

---

# PPPD - Lab. 13

Copyright ©2023 M. Śleszyńska-Nowak i in.

Zadanie punktowane, lab 13, 2023/2024, autor: Michał Własnowolski

---

**Uwaga:** Operując na listach nie wolno używać slice'ów, ani ujemnego indeksowania.

**Temat:** Usadzenie gości przy stole

## Treść zadania

Zadanie polega na napisaniu programu rozmieszczającego gości przy stole w taki sposób, by każdy znajdował się w bezpośrednim sąsiedztwie osób o wieku możliwie najbardziej zbliżonym do jego własnego.

## Etap pierwszy - wczytanie danych oraz tworzenie listy gości (2 p.)

- 1) Zaimplementuj klasę `Gosc`. Konstruktor tej klasy powinien przyjmować trzy argumenty: imię (typu `str`), nazwisko (typu `str`) i wiek (typu `int`). Klasa powinna zawierać listę atrybutów `__slots__`, ograniczającą dynamiczne tworzenie nowych atrybutów. Reprezentacja tekstowa obiektu tej klasy (`__str__`) powinna zwracać napis w formacie: "Imię Nazwisko, lat X".

Przykład użycia:

```
lista_gosci = wczytaj_gosci('lista_gosci.csv')
print(lista_gosci[0])
# Iza Wasilewska, lat 25
```

- 2) Napisz funkcję `wczytaj_gosci()`, która przyjmuje jako parametr ścieżkę do pliku tekstowego z listą gości. Poniżej opis pliku `'lista_gosci.csv'`, który funkcja powinna przetworzyć:
  - każdy wiersz odpowiada jednej osobie,
  - dane w wierszu są oddzielone przecinkami,
  - liczba gości jest parzysta,
  - Struktura danych w pliku: imię, nazwisko, wiek,
  - funkcja powinna zwracać listę instancji klasy `Gosc` zainicjowanych na podstawie danych z każdego wiersza pliku `'lista_gosci.csv'`,
  - użyj metody `line.strip().split(",")`, aby usunąć białe znaki (w tym znaki nowej linii) i podzielić każdy wiersz na listę danych.

## Etap drugi - inicjalizacja obiektów klasy `Stół` (1 p.)

Zaimplementuj klasę `Stol`, której konstruktor przyjmuje dwa argumenty:

- `lista_gosci`: lista obiektów klasy `Gosc` reprezentująca gości,
- `ksztalt`: parametr typu `str`, który przyjmuje jedną z dwóch wartości - 'okragly' lub 'prostokatny'.

Zależnie od wartości parametru `ksztalt`, klasa `Stol` powinna inicjalizować strukturę reprezentującą rozmieszczenie gości przy stole:

- dla `kształt == 'okrągły'`: lista usadzenia gości powinna być ustawiona jako jednoelementowa lista list, `[[gosc1, gosc2, ...]]`, symulująca miejsca dookoła okrągłego stołu.
- dla `kształt == 'prostokątny'`: lista usadzenia gości jest dzielona na pół, tworząc dwuelementową listę list, `[[gosc1_lewy, gosc2_lewy, ...], [gosc1_prawy, gosc2_prawy, ...]]`, reprezentującą gości siedzących po obu stronach prostokątnego stołu. Należy zauważyć, że nikt nie siedzi na szczytach stołu, a liczba gości jest zawsze parzysta.

Obiekt klasy `Stol` powinien dodatkowo przechowywać wartość `score` (typu *float*), która reprezentuje obliczoną ‘jakość usadzenia’ gości (szczegóły w etapie trzecim).

## Etap trzeci - oblicz jakość usadzenia (3 p.)

Celem jest zaimplementowanie metody `oblicz_score()` w klasie `Stol`, która oblicza i ustawia wartość “jakości usadzenia” gości przy stole. Metoda ta powinna stosować poniższy algorytm:

Dla każdego gościa na liście oblicz cząstkowy `score`, sumując wartości bezwzględne różnic wieku między danym gościem a jego sąsiadami. Następnie podziel tę sumę przez liczbę sąsiadów, aby uzyskać średnią różnicę wieku dla danego gościa. Po obliczeniu cząstkowego `score` dla każdego gościa, zsumuj wszystkie te wartości i podziel przez całkowitą liczbę gości, aby uzyskać końcowy `score` dla stołu. Ten wynikowy `score` powinien być następnie ustawiony jako atrybut obiektu klasy `Stol`, reprezentujący ogólną “jakość usadzenia” gości przy stole.

