Programowanie I

Zajęcia nr 5

Rafał Masełek

31. marca 2022r.

Zadanie 0 – zabawa słownikami

Celem tego zadania jest zrozumienie czym są "dictionary views". Napisz program, który realizuje poniższe kroki:

1. Stwórz słownik, w którym kluczami będą cyfry od 0 do 9, a wartościami napisy odpowiadające słownemu zapisowi cyfr, np. {1 : 'jeden', ...}
2. Skorzystaj z odpowiednich metod dla słowników i zapisz do dwóch zmiennych klucze i wartości.
3. Do dwóch nowych zmiennych wpisz ponownie klucze i wartości, ale tym razem dokonaj konwersji na listy.
4. Wypisz (czytelnie!) wszystkie cztery zmienne.
5. Usuń dowolne trzy elementy ze słownika.
6. Ponownie wypisz zawartość czterech zmiennych.

Powinieneś zobaczyć różnicę w wypisanych wartościach w zależności od tego, czy dokonano konwersji na listę czy nie. Metody ".keys()" oraz ".values()" zwracają tzw. "dictionary view", które są "podglądem" zawartości słownika **w obecnej chwili**, tzn. jeżeli zmienimy słownik to zobaczymy te zmiany w "dictionary views". Z drugiej strony, dokonując konwersji na listy tworzymy kopie wartości, które nie są już związane z zawartością oryginalnego słownika.

Zadanie 1 – rekurencja z zapamiętywaniem

Wykorzystaj program implementujący ciąg Collatza z pierwszych zajęć i dopisz do niego dekorator zapamiętujący policzone elementy ciągu. Możesz skorzystać z przykładu podanego na wykładzie dla ciągu Fibonacciego. Zmodyfikuj program tak, żeby w trakcie jednego wykonania użytkownik mógł podać kilka wartości k, w tym np. dwa razy taką samą. Za każdym razem zmierz i wyświetl czas, jaki zajmuje wykonanie fragmentu kodu liczącego wszystkie wyrazy ciągu aż do wystąpienia jedynki. Wypróbuj działanie i czas wykonania programu z dekoratorem i bez. Podając duże k powieneś być w stanie zauważyć zysk wynikający z użycia dekoratora do zapamiętywania policzonych elementów ciągu.

Zadanie 2 – baza danych w słowniku

Zaimplementuj bazę danych uczniów za pomocą słowników. Każdy uczeń będzie reprezentowany przez słownik z czterema kluczami-napisami:

- "imie" – odpowiada wartości: napis z imieniem ucznia,
- "nazwisko" – odpowiada wartości: napis z nazwiskiem ucznia,

- "erednia" – odpowiada wartości: liczba ze średnią ocen,
- "oceny" – odpowiada wartości: lista z ocenami (liczbami).

Uczniowie mają być umieszczeni w słowniku jako wartości z kluczami będącymi kolejnymi liczbami porządkowymi, tak jak w przykładzie poniżej:

```
uczniowie = {
1 : {'imie' : 'Jan', 'nazwisko' : 'Kowalski', 'srednia' : 4.5, 'oceny' : [4.0, 5.0] },
2 : {'imie' : 'Janina', 'nazwisko' : 'Nowak', 'srednia' : 3.5, 'oceny' : [3.0, 3.0, 4.0, 4.0]
}
```

Program ma komunikować się w jasny sposób z użytkownikiem i umożliwiać następujące czynności:

- Wypisanie listy uczniów w formacie "NUMER NAZWISKO, IMIE", gdzie lista jest uporządkowana wg. rosnących numerów.
- Dodanie nowego ucznia. Program powinien zapytać o imię i nazwisko. Nowy uczeń powinien dostać nowy numer, mieć ustawioną średnią na 0.0 i pustą tablicę ocen. Dla chętnych: spraw, żeby uczniowie zawsze mieli numery odpowiadające kolejności alfabetycznej nazwisk.
- Usunięcie istniejącego ucznia po podaniu jego numeru. Po usunięciu ucznia ze słownika należy zmienić numery-klucze pozostałych uczniów tak, żeby były po kolei. Np. jeżeli mamy 5 uczniów i usuniemy ucznia o numerze 1, to uczeń o numerze 2 dostanie nowy numer 1, uczeń o numerze 3 dostanie nowy numer 2 itd.
- Wypisanie szczegółów ucznia. Po wykonaniu tej komendy program powinien wyświetlić wszystkie informacje o uczniu w czytelny i jasny sposób.
- Dodać ocenę istniejącemu uczniowi. Nowa ocena powinna znaleźć się w liście ocen oraz wpłynąć na średnią. Dozwolone oceny są od 1 do 6 włącznie. Oceny niedozwolone powinny spowodować wyświetlenie stosownego komunikatu i nie powinny modyfikować słownika.
- Zakończyć działanie programu. Program powinien działać tak długo, aż użytkownik nie zdecyduje o jego zamknięciu.

