

# Języki Skryptowe

dokumentacja projektu zadania maturalnego z informatyki 2021

Kamil Grabowski, grupa 3/5

14 stycznia 2023

# Część I

## Opis programu

W folderze input każdy z plików zawiera jedną instrukcję na jedną linię takie jak:

- DOPISZ *litera* - oznacza, że na końcu napisu trzeba dopisać pojedynczą literę
- ZMIEN *litera* - oznacza, że ostatnią literę aktualnego napisu należy zmienić na podaną literę
- USUN 1 - oznacza, że należy usunąć ostatnią literę aktualnego napisu
- PRZESUN *litera* - oznacza, że pierwsze od lewej wystąpienie podanej litery w napisie należy zamienić na następną literę w alfabecie (jeśli litera to A, to należy zamienić na B, jeśli B, to na C itd.) Literę Z należy zamienić na A. Jeśli litera nie występuje w napisie, nie należy nic robić

## Zadania do wykonania:

1. Obliczyć całkowitą długość napisu po wykonaniu wszystkich instrukcji
2. Znaleźć najdłuższy ciąg występujących kolejno po sobie instrukcji tego samego rodzaju
3. Obliczyć, która litera jest najczęściej dopisywana
4. Podać napis, który powstanie po wykonaniu wszystkich instrukcji

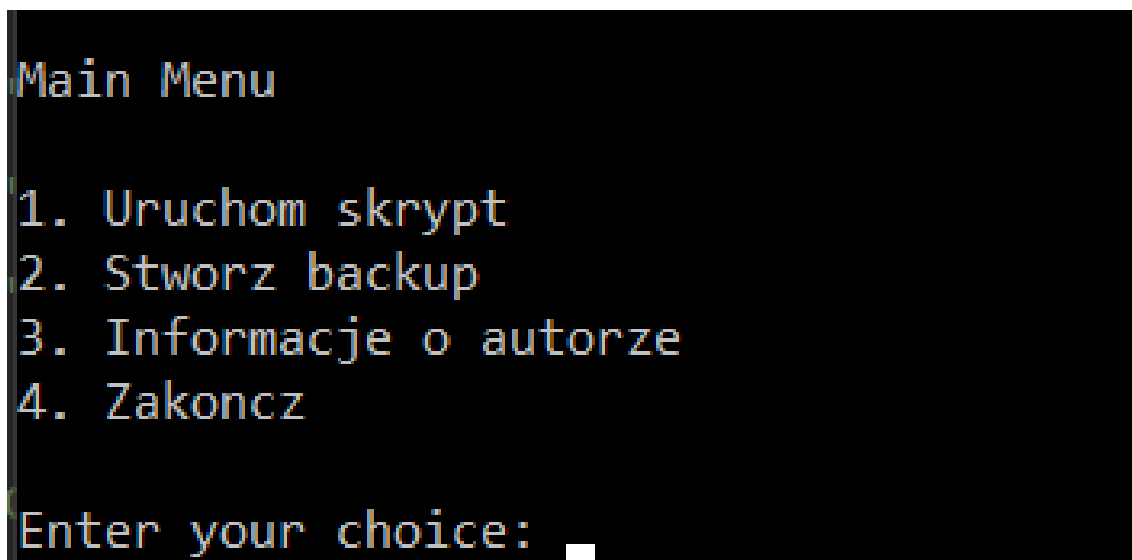
Dany jest następujący ciąg instrukcji:

- DOPISZ A
- DOPISZ B
- DOPISZ C
- USUN 1
- DOPISZ D
- ZMIEN B
- DOPISZ E
- PRZESUN B

Po wykonaniu pierwszych trzech instrukcji napis będzie miał postać ABC, potem AB, ABD, ABB, ABBE, wreszcie ostatnia instrukcja zamieni pierwsze B na C, więc ostatecznie powstały napis to ACBE.

## Instrukcja obsługi

Aby uruchomić program należy włączyć skrypt menu.bat otwierający menu obsługi naszego programu. Po uruchomieniu wyświetli nam się tekst z instrukcją obsługi programu, wymagający podania przez użytkownika liczby (1-4) w celu wykonania odpowiadającej mu funkcji.



