

Języki Skryptowe

dokumentacja projektu zaliczeniowego

Kacper Moll, grupa IIIE

6 grudnia 2021

Część I

Opis programu

ZADANIE 4 - ZEGAR

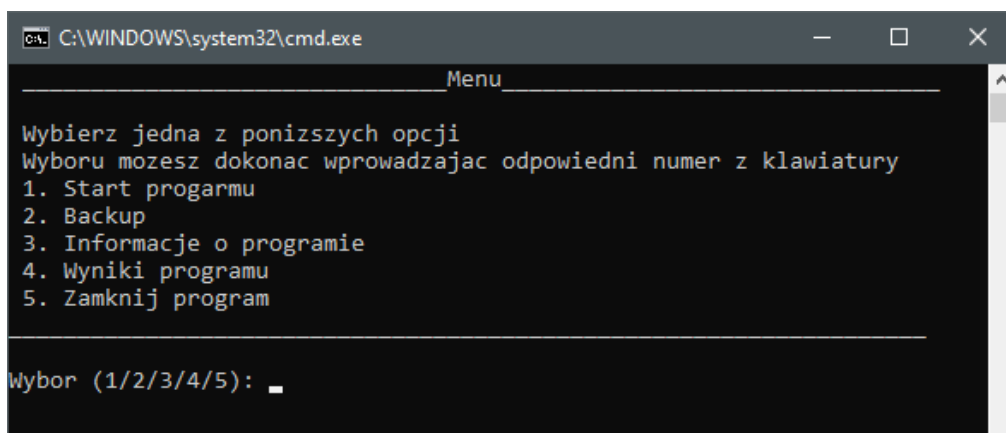
Pan Andrzej wchodząc do zakładu zegarmistrzowskiego spojrzął na zegarek, była dokładnie godzina 9.00. Na ścianie wisiały dwa zegary. Pierwszy z nich wskazywał na godzinę 8.00, a drugi 10.00. Pan Andrzej wiedział, że pierwszy zegar spieszy się 5 sekund na minutę, a drugi spóźnia o 5 sekund na minutę. Po jakimś czasie Pan Andrzej spojrzął na zegary, które wskazywały tę samą godzinę. Ile sekund upłynęło od czasu wejścia zegarmistrza do zakładu do momentu wskazania przez nie tej samej godziny. Należy napisać program obliczający czas w sekundach jaki upłynął od momentu startu zegara wskazującego godzinę 9.00 do momentu wskazania przez wszystkie zegary tej samej godziny. Wynik należy zapisać w pliku. Linia jest zakończona znakiem nowej linii.

Mała modyfikacja w celu rozbudowania programu:

Program został napisany w taki sposób, aby obsługiwał dowolne zegary (dowolna godzina oraz dowolne opóźnienie). Dodatkowo dane na temat zegarów są pobierane z pliku. Dzięki tym aktualizacją program stał się uniwersalny.

Instrukcja obsługi

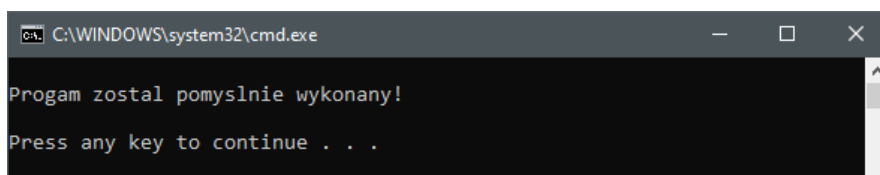
Obsługa programu odbywa się z poziomu wiersza poleceń, należy uruchomić plik „menu.bat”. W nim są zawarte wszystkie funkcje możliwe do wykonania w tym programie.



Rysunek 1: Okno startowe programu „menu.bat”, stąd użytkownik ma dostęp do wszystkich funkcjonalności. Program automatycznie wraca do okna startowego po wykonaniu każdej z nich.

1. Start programu

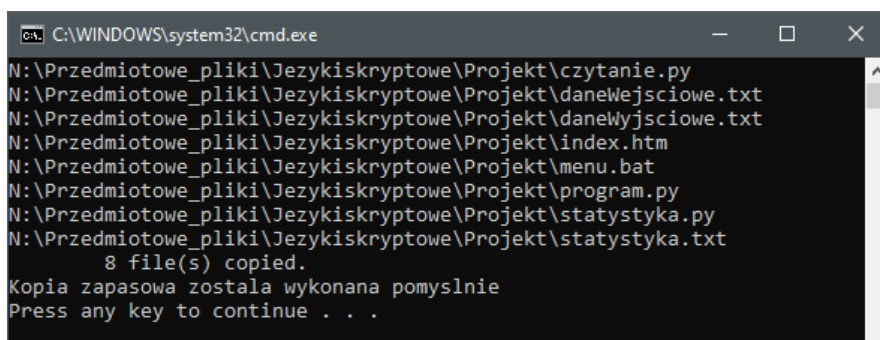
Zostaje uruchomiony plik „program.py” napisany w języku Python. Dokładne działanie tego programu opisane jest w dalszej części dokumentacji.



