

ГУАП

КАФЕДРА № 42

ОТЧЕТ  
ЗАЩИЩЕН С ОЦЕНКОЙ \_\_\_\_\_  
ПРЕПОДАВАТЕЛЬ

ассистент

\_\_\_\_\_  
должность, уч. степень, звание

\_\_\_\_\_  
подпись, дата

В.В. Жукалин

\_\_\_\_\_  
инициалы, фамилия

## ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ №5

Разработка командлетов PowerShell

по курсу: ИНСТРУМЕНТАЛЬНЫЕ СРЕДСТВА ИНФОРМАЦИОННЫХ  
СИСТЕМ

РАБОТУ ВЫПОЛНИЛ

СТУДЕНТ ГР. № \_\_\_\_\_ 4329

\_\_\_\_\_  
подпись, дата

Д.С. Шаповалова  
\_\_\_\_\_  
инициалы, фамилия

Санкт-Петербург 2025

## Содержание

1. Цель работы:.....	3
2. Задание:.....	3
3. Выполненные упражнения со скриншотами: .....	4
Упражнение 2.5:.....	4
Упражнение 2.6:.....	5
Упражнение 2.7:.....	6
Упражнение 2.8:.....	8
Упражнение 2.9:.....	9
Упражнение 2.10:.....	11
Упражнение 2.11:.....	13
4. Результаты выполнения задания по варианту 17: .....	14
Командлет Restart-Service. ....	14
Командлет Start-Service .....	16
Командлет Format-List.....	18
5. Вывод: .....	22
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ .....	23

## 1. Цель работы:

Изучение командлетов PowerShell.

## 2. Задание:

1. Изучить методические указания подраздела 2.3.
2. Выполнить все упражнения.
3. Результат выполнения зафиксировать на скриншотах.
4. Выполнить задания для самостоятельного изучения.
5. Сделать выводы о проделанной работе.

Задание для самостоятельного изучения:

Из ниже приведенного списка выбрать три командлета, изучить возможности их применения и привести примеры использования. При желании можно выбрать любые другие командлеты, примеры работы которых не рассмотрены в методических указаниях.

Вариант 17:

1. 17 - Restart-Service – перезапускает службу;
2. 22 - Start-Service – запускает службу;
3. 4 - Format-List – вывод результата команды в формате списка свойств, где на каждой новой строке отдельное свойство.

3. Выполненные упражнения со скриншотами:

Упражнение 2.5:

