String>] [-Encoding {ASCII | BigEndianUnicode | BigEndianUTF32 | Byte | Default | OEM | String | Unicode | Unknown | UTF7 | UTF8 | UTF32}] [-Exclude <System.String[]>] [-Filter <System.String[]>] [-Force] [-Include <System.String[]>] [-LiteralPath <System.String[]>] [-Raw] [-ReadCount <System.Int64>] [-Stream <System.String>] [-Tail <System.Int32>] [-TotalCount <System.Int64>] [-UseTransaction] [-Wait] [<CommonParameters>]

    Get-Content [-Path <System.String[]>] [-Credential <System.Management.Automation.PSCredential>] [-Delimiter <System.String>] [-Encoding {ASCII | BigEndianUnicode | BigEndianUTF32 | Byte | Default | OEM | String | Unicode | Unknown | UTF7 | UTF8 | UTF32}] [-Exclude <System.String[]>] [-Filter <System.String[]>] [-Force] [-Include <System.String[]>] [-Raw] [-ReadCount <System.Int64>] [-Stream <System.String>] [-Tail <System.Int32>] [-TotalCount <System.Int64>] [-UseTransaction] [-Wait] [<CommonParameters>]

ОПИСАНИЕ
    The 'Get-Content' cmdlet gets the content of the item at the location specified by the path, such as the text in a file or the content of a function. For files, the content is read one line at a time and returns a collection of objects, each representing a line of content.

    Beginning in PowerShell 3.0, 'Get-Content' can also get a specified number of lines from the beginning or end of an item.

ПАРАМЕТРЫ
    -Credential <System.Management.Automation.PSCredential>
        > [!NOTE] > This parameter isn't supported by any providers installed with PowerShell. To impersonate another > user, or elevate your credentials when running this cmdlet, use > Invoke-Command (./Microsoft.PowerShell.Core/Invoke-Command.md).

