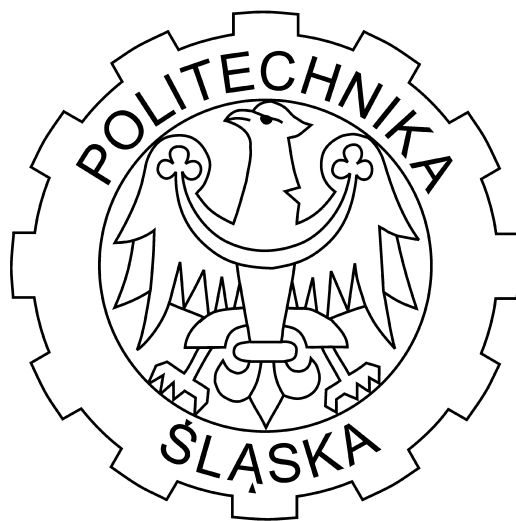


Projekt zaliczeniowy
Języki Skryptowe
”Licznik Znaków”

Artur Bednarczyk
Politechnika Śląska
Wydział Matematyki Stosowanej
Informatyka, semestr III

Styczeń 2018



Spis treści

1	Opis zadania	3
2	Algorytm	3
2.1	Opis	3
2.2	Schemat blokowy	4
3	Implementacja	5
3.1	Batch	5
3.2	C++	6
3.3	Python	7
4	Działanie	8
4.1	Schemat serwera	8
4.2	Prezentacja wyników	9
5	Testy	10
6	Podsumowanie	11
6.1	O projekcie	11
6.2	Przyszłość	11

1 Opis zadania

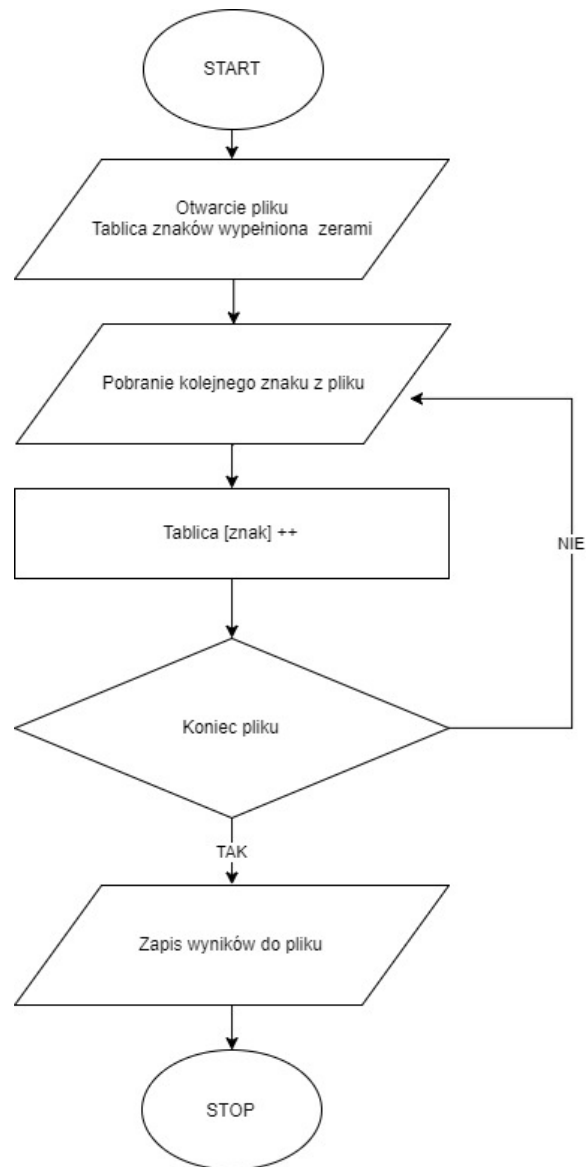
Celem zadania jest utworzenie skryptu systemowego, który będzie pełnił rolę menu. Menu powinno posiadać opcje takie jak utworzenie nowego raportu, utworzenie kopii zapasowej istniejącego raportu, wyjście. Nowy raport będzie generowany poprzez skrypt Python, na podstawie plików wyjściowych utworzonych przez aplikację napisaną w dowolnym języku programowania, która zlicza liczbę wystąpień znaków w plikach tekstowych, które są traktowane jako pliki wejściowe tej aplikacji.

