

# Praca domowa nr 3

Szanowni Państwo,

Tym razem w pracy domowej proszę o uzupełnienie gotowych programów. Mam nadzieję, że ułatwi to Państwu zaznajomienie się z argumentami *domyślnymi*, z *operacjami na plikach* oraz z dołączeniem i korzystaniem z *modułów*. Za każde zadania otrzymać można maksymalnie 1.5 punkta, przy czym wykonanie zadania 2b wymaga wykonania zadania 2a.

W razie pytań służę pomocą. Ponadto na najbliższych zajęciach omówimy pracę domową i wyjaśnimy wszystkie szczegóły związane z poruszonymi w niej zagadnieniami.

## Zadanie 1

*Korzystanie z funkcji.*

Uzupełnij kod programu dopisując funkcje *silnia* i *potęga*. Dodaj warunek w pętli *while*, który pozwoli zatrzymać program po wpisaniu wybranego przez Ciebie ciągu znaków (np. *stop*).

Zauważ, że funkcja *dzialanie\_matematyczne()* może przyjąć jeden lub dwa argumenty liczbowe, ponieważ argument *liczba\_2* przyjmuje wartość *domyślną None*.

```
def dzialanie_matematyczne(rodzaj, liczba_1, liczba_2=None):
    if rodzaj == 'suma':
        return liczba_1 + liczba_2
    elif rodzaj == 'silnia':
        return silnia(liczba_1)
    elif rodzaj == 'potega':
        return potega(liczba_1, liczba_2)
    else:
        return None

if __name__ == '__main__':
    while True:
        rodzaj = raw_input('podaj nazwe dzialania > ')
        a = float(raw_input('podaj pierwsza liczbe > '))
        b = float(raw_input('podaj druga liczbe > '))
        print dzialanie_matematyczne(rodzaj, a, b)
```

## Zadanie 2a

*Wartości domyślne argumentów funkcji. Wywoływanie funkcji w pętli.*

Rozszerz funkcję *wypisz\_zdanie()* w taki sposób, by przed dopełnieniem przyjmowała dodatkowy argument *przydawka*, który będzie określać dopełnienie zdania. *Domyślną* wartością przydawki ma być

ciąg znaków 'małego'.

```
def wypisz_zdanie(podmiot='Ala', orzeczenie='ma', dopelnienie='kota'):  
    print podmiot + ' ' + orzeczenie + ' ' + dopelnienie
```

Następnie napisz program, który w czterech *zagnieżdżonych* pętlach będzie przetwarzał cztery listy argumentów dla funkcji wypisz zdanie. Przykład dla dwóch pętli przetwarzających argumenty podmiot i orzeczenie znajduje się poniżej.

```
if __name__ == '__main__':  
    podmioty = ['Ala', 'Bob', 'Karol']  
    orzeczenia = ['ma', 'widzi', 'szuka']  
    for podmiot in podmioty:  
        for orzeczenie in orzeczenia:  
            wypisz_zdanie(podmiot, orzeczenie)
```

## Zadanie 2b

*Korzystanie z modułów języka Python.*

Aby uniknąć konieczności pisania wielu *zagnieżdżonych* pętli, twórcy języka Python przygotowali funkcję *product* w pakiecie *itertools*. Zmień swój program z zadania 2 wzorując się na poniższym przykładzie i zamień cztery zagnieżdżone pętle w jedną pętlę iterującą po *czterech* zmiennych.

```
import itertools  
  
def wypisz_zdanie(podmiot='Ala', orzeczenie='ma', dopelnienie='kota'):  
    print podmiot + ' ' + orzeczenie + ' ' + dopelnienie  
  
if __name__ == '__main__':  
    podmioty = ['Ala', 'Bob', 'Karol']  
    orzeczenia = ['ma', 'widzi', 'szuka']  
    for podmiot, orzeczenie in itertools.product(podmioty, orzeczenia):  
        wypisz_zdanie(podmiot, orzeczenie)
```

## Zadanie 3

*Język Python i pliki csv.*

Skorzystamy teraz z pakietu *csv*, aby wczytać z pliku szereg wartości, a następnie zliczyć ich wystąpienia i przedstawić statystyki zbiorów danych. Przykład z pracy domowej jest co prawda prosty, ale zasadnicze idee naukowego przetwarzania danych są w nim obecne. Zajmiemy się poniższą tabelą:

Imię	Płeć	Wzrost	Waga
Szymon	M	180	80

Anna	K	160	60
Karol	M	165	58
Tadeusz	M	170	90
Zofia	K	185	70
Helena	K	155	56
Bartosz	M	172	69
Jan	M	164	73
Judyta	K	167	59
Agata	K	163	53

Format *csv* (ang. *comma-separated values*) to popularny sposób zapisu danych tabelarycznych (więcej informacji np. tutaj: [http://pl.wikipedia.org/wiki/CSV\\_\(format\\_pliku\)](http://pl.wikipedia.org/wiki/CSV_(format_pliku))). Założymy, że jako separatora używać będziemy przecinka i że przecinek nie wystąpi jako fragment napisu lub liczby (znak oddzielający część dziesiętną).

Dane z powyższej tabeli zapiszemy zatem w następujący sposób:

```
Szymon,M,180,80
Anna,K,160,60
Karol,M,165,58
Tadeusz,M,170,90
Zofia,K,185,70
Helena,K,155,56
Bartosz,M,172,69
Jan,M,164,73
Judyta,K,167,59
Agata,K,163,53
```

Skopiuj powyższe dane do pliku `dane.csv`, a następnie przejdź do dalszej części zdania.

Aby przetworzyć plik `csv` w naszym programie, skorzystamy z operacji na plikach (funkcja `open()`) oraz z pakietu `csv`. Zadanie polega na uzupełnieniu poniższego programu tak, by obliczyć średni wzrost i wagę w grupach mężczyzn i kobiet. Dodatkowym atutem programu będzie znalezienie imion najniższego mężczyzny i najwyższej kobiety.

```
import csv

if __name__ == '__main__':
    mezczyzni = 0
    kobiety = 0

    plik = open('dane.csv', 'r')
    for imie,plec,wzrost,waga in csv.reader(plik):
        if plec == 'M':
            mezczyzni += 1
        else:
```

```
        kobiety += 1
print 'srednia waga wsrod mezczyzn to: '
print 'srednia waga wsrod kobiet to: '
```

Uwaga! W powyższej pętli zmienne *waga* i *wzrost* są typu *string*, zatem przed wykonaniem na nich operacji liczbowych należy *zrzutować* je na typ *int* lub *float*, np.:

```
suma_wag += float(waga)
```