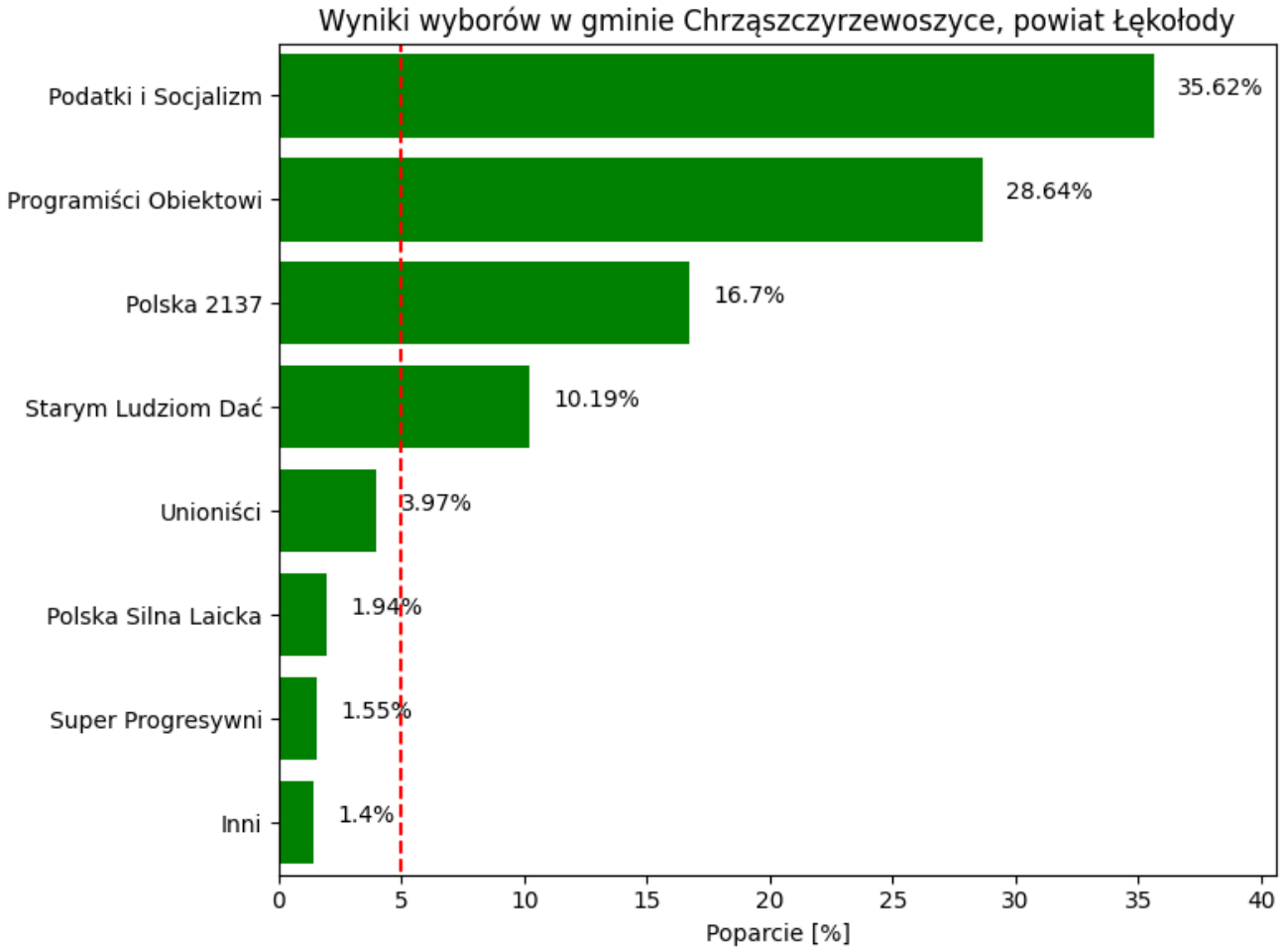
Program powinien sprawdzać, czy podany przez użytkownika numer ucznia występuje w słowniku celem uniknięcia błędu. Jak zawsze, program musi być idioto-odporny. Program powinien wyświetlać informację wyjaśniającą w jaki sposób można wykonać każdą z czynności. Zawsze gdy program oczekuje podania danych przez użytkownika, powinien pojawić się stosowny komunikat. Podany powyżej przykład implementacji słownika "uczniowie" można skopiować do programu, żeby mieć dwóch uczniów już w momencie uruchomienia.

Zadanie 3 – wybory w Chrząszczyrzewoszycach

W gminie Chrząszczyrzewoszyce (powiat Łękołody) odbyły się wybory parlamentarne. Wyniki poszczególnych komitetów wyborczych zebrano w poniższym słowniku:

```
wyniki = {
                        'Podatki i Socjalizm' :  9732,
                        'Unionisci' : 1084,
                        'Polska Silna Laicka' : 531,
                        'Polska 2137' : 4562,
                        'Programisci Obiektowi' : 7824,
                        'Starym Ludziom Dac' : 2783,
                        'Super Progresywni' : 423,
                        'Inni': 383,
                }
```

Korzystając z matplotlib.pyplot.barh() narysuj poziomy wykres słupkowy pokazujący procentowy wynik poszczególnych komitetów (patrz obrazek poniżej). Dodatkowo, używając matplotlib.pyplot.text() napisz na prawo od słupków wynik procentowy zaokrąglony do 2 miejsc po przecinku. Ponadto, korzystając z matplotlib.pyplot.vlines() narysuj pionową czerwoną przerywaną linię określającą próg wyborczy (5%). Zadbaj o estetykę wykresu, podpisz osie i dodaj tytuł. Słupki mają mieć kolor zielony. Słupki mają być wyświetlane w posortowany sposób, tj. partia z największym popraciem na górze, poniżej partia z najmniejszym popraciem itd. Skorzystaj z sortowania, nie ręcznego modyfikowania słownika! W tym celum może się przydać "sorted()" oraz "zip()".



Rysunek 1: Przykładowy wykres.