


POLITECHNIKA WROCŁAWSKA  Wydział Informatyki i Telekomunikacji	Wydział: Informatyki i Telekomunikacji Kierunek: Cyberbezpieczeństwo Rok Akademicki: 2024/2025 Rok studiów, semestr: I, 2 Grupa: 2 Termin: <i>poniedziałek, godz. 15.15</i>
Programowanie skryptowe – Laboratorium 6	
Prowadzący: mgr inż. Karolina Pfajfer Data wykonania ćwiczenia: <i>07.04.2025</i> Data oddania sprawozdania: <i>14.04.2025</i>	Autor: <i>Adam Dąbrowski, 283832</i>

```

2 $system = (Get-ComputerInfo).OsName
3 $wersja = (Get-ComputerInfo).OsVersion
4 $uzytkownik = $env:USERNAME
5 $ip = Get-NetIpAddress
6 Write-Host "$data"
7 Write-Host "$system"
8 Write-Host "$wersja"
9 Write-Host "$uzytkownik"

```

Część Praktyczna

1. Zmienne

Napisz skrypt wymagający podania długości i wysokości trójkąta w argumencie, obliczający pole powierzchni trójkąta. Wynik wyświetl na ekranie.

```

PS C:\Users\PC\Desktop\Prog-Skryptowe> .\lab6z1.ps1 5 4
Pole trojkata wynosi: 10

```

```

1 param (
2     [int]$dlu,
3     [int]$szer
4 )
5
6 $pole=0.5 * $dlu * $szer
7 Write-Host " Pole trojkata wynosi: $pole"

```

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 przeciwnym razie wyświetli komunikat "Liczba jest mniejsza od 10".

```

PS C:\Users\PC\Desktop\Prog-Skryptowe> .\lab6z2.ps1
Podaj liczbę: 7
liczba jest mniejsza od 10

PS C:\Users\PC\Desktop\Prog-Skryptowe> .\lab6z2.ps1
Podaj liczbę: 56
liczba jest wieksza od 10

```

```

1 $liczba = Read-Host " Podaj liczbę"
2
3 if ([int]$liczba -gt 10){
4     Write-Host "liczba jest wieksza od 10"
5 } else {
6     Write-Host " liczba jest mniejsza od 10"
7 }
8

```

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.

```

PS C:\Users\PC\Desktop\Prog-Skryptowe> .\lab6z3.ps1
Podaj nazwe uzytkownika: admin
dane poprawne
System.Security.SecureString

```

```

1 $login = Read-Host "Podaj nazwe uzytkownika"
2 $haslo = Read-Host "podaj haslo" -AsSecureString
3
4 $plainPassword = [Runtime.InteropServices.Marshal]::PtrToStringAuto([Runtime.InteropServices.Marshal]::SecureStringToBSTR($haslo))
5 if ($login -eq "admin" -and $plainPassword -eq "password") {
6     Write-Host "dane poprawne"
7 } else {
8     Write-Host "Nieporprawne dane logowania"
9 }
10 Write-Host $haslo

```

4. Pętle

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

```
PS C:\Users\PC\Desktop\Prog-Skryptowe> .\lab6z4.ps1
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
```

```
1 for ($i =1; $i -le 10; $i++) {
2     Write-Host "192.168.1.$i"
3 }
4 |
```

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.

Przykład: „Aktualna data na (Nazwa komputera) to X”

```
1 $komp = $env:COMPUTERNAME
2
3 function date {
4     Write-Host "-----"
5     Write-Host "Aktualna data na $komp to $(Get-Date) "
6     Write-Host "-----"
7 }
8 function wersja {
9     $wersja = $(Get-ComputerInfo | Select-Object -ExpandProperty WindowsVersion)
10    Write-Host "Wersja systemu na $komp to $wersja"
11    Write-Host "-----"
12 }
13 function uzytkownik {
14     Write-Host "Zalogowany uzytkownik na $komp to $env:USERNAME"
15     Write-Host "-----"
16 }
17 function ip {
18     $ip = ipconfig | Select-String -Pattern 'IPv4.\s'
19     Write-Host "Adres IP na $komp to $ip"
20     Write-Host "-----"
21 }
22 date
23 wersja
24 uzytkownik
25 ip
26
```

```
PS C:\Users\PC\Desktop\Prog-Skryptowe> .\lab6z5.ps1
-----
Aktualna data na DESKTOP-9ND2DTC to 04/14/2025 14:34:57
-----
Wersja systemu na DESKTOP-9ND2DTC to 2009
-----
Zalogowany użytkownik na DESKTOP-9ND2DTC to PC
-----
Adres IP na DESKTOP-9ND2DTC to   IPv4 Address. . . . . : 192.168.56.1   IPv4 Address. . . . .
. . : 192.168.0.236
-----
```

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:

- odczytywać zawartość pliku input_file.csv

- wykonywać 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.

```
Computername: DESKTOP-9ND2DTC
Manufacturer: LENOVO
Model: ThinkPad T480
SerialNumber: M32464TGFG65DSA
RAM: 32 GB
```

```
$csv = Import-Csv -Path "input_file.csv" -Delimiter ";"
$info = Get-ComputerInfo

$computerInfo = Get-CimInstance -ClassName Win32_ComputerSystem
$biosInfo = Get-CimInstance -ClassName Win32_BIOS
$osInfo = Get-CimInstance -ClassName Win32_OperatingSystem
$cpuInfo = Get-CimInstance -ClassName Win32_Processor
foreach ($row in $csv) {
    if ($row.GenerateReport -eq "True") {
        switch ($row.Component) {
            "Computername" { Write-Host "Computername: $env:COMPUTERNAME" }
            "Manufacturer" { Write-Host "Manufacturer: $($computerInfo.Manufacturer)" }
            "Model" { Write-Host "Model: $($computerInfo.Model)" }
            "SerialNumber" { Write-Host "SerialNumber: $($biosInfo.SerialNumber)" }
            "CpuName" { Write-Host "CpuName: $($cpuInfo.Name)" }
            "RAM" {
                $ramGB = [math]::Round($computerInfo.TotalPhysicalMemory / 1GB, 0)
                Write-Host "RAM: $ramGB GB" }
        }
    }
}
```