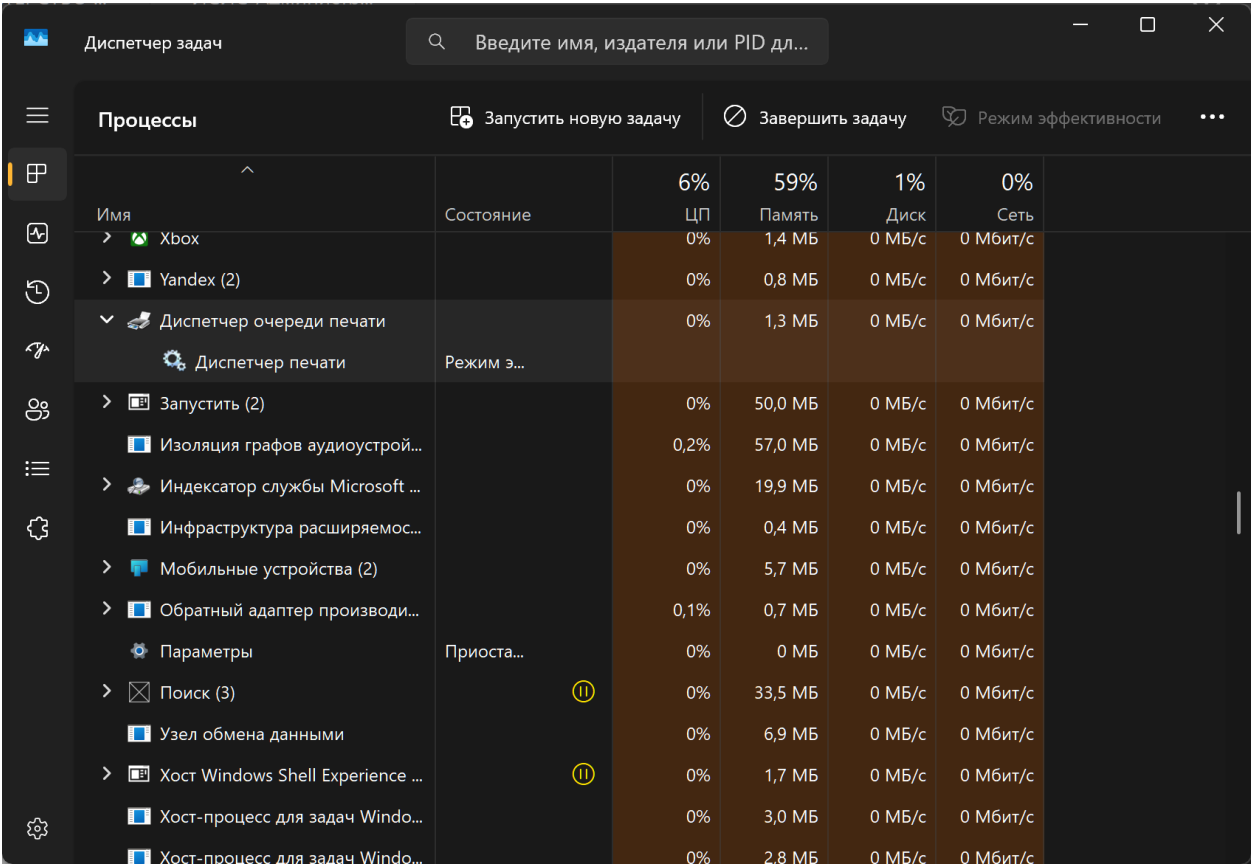


Рисунок 1.1.1 – Упражнение 2.5, часть 1

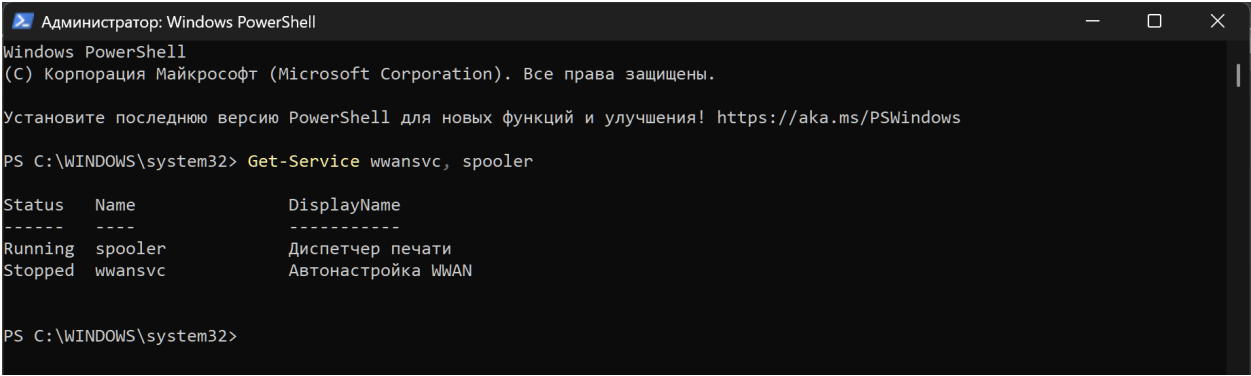
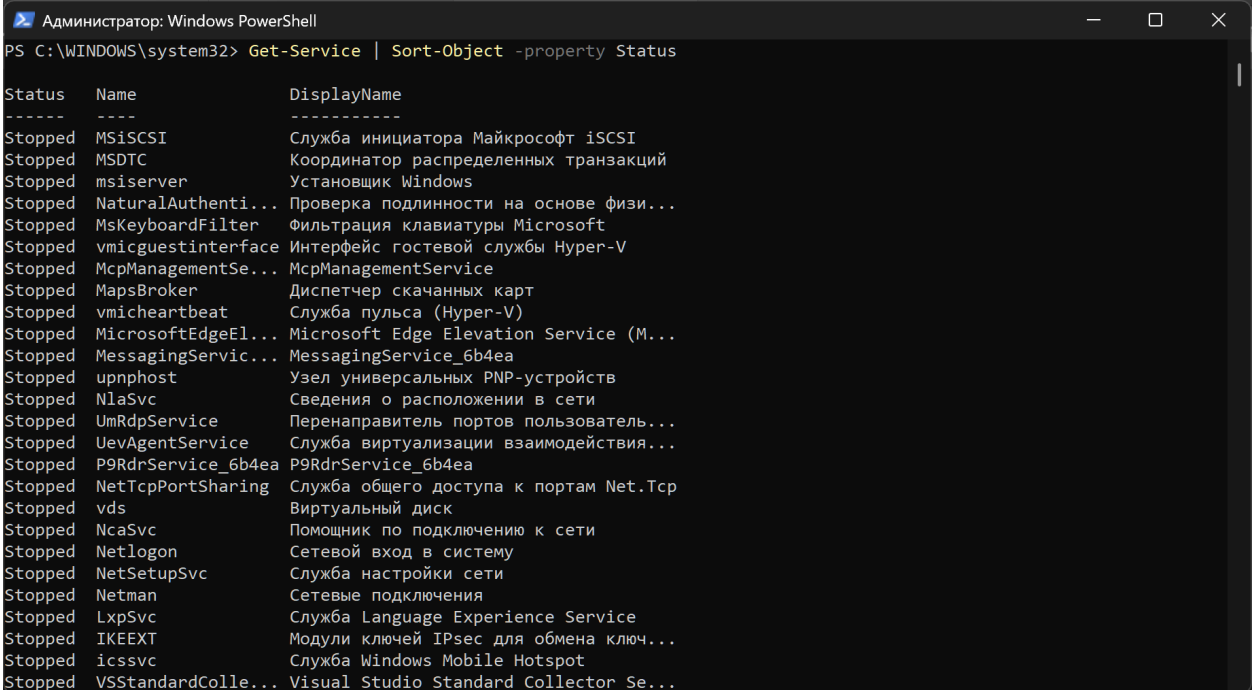


Рисунок 1.1.2 – Упражнение 2.5, часть 2

## Упражнение 2.6:



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

Status Name                DisplayName
-----
Stopped MSiSCSI              Служба инициатора Майкрософт iSCSI
Stopped MSDTC          Координатор распределенных транзакций
Stopped msiserver       Установщик Windows
Stopped NaturalAuthenti... Проверка подлинности на основе физи...
Stopped MsKeyboardFilter Фильтрация клавиатуры Microsoft
Stopped vmicguestinterface Интерфейс гостевой службы Hyper-V
Stopped McpManagementSe... MspManagementService
Stopped MapsBroker      Диспетчер скачанных карт
Stopped vmicheartbeat     Служба пульса (Hyper-V)
Stopped MicrosoftEdgeEl... Microsoft Edge Elevation Service (M...
Stopped MessagingServic... MessagingService_6b4ea
Stopped upnphost         Узел универсальных PNP-устройств
Stopped NlaSvc           Сведения о расположении в сети
Stopped UmRdpService     Перенаправитель портов пользователь...
Stopped UevAgentService  Служба виртуализации взаимодействия...
Stopped P9RdrService_6b4ea P9RdrService_6b4ea
Stopped NetTcpPortSharing Служба общего доступа к портам Net.Tcp
Stopped vds              Виртуальный диск
Stopped NcaSvc           Помощник по подключению к сети
Stopped Netlogon         Сетевой вход в систему
Stopped NetSetupSvc      Служба настройки сети
Stopped Netman           Сетевые подключения
Stopped LxpSvc           Служба Language Experience Service
Stopped IKEEXT           Модули ключей IPsec для обмена ключ...
Stopped icssvc           Служба Windows Mobile Hotspot
Stopped VSStandardColle... Visual Studio Standard Collector Se...
```

Рисунок 1.2.1 – Упражнение 2.6

## Упражнение 2.7:

Администратор: Windows PowerShell

PS C:\WINDOWS\system32> Get-Process

Handles	NPM(K)	PM(K)	WS(K)	CPU(s)	Id	SI	ProcessName
179	12	2128	2124	0,36	15548	1	AbletonPushCpl
1690	77	777552	524756	325,56	13240	1	Acrobat
181	13	2844	11052	0,19	20708	1	Acrobat
244	14	12372	21588	0,11	22448	1	Acrobat
210	14	2692	3008	0,13	9192	1	acrotray
373	15	6088	4572	21,34	16024	1	Adobe Crash Processor
174	13	2408	2032	2,48	15912	1	AdobeIPCBroker
241	14	4908	7492	23,47	7896	0	AggregatorHost
323	19	24492	41856	0,45	23876	1	ai
152	10	2508	2264	2,77	2964	0	amdfendrsr
520	21	5764	9400	227,05	14992	1	AMDRSServ
293	30	45516	5928	74,36	2668	1	AMDRSSrcExt
435	25	14484	26404	4,72	11904	1	ApplicationFrameHost
384	20	6116	11524	9,17	4644	0	AsusAppService
100	9	1432	1296	11,23	8908	1	AsusInputLocaleMonitor
159	13	2032	3504	2,00	4652	0	AsusNumPadService
283	13	2980	6000	2,83	3660	0	AsusOptimization
362	18	5920	9992	1,63	6968	1	AsusOptimizationStartupTask
178	15	4244	3120	0,38	16448	1	AsusOSD
553	18	5668	12196	6,97	4660	0	AsusSoftwareManager
3152	39	48040	20600	9,94	5248	1	AsusSoftwareManagerAgent
223	12	3240	5260	0,22	4692	0	AsusSwitch
597	20	6156	11688	4,63	4668	0	AsusSystemAnalysis
147	9	2172	11212	0,02	17436	1	AsusSystemAnalysis
2152	13	16008	4888	71,84	4768	0	AsusSystemDiagnosis
224	15	4072	7516	999,39	6744	0	AsusWiFiSmartConnect

Рисунок 1.3.1 – Упражнение 2.7, часть 1

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.EventArgs)
ErrorDataReceived	Event	System.Diagnostics.DataReceivedEventHandler ErrorDataReceived(System.Object, System.EventArgs)
Exited	Event	System.EventHandler Exited(System.Object, System.EventArgs)
OutputDataReceived	Event	System.Diagnostics.DataReceivedEventHandler OutputDataReceived(System.Object, System.EventArgs)
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 requestedType)
Dispose	Method	void Dispose(), void IDisposable.Dispose()
Equals	Method	bool Equals(System.Object obj)
GetHashCode	Method	int GetHashCode()
GetLifetimeService	Method	System.Object GetLifetimeService()

Рисунок 1.3.2 – Упражнение 2.7, часть 2