Rysunek 1: Główne menu programu

Możliwe wybory są następujące:

1. Uruchom skrypt - uruchamia skrypt python, który pobiera z każdego pliku w input dane i przetwarza na odpowiedzi. Odpowiedzi zapisuje w folderze output. Następnie tworzy raport.html

Calkowita dlugosc napisu	14	517	7	10	7
Najdluzszy ciag wystepujacy po sobie	DOPISZ 5	ZMIEN 6	DOPISZ 5	DOPISZ 8	DOPISZ 3
Litera najczesciej dopisywana	D	Z	D	D	A
Koncowy napis	ABDDXXDDABDDXDD	POZNIEJMONIONOZECZLO WIEKTENNADSZEDLODPOL	ABDDXDD	ABDDXCDXDD	BCCFXZZ
Input	<div> DOPISZ A DOPISZ B DOPISZ C DOPISZ D DOPISZ X PRZESUN C ZAMIEŃ X DOPISZ D DOPISZ W DOPISZ R USUN R ZMIEN D DOPISZ A DOPISZ B DOPISZ C DOPISZ D DOPISZ X PRZESUN C ZAMIEŃ X </div>	<div> DOPISZ F PRZESUN F ZMIEN V DOPISZ M ZMIEN Z DOPISZ E ZMIEN B ZMIEN I PRZESUN Z ZMIEN C DOPISZ H PRZESUN H PRZESUN C DOPISZ J ZMIEN J DOPISZ H DOPISZ O DOPISZ O </div>	<div> DOPISZ A DOPISZ B DOPISZ C DOPISZ D DOPISZ X PRZESUN C ZAMIEŃ X DOPISZ D DOPISZ W DOPISZ R USUN R ZMIEN D </div>	<div> DOPISZ A DOPISZ B DOPISZ C DOPISZ D DOPISZ X DOPISZ D DOPISZ X PRZESUN C ZAMIEŃ X DOPISZ D DOPISZ W DOPISZ R USUN R ZMIEN D </div>	<div> DOPISZ A DOPISZ B DOPISZ C ZAMIEŃ C PRZESUN B PRZESUN A DOPISZ F DOPISZ D USUN D DOPISZ X DOPISZ Y PRZESUN Y DOPISZ Z PRZESUN Z </div>
Nazwa pliku	data.txt	instrukcje.txt	pierwsze.txt	problemy.txt	raport.txt

Rysunek 2: Przykładowy raport programu

2. Stwórz Backup - Tworzy kopię zapasową danych w katalogu backup zawartość folderów input i output.

```
Enter your choice: 2
input\data.txt
input\instrukcje.txt
input\pierwsze.txt
input\problemy.txt
input\raport.txt
      5 file(s) copied.
output\09.01.2023  data.txt
output\09.01.2023  instrukcje.txt
output\09.01.2023  pierwsze.txt
output\09.01.2023  problemy.txt
output\09.01.2023  raport.txt
      5 file(s) copied.
```

