

DIE BESTE BACKUP-SOFTWARE



Das Backup Programm

GRUPPENNAME: BRIMSTONE FAN CLUB

GRUPPENMITGLIEDER: JOEL, LUCAS

DATUM: 15.06.2022

VERSION 1.0

Woche 1

In der Ersten Woche hatten wir ja weniger Zeit deswegen haben wir Informationen gesammelt und uns Gedanken darübergemacht wie wir so ein Projekt bewältigen wollen.

<# Unser Vorschlag #>

1. Als erstes erstellen wir die Textdateien und die Ordner. Diese sollen ja logische Namen haben also lassen wir diese der Reihe nach Generieren.
2. Dann wird der Backup durchgeführt
3. Ein Log wird erstellt. == In diesem Log soll stehen welche Dateien wo hin und wann verschoben wurde. Zusätzlich wollen wir in dieses ein Booleschen Wert reinschreiben welcher angibt ob die Datei erfolgreich angekommen ist.

<# Das Ziel #>

Das Programm soll einen Backup von der Quelle machen und dieses ins Ziel kopieren. Wenn die Dateien rüber kopiert werden wir eine Log Datei erstellt in welcher steht von wo die Sachen kopiert wurden wohin diese kopiert werden wann sie kopiert wurden und ob der Backup Prozess Erfolgreich oder nicht Erfolgreich war.

Woche 2

In der Zweiten Woche war Herr Lizio nicht da und wir haben der Klasse geholfen den Auftrag besser zu verstehen da noch vieles unklar war.

Liebe Herren Erni, Schulz und Beqirai

Ganz herzlichen Dank an Euch, dass Ihr in die Rolle der Lehrperson gesprungen seit und der Klasse weitergeholfen habt im Bereich PowerShell.

Ich schätze das sehr 🙏

Beste Grüße

Roland Bucher

Weil wir deshalb selber wenig machen konnten haben wir den ersten Schritt unseres Backup Programmes in Angriff genommen. Das Erstellen der Logisch nummerierten Textdateien und Ordnen mit Inhalten.

Um dieses Problem haben wir uns überlegt eine for Schleife zu benutzen.

```
function txterstellen
{
    For ($i = 1; $i -lt 3; $i++) {
        $i
    }
}
```

Wir haben uns im Unterricht ja schon mal mit so einer for Schleife auseinandergesetzt. Unsere Überlegung war dann die Erstellung der Textdatei mit dem \$i zu ersetzen.

<# txtersteller ohne Inhalt Generator #>

```
function txterstellen
{
    For ($i = 1; $i -lt 3; $i++){
        New-Item -Path "C:\Users\vmadmin\OneDrive - sluz\Desktop" -
Name "txt$i.txt" -ItemType "file"
    }
}
```

<# Finaler txtersteller mit Inhalt Generator #>

```
$Zielpfad = "C:\Test"
$AnzDateien = 70
$AnzZeich = 1000
$NameLenth = 10
$char = [char[]] 'abcdefghijklmnopqrstuvwxyz1234567890'
```

```

for($j = 1; $j -lt $AnzDateien+1; $j++){
    # Inhalt #
    $Inhalt = ""
    for($i = 0; $i -lt $AnzZeich; $i++){
        $Inhalt+=$char[(Get-Random -Maximum $char.Length -Minimum
0)];
    }





    # Datei erstellen #
    New-Item -Path $Zielpfad -Name "$j.txt" -Value "$Inhalt"
}

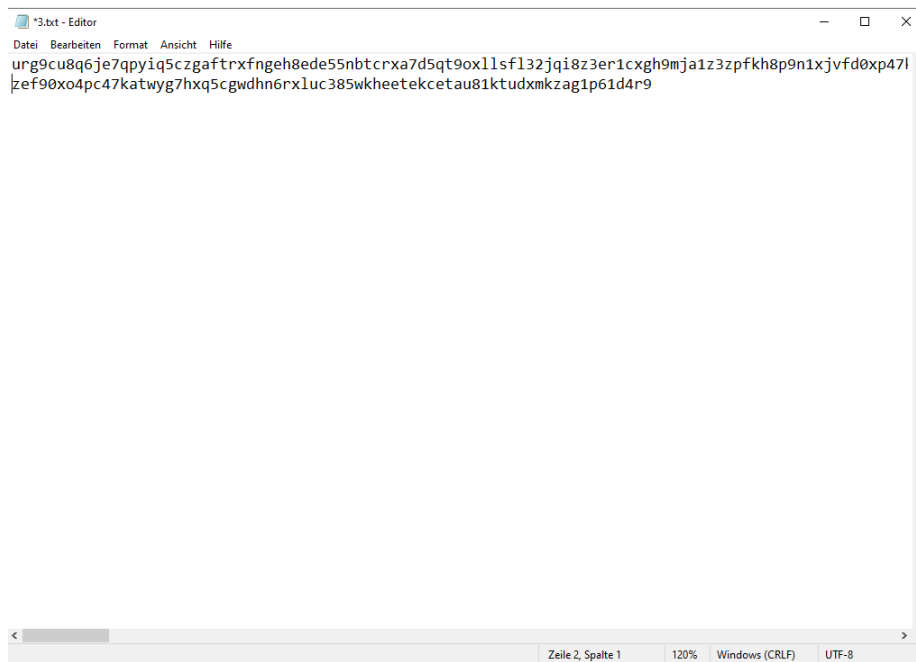
```