```

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.EventArgs)
ErrorDataReceived Event System.Diagnostics.DataReceivedEventHandler ErrorDataReceived(System.Object, System.EventArgs)
Exited    Event System.EventHandler Exited(System.Object, System.EventArgs)
OutputDataReceived Event System.Diagnostics.DataReceivedEventHandler OutputDataReceived(System.Object, System.EventArgs)
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 requestedType)
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()
<ПРОБЕЛ> следующая страница; <CR> следующая строка; Q выход
InitializeLifetimeService Method System.Object InitializeLifetimeService()
<ПРОБЕЛ> следующая страница; <CR> следующая строка; Q выход

```

Рисунок 1.3.3 – Упражнение 2.7, часть 3

## Упражнение 2.8:

Администратор: Windows PowerShell

PS C:\WINDOWS\system32> Get-Process | Sort-Object CPU

Handles	NPM(K)	PM(K)	WS(K)	CPU(s)	Id	SI	ProcessName
0	0	60	8		0	0	Idle
131	8	1340	644	0,02	4832	0	svchost
128	9	1704	764	0,02	2276	1	cncmd
48	4	572	256	0,02	15608	1	CCXProcess
147	9	2172	10432	0,02	17436	1	AsusSystemAnalysis
97	8	1236	976	0,02	15668	1	conhost
42	8	1984	268	0,03	1668	0	fontdrvhost
170	9	1736	9192	0,05	29540	0	svchost
118	8	1324	2524	0,05	7172	0	svchost
165	11	2328	12936	0,06	29260	1	MoNotificationUx
133	9	1508	3280	0,06	2400	0	wlanext
111	9	1260	828	0,08	6828	0	conhost
244	14	12372	14800	0,11	22448	1	Acrobat
165	10	1968	10044	0,11	27108	0	svchost
169	10	2048	3444	0,11	4816	0	service_update
201	18	16256	34440	0,13	10108	1	msedge
210	14	2692	3008	0,13	9192	1	acrotray
123	9	1688	2632	0,14	18348	1	dllhost
172	10	3948	15264	0,14	6792	0	svchost
58	4	1152	288	0,16	752	0	smss
199	13	11732	3084	0,16	14280	1	Notion_Calendar
205	11	1900	1628	0,17	6024	1	Wacom_TabletUser
206	13	1880	9888	0,17	24868	0	svchost
297	12	2132	3508	0,17	3144	0	WUDFHost
114	8	2012	7792	0,17	27564	0	svchost
181	13	2844	10596	0,19	20708	1	Acrobat
141	10	2032	2228	0,20	5700	0	service_update
257	15	17908	27020	0,20	28744	1	Spotify_Desktop
223	12	3240	4624	0,22	4692	0	AsusSwitch
105	7	1328	2040	0,23	15864	0	svchost
140	10	2036	4804	0,25	5224	1	UserOOBEBroker
117	9	1592	2260	0,27	17500	0	svchost

Рисунок 1.4.1 – Упражнение 2.8, часть 1

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	0	Idle
48	4	572	256	0,02	15608	1	CCXProcess
97	8	1236	976	0,02	15668	1	conhost
128	9	1704	764	0,02	2276	1	cncmd
131	8	1340	644	0,02	4832	0	svchost

Рисунок 1.4.2 – Упражнение 2.8, часть 2

Администратор: Windows PowerShell

PS C:\WINDOWS\system32> \$a = Get-Process | Sort-Object CPU | Select-Object -First 5

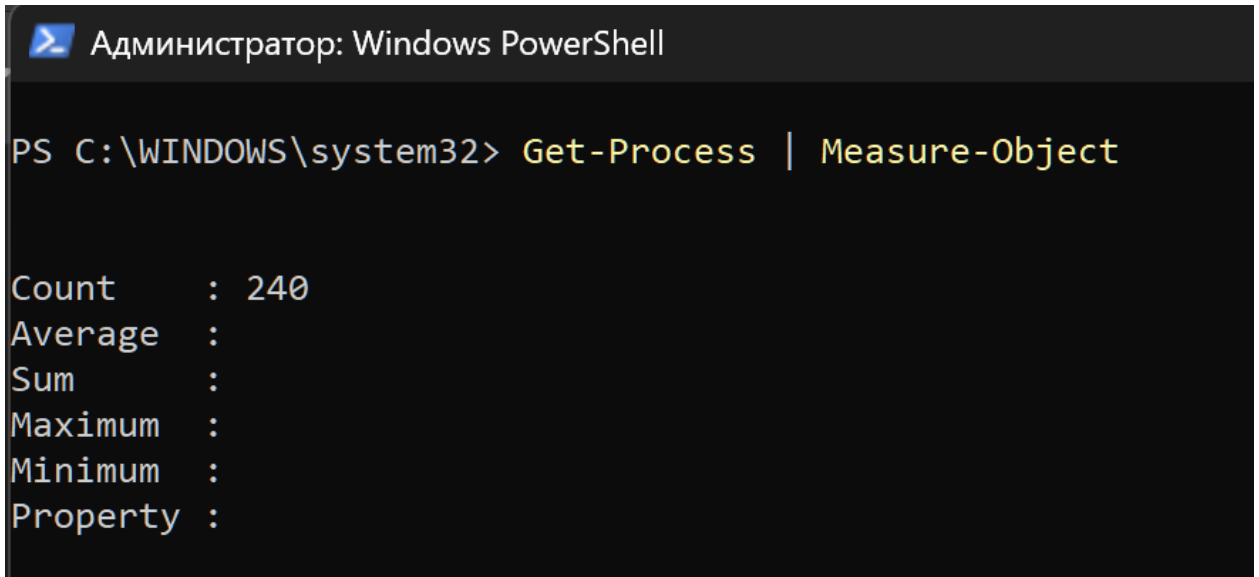
PS C:\WINDOWS\system32> \$a

Handles	NPM(K)	PM(K)	WS(K)	CPU(s)	Id	SI	ProcessName
0	0	60	8		0	0	Idle
48	4	572	256	0,02	15608	1	CCXProcess
128	9	1704	764	0,02	2276	1	cncmd
147	9	2172	10420	0,02	17436	1	AsusSystemAnalysis
131	8	1340	644	0,02	4832	0	svchost

Рисунок 1.4.3 – Упражнение 2.8, часть 3



### Упражнение 2.9:

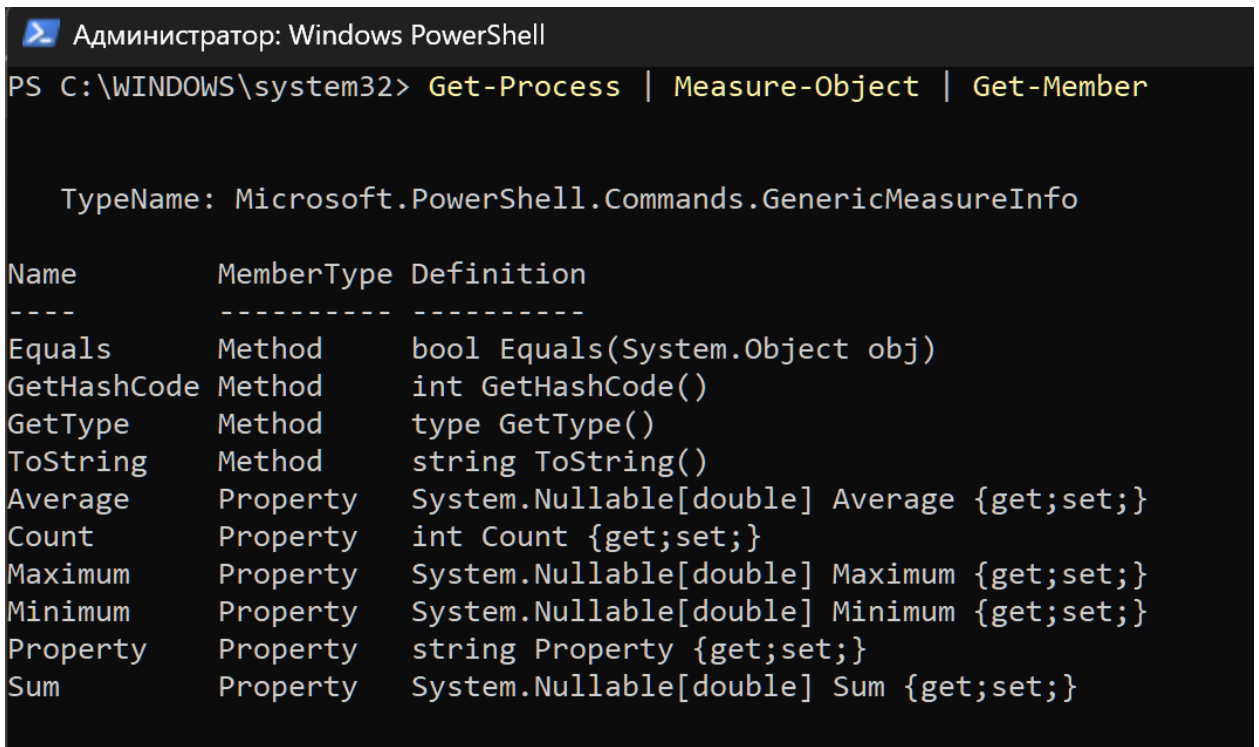


