**Лабораторная работа №14**

По дисциплине: «Операционные системы и среды»

Выполнил: студент 2-го курса

Группы ВТс-222

Горбатенко В. К.

Проверил: Пронин П. А.

**1.3.4. Работа с объектами**

1. Определение конвейера (запомнить). Привести пример применения конвейера.

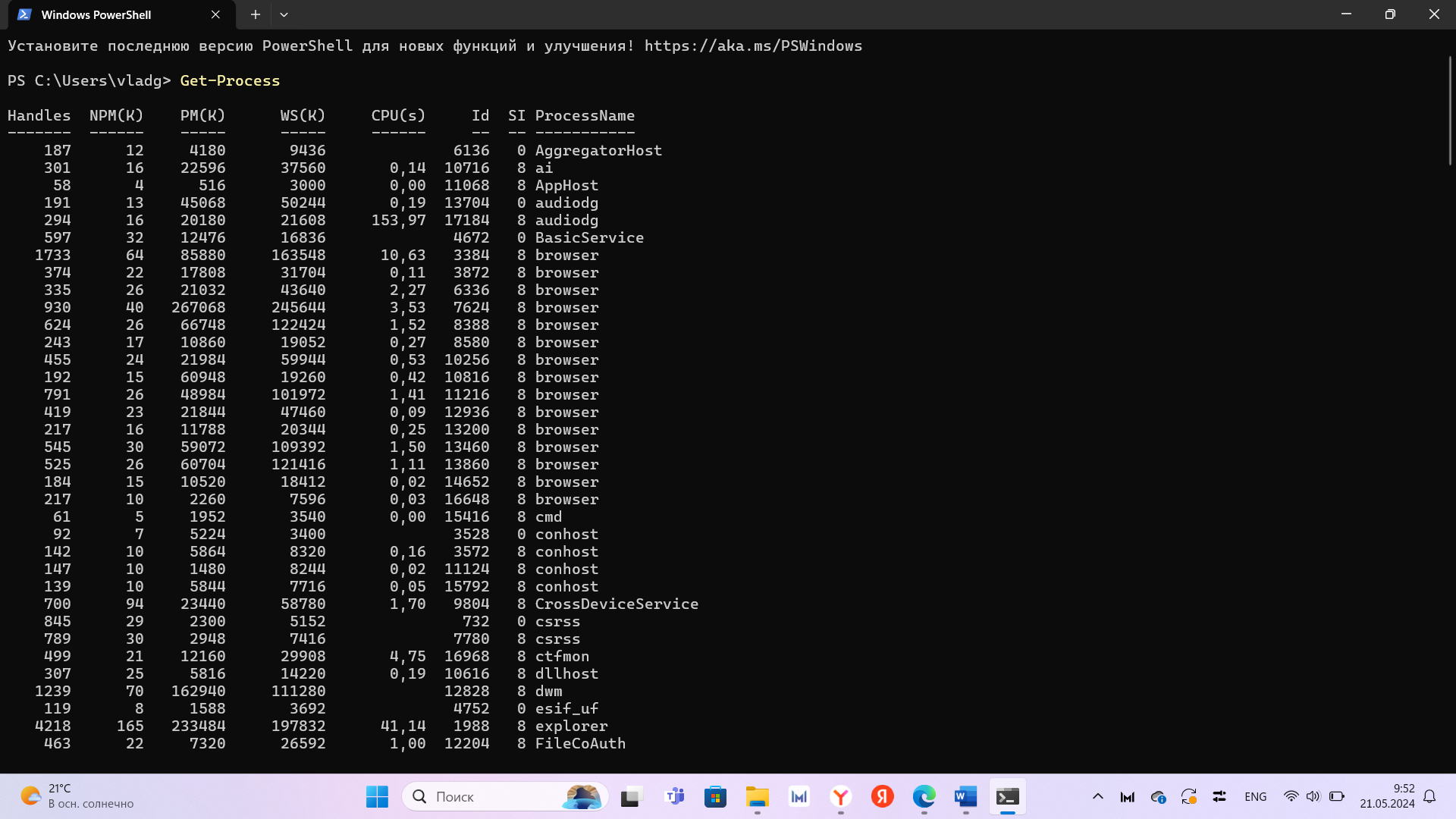
Конвейер – это последовательность команд, разделенных вертикальной чертой « | », в которой результаты обработки одной команды передаются другой команде.