Dla stołu o parametrze `kształt == 'okrągły'`, każdy gość ma dwóch sąsiadów - gości bezpośrednio przed nim (element `i-1`) oraz po nim (element `i+1`). Pierwszy i ostatni gość na liście są traktowani jako sąsiedzi. Dla stołu o parametrze `kształt == 'prostokątny'`, każdy gość ma do 5 sąsiadów - dwóch sąsiadujących bezpośrednio na tej samej stronie stołu (elementy `i-1` i `i+1`, jeśli istnieją), jednego naprzeciwko oraz po lewym i prawym skosie (elementy `i-1`, `i` oraz `i+1` znajdujące się po drugiej stronie stołu, jeśli istnieją).

Przykład sąsiadów przy okrągłym stole:

```
lista_gosci = [[gosc1, gosc2, gosc3, gosc4, ..., gosc10]]
sąsiadami dla gosc2 są gosc1 oraz gosc3
sąsiadami dla gosc1 są gosc2 oraz gosc10
```

Przykład sąsiadów przy prostokątnym stole:

```
lista_gosci = [[gosc1_lewy, gosc2_lewy, gosc3_lewy], [gosc1_prawy, gosc2_prawy, gosc3_prawy]]
sąsiadami dla gosc2_lewy są: gosc1_lewy, gosc3_lewy, gosc1_prawy, gosc2_prawy, gosc3_prawy
sąsiadami dla gosc1_lewy są: gosc2_lewy, gosc1_prawy, gosc2_prawy
```

## Etap czwarty - wyświetl Stół (2p.)

Zaimplementuj metodę `__str__()` w klasie `Stol`, która odpowiedzialna będzie za reprezentację tekstową obiektu. Metoda ta powinna:

- wyświetlać wszystkich gości usadzonych przy stole. Każdego gościa należy przedstawić w formacie: `Imię_Nazwisko_Wiek`,
- dla stołu o kształcie ‘okrągły’ lista gości powinna być wyświetlona w jednej kolumnie,
- dla stołu o kształcie ‘prostokątny’ lista gości powinna być wyświetlona w dwóch kolumnach reprezentujących dwie strony stołu (patrz *Przykłady użycia*).

## Etap piąty - main (2p. )

W funkcji `main` wykonaj następujące działania (patrz *Przykłady użycia*):

1) Wczytywanie listy gości:

- użyj funkcji `wczytaj_gosci('lista_gosci.csv')`, aby wczytać listę gości z podanego pliku,
- wyświetl wczytaną listę wszystkich gości.

## 2) Tworzenie i wyświetlanie obiektów klasy Stol:

- na podstawie wczytanej listy gości utwórz obiekty klasy Stol() z parametrami 'prostokątny' oraz 'okrągły'.
- wyświetl oba obiekty, aby zareprezentować rozmieszczenie gości przy obu kształtach stołu.

## 3) Wyszukiwanie optymalnego ustawienia:

- ustaw generator liczb pseudolosowych za pomocą random.seed(1234),
- przeprowadź 1000 symulacji, w każdej z nich zamieniając miejscami gości na liście za pomocą random.shuffle(lista). Wykonaj to zarówno dla stołu 'prostokątnego', jak i 'okrągłego'.

Przykład użycia funkcji shuffle:

```
lista = [1, 2, 3, 4]
random.seed(1234)
random.shuffle(lista)
print(lista)
# [3, 1, 4, 2]
```

- dla każdego typu stołu (prostokątnego i okrągłego) zapamiętaj obiekt Stol, który osiągnął najniższą wartość score i wyświetl go.

## Uwaga

- Jeśli program się nie kompiluje (interpretuje), ocena jest zmniejszana o połowę.
- Jeśli kod programu jest niskiej jakości (nieestetycznie formatowanie, mylące nazwy zmiennych itp.), ocena jest zmniejszana o 2p,
- Elementy rozwiązania korzystające ze slicingu lub indeksowania ujemnego będą ocenione na 0 punktów (cały element jest zerowany),

## Punktacja

Za poszczególne elementy można uzyskać następującą liczbę punktów:

- Wczytanie danych oraz stworzenie listy gości (2 p.)
- Inicjalizacja klasy Stół (1 p.)
- Obliczenie jakości usadzenia (3 p.)
- Wyświetlenie Stółów (2 p.)
- Funkcja main() (2 p.)