```
Администратор: Windows PowerShell

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

Count      : 240
Average    :
Sum        :
Maximum    :
Minimum    :
Property   :
```

Рисунок 1.5.1 – Упражнение 2.9, часть 1



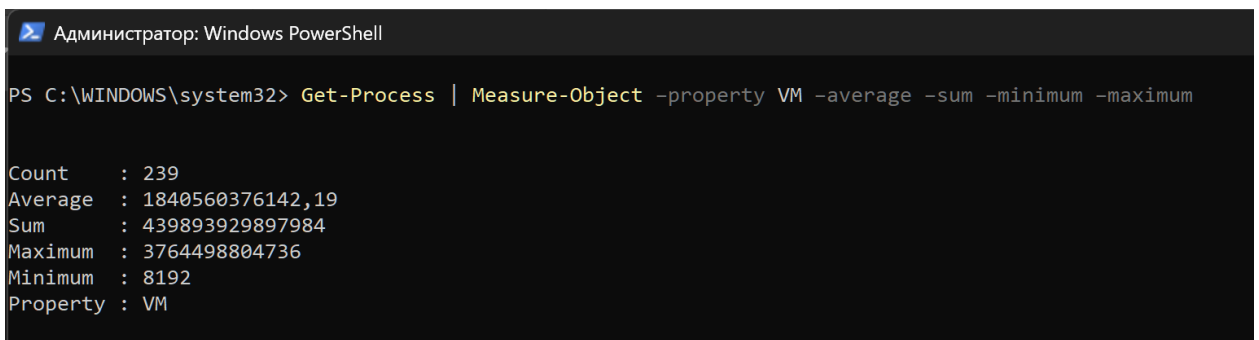
```
Администратор: Windows PowerShell

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 1.5.2 – Упражнение 2.9, часть 2



```
Администратор: Windows PowerShell

PS C:\WINDOWS\system32> Get-Process | Measure-Object -property VM -average -sum -minimum -maximum

Count      : 239
Average    : 1840560376142,19
Sum        : 439893929897984
Maximum    : 3764498804736
Minimum    : 8192
Property   : VM
```

Рисунок 1.5.3 – Упражнение 2.9, часть 3

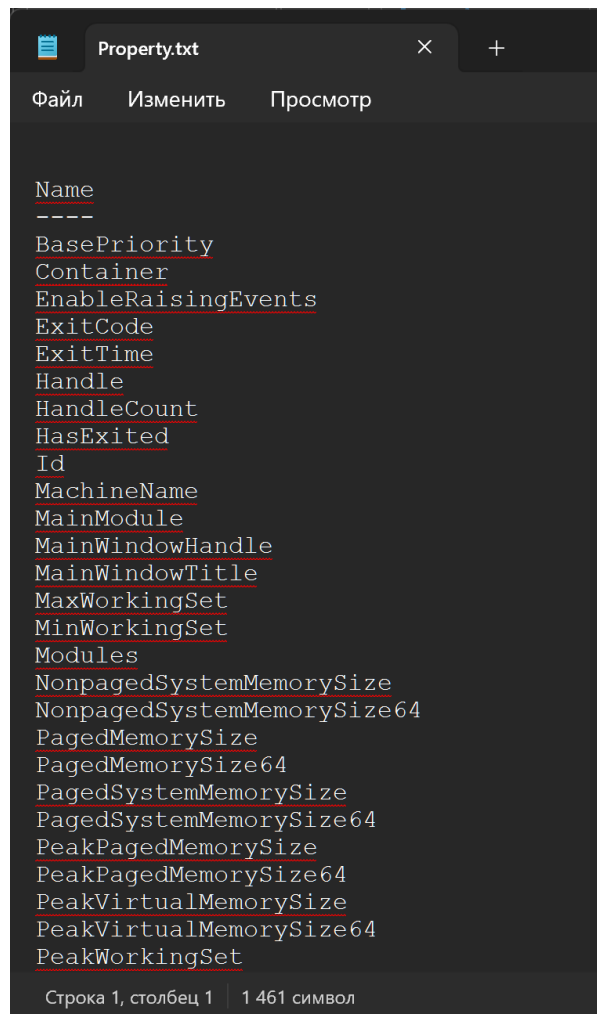


```
Администратор: Windows PowerShell

PS C:\WINDOWS\system32> Get-Process -id 0 | Get-Member -MemberType Property | Select-Object name
| Tee-Object -FilePath "C:\etoki's_dir\Property.txt" | Measure-Object

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

Рисунок 1.5.4 – Упражнение 2.9, часть 4



Property.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

Строка 1, столбец 1    1 461 символ

Рисунок 1.5.5 – Упражнение 2.9, часть 5

## Упражнение 2.10:

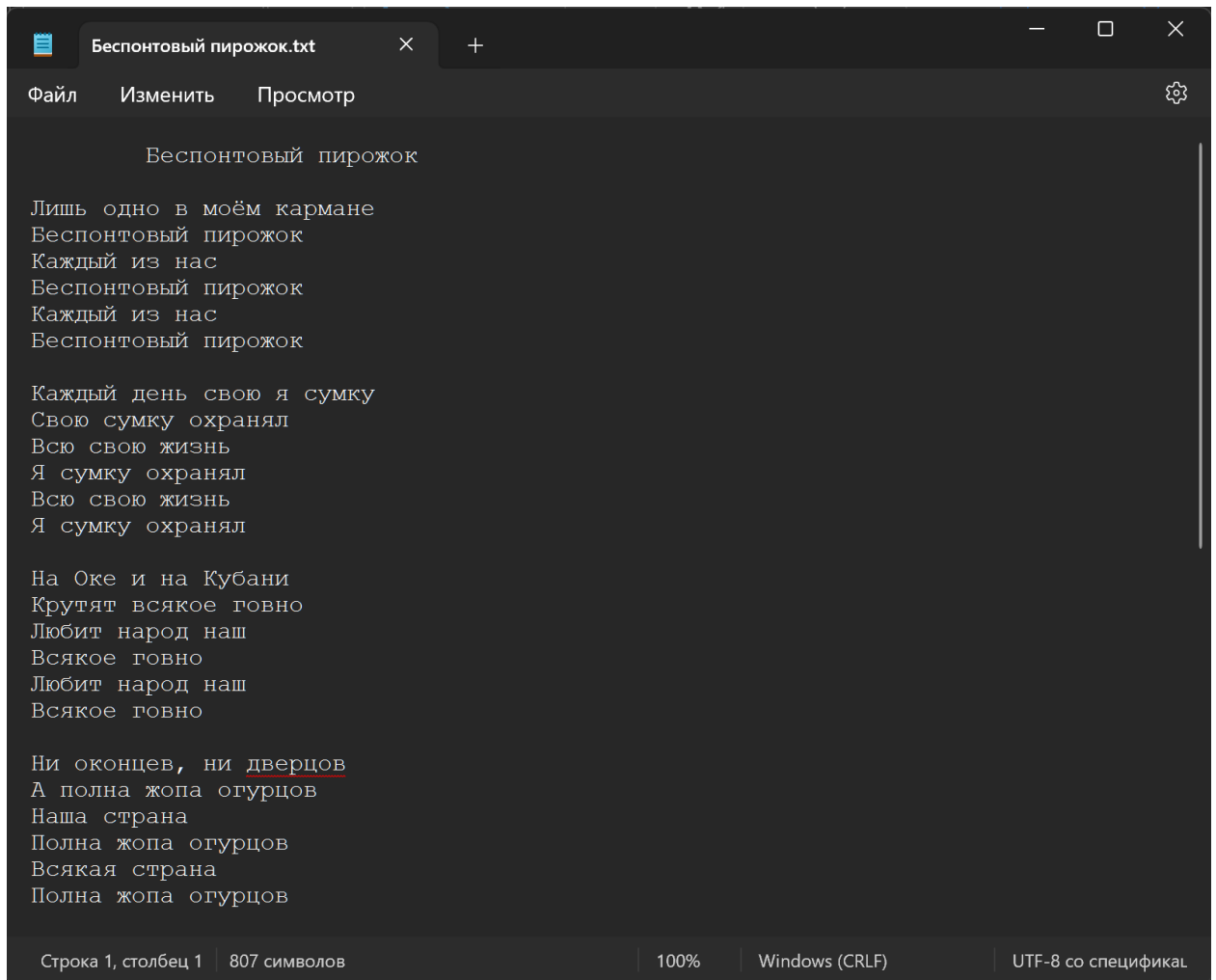


Рисунок 1.6.1 – Упражнение 2.10, часть 1

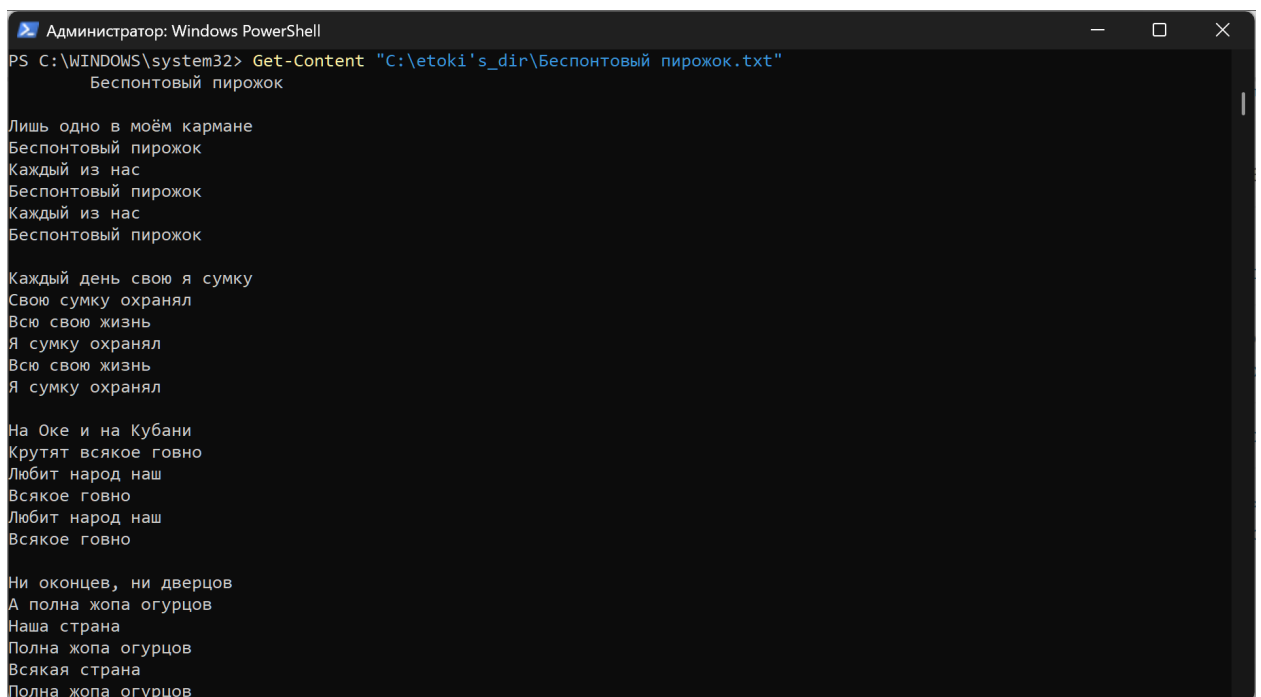


Рисунок 1.6.2 – Упражнение 2.10, часть 2

