

Programowanie w Python

Paweł Jamer
Mateusz Bieniek

Zadanie 1

Znajdujący się na tym slajdzie kod Pythonowy, oblicza w sposób przybliżony i uproszczony wartość zarobków netto, na podstawie zarobków brutto. Kod ten zawiera liczne błędy składniowe.

Popraw kod w taki sposób, żeby działał poprawnie. Oczekiwany wynik, jaki powinien zwrócić kod, to:

```
# Płaca brutto: 3200.000000
# * Ub. społeczne: 438.720000
# *** Ub. emeryt.: 312.320000
# *** Ub. rentow.: 48.000000
# *** Ub. chorob.: 78.400000
# * Ub. zdrowotne: 248.515200
# * Podatek doch.: 169.158400
# Płaca netto: 2343.606400
```

```
podatek_dochod = 0,17
skl_emeryt; skl_rentow; skl_chorob = 0.0976; 0.0150; 0.0245
ubezp_zdrow_staw1, ubezpie_zdrow_staw2 = 9%, 7.75%
```

```
koszt_uzyskania = 250.00
kwota_wolna = 43.76
```

```
placa_brutto = 3200
```

```
emeryt <- skl_emeryt * placa_brutto
rentow <- skl_rentow * placa_brutto
chorob <- skl_chorob * placa_brutto
spoleczn <- emeryt + rentow + chorob
```

```
przychod = placa_brutto - spoleczn
zdrow == ubezpie_zdrow_staw1 * przychod
```

```
podst_opod = przychod - koszt_uzyskania
podatek = podatek_dochod ** podst_opod - kwota_wolna
podatek -= ubezpie_zdrow_staw2 * przychod
```

```
placa_brutto - spoleczn - zdrow - podatek -> placa_netto
```

```
print('
    Płaca brutto: %f
    * Ub. społeczne: %f
    *** Ub. emeryt.: %f
    *** Ub. rentow.: %f
    *** Ub. chorob.: %f
    * Ub. zdrowotne: %f
    * Podatek doch.: %f
    Płaca netto: %f
' % (
    placa_brutto, spoleczn, emeryt, rentow, chorob, zdrow, podatek, placa_netto
))
```

Zadanie 2

Znajdujący się na tym slajdzie kod Pythonowy, oblicza punkty zdobyte przez studentów po drugich zajęciach (pts2), bazując na

- punktach, które zdobyli na pierwszych zajęciach (pts1),
- obecności na drugich zajęciach (pres2),
- ocenie pracy domowej, oddanej po drugich zajęciach (home2).

Odpowiedz na poniższe pytania.

1. W jaki sposób można uprościć 15-tą linię kodu?
2. Jak uprościć fragment kodu z linii od 17 do 22 do postaci jedno-linijkowej?
3. Dlaczego w linii 25 kodu okazuje się, że zawartość listy pts1 uległa zmianie, pomimo tego, że nie była nigdzie modyfikowana? Jak uniknąć modyfikacji zawartości listy str1?

```
1 pts1 = [3, 1, 3, 2, 1, 3, 2, 3, 2, 1, 3, 2, 1, 3, 3]
2
3 pres2 = [
4     True, True, True, True, False, True, True, True, True,
5     True, False, False, True, False, True
6 ]
7 home2 = [
8     'perf', 'perf', 'perf', 'good', 'good', 'good', 'perf',
9     'none', 'good', 'good', 'good', 'none', 'good', 'perf',
10    'good'
11 ]
12
13 pts2 = pts1
14 for i in range(len(pts2)):
15     pts2[i] += 1 if pres2[i] else 0
16
17     if home2[i] == 'perf':
18         pts2[i] += 2
19     elif home2[i] == 'good':
20         pts2[i] += 1
21     else:
22         pts2[i] += 0
23
24 print(pts2)
25 print(pts1)
```

Zadanie 3

Plik frankenstein.txt zawiera treść książki o tym samym tytule. Napisz fragment kodu, który:

1. wczyta zawartość tej książki do Pythona,
2. policzy ile razy w treści książki wystąpił każdy ze znajdujących się w niej symboli, takich jak litera, liczba, znak interpunkcyjny itp.,
3. wyświetli występujące w książce symbole w kolejności od najczęściej spotykanego, do najrzadziej spotykanego.