2 Algorytm

2.1 Opis

Po uruchomieniu skryptu systemowego ukazuje się menu, którego opcjami są: generowanie nowego raportu, kopia zapasowa istniejącego raportu, wyjście. Po wybraniu opcji "wyjście" skrypt kończy pracę i nic więcej się nie dzieje. Po wybraniu opcji "kopia zapasowa" skrypt sprawdza czy istnieje kopia zapasowa, jeśli tak to wyświetla komunikat, że istnieje, jeśli nie to tworzy nową kopię. Kopia zapasowa posiada w nazwie datę oraz godzinę wykonania, na podstawie której jest sprawdzane czy istnieje. Po wybraniu opcji "utwórz nowy raport", dla każdego pliku wejściowego wykonana jest program zliczający i na wyjściu daje plik z liczbą znaków. Na podstawie tych plików wyjściowych Python generuje raport.

2.2 Schemat blokowy



Rysunek 1: Schemat Blokowy

3 Implementacja

3.1 Batch

```
1 @echo off
2 :menu
3 cls
4 echo //////////////////////////////////
5 echo \\\          MENU          \\\
6 echo \\\      1. NEW RAPORT      \\\
7 echo \\\      2. BACKUP          \\\
8 echo \\\      3. EXIT            \\\
9 echo //////////////////////////////////
10 set /p in="Select option: "
11 if %in%==1 goto newRaport
12 if %in%==2 goto backup
13 if %in%==3 goto end
14 goto menu
15 :newRaport
16 cls
17 echo Wait...
18 if not exist input mkdir input
19 if not exist output mkdir output
20 if not exist report mkdir report
21 for /f %x in ('dir /b input\') do ( charCounter.exe %x
22 if errorlevel 0 echo DONE FOR: %x
23 )
24 main.py
25 echo Done...
26 pause
27 goto menu
28 :backup
29 cls
30 echo Wait...
31 if not exist backup mkdir backup
32 set backupName=backup_%date:~10,4%-~7,2%-~4,2%_%time:~0,2%-~3,2%.html
33 if exist "backup\%backupName%" echo Already exists
34 if not exist "backup\%backupName%" copy "report\report.html" "backup\%backupName%"
35 echo Done...
36 pause
37 goto menu
38 :end
```

Rysunek 2: Kod Batch

3.2 C++

```
1 #pragma once
2 #include <fstream>
3 #include <string>
4 class counter
5 {
6 private:
7     int table[127];
8     int size;
9     char input;
10    char* source;
11
12    void count();
13    void saveToFile();
14 public:
15     counter(char* source);
16     void operate();
17     ~counter();
18 };
```

(a) .h

```
1 #include "stdafx.h"
2 #include "counter.h"
3 #include <fstream>
4 #include <string>
5 counter::counter(char* source)
6 {
7     this->source = source;
8     for (int i = 0; i <= 127; i++) {
9         table[i] = 0;
10    }
11 }
12 counter::~counter()
13 {
14     delete table;
15     delete source;
16 }
17 void counter::count()
18 {
19     std::string fileSource = "input/";
20     fileSource += source;
21     std::ifstream file(fileSource, std::ios::in);
22     if (file.is_open()) {
23         while (file.good()) {
24             input = file.get();
25             table[(int)input]++;
26         }
27     }
28     file.close();
29 }
30 void counter::saveToFile()
31 {
32     std::string output="output/";
33     output.append(source);
34
35     std::ofstream outFile(output, std::ios::out);
36     for (int i = 32; i < 127; i++) {
37         outFile << (char)i << " --> " << table[i] << std::endl;
38     }
39 }
40 void counter::operate()
41 {
42     count();
43     saveToFile();
44 }
```

(b) .cpp

Rysunek 3: Kod C++

Utworzone funkcje `count()` oraz `saveToFile()` są odpowiedzialne odpowiednio za zliczanie znaków oraz zapis wyników do pliku. Liczba wystąpień danego znaku przechowywana jest w tablicy.