```
Администратор: Windows PowerShell
PS C:\WINDOWS\system32> 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>]

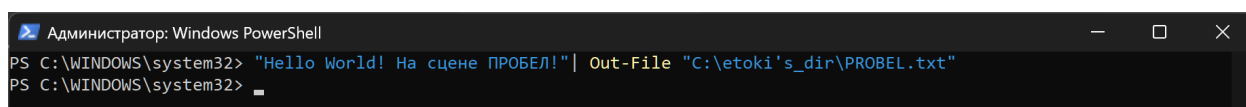
    Get-Content [-Path <System.String[]>] [-Credential <System.Management.Automation.PSCredential>] [-Delimiter <System.String>] [-Encoding {ASCII | BigEndianUnicode | BigEndianUTF32 | Byte | Default | OEM | String | Unicode | Unknown | UTF7 | UTF8 | UTF32}] [-Exclude <System.String[]>] [-Filter <System.String>] [-Force] [-Include <System.String[]>] [-LiteralPath <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.
```

Рисунок 1.6.3 – Упражнение 2.10, часть 3

## Упражнение 2.11:



```
Администратор: Windows PowerShell
PS C:\WINDOWS\system32> "Hello World! На сцене ПРОБЕЛ!" | Out-File "C:\etoki's_dir\PROBEL.txt"
PS C:\WINDOWS\system32>
```

Рисунок 1.7.1 – Упражнение 2.11, часть 1

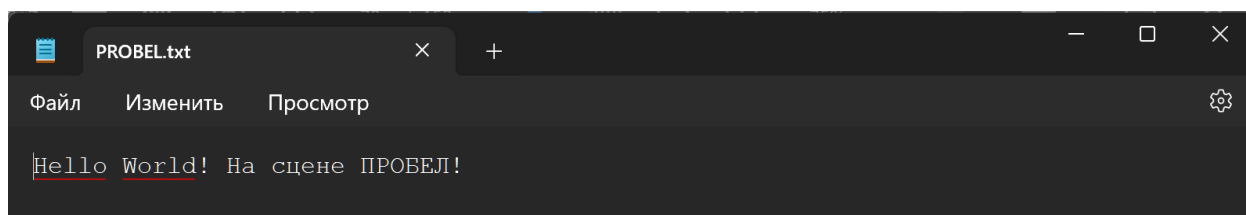


Рисунок 1.7.2 – Упражнение 2.11, часть 2

## 4. Результаты выполнения задания по варианту 17:

### Командлет Restart-Service.

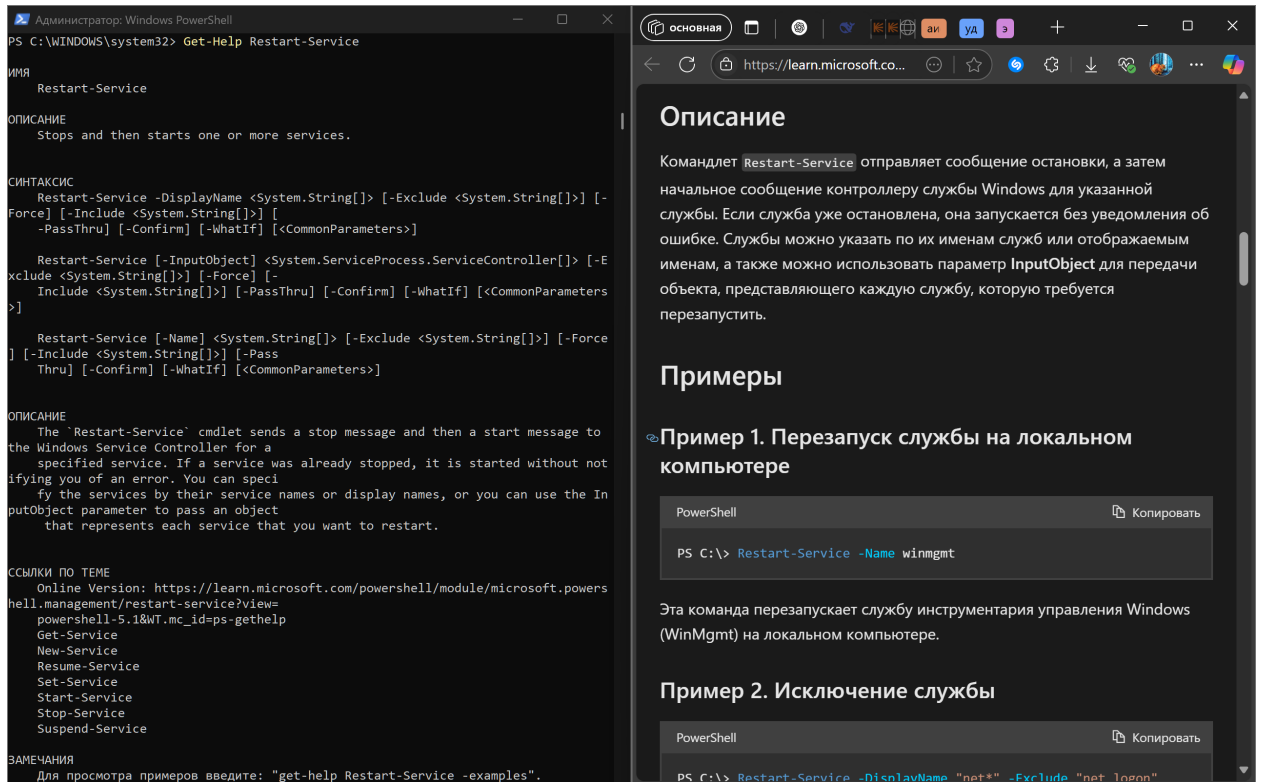


Рисунок 2.1 – Справка для Restart-Service

Из справки мы можем понять:

**Назначение:** перезапускает службу Windows.

**Синтаксис:** Restart-Service [-Name] <string[]> [-Force] [-PassThru] [-Confirm] [-WhatIf] [<CommonParameters>]

Основные параметры:

- Name – Имя службы (например, "wuauserv").
- DisplayName – Перезапуск по отображаемому имени (например, "Центр обновления Windows").
- Force – Принудительно останавливает зависимые службы.
- PassThru – Выводит объект службы после перезапуска.
- Confirm – Запрашивает подтверждение перед действием.
- WhatIf – Показывает, что произойдет без реального выполнения.

**Примеры:**