Rysunek 2: Okno po wybraniu numeru 1 z klawiatury wyświetla komunikat, który informuje użytkownika o pomyślnym wykonaniu pliku „program.py”

2. Backup

Tworzenie kopii zapasowej wszystkich plików projektu. Jako nazwę folderu z plikami zapasowymi program przyjmuje aktualną datę w formacie DD.MM.RRRR, a lokalizacja tego katalogu to Pulpit, aby nie było problemu ze zlokalizowaniem folderu.



Rysunek 3: Okno po wybraniu numeru 2 z klawiatury wyświetla wszystkie pliki skopiowane do folderu zapasowego.

Ten komputer > Pulpit > 02.12.2021					
Nazwa	Data modyfikacji	Typ	Rozmiar		
statystyka	01.12.2021 23:12	Python File	1 KB		
program	01.12.2021 22:31	Python File	3 KB		
czytanie	01.12.2021 21:32	Python File	2 KB		
menu	02.12.2021 00:41	Plik wsadowy Win...	3 KB		
statystyka	01.12.2021 23:12	Dokument tekstowy	1 KB		
daneWysciowe	02.12.2021 11:04	Dokument tekstowy	1 KB		
daneWejscowe	01.12.2021 22:30	Dokument tekstowy	1 KB		
index	01.12.2021 23:14	Brave HTML Docu...	1 KB		

Rysunek 4: Utworzonej kopii zapasowej.

3. Informacje o programie

Na ekranie konsoli zostają wyświetlone ogólne informacje na temat programu, w taki sposób, aby użytkownik był w stanie zrozumieć mechanikę działania oraz aby wiedział jak obsługiwać program.

```
C:\WINDOWS\system32\cmd.exe

Ogolne dzialanie programu:

Dzialanie programu opiera sie na trzech zegarach, posrod ktorych kazdy dziala inaczej

Wyobrazmy sobie nastepujaca sytuacje:
# pierwszy zegar spoznia o 5 sekund
# drugi zegar przyspiesza o 5 sekund
# trzeci zegar dziala poprawnie

Zadaniem tego programu jest obliczenie czasu, ktory uplynie az dane trzy zegary sie spotkaja

Dane wejscowe:
# zapisane sa w pliku daneWejscowe.txt
# format:
    "zegar1||zegar2||zegar3/n"
    "zegar1||zegar2||zegar3/n"
    itd.
Przy czym pod wartoscia zegar1/zegar2/zegar3 kryja sie informacje o danym zegarze w formacie:
# "H:M|-wartosc_opoznienia/+wartosc_przyspieszenia"
# w przypadku kiedy zegar dziala poprawnie nalezy wpisac "0"

Dla przykladu:
# "14:30|+5||16:30|-5||15:30|0"
# zegar1 przyspiesza 5 sekund
# zegar2 opoznia 5 sekund
# zegar3 dziala poprawnie

Dane wyjscowe:
# zapisane sa w pliku daneWyjscowe.txt
# wynik reprezentujacy czas, ktory minal jest podany w sekundach

Press any key to continue . . .
```

Rysunek 5: Okno po wybraniu numeru 3 z klawiatury wyświetla komunikat opisujący działanie programu

4. Wyniki programu

Resultat uruchomienia pliku „program.py” zostaje zapisany w pliku „daneWyjsciowe.txt”. Plik wsadowy systemu windows wywołuje plik „statystyka.py”, który oblicza średnią arytmetyczną wyników dla podanych zegarów. Następnie wszystkie dane są możliwe do podglądu w przeglądarce internetowej, która się automatycznie otwiera.

