

Tworzenie własnych programów do układania dawek pokarmowych i ich implementacja w serwisie sieciowym

Zad. 6.1

W Pythonie utwórz następujący skrypt (*File > New File*) pozwalający na obliczanie kaloryczności karmy (w kcal/100 g oraz kJ/100 g) na podstawie wyników jej analizy chemicznej (tab. 6.1). Uruchom skrypt za pomocą klawisza F5.

```
a = input("Podaj zawartość białka surowego [%]: ")
b = input("Podaj zawartość tłuszczu surowego [%]: ")
c = input("Podaj zawartość węglowodanów [%]: ")
e = 3.5 * float(a) + 8.5 * float(b) + 3.5 * float(c)
f = 14.64 * float(a) + 35.56 * float(b) + 14.64 * float(c)
print("kaloryczność karmy wynosi: {0} kcal/100 g \
lub: {1} kJ/100 g".format(round(e, 2), round(f, 2)))
```

Tabela 6.1 Wyniki analizy chemicznej przykładowej karmy

Białko surowe	27%
Tłuszcz surowy	10%
Węglowodany*	47%

*jako BAW – bezazotowe związki wyciągowe

Zad. 6.2

W Pythonie utwórz następujący skrypt pozwalający na obliczanie zawartości składników w suchej masie karmy. Przykładowe obliczenia wykonaj dla karm, których skład podany jest w tabeli 6.2. Do obliczenia zawartości poszczególnych składników w suchej masie wykorzystaj wzór:

$$a = b / sm * 100,$$

gdzie: a – zawartość składnika w suchej masie, b - oryginalna zawartość danego składnika, sm - zawartość suchej masy.

```
a = input("Podaj zawartość białka surowego [%]: ")
b = input("Podaj zawartość tłuszczu surowego [%]: ")
c = input("Podaj zawartość węglowodanów [%]: ")
d = input("Podaj zawartość popiołu [%]: ")
e = input("Podaj zawartość włókna [%]: ")
```

```

w = input("Podaj zawartość wody [%]: ")
sm = float(100) - float(w)
asm = float(a) / sm * 100
bsm = float(b) / sm * 100
csm = float(c) / sm * 100
dsm = float(d) / sm * 100
esm = float(e) / sm * 100

print("zawartość składników w s.m.:\n {0}% białka surowego\n \
{1}% tłuszczu surowego\n {2}% węglowodanów\n \
{3}% popiołu\n {4}% włókna".format(round(asm, 2),
round(bsm,2), round(csm, 2), round(dsm, 2), round(esm, 2)))

```

Tabela 6.2. Przykładowa analiza chemiczna wybranych karm dla psów dorosłych

Typ produktu	Białko [%]	Tłuszcz [%]	Popiół [%]	Włókno [%]	BAW [%]	Woda [%]
Karma mokra	10,0	7,0	1,6	0,5	2,9	78,0
Karma mokra	9,0	5,5	2,5	0,5	0,5	82,0
Karma sucha	20,0	13,0	8,0	3,0	48,0	8,0
Karma sucha	27,0	12,0	7,0	3,0	43,0	8,0

Zad. 6.3

W Pythonie utwórz następujący skrypt pozwalający na obliczanie ilości karmy (w g) jaką powinien otrzymywać dziennie pies określonej rasy przy danym poziomie aktywności ruchowej. Zapotrzebowanie spoczynkowe psa (RER – ang. *resting energy requirement*) oblicz na podstawie wzoru: $RER = 70 \cdot (\text{masa ciała})^{0,75}$, zaś wartość dziennego zapotrzebowania na energię metaboliczną (DER – ang. *daily energy requirement*) oblicz za pomocą formuły na podstawie współczynnika aktywności ruchowej psa odczytanego z tabeli 6.3. Ilość karmy (w g/dzień) wylicz z odpowiedniej proporcji. Przykładowe obliczenia wykonaj dla niekastrowanego, mieszkającego w bloku i wychodzącego na dłuższe spacerki raz dziennie psa (kategoria 2) o masie ciała 40 kg otrzymującego karmę o kaloryczności 344 kcal/100 g.

```

a = input("Podaj EM karmy [kcal/100 g]: ")
b = input("Podaj masę ciała psa [kg]: ")
print("kategorie:\n \
pies dorosły sterylizowany - 1\n \
pies dorosły niesterylizowany - 2\n \
pies otyły mało aktywny - 3\n \
odchudzanie - 4\n \