```
Администратор: Windows PowerShell
PS C:\WINDOWS\system32> Restart-Service -Name "wuauserv"
ПРЕДУПРЕЖДЕНИЕ: Ожидание остановки службы "Центр обновления Windows (wuauserv)"...
ПРЕДУПРЕЖДЕНИЕ: Ожидание остановки службы "Центр обновления Windows (wuauserv)"...
PS C:\WINDOWS\system32>
```

Рисунок 2.2 – Перезапуск по имени

```
Администратор: Windows PowerShell
PS C:\WINDOWS\system32> Restart-Service -Name "Spooler" -Confirm

Подтверждение
Вы действительно хотите выполнить это действие?
Выполнение операции "Restart-Service" над целевым объектом "Диспетчер печати (Spooler)".
[Y] Да - Y [A] Да для всех - A [N] Нет - N [L] Нет для всех - L
[S] Приостановить - S[?] Справка (значением по умолчанию является "Y"): Y
PS C:\WINDOWS\system32>
```

Рисунок 2.3 – Перезапуск с подтверждением

```
Администратор: Windows PowerShell
PS C:\WINDOWS\system32> Restart-Service -Name "WinRM" -Force
PS C:\WINDOWS\system32> Restart-Service -Name "Spooler" -Force
PS C:\WINDOWS\system32>
```

Рисунок 2.4 – Принудительный перезапуск

```
Администратор: Windows PowerShell
PS C:\WINDOWS\system32> Restart-Service -DisplayName "Диспетчер печати" -PassThru

Status      Name             DisplayName
-----
Running     Spooler          Диспетчер печати

PS C:\WINDOWS\system32>
```

Рисунок 2.5 – Перезапуск по отображаемому имени:

**Альтернативы:**

Stop-Service + Start-Service – отдельная остановка и запуск.

Get-Service | Restart-Service – перезапуск нескольких служб.

## Командлет Start-Service

The image shows two side-by-side windows. The left window is a Windows PowerShell terminal window titled 'Администратор: Windows PowerShell'. It displays the help text for the 'Start-Service' command. The right window is a web browser showing the Microsoft Learn page for 'Start-Service'.

**Администратор: Windows PowerShell**

```
PS C:\WINDOWS\system32> Get-Help Start-Service
```

**ИМЯ**

Start-Service

**ОПИСАНИЕ**

Starts one or more stopped services.

**СИНТАКСИС**

```
Start-Service -DisplayName <System.String[]> [-Exclude <System.String[]>] [-Include <System.String[]>] [-PassThru] [-Confirm] [-WhatIf] [<CommonParameters>]

Start-Service [-InputObject] <System.ServiceProcess.ServiceController[]> [-Exclude <System.String[]>] [-Include <System.String[]>] [-PassThru] [-Confirm] [-WhatIf] [<CommonParameters>]

Start-Service [-Name] <System.String[]> [-Exclude <System.String[]>] [-Include <System.String[]>] [-PassThru] [-Confirm] [-WhatIf] [<CommonParameters>]
```

**ОПИСАНИЕ**

The 'Start-Service' cmdlet sends a start message to the Windows Service Controller for each of the specified services. If a service is already running, the message is ignored without error. You can specify the services by their service names or display names, or you can use the InputObject parameter to supply a service object that represents the services that you want to start.

**ССЫЛКИ ПО ТЕМЕ**

Online Version: [https://learn.microsoft.com/powershell/module/microsoft.powershell.management/start-service?view=powershell-5.1&WT.mc\\_id=ps-gethelp](https://learn.microsoft.com/powershell/module/microsoft.powershell.management/start-service?view=powershell-5.1&WT.mc_id=ps-gethelp)

Get-Service  
New-Service  
Restart-Service  
Resume-Service  
Set-Service  
Stop-Service  
Suspend-Service

**ЗАМЕЧАНИЯ**

Для просмотра примеров введите: "get-help Start-Service -examples".  
Для получения дополнительных сведений введите: "get-help Start-Service -detailed".  
Для получения технических сведений введите: "get-help Start-Service -full".  
Для получения справки в Интернете введите: "get-help Start-Service -online"

**Описание**

Командлет `Start-Service` отправляет начальное сообщение контроллеру службы Windows для каждой из указанных служб. Если служба уже запущена, сообщение игнорируется без ошибок. Вы можете указать службы по именам служб или отображаемым именам или использовать параметр `InputObject` для предоставления объекта службы, представляющего службы, которые необходимо запустить.

**Примеры**

**Пример 1. Запуск службы с помощью его имени**

В этом примере запускается служба EventLog на локальном компьютере. Параметр имени определяет службу по имени службы.

```
PowerShell
Start-Service -Name "eventlog"
```

**Пример 2. Отображение сведений без запуска службы**

В этом примере показано, что произойдет при запуске служб, имеющих отображаемое имя, включающее "remote".

```
PowerShell
```

Рисунок 3.1 – Справка по Start-Service

Из справки можно понять:

**Назначение:** запускает остановленную службу Windows.

**Синтаксис:** `Start-Service [-Name] <string[]> [-DisplayName <string>] [-PassThru] [-Confirm] [-WhatIf] [<CommonParameters>]`

**Основные параметры:**

- Name – Запуск службы по её имени (например, "Spooler").
- DisplayName – Запуск по отображаемому имени (например, "Диспетчер печати").
- PassThru – Выводит объект службы после запуска (удобно для проверки состояния).
- Confirm – Запрашивает подтверждение перед запуском.
- WhatIf – Показывает, что произойдет без реального выполнения.

**Примеры использования:**

1. Запуск службы по имени
2. Запуск с выводом результата (-PassThru)
3. Запуск по отображаемому имени
4. Запуск с подтверждением (-Confirm)



```
Администратор: Windows PowerShell
PS C:\WINDOWS\system32> Start-Service -Name "Spooler"
PS C:\WINDOWS\system32> Start-Service -Name "WinRM" -PassThru

Status      Name                DisplayName
-----
Running     WinRM               Служба удаленного управления Window...