Hier sieht man das 10 Dateien mit einem Logischen Namen generiert wurden.

Verzeichnis: C:\Users\vmadmin\OneDrive - sluz\Desktop

Mode	LastWriteTime		Length	Name
----	-----	-----	-----	----
-a----	22.06.2022	15:07	999	1.txt
-a----	22.06.2022	15:07	999	2.txt
-a----	22.06.2022	15:07	999	3.txt
-a----	22.06.2022	15:07	999	4.txt
-a----	22.06.2022	15:07	999	5.txt
-a----	22.06.2022	15:07	999	6.txt
-a----	22.06.2022	15:07	999	7.txt
-a----	22.06.2022	15:07	999	8.txt
-a----	22.06.2022	15:07	999	9.txt
-a----	22.06.2022	15:07	999	10.txt


 1.txt		22.06.2022 15:07	Textdokument	1 KB
 2.txt		22.06.2022 15:07	Textdokument	1 KB
 3.txt		22.06.2022 15:07	Textdokument	1 KB
 4.txt		22.06.2022 15:07	Textdokument	1 KB
 5.txt		22.06.2022 15:07	Textdokument	1 KB
 6.txt		22.06.2022 15:07	Textdokument	1 KB
 7.txt		22.06.2022 15:07	Textdokument	1 KB
 8.txt		22.06.2022 15:07	Textdokument	1 KB
 9.txt		22.06.2022 15:07	Textdokument	1 KB
 10.txt		22.06.2022 15:07	Textdokument	1 KB



1

Zeichenzähler

Zeichen zählen & Wörter zählen



urg9cu8q6je7qpyiq5czgafrxfngeh8ede55nbtcrxa7d5qt9oxllsf132jq18z3er1cxgh9mja1z3zpfkh8p9n1xjvfd0xp47ksp380mvd3tmhum0txrfiqswidffj577zvi1p1eu5h7m9zomdjg81ekac1vvqslg2h5m55sp0jrpjqza6zf64vcqzsmfn5bynyrig3b8zox69nfm1lj9r3bp8tuyh228cwppi257cdyl9z3p95e3rqtevhm6o50s1nj72eemtielg0omxxevm60hbrqdtawsgemzywxjpwu6esq8dhlbmdgltq783btjwfm45nh0dhk7mqiqyp0dgmarr0wnerosrgtm9qghdfp7h727p913v9kbgc7380ocw27jqgdmimgbj8s1nmv6eq8r3x3iczcpdiy57a19c0wjzmzis3btt3z8se9v4ce2oi4qtad8s3mjoce4gq9mr1jtwmt9xdc59fbpri631f9roc8nn20sfj9aw6xkozl48nbpqeqdjh9v7k3rpm7sksc2frqp5n4l7g3che6f40dxwwoa3mccppmdeyxym3ly501l5ugxip5nclkk11aatrsu0x31z9xz0tj36hy0tundgkbd31mvk1ewlscd8qbwpvcvac66c9j4pegvbi7updwswso71l7owouy6n104yigsc0whfkip5rxz5c644yn47u83stne6nm60sgtvvuy232v8qzeu7470reswdce3582s1cyoeq4o4wkada0cy2xypxmdo0hptl8kk6l1qjwmy7d50yyvwwk11qeyestb1hjdjxk5kpjeqn48j0yhjlv0gm82muwzvlxrq5y0ey981vk4me1a8a58uladaapj175mn2ffqgsdombzxp71c1v0grp1yqxgf5q4imerr95x9wcun3zef90xo4pc47katwyg7hxq5cgwdhn6rxluc385wkheetekcetau81ktudxmkgzag1p61d4r9

1000 Zeichen

1 Wörter

1000 Ohne Leerzeichen

1 Absätze

2

¹ Die txt Datei mit 1000 Zeichen.