    -Delimiter <System.String>
        Specifies the delimiter that 'Get-Content' uses to divide the file into objects while it
```

Рисунок 16 – Вывод справки команды Get-Content

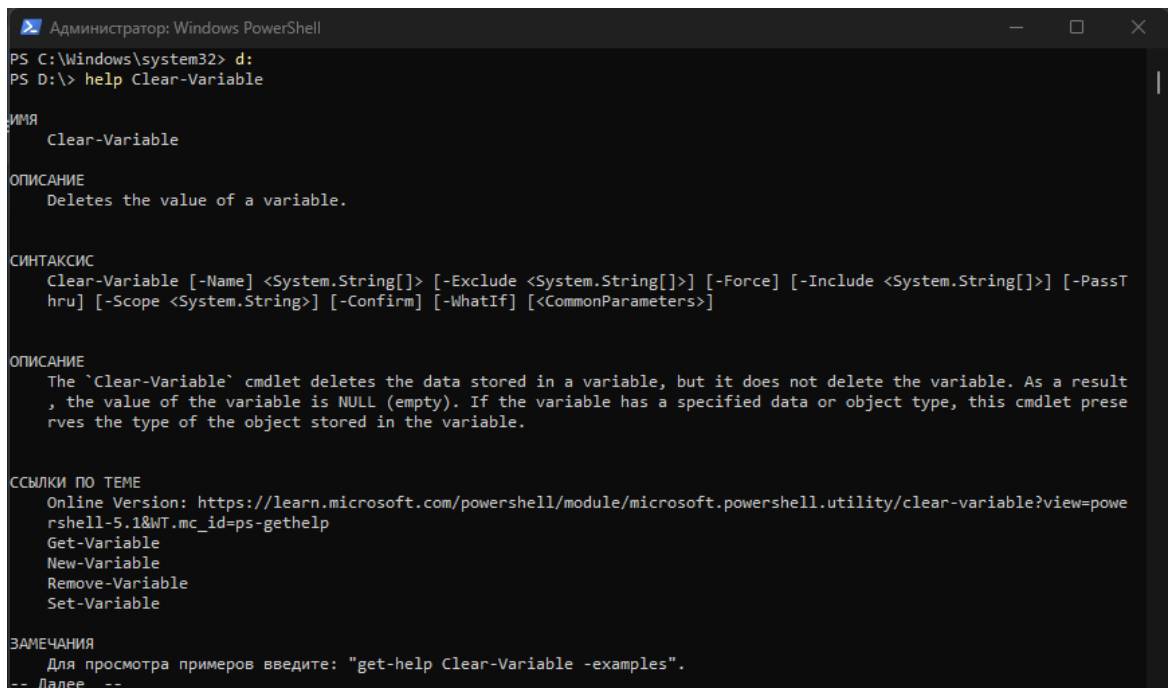
```
Администратор: Windows PowerShell
PS D:\> "Ouyuyuuu, SUAI!" | Out-File outfile.txt
PS D:\> type .\outfile.txt
Ouyuyuuu, SUAI!
PS D:\>
```

Рисунок 17 – Вывод справки команды Get-Content

3 Справочная информация о командах из индивидуального задания

В соответствии с вариантом 1 мне необходимо рассмотреть команды Clear-Variable, Start-Process, Group-Object. Первая команда из индивидуального задания — Clear-Variable. Это командлет PowerShell, предназначенный для удаления значений, присвоенных переменным в текущем сеансе. Он не удаляет саму переменную, а лишь очищает ее содержимое, то есть делает переменную «пустой», без значения. Это удобно, когда нужно сбросить значение переменной, не удаляя ее из области видимости скрипта или консоли. Команда может применяться как к одной переменной, так и к множеству переменных одновременно. Clear-Variable полезен при написании скриптов, в которых требуется обнуление данных перед повторным использованием переменных, что помогает избежать ошибок, связанных с сохранёнными ранее значениями.

Одним из ключей является -Name, который указывает имя переменной, значение которой нужно очистить. Например, Clear-Variable -Name myVar очистит содержимое переменной myVar. Вторым важным ключом — -Scope, определяющий область действия команды (например, текущая функция, скрипт или глобальная область). Это позволяет точно контролировать, где именно нужно произвести очистку значения. Третий ключ — -Force, он используется для очистки защищённых переменных, которые в обычных условиях могут быть недоступны для изменений. Использование этого ключа требует повышенных прав и должно применяться с осторожностью, чтобы не нарушить выполнение важных процессов скрипта. На рисунках 18, 19 изображена справка по командлету Clear-Variable.

A screenshot of a Windows PowerShell window titled "Администратор: Windows PowerShell". The command prompt shows the user at the C:\Windows\system32 directory, typing "d:" to switch to the drive, and then "help Clear-Variable". The help text is displayed in Russian. It includes the command name "Clear-Variable", a description "Deletes the value of a variable.", the syntax "Clear-Variable [-Name] <System.String[]> [-Exclude <System.String[]> [-Force] [-Include <System.String[]> [-PassThru] [-Scope <System.String>] [-Confirm] [-WhatIf] [<CommonParameters>]", a detailed explanation of the command's behavior, a list of related commands (Get-Variable, New-Variable, Remove-Variable, Set-Variable), and a note about using "get-help Clear-Variable -examples" for examples.

```
Администратор: Windows PowerShell
PS C:\Windows\system32> d:
PS D:\> help Clear-Variable

ИМЯ
    Clear-Variable

ОПИСАНИЕ
    Deletes the value of a variable.

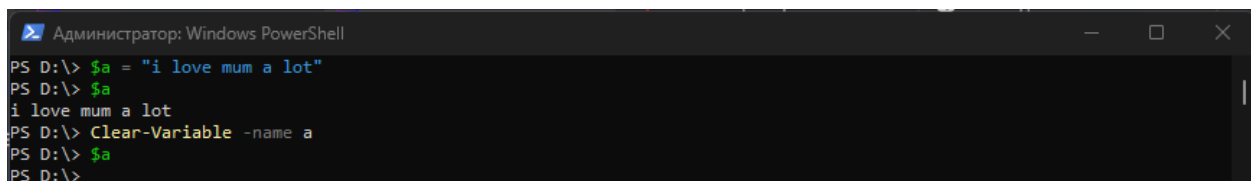
СИНТАКСИС
    Clear-Variable [-Name] <System.String[]> [-Exclude <System.String[]> [-Force] [-Include <System.String[]> [-PassThru] [-Scope <System.String>] [-Confirm] [-WhatIf] [<CommonParameters>]

ОПИСАНИЕ
    The 'Clear-Variable' cmdlet deletes the data stored in a variable, but it does not delete the variable. As a result, the value of the variable is NULL (empty). If the variable has a specified data or object type, this cmdlet preserves the type of the object stored in the variable.