Zadanie 4 2011, "Zegar"

Poniżej jest zaprezentowana statystyka działania programu

Wprowadzone dane	Wynik
9:00 +10 11:00 0 10:00 0 17:01 +5 18:59 -5 18:00 0 07:35 +5 08:35 0 9:35 -5 09:00 +2 11:00 -2 10:00 0	Zegary sie nigdy nie spotkaja 42480s 43200s 108000s

Średni czas, który potrzebują wprowadzone zegary to
64560.0

Rysunek 6: Po wybraniu numeru 4 z klawiatury, plik wsadowy „menu.bat” otwiera nowe okno w domyślnej przeglądarce, gdzie ta prezentuje zawartość pliku „index.htm” zawierający statystykę oraz wyniki programu

5. Zamknij program

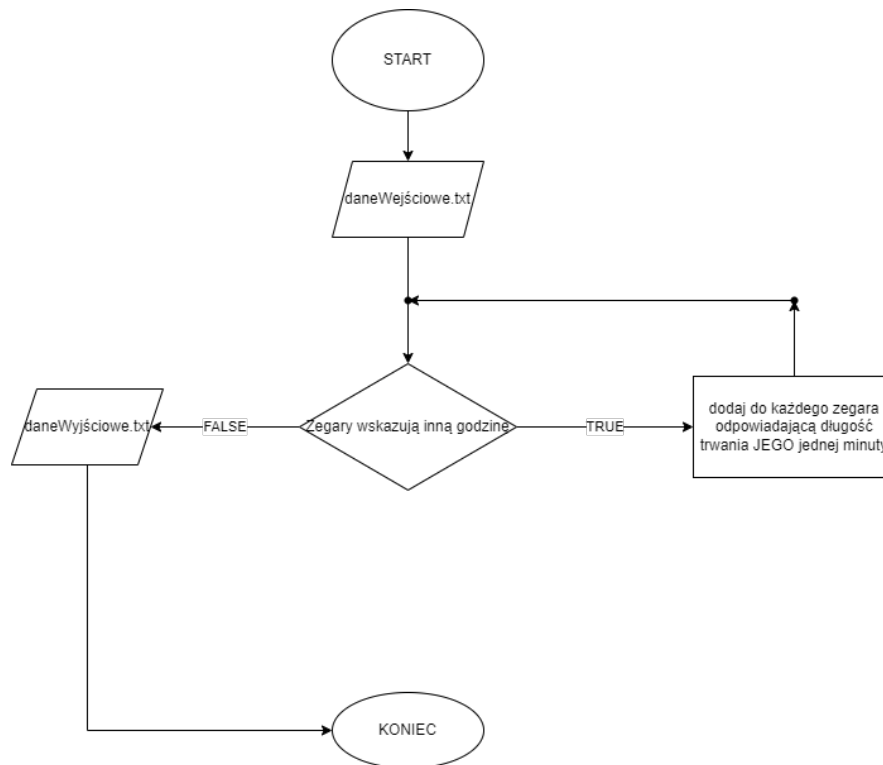
Zakończenie działania programu

Dodatkowe informacje

Aby program, będący rozwiązaniem zadania 4, pt. „Zegar”, działał poprawnie, należy go otworzyć za pomocą komputera z zainstalowanym Windowsem oraz wymagane jest aby było zainstalowane środowisko języka Python.

Część II

Opis działania



Rysunek 7: Schemat blokowy obrazuje działanie algorytmu wykonującego obliczenie potrzebnego czasu

Matematyczny wzór opisany w powyższym schemacie w bloku obliczeń:

$$z_1 = z_1 + 60 + o_1$$

$$z_2 = z_2 + 60 + o_2$$

$$z_3 = z_3 + 60 + o_3$$

Gdzie:

1. z_1, z_2, z_3 są wartościami pokazywanej godziny/czasu w sekundach dla kolejnych trzech zegarów
2. o_1, o_2, o_3 są wartościami opóźnień/przyspieszeń podanych w sekundach dla kolejnych trzech zegarów

Algorytm

Pseudokod stworzony w L^AT_EX, który prezentuje algorytm obliczenia szukanego wyniku.

Data: Dane wejściowe: *godzina* oraz *opoznienie*

Result: Dane wyjściowe: *czas spotkania podany w sekundach*

minutes = []

i = 0

while (*i* < *number_of_lines_in_inputfile*) **do**

 Dodaj element do listy „minutes”

if (*int(clock1_delay)* == *int(clock2_delay)* ||

int(clock1_delay) == *int(clock3_delay)* ||

int(clock3_delay) == *int(clock2_delay)*)

then

 | i-ty element w liście „minutes” zamień na: „Zegary się nigdy nie spotkają”

else

while (*clock1_time* != *clock2_time* != *clock3_time*) **do**

 zwiększ *clock_time* w każdym zegarze o długość trwania jednej minuty

 danego zegara (dla każdego musi być inna) Następnie powiększ i-ty

 element w liście „minutes” dodając 1

end

end

end

return *minutes*

Algorithm 1: Algorytm obliczania czasu, który upłynie do momentu aż zegary będą wskazywać tę samą godzinę.