PS C:\WINDOWS\system32> Start-Service -DisplayName "Центр обновления Windows"
PS C:\WINDOWS\system32> Start-Service -Name "Dhcp" -Confirm

Подтверждение
Вы действительно хотите выполнить это действие?
Выполнение операции "Start-Service" над целевым объектом "DHCP-клиент (Dhcp)".
[Y] Да - Y  [A] Да для всех - A  [N] Нет - N  [L] Нет для всех - L
[S] Приостановить - S[?] Справка (значением по умолчанию является "Y"): Y
PS C:\WINDOWS\system32>
```

Рисунок 3.2 – Примеры использования

## Командлет Format-List

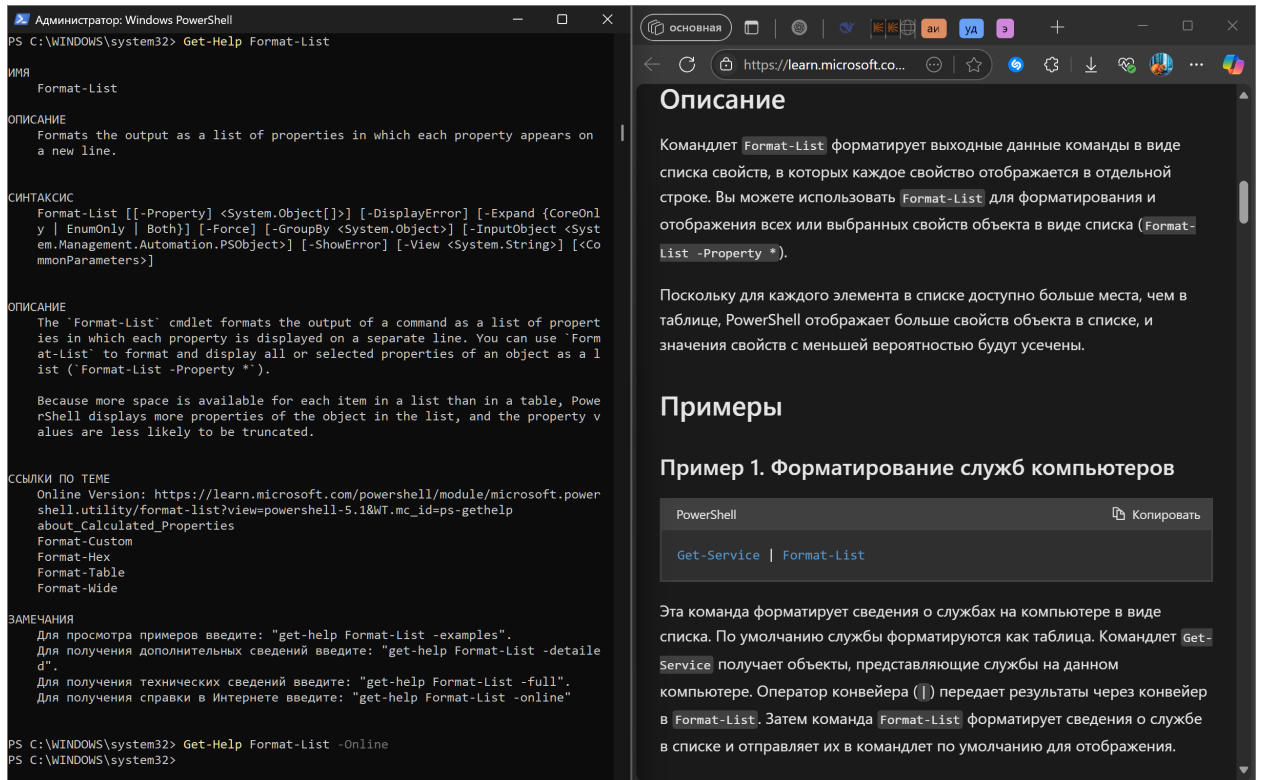


Рисунок 4.1 – Справка по Format-List

Из справки можно понять:

**Назначение:** форматирует вывод команд PowerShell в виде списка свойств (каждое свойство на новой строке).

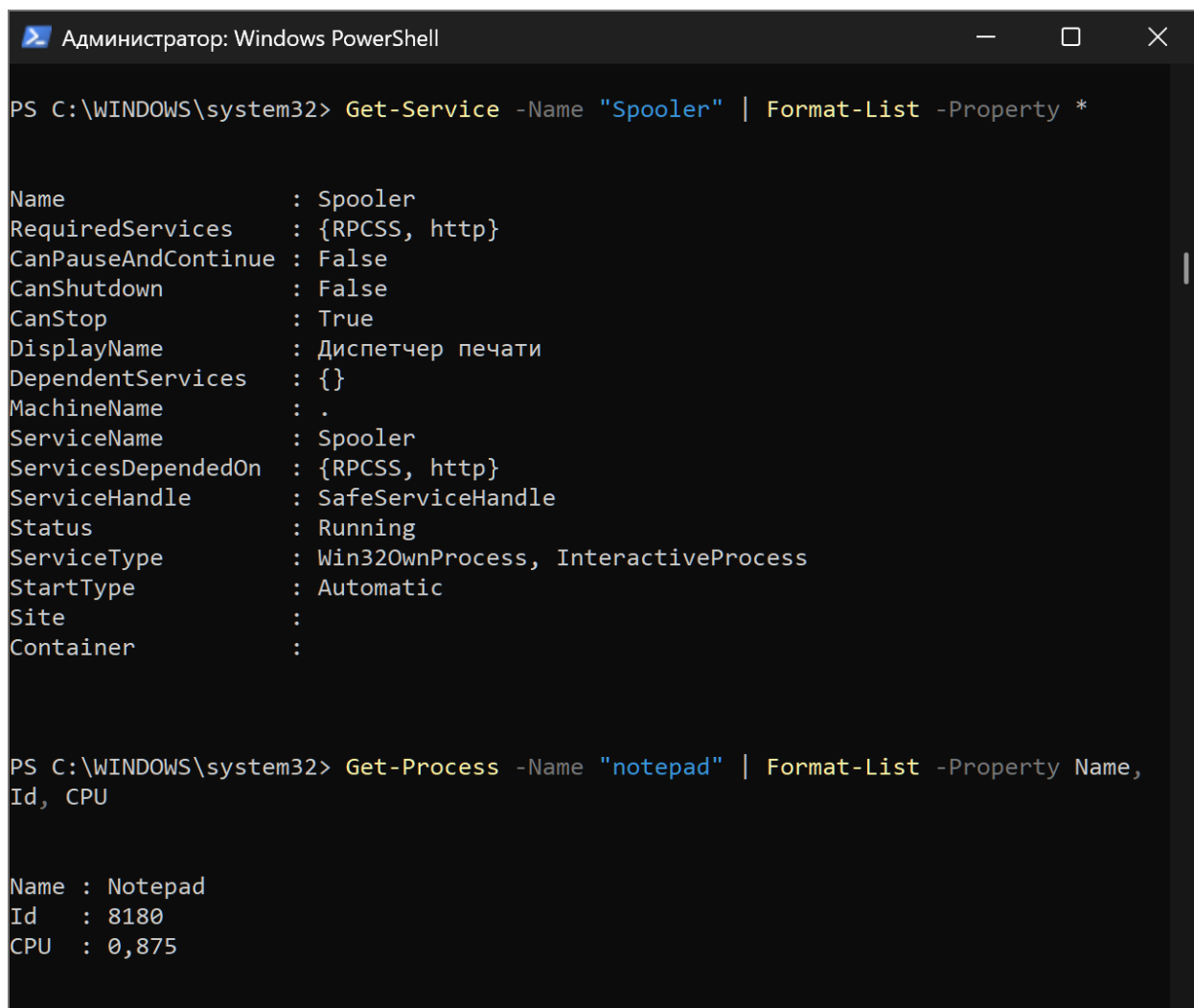
**Синтаксис:** `Format-List [[-Property] <Object[]> [-GroupBy <Object>] [-View <string>] [-ShowError] [-DisplayError] [-Force] [-Expand <string>] [-InputObject <psobject>] [<CommonParameters>]`

### Основные параметры:

- Property – Выводит только указанные свойства (например, Name, Status).
- GroupBy – Группирует объекты по указанному свойству (например, Status).
- View – Использует предопределённое представление (если есть).
- Force – Показывает все свойства, включая системные (например, PSSStandardMembers).
- Expand – Раскрывает свойства коллекций (All, CoreOnly, Both).

### Примеры использования:

1. Вывод всех свойств объекта
2. Вывод конкретных свойств
3. Группировка по свойству
4. Раскрытие вложенных свойств