ССЫЛКИ ПО ТЕМЕ
    Online Version: https://learn.microsoft.com/powershell/module/microsoft.powershell.utility/clear-variable?view=powershell-5.1&WT.mc_id=ps-gethelp
    Get-Variable
    New-Variable
    Remove-Variable
    Set-Variable

ЗАМЕЧАНИЯ
    Для просмотра примеров введите: "get-help Clear-Variable -examples".
-- Далее --
```

Рисунок 18 – справка по Clear-Variable

A screenshot of a Windows PowerShell window titled "Администратор: Windows PowerShell". The user sets a variable \$a to the string "i love mum a lot" and then uses the Clear-Variable command with the -name parameter to clear the variable. The prompt shows the variable's value before and after the command, confirming it has been cleared.

```
Администратор: Windows PowerShell
PS D:\> $a = "i love mum a lot"
PS D:\> $a
i love mum a lot
PS D:\> Clear-Variable -name a
PS D:\> $a
```

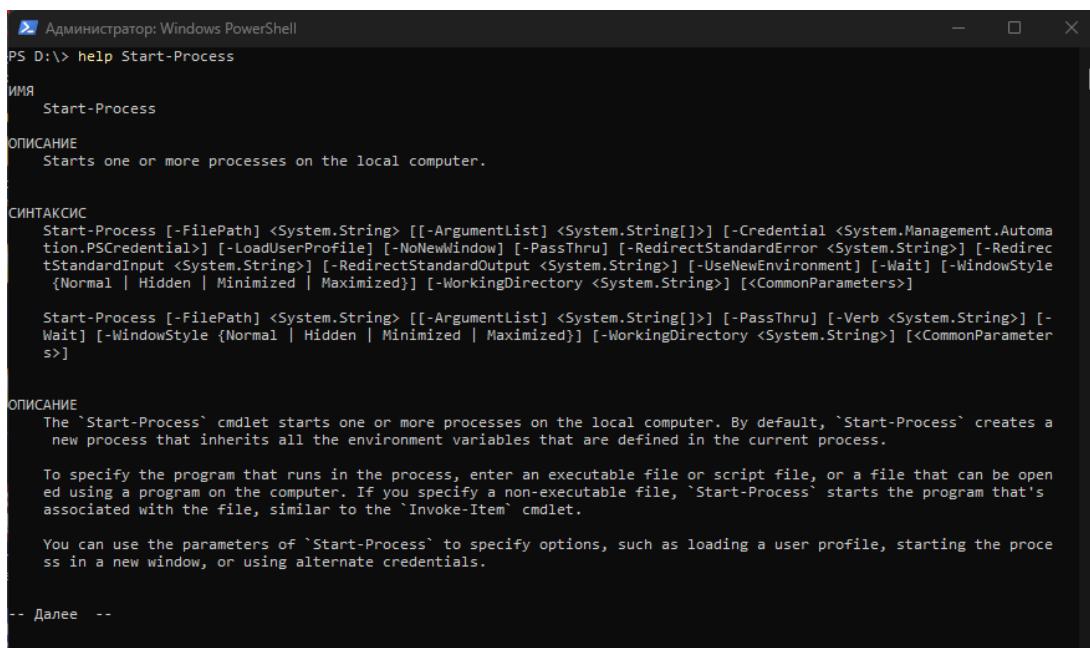
Рисунок 19 – пример использования Clear-Variable

Вторая команда из индивидуального задания — Start-Process. Этот командлет используется для запуска новых процессов в операционной системе Windows. Он предоставляет гибкие возможности по запуску внешних программ, сценариев, приложений или команд с различными параметрами. Командлет Start-Process особенно полезен в автоматизированных скриптах и административных задачах, когда необходимо инициировать выполнение программ с определёнными условиями, например, в фоновом режиме, с повышенными правами или в указанной рабочей директории.

Командлет может быть использован для запуска как простых программ, вроде блокнота (Start-Process notepad), так и сложных процессов с аргументами и настройками. Это делает его мощным инструментом для администраторов и разработчиков. Кроме того, он позволяет задать, следует

ли дожидаться завершения процесса, что особенно важно при последовательной автоматизации действий.

Ключ `-FilePath` является обязательным и указывает путь к исполняемому файлу или приложению, которое нужно запустить. Например, `Start-Process -FilePath "notepad.exe"` иницирует запуск блокнота. Второй важный ключ — `-ArgumentList`, который позволяет передавать аргументы запускаемой программе. Это полезно при работе с программами, поддерживающими параметры командной строки, например: `Start-Process -FilePath "cmd.exe" -ArgumentList "/c echo Hello"`. Третий ключ — `-Wait` — указывает PowerShell ждать завершения процесса перед выполнением следующих команд. Это особенно важно при написании скриптов, где действия должны выполняться последовательно. На рисунке 20-21 изображена справка по командлету `Start-Process`.