Testy

Dla sprawdzenia poprawności działania programu zostały wykonane różne sytuacje, program obliczył poprawny wynik. Wyniki te prezentowane są po wyborze opcji nr. 4 w „menu.bat”

<i>Wprowadzone dane</i>	<i>Wynik</i>
9:00 +10 11:00 0 10:00 0 17:01 +5 18:59 -5 18:00 0 07:35 +5 08:35 0 9:35 -5 09:00 +2 11:00 -2 10:00 0	Zegary sie nigdy nie spotkaja 42480 43200 108000

Rysunek 8: Otrzymane wyniki testów

Pełen kod aplikacji

program.py

```
1 from datetime import datetime
2 import czytanie
3
4 # returns the time in seconds
5 def total_seconds(list):
6     for i in range(0, czytanie.set_of_three_clocks):
7         for j in range(0, czytanie.clocks_number):
8             time = datetime.strptime(list[i][j][0], "%H:%M")
9             time_in_seconds = time.minute * 60 + time.hour * 3600
10            list[i][j][0] = time_in_seconds
11    return list
12
13
14 def sort_by_delay(list):
15     for i in range(0, czytanie.set_of_three_clocks):
16         # reverse = True ---> sorts in descending order
17         list[i] = sorted(list[i], key=lambda l: int(l[1]), reverse=True)
18    return list
19
20
21 def time_to_meet(list):
22     minutes = []
23     for i in range(0, czytanie.set_of_three_clocks):
24         minutes.append(0)
25         clock1_time = list[i][0][0]
26         clock1_delay = list[i][0][1]
27         clock2_time = list[i][1][0]
28         clock2_delay = list[i][1][1]
29         clock3_time = list[i][2][0]
30         clock3_delay = list[i][2][1]
31
32         if (
33             int(clock1_delay) == int(clock2_delay)
34             or int(clock1_delay) == int(clock3_delay)
35             or int(clock3_delay) == int(clock2_delay)
36         ):
37             minutes[i] = "Zegary sie nigdy nie spotkaja"
38
39         else:
40             while not (clock1_time == clock2_time == clock3_time):
41                 clock1_time += 60 + int(clock1_delay)
42                 clock2_time += 60 + int(clock2_delay)
43                 clock3_time += 60 + int(clock3_delay)
44                 minutes[i] += 1
45    return minutes
46
47
48 seconds_array3D = total_seconds(czytanie.array3D)
49 sorted_seconds_array3D = sort_by_delay(seconds_array3D)
50 seconds_to_meet = time_to_meet(sorted_seconds_array3D)
51
```

```

52 myFile = open("daneWyjsciowe.txt", "w")
53 myFile.write("")
54 myFile.close()
55
56 myFile = open("daneWyjsciowe.txt", "a")
57 for i in range(len(seconds_to_meet)):
58     if seconds_to_meet[i] == "Zegary sie nigdy nie spotkaja":
59         myFile.write(seconds_to_meet[i] + "\n")
60     else:
61         myFile.write(str(seconds_to_meet[i] * 60) + "\n")
62 myFile.close()
63
64 print("\nProgarm zostal pomyslnie wykonany! \n")

```

statystyka.py

```

1 from typing import Counter, Text
2
3
4 f = open("daneWyjsciowe.txt", "r")
5 text = f.readlines()
6 f.close()
7
8 # removing suffix "\n"
9 text1 = []
10 for i in text:
11     text1.append(i.removesuffix("\n"))
12
13 result = 0
14 counter = 0
15 for i in text1:
16     if i == "Zegary sie nigdy nie spotkaja":
17         continue
18     else:
19         result += int(i)
20         counter += 1
21
22 mean_average = result / counter
23 f = open("statystyka.txt", "w")
24 f.write(str(mean_average))
25 f.close()

```

czytanie.py

```

1 # this file is responsible for loading data from a file
2 # and then assigning them to a 3D list accordingly
3 import pprint
4
5
6 def ThreeD(a, b, c):
7     lst = [[[None for col in range(a)] for col in range(b)] for row in
8             range(c)]
9     return lst
10
11 # reading from file