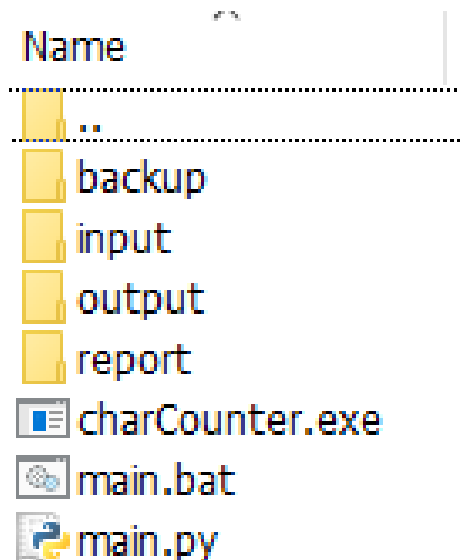
3.3 Python

```
1 import datetime
2 import os
3 currentTime = datetime.datetime.now().strftime("%Y-%m-%d %H:%M:%S")
4 reportFile = open('report/report.html', 'w')
5 html = """<html>
6 <head><title>Report</title>
7 <style>
8 body{background:#252525;color:#ACAC11; font-family: 'Trocchi', serif;font-weight: bold;}
9 .title{color: khaki;font-family: 'Trocchi', serif;font-size: 48px;font-weight: normal;line-height: 48px;margin: auto;}
10 .div_table{height:75%;width:75%;margin:auto;overflow:auto;}
11 .titleText{color: #c0c0c0;font-family: 'Trocchi', serif;font-size: 32px;font-weight: normal;line-height: 48px;margin: auto;}
12 .reportTable{margin: auto;background: #333333;border: solid khaki;text-align:center;overflow:auto;}
13 .reportTable td{border: solid black thin;width: 75px;}
14 table.reportTable > tbody > tr:first-child > td:first-child{background: grey;width:40px;}
15 table.reportTable > tbody > tr > td:first-child, table.reportTable > tbody > tr:first-child{background: green;color:#d0d0d0;}
16 table.reportTable td:hover{background: silver;color: #333333;}
17 </style>
18 </head>
19 <body><div align="center" class="title">Characters counter</div><div align="center" class="titleText">Report was generated:<br/> """
20 reportFile.write(html + str(currentTime) + """</div> """
21 indir = 'output/'
22 firstFile = False
23 chars = []
24 count = []
25 allCount = []
26 reportFile.write("""<div class="div_table"><table class="reportTable">""")
27 reportFile.write("""<tr><td></td></tr>""")
28 for files in os.walk(indir):
29     for f in files[2]:
30         link = "output\\"
31         link += f
32         reportFile.write("""<td>"""+f+"""/td>""")
33         if firstFile == False:
34             out = open(link,'r')
35             for linesF in out:
36                 chars.append(linesF[:1])
37                 firstFile = True
38             out.close()
39             out = open(link,'r')
40             for lines in out:
41                 count.append((str(lines[6:])).strip("\n"))
42             allCount.append(count)
43             count = []
44             out.close()
45 reportFile.write("""</tr>""")
46 try:
47     chars[0]="Space"
48 except IndexError:
49     print("Maybe there is no files?")
50 for x in range(0,len(chars)):
51     reportFile.write("""<tr><td>"""+chars[x]+"""/td>""")
52     for y in range(0,len(allCount)):
53         reportFile.write("""<td>"""+allCount[y][x]+"""/td>""")
54     reportFile.write("""</tr>""")
55 reportFile.write("""</table></div></body></html>""")
56 reportFile.close()
```