Rysunek 3: Informacje o programie

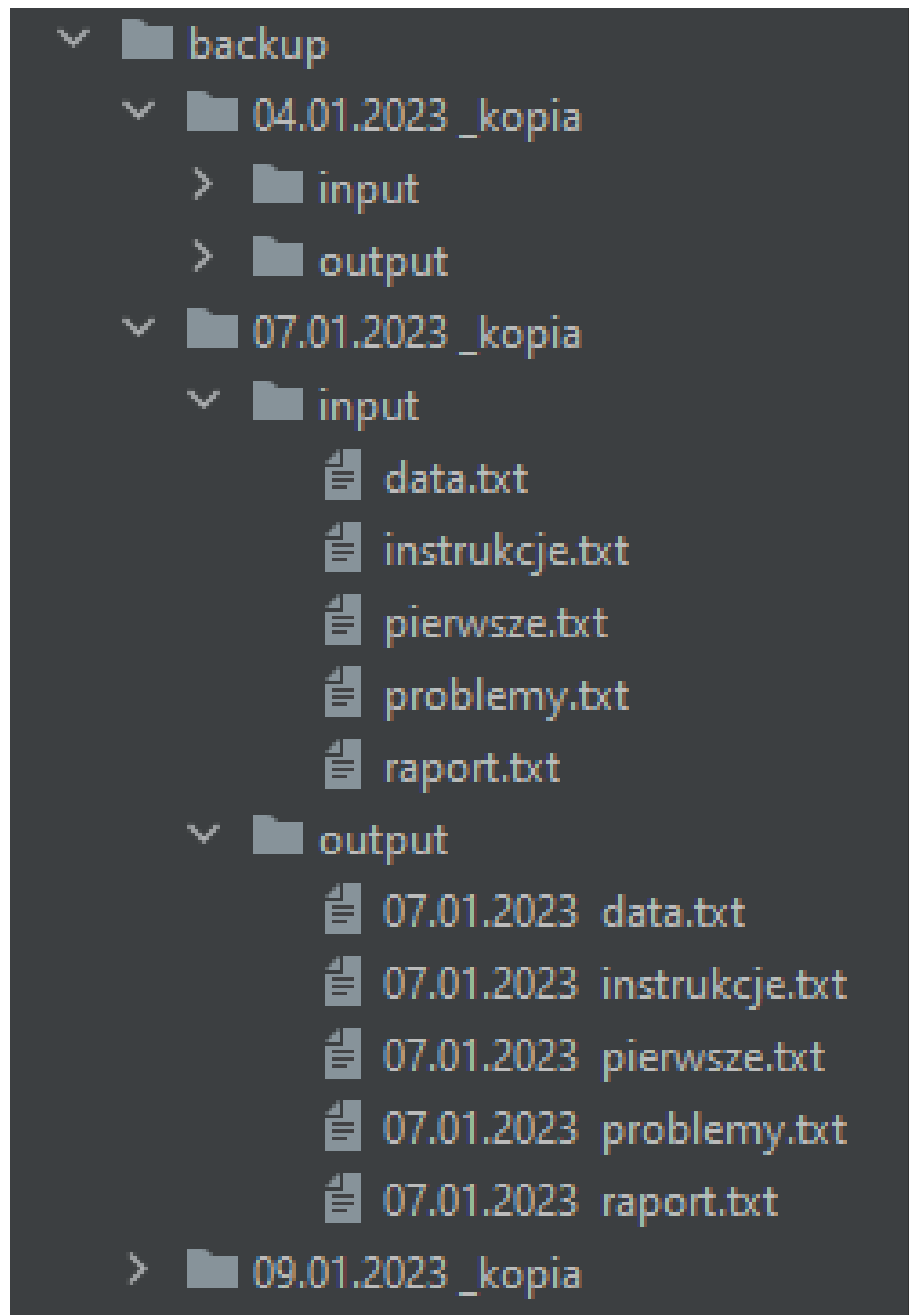
3. Informacje o autorze - krótki opis projektu oraz jego autora

```
Enter your choice: 3
Projekt stworzony przez Kamil Grabowskiego studenta Politechniki Slaskiej na wydziale Matema
tyki Stosowanej na kierunku informatyka, sem. III gr 3.
Program wykonuje operacje dla tekstu jak
DOPISZ (dopisuje litere),
USUN (usuwa litere),
PRZESUN (przesuwa litere na kolejna w alfabecie),
ZAMIEN (zamienia litere na inna)
```

Rysunek 4: Komunikat o dokonanym backupie

4. Zakoncz - kończy program.

Podanie innej liczby lub znaku, skutkuje powiadomieniem o wprowadzeniu niepoprawnego polecenia