- Просмотр структуры объектов.

-Фильтрация объектов в конвейере. Удаление из конвейера объектов, не удовлетворяющих определённому условию. Эту функциональность обеспечивает командлет Where-Object.

-Сортировка объектов. Сортировка элементов конвейера осуществляется с помощью командлета Sort-Object.

1. Получить начальную информацию об активных процессах, используя оболочку PowerShell. Расшифровать обозначения: Handles; NPM(K); WS(K); VM(M).



Handles - количество открытых дескрипторов файлов

NPM(K) - количество физической памяти используемой процессом (Кб)

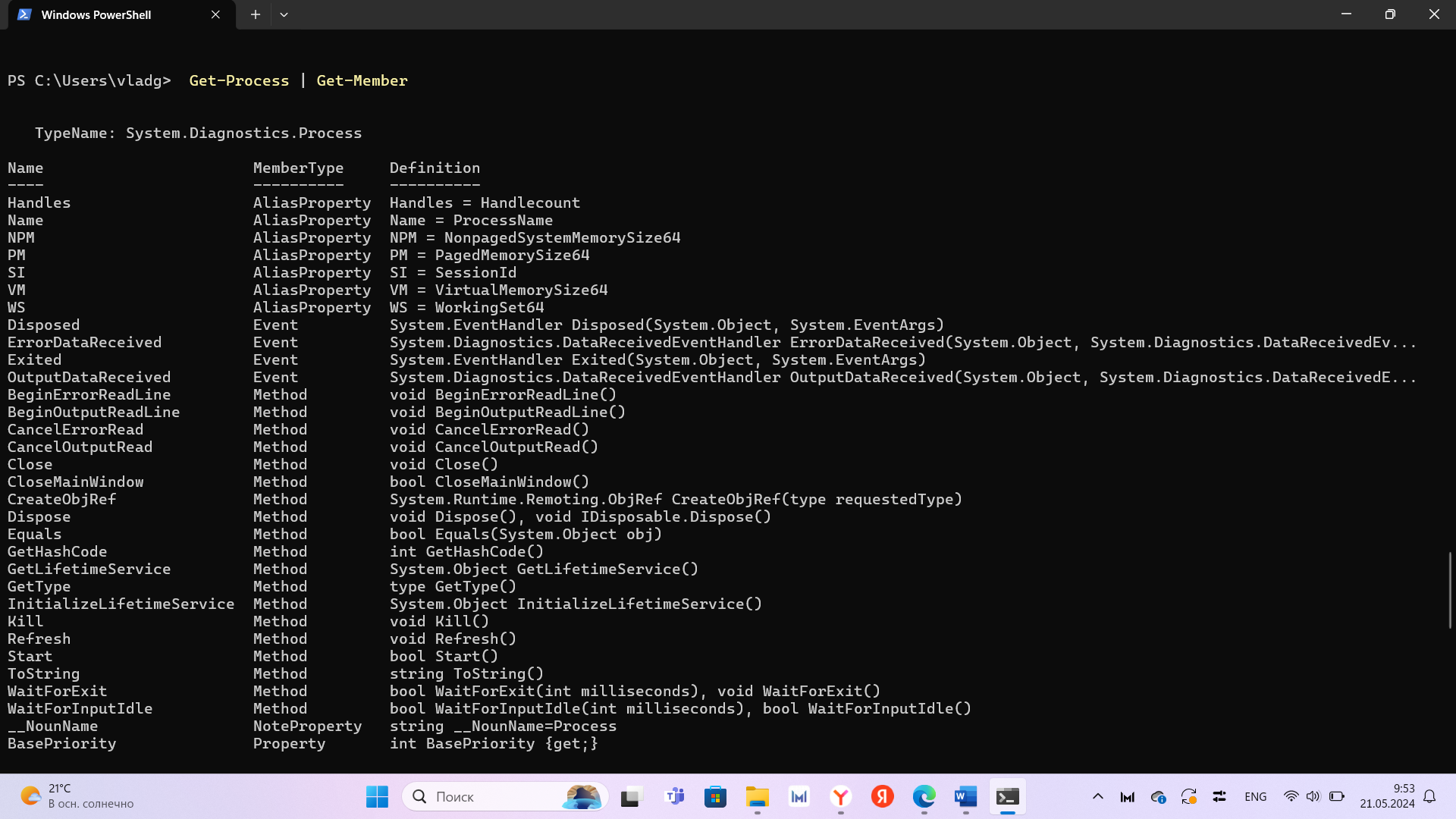
WS(K) - количество физической памяти, выделенной процессу (Кб)

