

# PROGRAMOWANIE FUNKCYJNE W PYTHONIE

## *ĆWICZENIA DO PREZENTACJI*

### MODUŁ 4

AltKom Akademia S.A., materiały własne

# 1 PROGRAMOWANIE FUNKCYJNE

## ĆWICZENIE 1.1:

### Wieczny kalendarz – użycie funkcji

#### UMIEJĘTNOŚCI:

- Po wykonaniu ćwiczenia zdobędziesz umiejętności:
  - wykonywania operacji na tekstach
  - definiowania i korzystania z funkcji

#### CELE I ZADANIA:

- Napisz funkcję, która na podstawie daty podanej jako tekst, określi jaki to dzień tygodnia
- Wykorzystaj algorytm z ćwiczenia 2.2 z modułu 3
- W rozwiązaniu zdefiniuj dwie funkcje pomocnicze:
  - jedną – określającą numer dnia tygodnia
  - drugą – określającą na podstawie numeru dnia – nazwę dnia tygodnia

#### ALGORYTM WYKONANIA:

- Korzystając z algorytmu zaimplementowanego w ćwiczeniu 2.2 z modułu 3 utwórz funkcję pomocniczą *numer\_dnia\_tygodnia*
  - funkcja powinna posiadać dwa argumenty:
    - \* tekst zawierający datę (w kolejności: dzień, miesiąc, rok)
    - \* użyty separator do oddzielenia elementów daty
  - zgodnie z zaimplementowanym algorytmem funkcja powinna zwrócić numer dnia tygodnia
- Zastanów się, jak efektywnie wyodrębnić z daty: dzień, miesiąc i rok i skonwertować je na wartości liczbowe
- Utwórz drugą funkcję pomocniczą *nazwa\_dnia\_tygodnia*
  - funkcja powinna poprzez argument przyjąć numer dnia tygodnia
  - w oparciu o dane na wejściu powinna zwrócić tekst z nazwą dnia tygodnia
- Utwórz funkcję *jaki\_to\_dzien*, która na podstawie tekstu z datą oraz użytego separatora wypisze nazwę dnia tygodnia
- Podanie parametru separatora powinno być opcjonalne
- W implementacji wykorzystaj utworzone wcześniej funkcje pomocnicze
- Wywołaj utworzoną funkcję dla wczytanej wcześniej daty i sprawdź jej działanie

## ĆWICZENIE 1.2:

### “Baza” osób – funkcje własne i wbudowane

#### UMIEJĘTNOŚCI:

- Po wykonaniu ćwiczenia zdobędziesz umiejętności:
  - symulowania kolekcji danych za pomocą słowników oraz list (namiastka bazy danych)
  - wydobywania potrzebnych danych z utworzonej struktury
  - definiowania i użycia funkcji

#### CELE I ZADANIA:

- Wzorując się na rozwiązaniu ćwiczenia 2.11 z modułu 3 zrealizuj te same zapytania, tym razem wykorzystując wbudowane funkcje *filter* oraz *map*
- W rozwiązaniu wykorzystaj własne funkcje reprezentujące warunki (predykaty) przekazywane jako argument do funkcji *filter* oraz funkcję przekształcenia przekazywaną jako argument do funkcji *map*
- Sprawdź, czy wyniki są identyczne

#### ALGORYTM WYKONANIA:

- Z ćwiczenia 2.11 z modułu 3 skopiuj dane wejściowe (słowniki i listę)
- Możesz też utworzyć dodatkową funkcję *utworz\_osobe*, która pobierze wszystkie wymagane dane opisujące osobę i zwróci gotowy słownik – w ten sposób zmniejszymy ryzyko pomyłki podczas tworzenia słowników, gdyż funkcja zadba, aby we wszystkich słownikach były te same klucze
- Utwórz własne funkcje:
  - predykaty, które reprezentują warunki wykorzystywane w zapytaniach (występujące w wyrażeniach listowych)
  - funkcje, które przekształcają dane sekwencji (patrz: ostatnie zapytanie)
- Wykorzystaj utworzone funkcje oraz wbudowane funkcje *filter* oraz *map* do zrealizowania tych samych zapytań co w ćwiczeniu 2.11 z modułu 3 – w ten sposób uda się wyeliminować wyrażenia listowe
- Porównaj otrzymane wyniki ze wcześniejszymi

## ĆWICZENIE 1.3:

### “Baza” osób – sortowanie

#### UMIEJĘTNOŚCI:

- Po wykonaniu ćwiczenia zdobędziesz umiejętności:
  - symulowania kolekcji danych za pomocą słowników oraz list (namiastka bazy danych)
  - sortowania danych wg zadanych kryteriów

#### CELE I ZADANIA:

- Korzystając z danych z poprzedniego ćwiczenia 1.2 wypisz na ekranie aktualne dane wszystkich osób znajdujących się w liście
- Uporządkuj dane wg różnych, zadanych kryteriów i wypisz je na ekranie, aby sprawdzić, czy zostały posortowane

#### ALGORYTM WYKONANIA:

- Skopiuj rozwiązanie poprzedniego ćwiczenia 1.2
- Pozostaw tylko dane (słowniki i listę)
- Wypisz na ekranie aktualny stan listy
- Posortuj dane wg zadanych kryteriów, np.:
  - wg nazwisk
  - wg płci i wieku
  - ...
- Wypisz ponownie te dane, aby sprawdzić, czy zostały one faktycznie posortowane

## ĆWICZENIE 1.4:

### “Baza” osób – funkcje lambda

#### UMIEJĘTNOŚCI:

- Po wykonaniu ćwiczenia zdobędziesz umiejętności:
  - symulowania kolekcji danych za pomocą słowników oraz list (namiastka bazy danych)
  - wydobywania potrzebnych danych z utworzonej struktury
  - definiowania i użycia wyrażeń lambda

#### CELE I ZADANIA:

- Wzorując się na rozwiązaniu ćwiczenia 1.3 zastąp funkcje przekazywane jako argumenty do funkcji *filter* i *map* funkcjami lambda

#### ALGORYTM WYKONANIA:

- Skopiuj rozwiązanie ćwiczenia 1.3
- Usuń funkcje predykatów oraz funkcje konwersji
- Zastąp je funkcjami lambda
- Sprawdź działanie tak zmodyfikowanego programu
- Porównaj otrzymane wyniki ze wcześniejszymi
- Jakie zalety/wady ma to rozwiązanie?