Licznik Znaków

Artur Bednarczyk

Styczeń 2018

Spis treści

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.

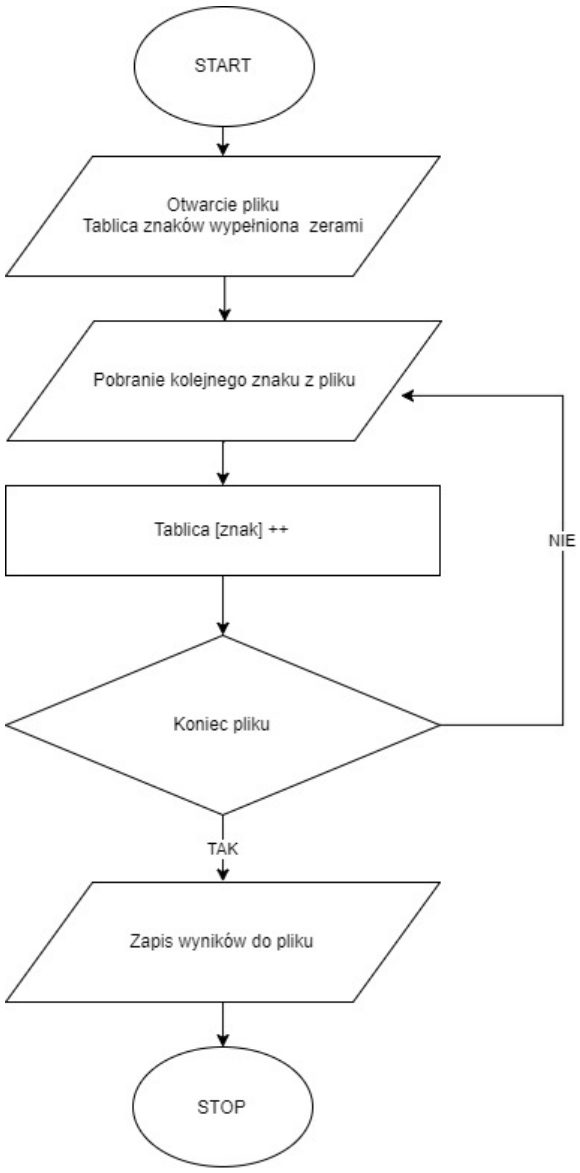
2 Model Matematyczny Aplikacji

3 Algorytm

3.1 Opis

Po uruchomieniu skryptu systemowego ukazuje się menu, którego opcjami są: generowanie nowego raportu, kopia zapasowa istniejącego raportu, wyśjcie. Po wybraniu opcji "wyjście" skrypt kończy pracę i nic wiecej się nie dzieje. Po wybraniu opcji "kopia zapasowa" skrypt sprawdza czy isnieje 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 sprawdzanie czy istnieje. Po wybraniu opcji opcji "utwórz nowy raport", dla każdego pliku wejściowego wykona się program zliczający i na wyjściu da plik z liczbą znaków. Na podstawie tych plików wyjściowych Python generuje raport.

3.2 Schemat blokowy



Rysunek 1: Schemat Blokowy

4 Implementacja

4.1 Batch

```
@echo off
:menu
cls
echo \\\\\\\\\\\\\\\\\\\\\\\\\\\
echo \\         MENU          \\
echo \\    1. NEW RAPORT      \\
echo \\    2. BACKUP          \\
echo \\    3. EXIT            \\
echo \\\\\\\\\\\\\\\\\\\\\\\\\\\
set /p in="Select option: "
if %in%==1 goto newRaport
if %in%==2 goto backup
if %in%==3 goto end
goto menu
:newRaport
cls
echo Wait...
if not exist input mkdir input
if not exist output mkdir output
if not exist report mkdir report
for /f %%x in ('dir /b input\') do ( charCounter.exe %%x
if errorlevel 0 echo DONE FOR: %%x
)
main.py
echo Done...
pause
goto menu
:backup
cls
echo Wait...
if not exist backup mkdir backup
set backupName=backup_%date:~10,4%-%date:~7,2%-%date:~4,2%_%time:~0,2%-%time:~3,2%.html
if exist "backup\%backupName%" echo Already exists
if not exist "backup\%backupName%" copy "report\report.html" "backup\%backupName%"
echo Done...
pause
goto menu
:end
```