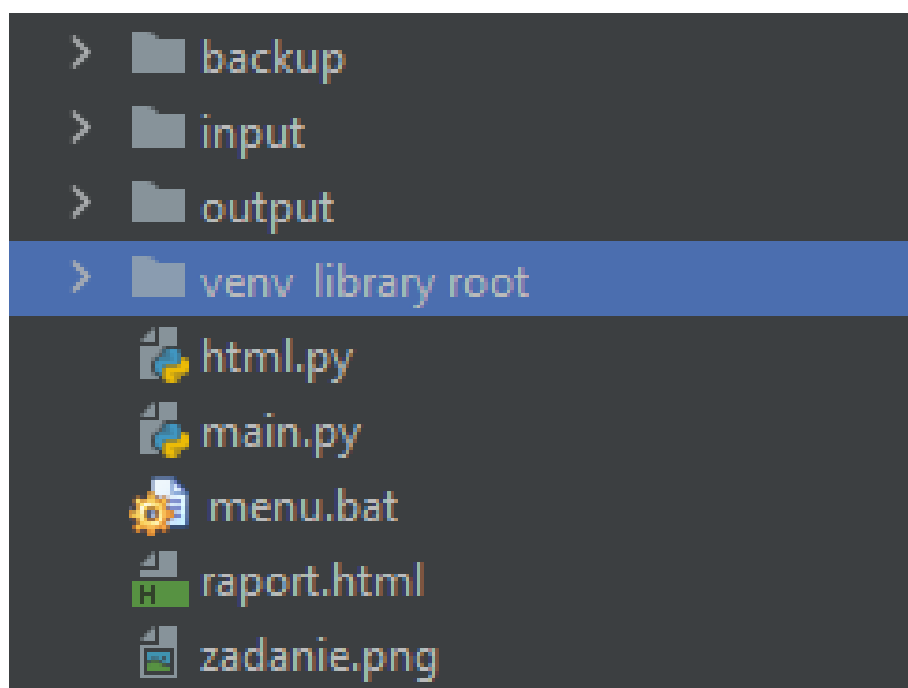


Rysunek 5: struktura zapisu backupu

## Struktura danych programu

Program składa się z następującej struktury danych, wymaganych do prawidłowego uruchomienia aplikacji:

- **menu.bat** - Skrypt batch będący menu, którym uruchamia program.
- **main.py** - Skrypt python zawierający główny program, pobierający pliki wejściowe, tworzący plik wyjścia.
- **html.py** - Skrypt python pobierający dane z plików wejścia oraz wyjścia i generujący plik raport.html zawierający raport wszystkich danych w postaci tabeli
- **input** => Każdy plik w folderze zawiera instrukcje
- **output** - folder do zapisu outputu z inputa
- **raport.html** - plik html zawierający raport



Rysunek 6: Struktura danych programu w formie drzewa

## Część II

### Opis działania

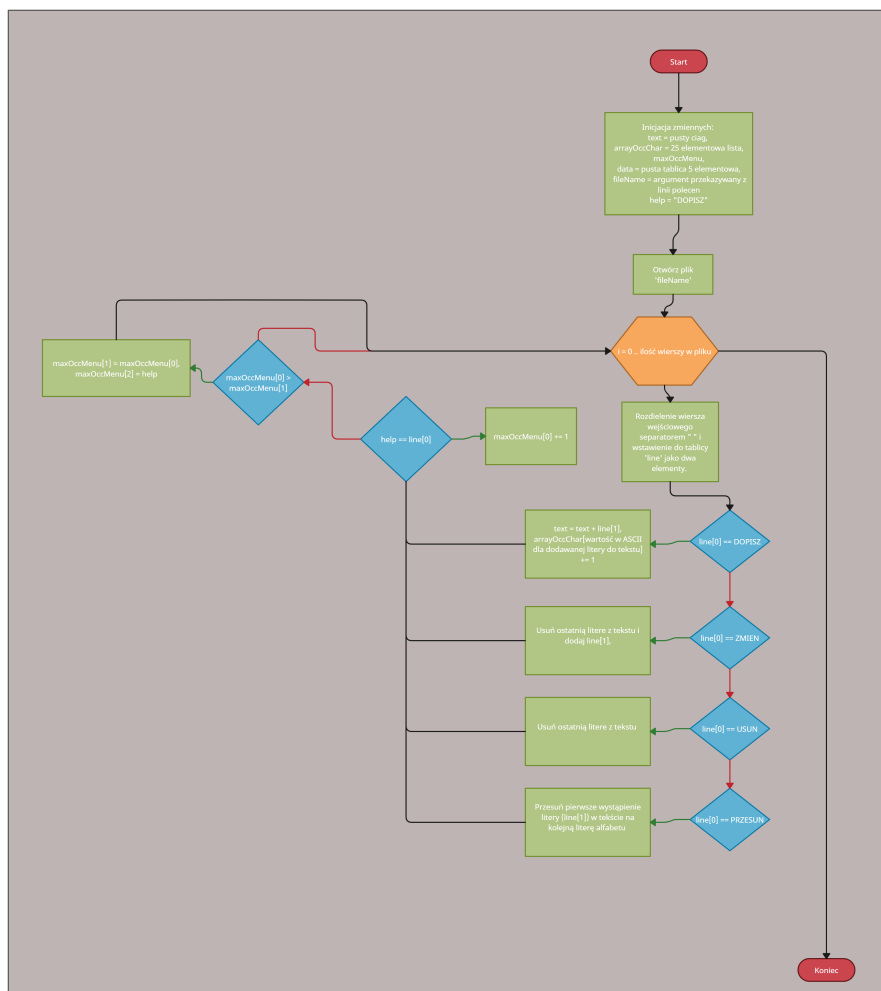
Skrypt *menu.bat* wyświetla menu z którego następnie możemy uruchomić skrypt *main.py*. Ów skrypt pobiera dla każdego pliku z input dane i przetwarza je na odpowiedzi. Następnie skrypt batch'owy wysyła odpowiedzi do pliku oraz do folderu output. Po poprawnym zapisanie do folderu output uruchamia się skrypt *html.py*, który pobiera dane z folderu input oraz output i przetwarza na tabelkę w html.