² Zeichenzähler um zu zeigen das es wirklich 1000 Zeichen sind.

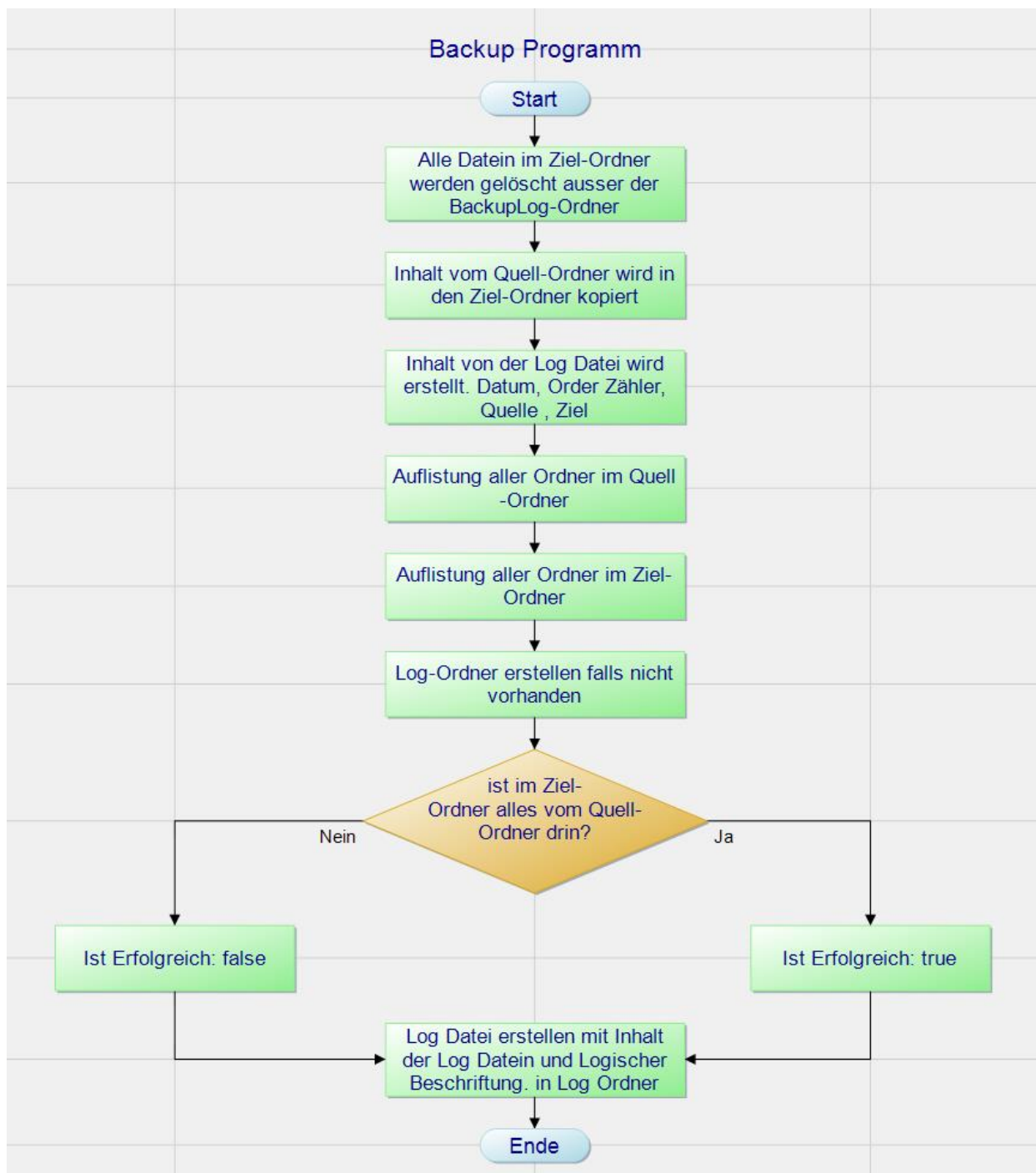
Woche 3

Heute haben wir angefangen das Backup Programm zu Programmieren

<# Erster Backup Versuch #>

```
function BackupProgramm([string]$source, [string]$BackupOrdnerPfad){  
    Copy-Item -Path $source -Destination $BackupOrdnerPfad -Recurse  
}
```