Rysunek 2: Kod Batch

4.2 C++

```h
#pragma once
#include <fstream>
#include <string>
class counter
{
private:
	int table[127];
	int size;
	char input;
	char* source;

	void count();
	void saveToFile();
public:
	counter(char* source);
	void operate();
	~counter();
};
```

(a) .h

```cpp
#include "stdafx.h"
#include "counter.h"
#include <fstream>
#include <string>
counter::counter(char* source)
{
	this->source = source;
	for (int i = 0; i <= 127; i++) {
		table[i] = 0;
	}
}
counter::~counter()
{
	delete table;
	delete source;
}
void counter::count()
{
	std::string fileSource = "input/";
	fileSource += source;
	std::ifstream file(fileSource, std::ios::in);
	if (file.is_open()) {
		while (file.good()) {
			input = file.get();
			table[(int)input]++;
		}
	}
	file.close();
}
void counter::saveToFile()
{
	std::string output="output/";
	output.append(source);

	std::ofstream outFile(output, std::ios::out);
	for (int i = 32; i < 127; i++) {
		outFile << (char)i << " --> " << table[i] << std::endl;
	}
}
void counter::operate()
{
	count();
	saveToFile();
}
```

(b) .cpp

Rysunek 3: Kod C++

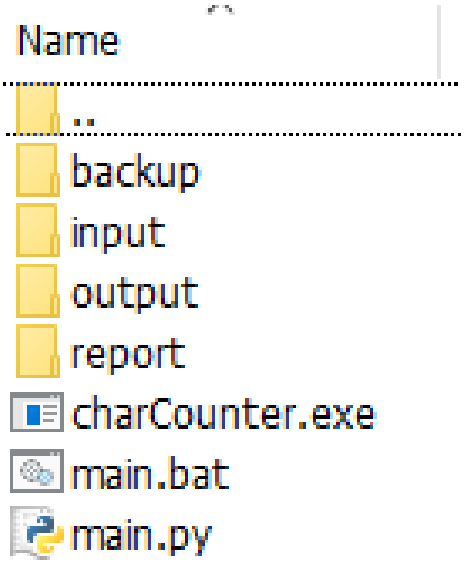
4.3 Python

```python
import datetime
import os
currentTime = datetime.datetime.now().strftime("%Y-%m-%d %H:%M:%S")
reportFile = open('report/report.html', 'w')
html = """<html>
<head><title>Report</title>
<style>
body{background:#252525;color:#ACAC11; font-family: 'Trocchi', serif;font-weight: bold;}
.title{color: khaki;font-family: 'Trocchi', serif;font-size: 48px;font-weight: normal;line-height: 48px;margin: auto;}
.div_table{height:75%;width:75%;margin:auto;overflow:auto;}
.titleText{color: #c0c0c0;font-family: 'Trocchi', serif;font-size: 32px;font-weight: normal;line-height: 48px;margin: auto;}
.reportTable{margin: auto;background: #333333;border: solid khaki;text-align:center;overflow:auto;}
.reportTable td{border: solid black thin;width: 75px;}
table.reportTable > tbody > tr:first-child > td:first-child{background: grey;width:40px;}
table.reportTable > tbody > tr > td:first-child, table.reportTable > tbody > tr:first-child{background: green;color:#d0d0d0;}
table.reportTable td:hover{background: silver;color: #333333;}
</style>
</head>
<body><p align="center" class="title">Characters counter</p><p align="center" class="titleText">Report was generated:<br/> """
reportFile.write(html + str(currentTime) + """</p> """)
indir = 'output/'
firstFile = False
chars = []
count = []
allCount = []
reportFile.write("""<div class="div_table"><table class="reportTable">""")
reportFile.write("""<tr><td></td>""")
for files in os.walk(indir):
    for f in files[2]:
        link = "output\\"
        link += f
        reportFile.write("""<td>"""+f+"""</td>""")
        if firstFile == False:
            out = open(link,'r')
            for linesF in out:
                chars.append(linesF[:1])
                firstFile = True
            out.close()
        out = open(link,'r')
        for lines in out:
            count.append((str(lines[6:])).strip("\n"))
        allCount.append(count)
        count = []
        out.close()
reportFile.write("""</tr>""")
try:
    chars[0]="Space"
except IndexError:
    print("Maybe there is no files?")
for x in range(0,len(chars)):
    reportFile.write("""<tr><td>"""+chars[x]+"""</td>""")
    for y in range(0,len(allCount)):
        reportFile.write("""<td>"""+allCount[y][x]+"""</td>""")
    reportFile.write("""</tr>""")
reportFile.write("""</table></div></body></html>""")
reportFile.close()
```