## Algorytm

1. Inicjalizuj zmienne:
  - I. *text* - jako pusty ciąg znaków
  - II. *arrayOccChar* - jako 25-elementową listę pustą
  - III. *maxOccMenu* - jako listę zawierającą 0, 0 i pusty ciąg znaków
  - IV. *data* - jako pustą listę
  - V. *fileName* - jako pierwszy argument przekazywany z linii poleceń
2. Otwórz plik określony przez *fileName* w katalogu "input"
3. Przejdź przez każdy wiersz pliku
  - I. Podziel wiersz na listę słów używając ' ' jako separatora.
  - II. Sprawdź pierwsze słowo na liście:
    - (a) Jeśli jest to "DOPISZ", dopisz do końca zmiennej *text* drugie słowo i zwiększ odpowiedni element w liście *arrayOccChar* o 1
    - (b) Jeśli jest to "ZMIEN", zamień ostatni znak zmiennej *text* na drugie słowo
    - (c) Jeśli jest to "USUN", usuń ostatni znak zmiennej *text*
    - (d) Jeśli jest to "PRZESUN", zamień pierwsze wystąpienie drugiego słowa w zmiennej *text* na następny znak w alfabecie.
  - III. Zaktualizuj zmienną *maxOccMenu*:
    - (a) Jeśli obecne słowo jest takie same jak poprzednie, zwiększ pierwszy element *maxOccMenu* o 1
    - (b) Jeśli obecne słowo jest różne od poprzedniego, sprawdź czy pierwszy element *maxOccMenu* jest większy od drugiego. Jeśli tak, zaktualizuj drugi element na taki sam jak pierwszy i zaktualizuj trzeci element na poprzednie słowo. Ustaw pierwszy element na 0 i zaktualizuj poprzednie słowo na obecne
  - IV. Zamknij plik.



## Schemat blokowy



Rysunek 7: Schemat blokowy algorytmu

sys

```
1 input = str(sys.argv[1])
2 //Pobiera pierwszy argument podany podczas uruchamiania skryptu python
3
4 sys.exit("Nie znaleziono pliku slownikowego")
5 //Wyswietla w konsoli podany komunikat i przerywa dzialanie skryptu
  python
```

---

## Funkcje zawarte w main.py

```
1 def zad1(text):
2     return len(text)
3 //Funkcja zwracajaca dlugosc napisu po wszystkich operacjach
```

---

```
1 def zad2(text):
2     return str(text[2]) + " " + str(text[1])
3 //Funkcja zwracajaca najdluzszy ciag wystepujacych kolejno po sobie
```

