

Część Praktyczna

1. Zmienne

Napisz skrypt wymagający podania długości i wysokości trójkąta w argumentach. Oblicza pole powierzchni trójkąta i wyświetli wynik na ekranie.

```
1 param([
2     [double]$podstawa,
3     [double]$wysokosc
4 ])
5
6 $pole = 0.5 * $podstawa * $wysokosc
7 Write-Output "Pole trójkąta o podstawie $podstawa i wysokości $wysokosc wynosi $pole"
```

```
PS> ./nowyskrypt.ps1 -podstawa 10 -wysokosc 10
Pole trójkąta o podstawie 10 i wysokości 10 wynosi 50
```

2. Instrukcje warunkowe

Napisz skrypt, który zapyta użytkownika o liczbę i sprawdzi, czy jest większa od 10. Jeśli tak, wyświetli komunikat "Liczba jest większa od 10", w

```
(kali@kali)-[/home/kali/nowykatalog]
PS> ./nowyskrypt.ps1 -liczba 1
liczba jest mniejsza od 10
```

3. Operatory logiczne

Napisz skrypt, który poprosi użytkownika o podanie nazwy użytkownika i hasła, a następnie sprawdzi, czy podane dane są poprawne (np. nazwa użytkownika to "admin" i hasło to "password"). **Zastosuj maskowanie wprowadzanego hasła**

KONTAKT

Adrian Florek

adrian.florek@pwr.edu.pl

PROGRAMOWANIE SKRYPTOWE

```
# Pobranie loginu
$login = Read-Host "Podaj nazwe uzytkownika"

# Pobranie hasła z maskowaniem
$securePassword = Read-Host "Podaj haslo" -AsSecureString

# Konwersja SecureString do zwykłego tekstu (do porównania)
$plainPassword = [Runtime.InteropServices.Marshal]::PtrToStringAuto(
    [Runtime.InteropServices.Marshal]::SecureStringToBSTR($securePassword)
)

# Sprawdzenie danych
if ($login -eq "admin" -and $plainPassword -eq "password") {
    Write-Host "Zalogowano pomyslnie!"
} else {
    Write-Host "Blad logowania. Sprobuj ponownie."
}
```

```
Podaj nazwe uzytkownika: admin
Podaj haslo: *****
Zalogowano pomyslnie!
```

4. Pętle

Napisz skrypt generujący 10 adresów IP za pomocą pętli, które będą różniły się czwartym oktetem rosnąco w adresie 192.168.1.X

```
$liczba = "192.168.1."
for ($i=0;$i -le 10;$i++) {
    $wyjście = $liczba + $i
    write-output $wyjście
}
```

KONTAKT

Adrian Florek

adrian.florek@pwr.edu.pl

PROGRAMOWANIE SKRYPTOWE

```
(kali㉿kali)-[/home/kali/nowy katalog]
└─$ PS> ./nowyskrypt.ps1
192.168.1.0
192.168.1.1
192.168.1.2
192.168.1.3
192.168.1.4
192.168.1.5
192.168.1.6
192.168.1.7
192.168.1.8
192.168.1.9
192.168.1.10
```

5. Funkcje

Utwórz skrypt wyświetlający informacje o systemie: Aktualna data, Wersja systemu, Użytkownik, Adres IP.

Każda informacja powinna znajdować się w osobnej funkcji. Użyj jednej zmiennej z nazwą komputera w każdej funkcji.

```
34 Get-UserName -computerName $computerName
35 Get-IPAddress -computerName $computerName
```

```
Adres IP komputera (kali) to 10.0.2.15 i000::D13d:8ba1:380c:c716
```

KONTAKT

Adrian Florek
adrian.florek@pwr.edu.pl

PROGRAMOWANIE SKRYPTOWE

6. Zadania połączone

Utwórz plik CSV o nazwie input_file.csv z zawartością:

Component;GenerateReport

Computername;True

Manufacturer;True

Model;True

SerialNumber;True

CpuName;False

RAM;True

Napisz skrypt powershell, który będzie:

- odczyta zawartość pliku input_file.csv
- wykona iterację po wszystkich wierszach w pliku input_file.csv i zwróci informacje o komputerze/laptopie tylko wtedy, gdy w danym wierszu opcja GenerateReport ma wartość True.

Dane wyjściowe powinny wyglądać następująco:

Computername: ABWI45774RZ

Manufacturer: LENOVO

Model: ThinkPad T123

SerialNumber: M4564XDCG54DSA

RAM: 32 GB

Zauważ, że CpuName nie jest wyświetlane, ponieważ GenerateReport jest ustawione na False w tym wierszu. Spróbuj zmienić wartości True|False w kolumnie GenerateReport pliku input_file.csv, wykonaj skrypt ponownie.

Komenda Get-ComputerInfo jest Twoim przyjacielem.

KONTAKT

Adrian Florek

adrian.florek@pwr.edu.pl

PROGRAMOWANIE SKRYPTOWE

```
$data = Import-Csv -Path "./input_file.csv" -Delimiter ";"
$output = @()

foreach ($row in $data) {
    if ($row.GenerateReport -eq "True") {
        switch ($row.Component) {
            "Computername" {
                $name = hostname
                $output += "Computername: $name"
            }
            "Manufacturer" {
                $man = (sudo dmidecode -s system-manufacturer ).Trim()
                $output += "Manufacturer: $man"
            }
            "Model" {
                $model = (sudo dmidecode -s system-product-name ).Trim()
                $output += "Model: $model"
            }
            "SerialNumber" {
                $serial = (sudo dmidecode -s system-serial-number ).Trim()
                $output += "SerialNumber: $serial"
            }
            "CpuName" {
                $cpu = (lscpu | grep 'Model name' | awk -F ':' '{print $2}' ).Trim()
                $output += "CpuName: $cpu"
            }
            "RAM" {
                $mem = "{0:N0}" -f ((free -g | grep Mem | awk '{print $2}')) + " GB"
                $output += "RAM: $mem"
            }
        }
    }
}

$output -join "`n"|
```

```
(kali㉿kali)-[/home/kali/nowykatalog]
PS> ./nowyskrypt2.ps1
Computername: kali
Manufacturer: innotek GmbH
Model: VirtualBox
SerialNumber: 0
RAM: 3 GB
```

KONTAKT

Adrian Florek

adrian.florek@pwr.edu.pl

PROGRAMOWANIE SKRYPTOWE

Zadanie dodatkowe

Zapisz dane wyjściowe do pliku. Nazwa pliku powinna być jak w przykładzie: "ComputerReport_YYYYMMdd-HHmmss.txt", gdzie:

RRRR - bieżący rok

MM - bieżący miesiąc

dd - bieżący dzień

HH - bieżąca godzina

mm - bieżąca minuta

ss - bieżąca sekunda

Skrypt powinien wyświetlić komunikat w konsoli:

Raport został zapisany w pliku: <pełna ścieżka do pliku>.