Rysunek 4: Kod Python

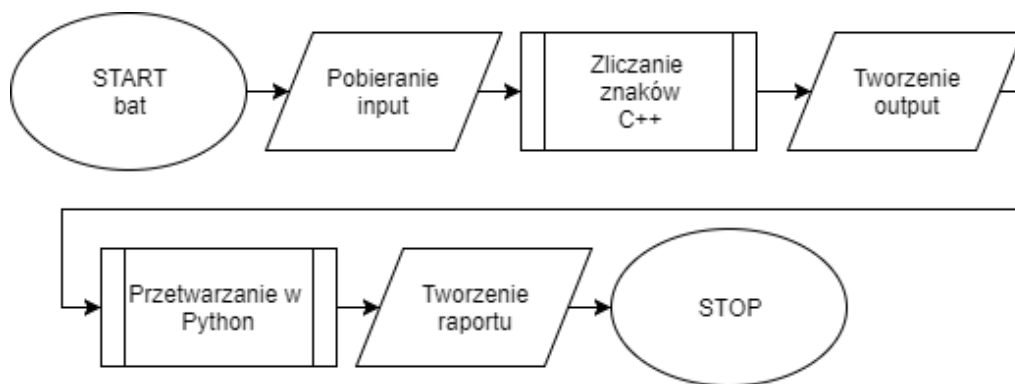
4 Działanie

4.1 Schemat serwera



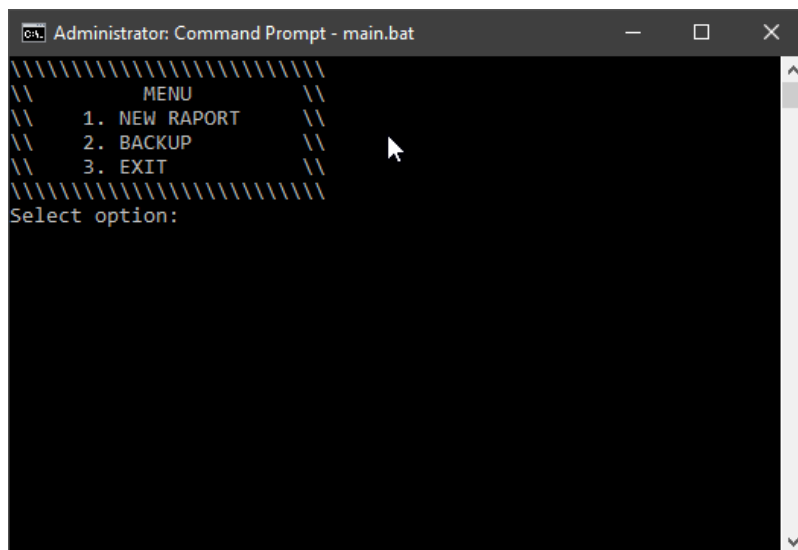
Rysunek 5: Foldery

W folderze input znajdują się pliki wejściowe, czyli pliki, w których będą zliczane znaki. W folderze output znajdują się wyniki obliczeń. Folder backup zawiera kopie zapasowe raportów, a w folderze report znajduje się ostatni wykonany raport.