---

```
1 def zad3(arrayOccChar):
2     return chr(arrayOccChar.index(max(arrayOccChar)+65))
3 //Funkcja zwracajaca najczesciej dopisywana litere
```

---

```
1 def zad4(text):
2     return text
3 //Funkcja zwracajaca finalny napis
```

---

# Pełen kod aplikacji

## main.py

```
1 import sys
2
3 text = ""
4 arrayOccChar = [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
    0, 0, 0, 0, 0, 0, 0] #25 elementów
5 maxOccMenu = [0,0,""] #[0] - ilość wystąpień w danym momencie, [1] -
    ilość maksymalna wystąpień, [2] - co występuje najwięcej razy
6 data = []
7 fileName = sys.argv[1]
8
9 help = "DOPISZ" #pomocna zmienna do pierwszego wyszukiwania. Pierwsza
    instrukcja zawsze powinna być DOPISZ
10 with open("input/" + fileName) as file: #zczytywanie z pliku w danych i
    przetwarzanie
11     for x in file:
12         line = x.strip().split(' ')
13         if(line[0] == 'DOPISZ'): #wstawienie nowej litery na koniec
            tekstu
14             text = text + line[1]
15             arrayOccChar[ord(line[1])-65] += 1
16         elif(line[0] == "ZMIEN"):
17             text = text[:-1] + line[1] #Usunięcie ostatniej litery z
            tekstu i wstawienie nowej
18         elif(line[0] == "USUN"): #usunięcie ostatniej litery
19             text = text[:-1]
20         elif(line[0] == "PRZESUN"): #przesunięcie pierwszej litery w
            tekście o kolejną literę w alfabecie
21             ascii = (ord(line[1])-65)
22             if(ascii == 25):
23                 replaceChar = 'A'
24             else:
25                 replaceChar = chr(ascii+66)
26             text = text.replace(line[1], replaceChar, 1) #zamiana z kodem
                ASCII na literę
27
28         if (help == line[0]): #Do obliczania najczęściej występującej
            instrukcji
29             maxOccMenu[0] += 1
30         else:
31             if(maxOccMenu[0] > maxOccMenu[1]):
32                 maxOccMenu[1] = maxOccMenu[0]
33                 maxOccMenu[2] = help
34             maxOccMenu[0] = 0
35             help = line[0]
36
37 def zad1(text): # długość napisu po wykonaniu wszystkich instrukcji
38     return len(text)
39
40 def zad2(a): # najdłuższy ciąg występujących kolejno po sobie
41     return str(a[2]) + " " + str(a[1])
```

```

42
43 def zad3(arrayOccChar): # litera jest najczesciej dopisywana
44     return chr(arrayOccChar.index(max(arrayOccChar))+65)
45
46 def zad4(text): # finalny napis
47     return text
48
49 data.extend([zad1(text), zad2(maxOccMenu), zad3(arrayOccChar), zad4(text)])
50
51 print(zad1(text))
52 print(zad2(maxOccMenu))
53 print(zad3(arrayOccChar))
54 print(zad4(text))
55 print(fileName)

```

---

## html.py

```

1 import os
2 from jinja2 import Template #biblioteka do tworzenia stron w python
3
4 data = []
5 textLines = []
6 folder = "output"
7 count = 0
8 for filename in os.listdir(folder): #odczytuje plik w danym folderze
9     count += 1
10    with open("output/" + filename) as file: #pobiera dane z pliku
11        output
12        for l in file:
13            line = l.strip()
14            data.append(line) #wstawia do tablicy
15            print(line)
16    with open("input/" + data[len(data)-1], 'r') as f:
17        text = f.read()
18        lines = text.strip()
19        textLines.append(lines)
20
21 print(data)
22 print(count)
23 inputText = ""
24 #kod html z petlami wbudowanymi do automatycznego tworzenia raportu
25 html_template = """
26 <html>
27     <head>
28         <title>Raport</title>
29         <meta charset="UTF-8">
30         <style>
31             table,th,td{
32                 border: 1px solid black;
33                 justify-content: center;
34                 text-align:center;
35                 font-size: 21px;