<# Das PAP zu unserem Projekt #>



Woche 4

Nachdem wir beide mehrere Test Backup Programme programmiert haben, haben wir uns für eine Lösung von Joel entschieden welche wir als einfachste und beste fanden.

<# Fertiger Backup Code #>

```
$LogPath = $BckPath
[bool]$IsErfolgreich = 1
[string]$inhalt = ""
$newline = "`r`n"
$tab = "`t"
$FileCount = (Get-ChildItem $SrcPath | Measure-Object).Count

#Alle Dateien im Ziel-Ordner löschen, ausser Backuplog-Ordner#
Remove-Item -Path "$BckPath\*" -Exclude "BackupLog"
for($i = 0; $i -lt ((Get-ChildItem $SrcPath | Measure-Object).Count); $i++){
    Copy-Item ($SrcPath+(Get-ChildItem $SrcPath).BaseName[$i]) -Destination
    $BckPath -Recurse
}

#Inhalt von der Log-Datei#
$date = Get-Date
$inhalt+= [string]$date+$newline+[string]$FileCount+" wurden von '"+$SrcPath+"'
nach '"+$BckPath+"' verschoben:"+$newline+$newline+$SrcPath+": "+$newline

#Dateinamen vom Quell-Ordner#
for($i = 0; $i -lt $FileCount; $i++){
    $inhalt+=$tab+(Get-ChildItem -Path $SrcPath).BaseName[$i] +$newline
}

#Dateinamen vom Ziel-Ordner#
$inhalt+=$newline
$inhalt+=$BckPath+": "+$newline;
for($i = 0; $i -lt ((Get-ChildItem $BckPath | Measure-Object).Count); $i++){
    $inhalt+=$tab+(Get-ChildItem -Path $BckPath).BaseName[$i] +$newline
}
$inhalt+=$newline

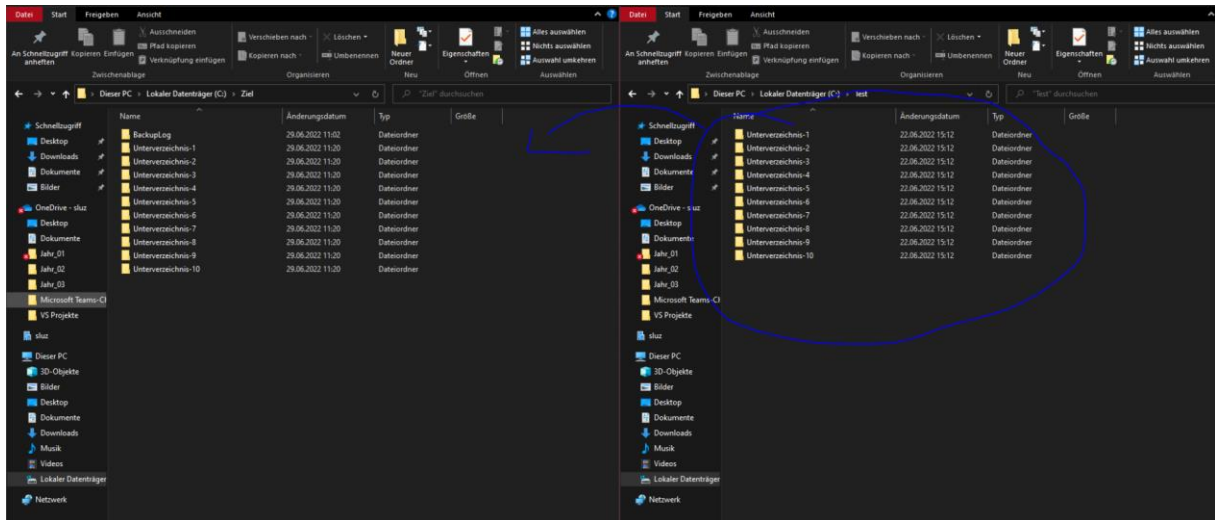
#Vergleich, ob alles in ordnung ist#
if(Compare-Object (Get-ChildItem -Path $SrcPath) (Get-ChildItem -Path $BckPath -
Exclude "BackupLog") -Property Name, Length){
    $IsErfolgreich=0
}

#"Ist Erfolgreich" in Log.txt#
$inhalt+="Ist erfolgreich: "
if($IsErfolgreich -eq 1){
    $inhalt+="True"
}else{
    $inhalt+="False"
}

#Log-Ordner erstellen, falls nicht vorhanden#
if(-not (Test-Path -Path "$LogPath\BackupLog")){
    New-Item -Path $LogPath -Name "BackupLog" -ItemType "directory"
}

#Log-Datei in Log-Ordner erstellen#
New-Item -Path "$LogPath\BackupLog" -Name ("log-"+$date.Year+"-"+$date.Month+"-"+$date.Day+"-"+$date.Hour+"-"+$date.Minute+"-"+$date.Second+".txt") -ItemType "file" -Value $inhalt
Write-Host $inhalt
```