```
Администратор: Windows PowerShell
PS D:\> help Start-Process

ИМЯ
    Start-Process

ОПИСАНИЕ
    Starts one or more processes on the local computer.

СИНТАКСИС
    Start-Process [-FilePath] <System.String> [[-ArgumentList] <System.String[]>] [-Credential <System.Management.Automation.PSCredential>] [-LoadUserProfile] [-NoNewWindow] [-PassThru] [-RedirectStandardError <System.String>] [-RedirectStandardInput <System.String>] [-RedirectStandardOutput <System.String>] [-UseNewEnvironment] [-Wait] [-WindowStyle {Normal | Hidden | Minimized | Maximized}] [-WorkingDirectory <System.String>] [<CommonParameters>]

    Start-Process [-FilePath] <System.String> [[-ArgumentList] <System.String[]>] [-PassThru] [-Verb <System.String>] [-Wait] [-WindowStyle {Normal | Hidden | Minimized | Maximized}] [-WorkingDirectory <System.String>] [<CommonParameters>]

ОПИСАНИЕ
    The 'Start-Process' cmdlet starts one or more processes on the local computer. By default, 'Start-Process' creates a new process that inherits all the environment variables that are defined in the current process.

    To specify the program that runs in the process, enter an executable file or script file, or a file that can be opened using a program on the computer. If you specify a non-executable file, 'Start-Process' starts the program that's associated with the file, similar to the 'Invoke-Item' cmdlet.

    You can use the parameters of 'Start-Process' to specify options, such as loading a user profile, starting the process in a new window, or using alternate credentials.