Zadanie 4

Wyjaśnij krok po kroku, co stanie się po wykonaniu linii 7 poniższego kodu.

```
1 # HOSPITALIZACJE OSÓB CHORYCH NA COVID-19
2 hosp = (15444, 16144, 16427, 17223, 18160, 18654, 19114)
3
4 def transform2(ts, func):
5     return [func(v0, v1) for v0, v1 in zip(hosp[:-1], hosp[1:])]
6
7 transform2(hosp, lambda x0, x1: x1 - x0)
```

Wykorzystaj funkcję transform2 do policzenia dobowej zmiany procentowej liczby osób hospitalizowanych.

Zadanie 5

Stwórz klasę `Sample`, reprezentującą jednowymiarową próbkę statystyczną. Klasa ta powinna zostać zaimplementowana w taki sposób, żeby na jej obiektach możliwe było wykonanie zamieszczonego na tym slajdzie fragmentu kodu i uzyskanie wyników analogicznych do zaprezentowanych.

```
# Utworzenie próbki.  
s = Sample([1, 2, 3])  
  
# Zwrócenie wszystkich wartości próbki.  
s.get_vals()  
[1, 2, 3]  
  
# Zwrócenie wartości i-tej obserwacji z próbki.  
s.get_val(1)  
1  
  
# Ustawienie wartości i-tej obserwacji w próbce.  
s.set_val(1, 222)  
s.get_vals()  
[222, 2, 3]  
  
# Dodanie nowej obserwacji na koniec próbki.  
s.add_val(12345)  
s.get_vals()  
[222, 2, 3, 12345]
```

Zadanie 6

Stwórz klasę ExtSample, która rozszerza funkcjonalność klasy Sample. Klasa ExtSample, poza funkcjonalnościami klasy Sample, powinna dodatkowo oferować te funkcjonalności, które prezentuje kod zawarty na tym slajdzie.

```
es = ExtSample([1, 2, 3, 4])
es.get_vals()
[1, 2, 3, 4]

es.add_val(5)
es.get_vals()
[1, 2, 3, 4, 5]

# Policzenie sumy z wartości obserwacji
# będących w próbce.
es.sum()
15

# Policzenie iloczynu z wartości obserwacji
# będących w próbce.
es.mul()
120

# Policzenie średniej z wartości obserwacji
# będących w próbce.
es.avg()
3.0
```

Zadanie 7

Stwórz klasę Vector, reprezentującą obiekt wektora matematycznego.

Klasa ta powinna zostać zaimplementowana w taki sposób, żeby na jej obiektach możliwe było wykonanie działania dodawania wektorów za pomocą operatora + oraz mnożenia skalarnego wektorów, za pomocą operatora *.

```
v1 = Vector([1, 2, 1, 5])  
v2 = Vector([2, 3, 1, 4])
```

```
# [1+2, 2+3, 1+1, 5+4]  
v3 = v1 + v2  
v3.get_vals()  
[3, 5, 2, 9]
```

```
# 1*2 + 2*3 + 1*1 + 5*4  
v1 * v2  
29
```


Zadanie 8

Załaduj plik **bank_register.rds** używając Pythona.

Plik **bank_register.rds** zawiera dane dotyczące klientów pewnego banku w postaci następującej struktury.

id	date	income	demographic	products
<i>Identyfikator klienta i umowy.</i>	<i>Data dokonania wpisu.</i>	<i>Roczny dochód klienta.</i>	<i>Płeć, wiek i liczba dzieci.</i>	<i>Posiadane obecnie produkty bankowe.</i>
463_1	Jul 21, 2018	15.331,22\$	F,37,0	DEP,MOR

Przekształć te dane do poniższej postaci.

client_id	agreement_id	date	income	sex	age	child	dep	cre	mor
<i>Id klienta.</i>	<i>Id umowy.</i>	<i>Data.</i>	<i>Dochód.</i>	<i>Płeć.</i>	<i>Wiek.</i>	<i>Dzieci.</i>	<i>Czy posiada depozyt?</i>	<i>Czy posiada kredyt?</i>	<i>Czy posiada kredyt hipoteczny?</i>
463	1	2018-06-21	15331.22	F	37	0	TRUE	FALSE	TRUE

Zadanie 9

Załaduj plik **ugly_diamonds.csv** do Pythona i:

- Wyznacz średnią, medianę oraz odchylenie standardowe cen w grupach wg kolumn *cut* oraz *color*
- Wybierz z tabeli tylko te wiersze, gdzie *cut* = Premium i skopiuj do nowej zmiennej
- Utwórz kopię tabeli, a w niej nową kolumnę $V = x * y * z$
- Oblicz ile obserwacji zawierają grupy z kolumny *clarity*