```
Администратор: Windows PowerShell

PS C:\WINDOWS\system32> Get-Service -Name "Spooler" | Format-List -Property *

Name                : Spooler
RequiredServices    : {RPCSS, http}
CanPauseAndContinue : False
CanShutdown         : False
CanStop             : True
DisplayName         : Диспетчер печати
DependentServices   : {}
MachineName         : .
ServiceName         : Spooler
ServicesDependedOn  : {RPCSS, http}
ServiceHandle       : SafeServiceHandle
Status              : Running
ServiceType         : Win32OwnProcess, InteractiveProcess
StartType           : Automatic
Site                :
Container           :

PS C:\WINDOWS\system32> Get-Process -Name "notepad" | Format-List -Property Name,
Id, CPU

Name : Notepad
Id   : 8180
CPU  : 0,875
```

Рисунок 4.2 – Примеры использования, часть 1

```
Администратор: Windows PowerShell
PS C:\WINDOWS\system32> Get-Service | Format-List -Property Name, Status -GroupBy Status

Status: Stopped

Name : AarSvc_6b4ea
Status : Stopped

Name : AdobeARMservice
Status : Stopped

Name : AGSService
Status : Stopped

Name : ALG
Status : Stopped

Status: Running

Name : AMD Crash Defender Service
Status : Running

Name : AMD External Events Utility
Status : Running

Status: Stopped

Name : AppIDSvc
Status : Stopped

Status: Running

Name : Appinfo
Status : Running
```

Рисунок 4.3 – Примеры использования, часть 2 (группировка по свойству)

```
Администратор: Windows PowerShell
PS C:\WINDOWS\system32> Get-Process | Format-List -Property Name, Modules -Expand Both

Name      : AbletonPushCpl
Modules   : {System.Diagnostics.ProcessModule (AbletonPushCpl.exe), System.Dia
            gnostics.ProcessModule (ntdll.dll), System.Diagnostics.ProcessModule (KERNEL
            32.DLL), System.Diagnostics.ProcessModule (KERNELBASE.dll)...}

Name      : Acrobat
Modules   : {System.Diagnostics.ProcessModule (Acrobat.exe), System.Dia
            gnostics.ProcessModule (ntdll.dll), System.Diagnostics.ProcessModule (KERNEL32.DLL)
            , System.Diagnostics.ProcessModule (KERNELBASE.dll)...}

Name      : Acrobat
Modules   : {System.Diagnostics.ProcessModule (Acrobat.exe), System.Dia
            gnostics.ProcessModule (ntdll.dll), System.Diagnostics.ProcessModule (wow64.dll), S
            ystem.Diagnostics.ProcessModule (wow64base.dll)...}

Name      : Acrobat
Modules   : {System.Diagnostics.ProcessModule (Acrobat.exe), System.Dia
            gnostics.ProcessModule (ntdll.dll), System.Diagnostics.ProcessModule (wow64.dll), S
            ystem.Diagnostics.ProcessModule (wow64base.dll)...}

Name      : acrotray
Modules   : {System.Diagnostics.ProcessModule (acrotray.exe), System.Dia
            gnostics.ProcessModule (ntdll.dll), System.Diagnostics.ProcessModule (KERNEL32.DLL)
            }, System.Diagnostics.ProcessModule (KERNELBASE.dll)...}

Name      : Adobe Crash Processor
Modules   : {System.Diagnostics.ProcessModule (Adobe Crash Processor.exe), System.D
            iagnostics.ProcessModule (ntdll.dll), System.Diagnostics.ProcessModule
            (KERNEL32.DLL), System.Diagnostics.ProcessModule (KERNELBASE.dll)...}

Name      : AdobeIPCBroker
Modules   : {System.Diagnostics.ProcessModule (AdobeIPCBroker.exe), System.Dia
            gnostics.ProcessModule (ntdll.dll), System.Diagnostics.ProcessModule (wow64.
            dll), System.Diagnostics.ProcessModule (wow64base.dll)...}

Name      : AggregatorHost
Modules   : {System.Diagnostics.ProcessModule (AggregatorHost.exe), System.Dia
            gnostics.ProcessModule (ntdll.dll), System.Diagnostics.ProcessModule (KERNEL
            32.DLL), System.Diagnostics.ProcessModule (KERNELBASE.dll)...}

Name      : ai
Modules   : {System.Diagnostics.ProcessModule (ai.exe), System.Dia
            gnostics.ProcessModule (ntdll.dll), System.Diagnostics.ProcessModule (KERNEL32.DLL), S
            ystem.Diagnostics.ProcessModule (KERNELBASE.dll)...}
```

Рисунок 4.4 – Примеры использования, часть 3 (раскрытие вложенных свойств)

### Важные особенности

Format-List предназначен только для отображения. Для сохранения используется Out-File или Export-Csv

**Сокращение:** Вместо Format-List можно писать FL: Get-Service -Name "WinRM" | FL

## 5. Вывод:

В данной работе я изучила командлеты PowerShell. Для этого были выполнены упражнения по основным командам, а также индивидуальное задание.

Изученные команды:

- Get-Process – просматривать и обрабатывать запущенные процессы.
- Get-Service – выводит информацию о запущенных службах.
- Sort-Objects – позволяет сортировать объекты по определенным параметрам.
- Format-List (FL) – отображение объектов в виде списка свойств.
- Start-Service – запуск служб Windows.
- Restart-Service – перезапуск служб Windows.

Таким образом, я освоила базовые команды PowerShell, научилась управлять службами (Start-Service, Restart-Service) и форматировать вывод (Format-List). Полученные навыки позволяют автоматизировать задачи в ОС Windows и эффективно работать с консолью.

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Restart-Service (Microsoft.PowerShell.Management) - PowerShell | Microsoft Learn – URL: [https://learn.microsoft.com/ru-ru/powershell/module/microsoft.powershell.management/restart-service?view=powershell-5.1&WT.mc\\_id=ps-gethelp](https://learn.microsoft.com/ru-ru/powershell/module/microsoft.powershell.management/restart-service?view=powershell-5.1&WT.mc_id=ps-gethelp) (дата обращения 15.05.2025)
2. Start-Service (Microsoft.PowerShell.Management) - PowerShell | Microsoft Learn – URL: [https://learn.microsoft.com/ru-ru/powershell/module/microsoft.powershell.management/start-service?view=powershell-5.1&WT.mc\\_id=ps-gethelp](https://learn.microsoft.com/ru-ru/powershell/module/microsoft.powershell.management/start-service?view=powershell-5.1&WT.mc_id=ps-gethelp) (дата обращение 15.05.2025)
3. Format-List (Microsoft.PowerShell.Utility) - PowerShell | Microsoft Learn – URL: [https://learn.microsoft.com/ru-ru/powershell/module/microsoft.powershell.utility/format-list?view=powershell-5.1&WT.mc\\_id=ps-gethelp](https://learn.microsoft.com/ru-ru/powershell/module/microsoft.powershell.utility/format-list?view=powershell-5.1&WT.mc_id=ps-gethelp) (дата обращение 15.05.2025)