-- Далее --
```

Рисунок 20 — справка по `Start-Process`

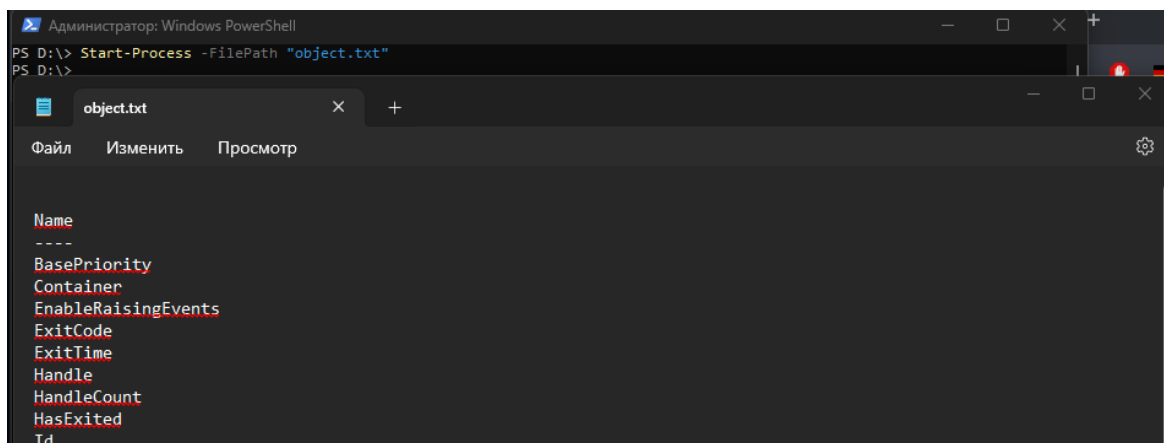


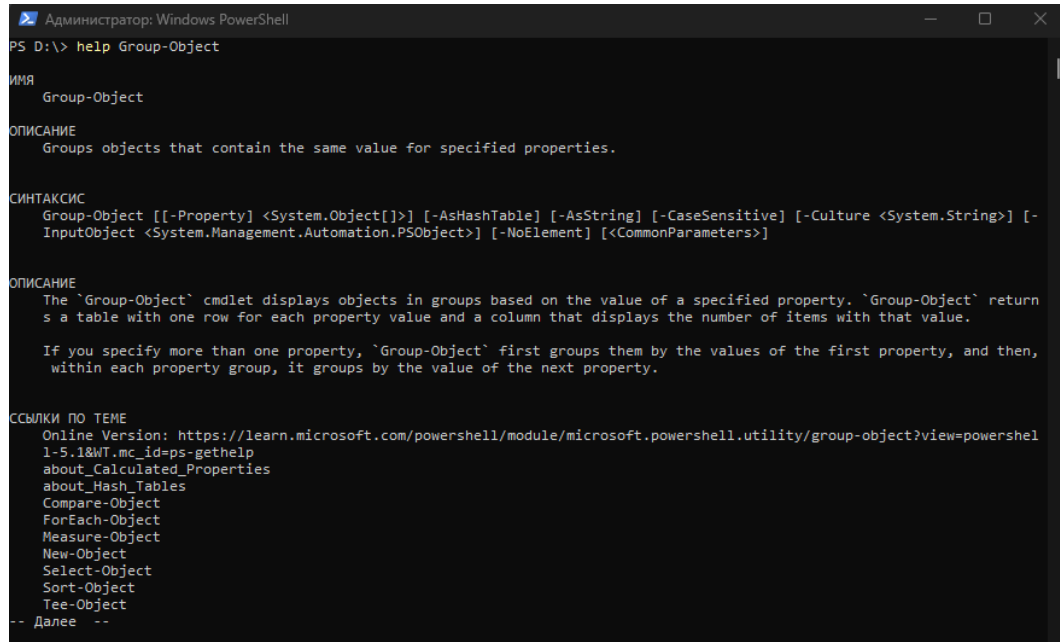
Рисунок 21 – вывод файла со свойством FilePath

Третья команда из индивидуального задания — Group-Object. Этот командлет PowerShell используется для группировки объектов по значению одного или нескольких их свойств. Он часто применяется при анализе больших массивов данных, полученных из различных источников, таких как списки процессов, файлы, службы и другие системные объекты. Командлет создает на выходе сгруппированный результат, где для каждого уникального значения свойства указывается количество соответствующих объектов и сами объекты, попавшие в эту группу. Это позволяет быстро выявлять повторяющиеся элементы и проводить статистический анализ.

Например, если получить список процессов и сгруппировать их по имени, можно легко узнать, сколько экземпляров каждой программы запущено. Или при анализе данных из файла можно сгруппировать записи по какому-либо полю, чтобы увидеть структуру распределения значений.

Один из ключевых параметров — -Property, который определяет имя свойства, по которому производится группировка. Это основной ключ, без которого командлет не будет работать осмысленно. Пример использования: `Get-Process | Group-Object -Property ProcessName`. Второй ключ — -AsString, который указывает, что свойство следует рассматривать как строку. Это полезно, если значения по умолчанию интерпретируются как объекты, а не как текст. Третий ключ — -NoElement, при использовании которого в результатах

группировки не отображаются сами объекты, только имя группы и количество элементов. Это удобно, если интересует только статистика, без лишней детализации. На рисунке 22-23 изображена справка по командлету Group-Object.



```
Администратор: Windows PowerShell
PS D:\> help Group-Object

ИМЯ
    Group-Object

ОПИСАНИЕ
    Groups objects that contain the same value for specified properties.