## Przykłady działania kodu

Lista gości:

Iza Wasilewska, lat 25  
Nikola Kowalska, lat 83  
Agata Kaźmierczak, lat 77  
Adela Lis, lat 49  
Agnieszka Szulc, lat 31  
Alice Zielińska, lat 32  
Ewa Malinowska, lat 40  
Jolanta Borkowska, lat 77  
Konstancja Mazur, lat 36  
Bianka Urbańska, lat 26  
Kryspin Laskowski, lat 22  
Amir Urbański, lat 44  
Kamil Marciniak, lat 67

---

Denis Jaworski, lat 11  
Przemysław Laskowski, lat 90  
Eryk Tomaszewski, lat 74  
Ireneusz Duda, lat 49  
Korneliusz Sadowski, lat 64  
Albert Sawicki, lat 15  
Zbigniew Wodecki, lat 40

Stół prostokątny:

Iza\_Wasilewska\_25 || Kryspin\_Laskowski\_22  
Nikola\_Kowalska\_83 || Amir\_Urbański\_44  
Agata\_Kaźmierczak\_77 || Kamil\_Marciniak\_67  
Adela\_Lis\_49 || Denis\_Jaworski\_11  
Agnieszka\_Szulc\_31 || Przemysław\_Laskowski\_90  
Alice\_Zielińska\_32 || Eryk\_Tomaszewski\_74  
Ewa\_Malinowska\_40 || Ireneusz\_Duda\_49  
Jolanta\_Borkowska\_77 || Korneliusz\_Sadowski\_64  
Konstancja\_Mazur\_36 || Albert\_Sawicki\_15  
Bianka\_Urbańska\_26 || Zbigniew\_Wodecki\_40  
Score:28.586666666666666

Stół okrągły:

Iza\_Wasilewska\_25  
Nikola\_Kowalska\_83  
Agata\_Kaźmierczak\_77  
Adela\_Lis\_49  
Agnieszka\_Szulc\_31  
Alice\_Zielińska\_32  
Ewa\_Malinowska\_40  
Jolanta\_Borkowska\_77  
Konstancja\_Mazur\_36  
Bianka\_Urbańska\_26  
Kryspin\_Laskowski\_22  
Amir\_Urbański\_44  
Kamil\_Marciniak\_67  
Denis\_Jaworski\_11  
Przemysław\_Laskowski\_90  
Eryk\_Tomaszewski\_74  
Ireneusz\_Duda\_49  
Korneliusz\_Sadowski\_64  
Albert\_Sawicki\_15  
Zbigniew\_Wodecki\_40  
Score:26.8

Najlepsze usadzenie:

Stół prostokątny:

Jolanta\_Borkowska\_77 || Przemysław\_Laskowski\_90  
Eryk\_Tomaszewski\_74 || Kamil\_Marciniak\_67  
Korneliusz\_Sadowski\_64 || Nikola\_Kowalska\_83  
Ewa\_Malinowska\_40 || Albert\_Sawicki\_15  
Iza\_Wasilewska\_25 || Bianka\_Urbańska\_26  
Adela\_Lis\_49 || Agata\_Kaźmierczak\_77  
Konstancja\_Mazur\_36 || Zbigniew\_Wodecki\_40

---

Ireneusz\_Duda\_49 || Amir\_Urbański\_44  
Alice\_Zielińska\_32 || Kryspin\_Laskowski\_22  
Denis\_Jaworski\_11 || Agnieszka\_Szulc\_31  
Score:18.266666666666662

Stół okragly:  
Iza\_Wasilewska\_25  
Zbigniew\_Wodecki\_40  
Ireneusz\_Duda\_49  
Nikola\_Kowalska\_83  
Jolanta\_Borkowska\_77  
Eryk\_Tomaszewski\_74  
Agata\_Kaźmierczak\_77  
Korneliusz\_Sadowski\_64  
Przemysław\_Laskowski\_90  
Amir\_Urbański\_44  
Agnieszka\_Szulc\_31  
Albert\_Sawicki\_15  
Denis\_Jaworski\_11  
Alice\_Zielińska\_32  
Bianka\_Urbańska\_26  
Kamil\_Marciniak\_67  
Adela\_Lis\_49  
Kryspin\_Laskowski\_22  
Ewa\_Malinowska\_40  
Konstancja\_Mazur\_36  
Score:16.7