

PowerShell – Zadania

1. Poniższy skrypt wyświetli nazwy wszystkich aktywnych zadań oraz ich liczbę. Rozszerzyć funkcjonalność skryptu tak, aby wyświetlał dodatkowo **łącznie** zajęta przez zadania **pamięć** (właściwość .PM) oraz nazwę procesu który zajmuje **najwięcej miejsca** w pamięci.

```
$procesy = Get-Process
foreach ($proces in $procesy)
{
    Write-Host Proces: $proces.Name
}
Write-Host Procesów łącznie: $procesy.Count
```

2. Poniższy skrypt pobiera od użytkownika liczbę (minimalny rozmiar pliku), a następnie wyświetla wszystkie pliki z katalogu bieżącego i jego podkatalogów o rozmiarze większym niż podana liczba. Rozszerz funkcjonalność skryptu tak, aby wyświetlał pliki o rozmiarze z podanego przez użytkownika przedziału i wyświetl wyniki w postaci tabeli (podpowiedź: | Out-GridView).

```
[long]$l1 = Read-Host "Minimalny rozmiar pliku (bajty)"
Get-ChildItem -Recurse | Where-Object { $_.Length -gt $l1 }
```

3. Poniższy skrypt sprawdza stan danego dnia (czy jest dniem wolnym i jakie święto obowiązuje). Rozszerzyć funkcjonalność skryptu tak, aby:

- sprawdzał inne święta z innych miesięcy, wykorzystując także zakresy dni (strona 51 – instrukcje),
- sprawdzał dzień tygodnia i odpowiednio uaktualniał zmienną \$wolne (\$teraz.DayOfWeek),

```
$teraz = Get-Date
[bool]$wolne = 0
$swieto = "brak"
switch($teraz.Month) {
    1 { # STYCZEŃ
        switch($teraz.Day) {
            1 { $wolne = 1; $swieto = "Nowy rok" }
            6 { $wolne = 1; $swieto = "Trzech króli" }
        }
    }
}
Write-Host Dzisiejsze swieto: $swieto
Write-Host Czy dzien wolny: $wolne
```

4*. Przerobić skrypt z zadania 3 w funkcję przyjmującą dwa argumenty (miesiąc oraz dzień) i zwracającą informację o tym czy dany dzień jest dniem wolnym. Wywołać kolejno funkcję dla kolejnych dni z kolejnych miesięcy wyświetlając wyniki w formie kalendarza (dni wolne wyświetlić na czerwono).

Skrypty .bat – Zadania

1. Poniższy skrypt sumuje liczby od 1 do 10. Wyrażenie *setlocal enabledelayedexpansion* pozwala na uaktualnianie wartości zmiennych globalnych wewnątrz pętli (wewnątrz pętli należy używać !suma! zamiast %suma%). Rozszerzyć funkcjonalność poniższego skryptu tak, aby obliczana była również wartość średnia (w tym celu zliczyć iteracje pętli).

```
setlocal enabledelayedexpansion
echo off
set suma=0
for /L %%i in (1,1,10) do (
    echo %%i
    set /A suma=!suma!+%%i
)
echo Suma: %suma%
```

2. Zaznajomić się z przykładowymi skryptami programu **xBATED** (do pobrania na stronie instytutu - Działalność dydaktyczna – Informatyka - Studia stacjonarne - Systemy operacyjne).