VM(M) - количество виртуальной памяти используемой процессом (Мб)

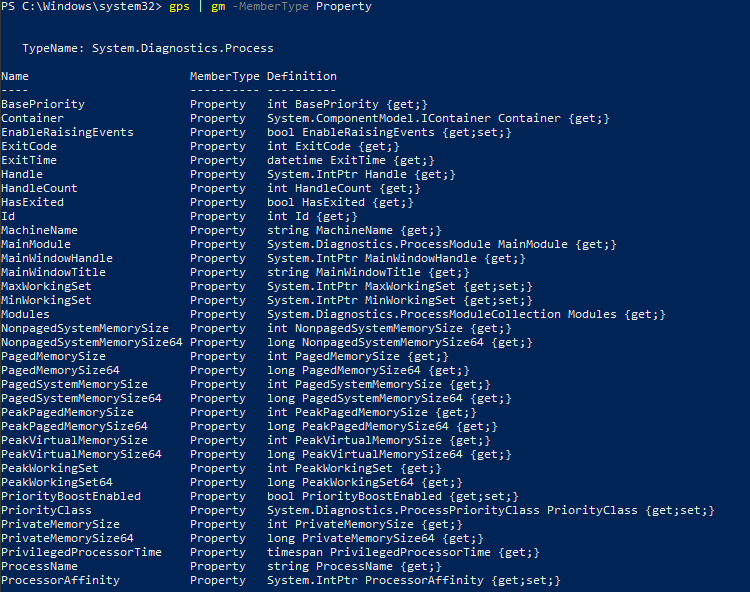
1. Привести пример использования конвейера из двух командлетов «Get-Process | Get-Member» в полном и сокращенном вариантах. Расшифровать столбцы: « MemberType»; «Definition».

MemberType: Тип члена (например, Property - свойство, Method - метод, ScriptMethod - скриптовый метод и т.д.)

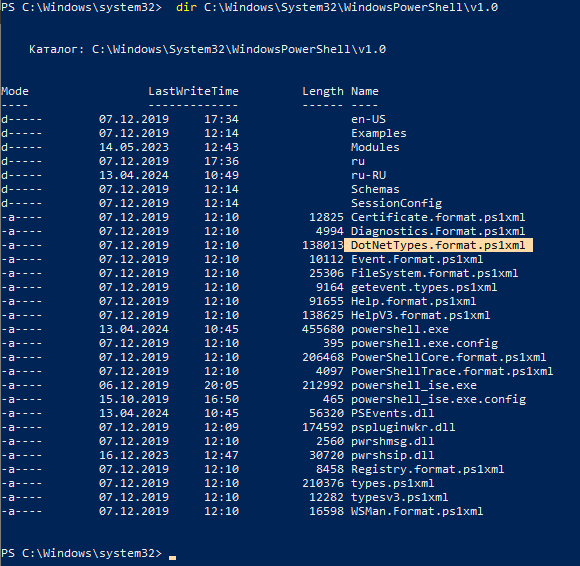
Definition: Определение (значение или содержание) члена (например, код метода, свойства или скрипта).



1. Получить перечень свойств процессов для заданной категории свойств. Категория свойств процессов «-MemberType Property».