```

```

12 myFile = open("daneWejscowe.txt")
13 text = myFile.read()
14 myFile.close()
15
16 # splitting the file content into lines
17 ls_lines = text.split("\n")
18
19 # clock = 2 because each clock consists of two pieces of information:
20 # - the time
21 # - the duration of one minute
22 clock = 2
23
24 # clocs_number = 3 because we are considering a situation with three
    clocs
25 clocs_number = 3
26
27 # set_of_three_clocks is equal number of lines in the file
28 set_of_three_clocks = len(ls_lines)
29
30 # creating a 3D list
31 array3D = ThreeD(clock, clocs_number, set_of_three_clocks)
32
33
34 ls_each_clock = []
35 ls_three_clocks = []
36
37
38 for i in ls_lines:
39     ls_three_clocks = i.split("||", 2)
40     for j in ls_three_clocks:
41         ls_each_clock += j.split("|", 1)
42
43
44 for i in range(0, set_of_three_clocks): # row times
45     for j in range(0, clocs_number): # three times
46         for k in range(0, clock): # two times
47             array3D[i][j][k] = ls_each_clock.pop(0)
48
49 # deleting unnecessary data from memory
50 ls_lines.clear()
51 ls_three_clocks.clear()

```

menu.bat

```

1 @echo off
2 cd N:\Przedmiotowe_pliki\Jezykiskryptowe\Projekt
3 :menu
4 cls
5 echo
6
7 -----Menu-----
8 echo .
9 echo Wybierz jedna z ponizszych opcji
10 echo Wyboru mozesz dokonac wprowadzajac odpowiedni numer z klawiatury
11 echo 1. Start progarmu
12 echo 2. Backup

```

```

11 echo 3. Informacje o programie
12 echo 4. Wyniki programu
13 echo 5. Zamknij program
14 echo -----
15 echo.
16 set /p choice="Wybor (1/2/3/4/5): "
17
18 IF %choice%==1 GOTO first
19 IF %choice%==2 GOTO second
20 IF %choice%==3 GOTO third
21 IF %choice%==4 GOTO fourth
22 IF %choice%==5 GOTO fifth
23
24 echo.
25 echo Nie ma takiej opcji, sprobuj ponownie!
26 pause
27 GOTO :menu
28
29 :first
30 cls
31 call program.py
32 pause
33 GOTO :menu
34
35
36 :second
37 cls
38 mkdir C:\Users\user\Desktop\%date%
39 copy N:\Przedmiotowe_pliki\Jezykiskryptowe\Projekt C:\Users\user\Desktop
    \%date%
40 echo Kopia zapasowa zostala wykonana pomyslnie
41 pause
42 GOTO :menu
43
44
45 :third
46 cls
47 echo.
48 echo
    -----
49 echo.
50 echo Ogolne dzialanie programu:
51 echo.
52 echo Dzialanie programu opiera sie na trzech zegarach, posrod ktorych
    kazdy dziala inaczej
53 echo.
54 echo Wyobrazmy sobie nastepujaca sytuacje:
55 echo # pierwszy zegar spoznia o 5 sekund
56 echo # drugi zegar przyspiesza o 5 sekund
57 echo # trzeci zegar dziala poprawnie
58 echo.
59 echo Zadaniem tego programu jest obliczenie czasu, ktory uplynie az dane
    trzy zegary sie spotkaja
60 echo.

```

```

61 echo
-----

62 echo .
63 echo Dane wejscowe:
64 echo # zapisane sa w pliku daneWejscowe.txt
65 echo # format:
66 echo         "zegar1||zegar2||zegar3/n"
67 echo         "zegar1||zegar2||zegar3/n"
68 echo         itd.
69 echo         Przy czym pod wartoscia zegar1/zegar2/zegar3 kryja sie
informacje o danym zegarze w formacie:
70 echo         # "H:M|-wartosc_opoznienia/+wartosc_przyspieszenia"
71 echo         # w przypadku kiedy zegar dziala poprawnie nalezy wpisac "0"
72 echo .
73 echo         Dla przykladu:
74 echo         # "14:30|+5||16:30|-5||15:30|0"
75 echo         # zegar1 przyspiesza 5 sekund
76 echo         # zegar2 opoznia 5 sekund
77 echo         # zegar3 dziala poprawnie
78 echo .
79 echo
-----

80 echo .
81 echo Dane wyjscowe:
82 echo # zapisane sa w pliku daneWyjscowe.txt
83 echo # wynik reprezentujacy czas, ktory minal jest podany w sekundach
84 echo .
85 pause
86 GOTO :menu
87
88 :fourth
89 cls
90 call statystyka.py
91 start index.htm
92 GOTO :menu
93
94 :fifth
95 exit

```