Rysunek 4: Kod Python

5 Działanie

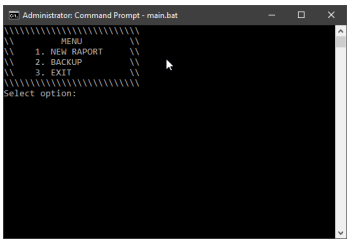
5.1 Schemat serwera



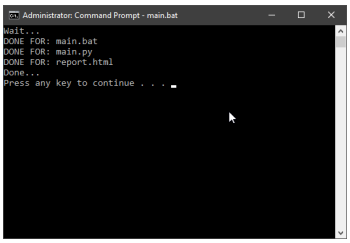
Rysunek 5: Foldery

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.

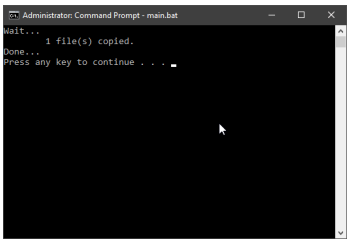
5.2 Prezentacja wyników



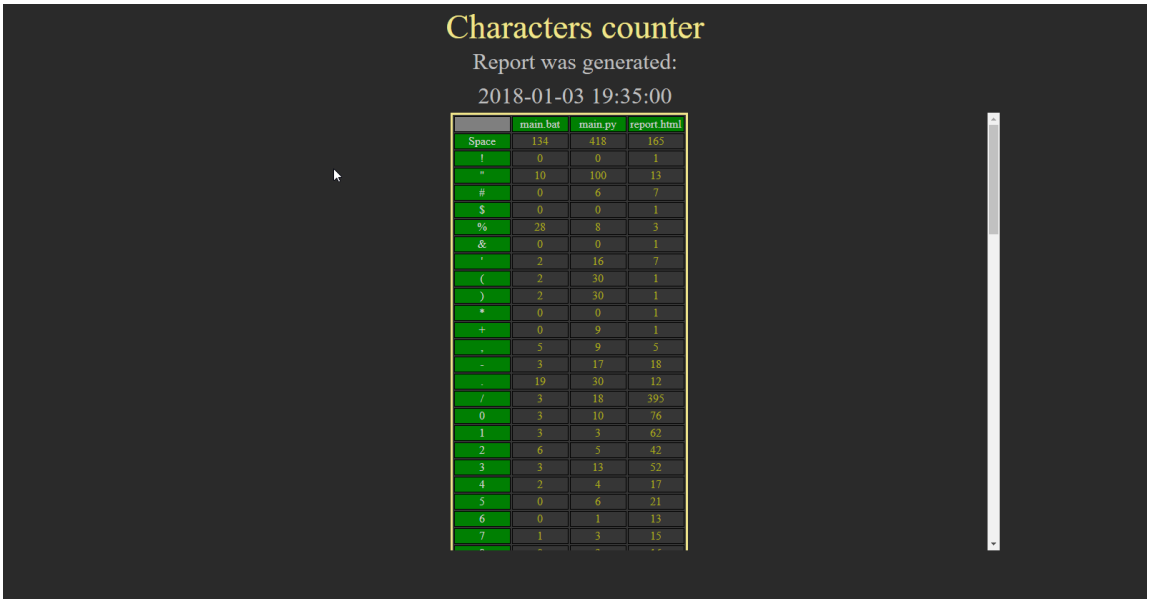
(a) Menu



(b) Nowy raport



(c) Kopia zapasowa



Rysunek 7: Przykładowy raport

6 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.

7 Podsumowanie

7.1 O projekcie

Projekt został wykonany z wykorzystaniem:

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

7.2 Przyszłość

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