Mit diesem Backup Programm kann man die Quelle, das Ziel und es wird eine Logdatei erstellt welche alle relevanten Punkte in sich beinhaltet.



3

BackupLog	29.06.2022 11:02	Dateiordner
Unterverzeichnis-1	29.06.2022 11:20	Dateiordner
Unterverzeichnis-2	29.06.2022 11:20	Dateiordner
Unterverzeichnis-3	29.06.2022 11:20	Dateiordner
Unterverzeichnis-4	29.06.2022 11:20	Dateiordner
Unterverzeichnis-5	29.06.2022 11:20	Dateiordner
Unterverzeichnis-6	29.06.2022 11:20	Dateiordner
Unterverzeichnis-7	29.06.2022 11:20	Dateiordner
Unterverzeichnis-8	29.06.2022 11:20	Dateiordner
Unterverzeichnis-9	29.06.2022 11:20	Dateiordner
Unterverzeichnis-10	29.06.2022 11:20	Dateiordner

4

log-2022-06-29_10-58-02.txt	29.06.2022 10:58	Textdokument	1 KB
log-2022-06-29_10-58-39.txt	29.06.2022 10:58	Textdokument	1 KB
log-2022-06-29_11-01-49.txt	29.06.2022 11:01	Textdokument	1 KB
log-2022-06-29_11-02-26.txt	29.06.2022 11:02	Textdokument	1 KB
log-2022-06-29_11-20-37.txt	29.06.2022 11:20	Textdokument	1 KB

5

³ Auf dem Bild kann man erkennen wie die Ordner von der Quelle zum Ziel kopiert werden.

⁴ Wenn die Ordner kopiert werden erstellt sich ein BackupLog in welchem sich die Log Dateien befinden

⁵ So sieht es im BackupLog Ordner aus. Die Log Dateien sind beschriftet mit dem Jahr, Datum und der Uhrzeit



```
log-2022-06-29_11-20-37.txt - Editor
Datei Bearbeiten Format Ansicht Hilfe
06/29/2022 11:20:37
10 wurden von 'C:\Test\' nach 'C:\Ziel\' verschoben:

C:\Test\:
  Unterverzeichnis-1
  Unterverzeichnis-10
  Unterverzeichnis-2
  Unterverzeichnis-3
  Unterverzeichnis-4
  Unterverzeichnis-5
  Unterverzeichnis-6
  Unterverzeichnis-7
  Unterverzeichnis-8
  Unterverzeichnis-9

C:\Ziel\:
  BackupLog
  Unterverzeichnis-1
  Unterverzeichnis-10
  Unterverzeichnis-2
  Unterverzeichnis-3
  Unterverzeichnis-4
  Unterverzeichnis-5
  Unterverzeichnis-6
  Unterverzeichnis-7
  Unterverzeichnis-8
  Unterverzeichnis-9

Ist erfolgreich: True

Zeile 1, Spalte 1    80%    Windows (CRLF)    UTF-8
```

6

<# Anleitung für unser Programm #>

Als erstes gibt man im Skript das Quellen, Ziel und das BackupLog Verzeichnis an wo man die verschiedenen Ordner hat oder erstellen will dann führt man das Script nur noch aus und man ist fertig.

⁶ Und so sind die Log Dateien Strukturiert