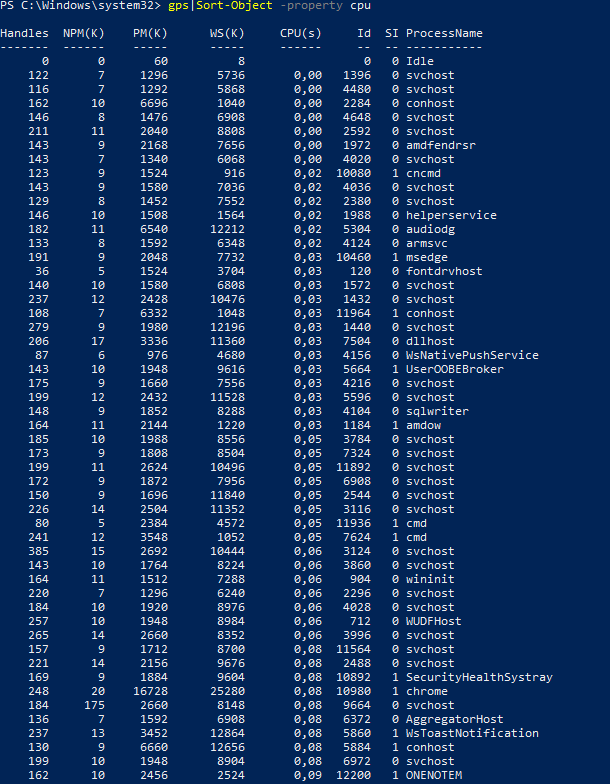
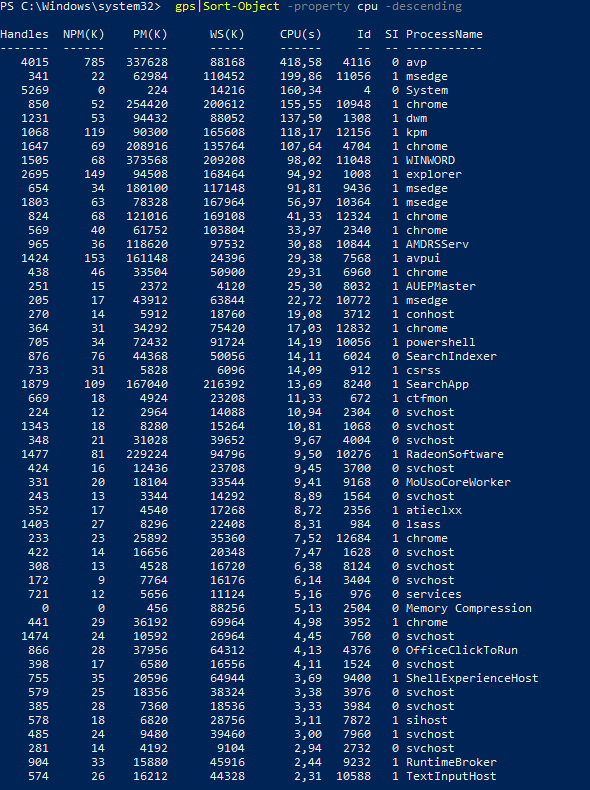
1. С помощью оболочки «PowerShell» найти несколько конфигурационных файлов, заканчивающихся на «\*format.pslxml». Выделить файл «DotNetTypes. format.pslxml.



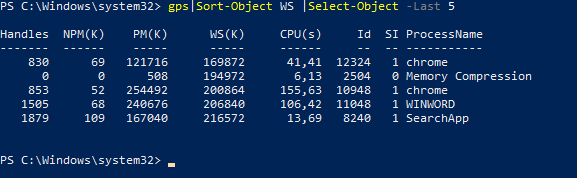
1. Пояснить «Методы дихотомии». Определение?

Дихотоми́ я – раздвоенность, последовательное деление на две части, не связанные между собой. Способ логического деления класса на подклассы, который состоит в том, что делимое понятие полностью делится на два взаимодополняющих понятия. Дихотомическое деление в математике является способом образования подразделов одного понятия или термина и служит для образования классификации элементов.