```

```

36         table{
37         }
38         th{
39         }
40         td{
41             overflow: hidden;
42         }
43         .inou{
44             display: block;
45         }
46     </style>
47 </head>
48 <body>
49     <table>
50         <tr>
51             <td>Calkowita dlugosc napisu</td>
52             {% for item in range(0,data|length, 5) %}
53             <td>{{ data[item] }}</td>
54             {% endfor %}
55         </tr>
56         <tr>
57             <td>Najdluzszy ciag wystepujacy po sobie</td>
58             {% for item in range(1,data|length, 5) %}
59             <td>{{ data[item] }}</td>
60             {% endfor %}
61         </tr>
62         <tr>
63             <td>Litera najczesciej dopisywana</td>
64             {% for item in range(2,data|length, 5) %}
65             <td>{{ data[item] }}</td>
66             {% endfor %}
67         </tr>
68         <tr>
69             <td>Koncowy napis</td>
70             {% for item in range(3, data|length, 5) %}
71             <td><textarea>{{ data[item] }}</textarea></td>
72             {% endfor %}
73         </tr>
74         <tr>
75             <td>Input</td>
76             {% for item in range(textLines|length) %}
77             <td>
78                 <textarea style="width: 120px;height: 300px;
79                     resize: none">
80                     {{ textLines[item] }}
81                 </textarea>
82             </td>
83             {% endfor %}
84         </tr>
85         <td>Nazwa pliku</td>
86         {% for item in range(4,data|length, 5) %}
87         <td>{{ data[item] }}</td>
88         {% endfor %}
89     </table>
</body>

```

```

90
91 </html>
92 """
93 # template = Template(html_template)
94
95
96 # Tworzenie szablonu za pomoc jinja
97 template = Template(html_template)
98
99 # Wyplacenie szablonu danymi z listy
100 html = template.render(data=data, inputText=inputText, textLines=
    textLines)
101
102 # Zapisywanie plik HTML
103 with open('raport.html', 'w') as f:
104     f.write(html)

```

---

## menu.bat

```

1 @echo off
2 //Zapisywanie do zmiennej aktualnego czasu
3 for /f "tokens=1-3 delims=" %%a in ('date /t') do set dataHelp=%%c%%a%%b
4 set current_time=%dataHelp%
5 echo %current_time%
6 del /F /Q output\*
7
8 :menu
9 echo .
10 echo Main Menu
11 echo .
12 echo 1. Uruchom skrypt
13 echo 2. Stworz backup
14 echo 3. Informacje o autorze
15 echo 4. Zakoncz
16 echo .
17 set /p choice=Enter your choice: //Wybor opcji
18
19 if "%choice%"=="1" goto option1
20 if "%choice%"=="2" goto option2
21 if "%choice%"=="3" goto option3
22 if "%choice%"=="4" goto option4
23
24 echo Invalid choice.
25 goto menu
26
27 :option1 //przetwarzanie danych dla kazdego pliku w input i zapisywanie
    do output
28 for /f "delims=" %%a in ('dir /b /a-d "input"') do (
29     python3 main.py %%a > "output\%current_time% %%a
30 )
31 goto menu
32
33 :option2 //Tworzenie kopii zapasowej

```

```
34 md "backup\%current_time%_kopia"
35 md "backup\%current_time%_kopia\input" "backup\%current_time%_kopia\
  output"
36 copy "input\*" "backup\%current_time%_kopia\input"
37 copy "output\*" "backup\%current_time%_kopia\output"
38 goto menu
39
40 :option3 //Informacje o autorze
41 echo Projekt stworzony przez Kamil Grabowskiego studenta Politechniki
  Slaskiej na wydziale Matematyki Stosowanej na kierunku informatyka,
  sem. III gr 3.
42 echo Program wykonuje operacje dla tekstu jak
43 echo DOPISZ (dopisuje litere),
44 echo USUN (usuwa litere),
45 echo PRZESUN (przesuwa litere na kolejna w alfabecie),
46 echo ZAMIEN (zamienia litere na inna)
47 goto menu
48
49 :option4 //Zakonczenie dzialania programu
50 goto end
51
52 :end
53 echo.
54 echo Press any key to exit.
55 pause >nul
```

---