



## 2020/21 - Wprowadzenie do programowania (C#) - LAB (K. Molenda)

Kokpit / Moje kursy / wdp-2020-21-lab-kmolenda / Sprawdzian - programowanie obiektowe / Sprawdzian OOP - PRÓBNY

Rozpoczęto Monday, 28 June 2021, 21:27 PM Stan Ukończono Monday, 28 June 2021, 22:57 PM Wykorzystany 1 godzina 30 min. czas Ocena 10,00 pkt. na 15,00 pkt. możliwych do uzyskania (67%)

Pytanie 1 Poprawnie Ocena: 5,00 z

≡

5,00 ℙ Oflaguj pytanie

Twoim zadaniem jest wykonanie analizy logowań do pewnego systemu webowego. Otrzymujesz wielowierszowy napis (typu string) zawierający, w kolejnych liniach: datę i czas logowania, login użytkownika oraz numer IP, np.:

```
2021-06-10 5:20 mirek 180.40.41.120
2021-06-10 10:20 adam 80.40.41.120
2021-06-10 10:21 admin 192.168.4.10
2021-06-11 8:09 maciek 149.40.53.12
2021-06-11 8:10 adam 80.40.41.120
2021-06-11 18:10 mirek 180.40.41.120
```

Napisz procedurę o sygnaturze

public static void Analyze(string logs)

wypisującą na standardowe wyjście, w porządku rosnącym, loginy użytkowników, którzy w badanym okresie zalogowali się do serwisu przynajmniej jeden raz każdego dnia.

- Nie wypisuj duplikatów.
- Loginy użytkowników wypisz w jednej linii, oddzielając je przecinkiem i pojedynczą spacją.
- W przypadku braku użytkowników spełniających warunki zadania wypisz słowo empty.

⚠ Rozwiąż zadanie wykorzystując właściwie dobrane do problemu standardowe kolekcje C#. Zabronione jest używanie operatorów LINQ.

## Na przykład:

Dane wejściowe		Wynik
2021-06-10	5:20 mirek 180.40.41.120	adam, mirek
2021-06-10	10:20 adam 80.40.41.120	
2021-06-10	10:21 admin 192.168.4.10	
2021-06-11	8:09 maciek 149.40.53.12	
2021-06-11	8:10 adam 80.40.41.120	
2021-06-11	18:10 mirek 180.40.41.120	
2021-06-10	5:20 mirek 180.40.41.120	empty
2021-06-10	10:20 adam 80.40.41.120	
2021-06-10	10:21 admin 192.168.4.10	
2021-06-11	8:09 maciek 149.40.53.12	
2021-06-11	8:10 kazik 80.40.41.120	

**Odpowiedź:** (system kar: 0, 0, 1, 5, 10, 20, ... %)

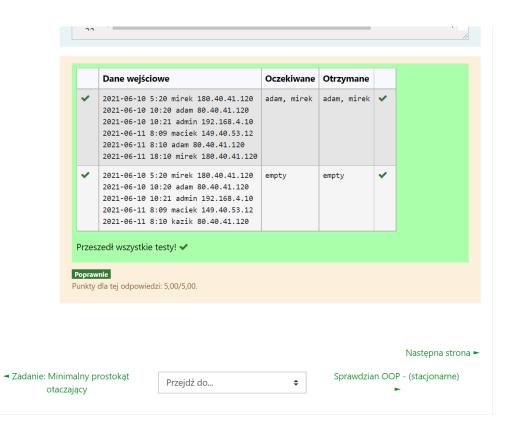
## Zresetuj odpowiedź

```
public static void Analyze(string logs)
             {
 3
                  string[] rozdzieloneLinie = logs.Split("\n");
 4
                  string[] rozdzieloneDaty = new string[rozdzieloneLinie.Lengt
                  string[] rozdzieloneLoginy = new string[rozdzieloneLinie.Len
                  Dictionary<string, int> listaDni = new Dictionary<string, in
                  for (int i = 0; i < rozdzieloneLinie.Length; i++)
 8
                      rozdzieloneDaty[i] = rozdzieloneLinie[i].Split(" ")[0];
rozdzieloneLoginy[i] = rozdzieloneLinie[i].Split(" ")[2]
10
                           Console.WriteLine(rozdzieloneDaty[i]);
11
12
                           Console.WriteLine(rozdzieloneLoginy[i]);
13
                      if (listaDni.ContainsKey(rozdzieloneDaty[i]))
14
15
                          listaDni[rozdzieloneDaty[i]] += 1;
16
17
                          listaDni.Add(rozdzieloneDaty[i], 1);
18
19
20
                  SortedDictionary<string, int> listaLoginow = new SortedDicti
21
```

Nawigacja w teście



Pokaż wszystkie pytania na stronie Zakończ przegląd



Jesteś zalogowany(a) jako <u>Petek Krystian</u> (<u>Wyloguj</u>) <u>wdp-2020-21-lab-kmolenda</u> <u>Podsumowanie zasad przechowywania danych</u>