Programowanie 1R – Ćwiczenia 3
Słowniki i zbiory

Mikołaj Marcinkowski

10 marca 2026

Zadania możesz wykonywać jako oddzielne skrypty `.py` lub w notatniku jupytera `.ipynb`, chyba że treść zadania wskazuje inaczej.

Zadania wzorowane na Serii 3 zadań autorstwa Bartłomieja Zglinickiego (`https://www.fuw.edu.pl/~bzglinicki/teaching/p1r/zadania/seria-3/`), udostępnionej na licencji CC BY 4.0.

Zadanie 1. Lotto

Napisz program, który losuje sześć różnych liczb całkowitych z przedziału 1 do 49 włącznie i je wypisuje. Do losowania możesz wykorzystać funckję `random.randint` z biblioteki `random`.

Zadanie 2. Liczenie elementów

Napisz funkcję, która przyjmuje listę, liczy, ile razy występuje w niej każdy jej element, po czym zwraca słownik, w którym kluczami są elementy listy, a odpowiadającymi im wartościami krotność danego elementu w liście.

Zadanie 3. Liczba ludności

W Pythonie do przechowywania danych często wykorzystuje się pliki `json`. Są to pliki tekstowe, w których, używając (prawie) identycznej notacji jak kodzie, można zapisać słownik lub listę, zawierające liczby, stringi czy inne słowniki i listy.

W załączonym do zadania pliku `ludnosc.json` znajdują się informacje na temat liczby ludności i pola powierzchni (w km^2) województw na przestrzeni lat. Zapisane są w postaci zagnieżdżonych słowników:

```
{rok:{województwo:{"powierzchnia km2":powierzchnia,
                   "ludność":ludność}}}
```

Możesz otworzyć ten plik w dowolnym edytorze, by lepiej zapoznać się z jego strukturą.

Napisz program, który wczytuje dane z pliku `ludnosc.json`, po czym prosi użytkownika o wybór województwa i statystyki: liczby ludności, powierzchni lub gęstości zaludnienia. Po wyborze przygotowuje wykres pokazujący wartości danej statystyki w danym województwie na przestrzeni lat. Oprócz województw, program powinien też pozwalać na wyświetlanie statystyk dla całej Polski.

Dane z pliku `json` można wczytać na przykład kodem:

```
import json

plik = open("ścieżka/do/pliku/ludnosc.json")
dane = json.load(plik)
plik.close()
```

Wtedy obiekt `dane` będzie wczytanym słownikiem.

Dane pochodzą z GUS.

Zadanie 4. Histogram

a) Napisz funkcję `przygotuj_histogram`, która przyjmuje dwie listy liczb: `granice` i `dane`. O liście `granice` załóż, że jest posortowana rosnąco. Jeśli oznaczymy elementy `granice` jako g_0, g_1, …, g_n, to funkcja ta powinna policzyć, ile liczb x z `dane` mieści się w przedziałach $g_0 \leqslant x < g_1$, $g_1 \leqslant x < g_2$ itd. Powinna też policzyć liczby poniżej g_0 i powyżej (lub równe) g_n. Funkcja ta powinna zwrócić słownik, w którym kluczami są nazwy określające poszczególne przedziały (np. "0-10", "10-20", "<0"), a wartościami liczba liczb x, które mieszczą się w każdym z przedziałów.

b) Następnie napisz funkcję `narysuj_histogram`, która przyjmuje słownik jak ten zwracany przez `przygotuj_histogram` i tworzy wykres słupkowy, którego słupki podpisane są nazwami przedziałów, a ich wysokości to liczba zliczeń z przedziału. Do wykonania rysunku skorzystaj z wykresu `ax.bar` z biblioteki `matplotlib` (dokumentacja, tutorial).

c) W załączonym do zadania pliku `D-42.json` znajdują się przykładowe granice i dane. Wykonaj na ich podstawie histogram. Na wypadek, gdyby podpisy słupków na siebie nachodziły, można je obrócić o zadany w stopniach `kąt` kodem:

```
for label in ax.get_xticklabels():
    label.set_rotation(kąt)
```

Dziękuję Zuzannie Marcinkowskiej za pomysł na zadanie 4 i Tomaszowi Machtylowi za pomoc w przygotowaniu danych do histogramu.