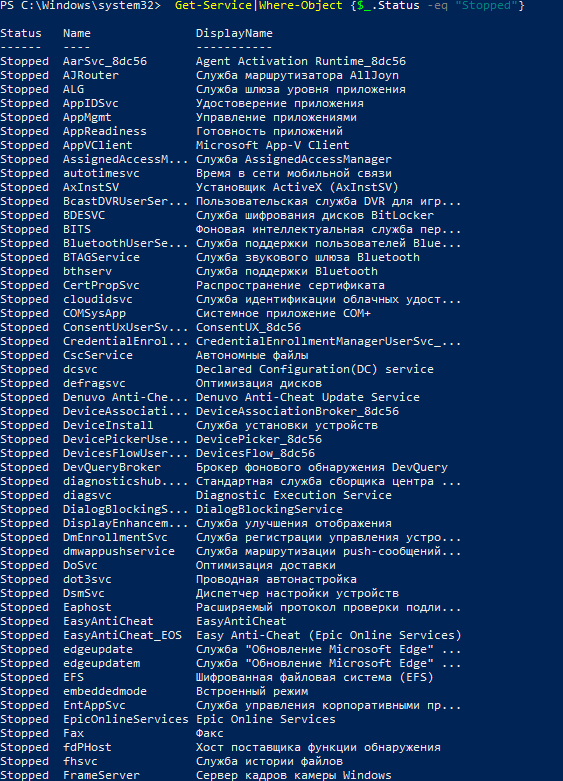
1. Вывести список процессов, упорядоченных по процессорному времени.

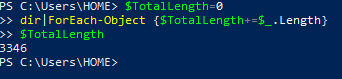
1. Сформировать команду-конвейер для получения пяти процессов, использующих наибольшие объемы памяти.



1. Отфильтровать данные по остановленным службам, используя команду-конвейер из двух командлет (свойство «Status» равно «Stopped»).



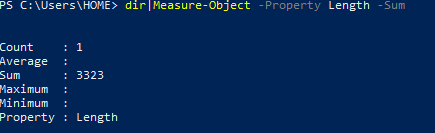
1. Подсчитать объем памяти, занимаемый каталогом (каталог по выбору). Решение задачи – обеспечить последовательностью команд.

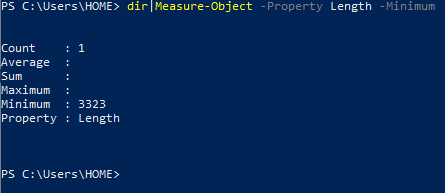


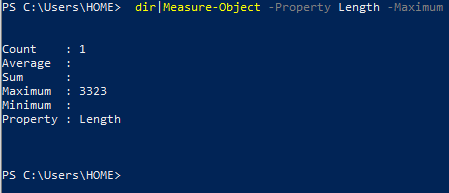
1. Подсчитать объем памяти, занимаемый каталогом (каталог по выбору. Решение задачи обеспечить командлетом «Measure-Object» в двух вариантах:

Вариант 1. Расчет памяти, занимаемой каталогом.

Вариант 2. Расчет памяти, занимаемой каталогом с минимумом и максимумом.







**Командлеты PowerShell**

Командлеты являются основными строительными блоками оболочки PowerShell. Они представляют собой небольшие программы, которые выполняют определенные задачи и возвращают результаты. Командлеты обычно имеют имя глагола-существительного, например Get-Command или Stop-Process.

**Работа с объектами PowerShell**

Объекты в PowerShell являются контейнерами для данных. Они могут содержать свойства (по именованным ключам) и методы (по именованным функциям). Объекты могут быть созданы с помощью командлетов или оператора New-Object.

Для доступа к свойствам объекта используется оператор ., например:

$computer = Get-Computer -Name "MyComputer"

$osVersion = $computer.OperatingSystem

Для вызова методов объекта используется оператор (), например:

$computer.Shutdown()

**1.3.4. Работа с объектами PowerShell**

В этом разделе представлено несколько примеров использования командлетов и объектов PowerShell:

**Создание объекта:**

$newObject = New-Object -TypeName System.Object

**Доступ к свойствам объекта:**

$propertyValue = $object.PropertyName

**Вызов методов объекта:**

$object.MethodName()

**Получение типа объекта:**

$typeName = $object.GetType().Name

**Преобразование объекта в строку:**

$stringRepresentation = $object.ToString()

**Копирование объекта:**

$copy = $object.Clone()

**Сравнение объектов:**

$isEqual = ($object1 -eq $object2)

**Перебор свойств объекта:**

foreach ($property in $object.Properties) {

Write-Host "$($property.Name): $($property.Value)"

}