

Pętla FOR - KLUCZ

1.

```
# Upewnij się, że katalog istnieje
$directory = "C:\temp"
if (-not (Test-Path -Path $directory)) {
    New-Item -Path $directory -ItemType Directory
}
# Utwórz kilka plików w katalogu
$fileName = @( "file1.txt", "file2.txt", "file3.txt", "file4.txt", "file5.txt" )
foreach ($fileName in $fileNames) {
    $filePath = Join-Path -Path $directory -ChildPath $fileName
    New-Item -Path $filePath -ItemType File -Force
}
# Potwierdź utworzenie plików
Get-ChildItem -Path $directory
```

2. `$files = Get-ChildItem C:\temp`

3. `foreach($f in $files)`

```
{
    $f.Encrypt()
    $f.IsReadOnly = $true
}
```

W podanym kodzie problem wynika z tego, że \$f jest instancją System.IO.DirectoryInfo, a nie System.IO.FileInfo. Metody i właściwości Encrypt() i IsReadOnly są dostępne tylko dla plików (System.IO.FileInfo), a nie dla katalogów (System.IO.DirectoryInfo).

Aby rozwiązać ten problem, upewnij się, że \$files zawiera listę plików, a nie katalogów. Można to zrobić, używając metody GetFiles() zamiast GetDirectories() lub podobnej metody. Poprawiona wersja kodu:

```
$files = Get-ChildItem C:\temp -File
foreach($f in $files)
```

```
{  
    $f.Encrypt()  
    $f.IsReadOnly = $true  
}
```

4. \$files = Get-ChildItem C:\temp -File

```
foreach($f in $files)
```

```
{  
    $f.Decrypt()  
    $f.IsReadOnly = $false  
}
```

5. for(\$i=0;\$i -lt 5; \$i++)

```
{  
    [console]::beep(500,300)  
    Start-Sleep -Milliseconds 1000  
}
```