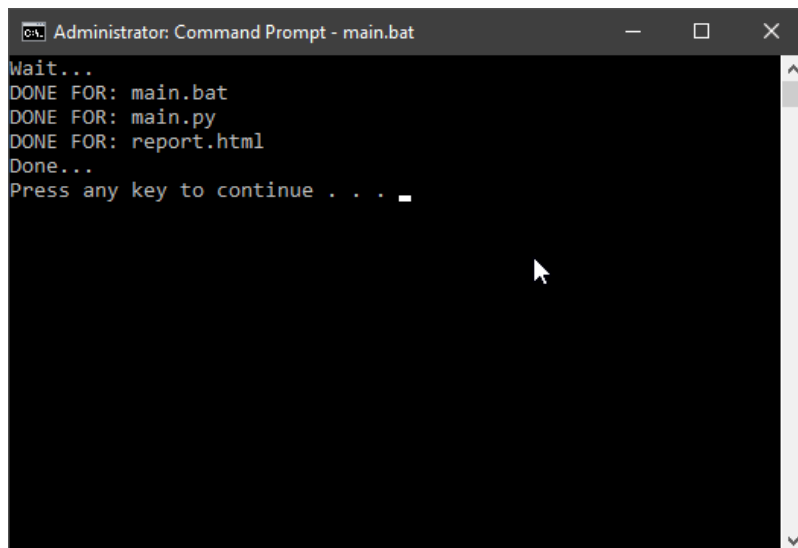


Rysunek 6: Flowchart

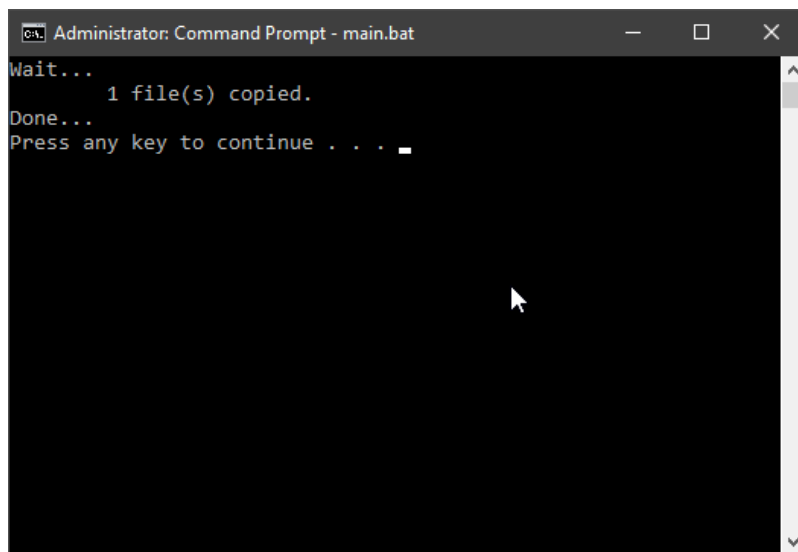
4.2 Prezentacja wyników



Rysunek 7: Menu



Rysunek 8: Nowy raport



Rysunek 9: Kopia zapasowa

Characters counter

Report was generated:
2018-01-03 19:35:00

	main.bat	main.py	report.html
Space	134	418	165
!	0	0	1
"	10	100	13
#	0	6	7
\$	0	0	1
%	28	8	3
&	0	0	1
'	2	16	7
(2	30	1
)	2	30	1
*	0	0	1
+	0	9	1
,	5	9	5
-	3	17	18
.	19	30	12
/	3	18	395
0	3	10	76
1	3	3	62
2	6	5	42
3	3	13	52
4	2	4	17
5	0	6	21
6	0	1	13
7	1	3	15

Rysunek 10: Przykładowy raport

5 Testy

W ramach testów, były usuwane foldery, pliki wejściowe. Przeprowadzone testy, pozwoliły na zabezpieczenie projektu przed problemami z brakującymi plikami. Gdy użytkownik usunie jakiś folder, zostanie on utworzony podczas działania skryptu.

6 Podsumowanie

6.1 O projekcie

Projekt został wykonany z wykorzystaniem:

- Windows 10
- Python 3.5
- C++ (Microsoft Visual Studio 2017 Enterprise)
- HTML + CSS

6.2 Przyszłość

W przyszłości projekt może zostać rozwinięty o obsługę innych znaków, dodatkowe statystyki, wyszukiwarkę w raporcie.