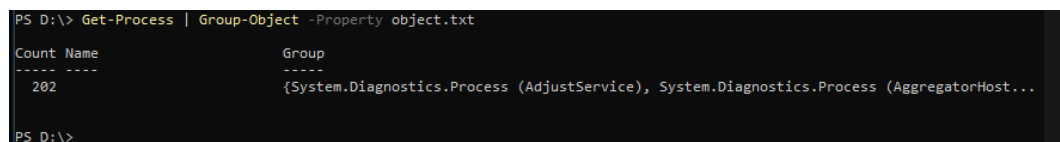
СИНТАКСИС
    Group-Object [[-Property] <System.Object[]>] [-AsHashTable] [-AsString] [-CaseSensitive] [-Culture <System.String>] [-
    InputObject <System.Management.Automation.PSObject>] [-NoElement] [<CommonParameters>]

ОПИСАНИЕ
    The 'Group-Object' cmdlet displays objects in groups based on the value of a specified property. 'Group-Object' return
    s a table with one row for each property value and a column that displays the number of items with that value.

    If you specify more than one property, 'Group-Object' first groups them by the values of the first property, and then,
    within each property group, it groups by the value of the next property.

ССЫЛКИ ПО ТЕМЕ
    Online Version: https://learn.microsoft.com/powershell/module/microsoft.powershell.utility/group-object?view=powershel
    1-5.1&WT.mc_id=ps-gethelp
    about_Calculated_Properties
    about_Hash_Tables
    Compare-Object
    ForEach-Object
    Measure-Object
    New-Object
    Select-Object
    Sort-Object
    Tee-Object
    -- Далее --
```

Рисунок 22 – справка Group-Object



```
PS D:\> Get-Process | Group-Object -Property Name

Count Name Group
-----
    202 {System.Diagnostics.Process (AdjustService), System.Diagnostics.Process (AggregatorHost...
```

Рисунок 23 – вывод с помощью Group-Object

4 Выводы о проделанной работе

В ходе выполнения лабораторной работы была достигнута основная цель — изучение и практическое применение командлетов PowerShell. На этапе выполнения упражнений были исследованы базовые команды оболочки PowerShell, такие как `Get-Service`, `Sort-Object`, `Get-Process`, `Measure-Object`, `Select-Object`, `Get-Content` и другие. Они продемонстрировали принципы работы с объектами, конвейерной обработки данных, сортировки, фильтрации, подсчета объектов и вывода информации в файлы. Были получены практические навыки управления службами, анализа запущенных процессов, работы с переменными и текстовыми файлами. Эти упражнения позволили освоить синтаксис командлетов, научиться комбинировать их с помощью конвейеров и использовать параметры для получения нужных результатов.

В рамках индивидуального задания были подробно изучены три команды: `Clear-Variable`, `Start-Process` и `Group-Object`. Каждая из них играет важную роль в автоматизации административных задач. `Clear-Variable` позволяет очищать значения переменных без их удаления, что необходимо при повторном использовании переменных в скриптах. `Start-Process` обеспечивает запуск внешних процессов с передачей аргументов и управлением их выполнением, что полезно при написании универсальных скриптов. `Group-Object` служит для группировки объектов по значению их свойств, позволяя эффективно анализировать и обрабатывать данные. Для каждого командлета были рассмотрены ключевые параметры и приведены пояснения к их применению.

Полученные знания позволяют уверенно ориентироваться в среде PowerShell, создавать скрипты для автоматизации типовых задач и эффективно управлять объектами операционной системы. Это создаёт прочную основу для дальнейшего изучения инструментальных средств информационных систем и их применения в профессиональной деятельности.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. ГУАП, документация для учебного процесса. – URL: <https://guap.ru/regdocs/docs/uch> (дата обращения 22.04.2025)
2. Полезные команды в cmd на Windows – URL: <https://htmlacademy.ru/blog/soft/windows-command-line> (дата обращения 26.04.2025)
3. CMD Полная документация для Windows – URL: <https://learn.microsoft.com/ru-ru/windows-server/administration/windows-commands/cmd> (дата обращения 26.04.2025)