```

```

odzywianie w intensywnej terapii - 5\n \
odzyskanie masy ciała w okresie rekonwalescencji - 6")
c = input("Podaj kategorię: ")
rer = 70 * float(b)**(0.75)
kat = {"1": 1.6, "2": 1.8, "3": 1.4, "4": 1.0, "5": 1.0, "6": 1.3}
der = kat[c] * rer
ilosc = der * 100 / float(a)
print("RER: {0} kcal/dzień".format(round(rer, 2)))
print("DER: {0} kcal/dzień".format(round(der, 2)))
print("Ilość karmy: {0} g/dzień".format(round(ilosc, 2)))

```

Tabela 6.3 Dienne zapotrzebowanie na energię metaboliczną (DER – ang. *daily energy requirement*) psów dorosłych

kategoria	znaczenie	DER (·RER) [kcal/dzień]
1	pies dorosły sterylizowany	1,6
2	pies dorosły niesterylizowany	1,8
3	pies otyły mało aktywny	1,4
4	odchudzanie	1,0
5	odżywianie w intensywnej terapii	1,0
6	odzyskanie masy ciała w okresie rekonwalescencji	1,2 - 1,4

Zad. 6.4

Na podstawie zadania 6.3 utwórz w Pythonie podobny skrypt, który pozwalałby na obliczenie dziennej ilości karmy (w g) jaką powinien otrzymywać pies o określonej masie ciała przy uwzględnieniu stadium wzrostu organizmu (tab. 6.4). Przykładowe obliczenia wykonaj dla suki w wieku 4 miesięcy o masie ciała 5,5 kg (50% masy ciała osobnika dorosłego) otrzymującej karmę o kaloryczności 358 kcal/100 g.

Tabela 6.4 Dienne zapotrzebowanie na energię metaboliczną (DER) psów rosnących

kategoria	DER (·RER)
od odsadzenia do uzyskania 50% masy ciała psa dorosłego	3,0
od uzyskania 50% masy ciała psa dorosłego do uzyskania 80% masy ciała psa dorosłego	2,5
od uzyskania 80% masy ciała psa dorosłego do uzyskania 100% masy ciała psa dorosłego	2,0
100% masy ciała psa dorosłego	1,8

Zad. 6.5

W Pythonie utwórz następujący skrypt pozwalający na obliczanie dziennego zapotrzebowania na witaminę B₁₂ (µg/kg) dorosłego psa na podstawie wartości zalecanego dziennego pobrania (tab. 6.5).

Przykładowe obliczenia wykonaj dla dorosłego niekastrowanego psa mieszańca w wieku 4 lat, o masie ciała 22 kg, biegającego około 1,5 godziny dziennie (kategoria 2; kaloryczność karmy 330 kcal/100 g).

```

a = input("Podaj EM karmy [kcal/100 g]: ")
b = input("Podaj masę ciała psa [kg]: ")
print("kategorie:\n \
pies dorosły sterylizowany - 1\n \
pies dorosły niesterylizowany - 2\n \
pies otyły mało aktywny - 3\n \
odchudzanie - 4\n \
odżywianie w intensywnej terapii - 5\n \
odzyskanie masy ciała w okresie rekonwalescencji - 6")
c = input("Podaj kategorię: ")
d = input("Podaj min. zapotrzebowanie na witaminę B12 [ug/1000 kcal]: ")
e = input("Podaj max. zapotrzebowanie na witaminę B12 [ug/1000 kcal]: ")
rer = 70 * float(b)**(0.75)
kat = {"1": 1.6, "2": 1.8, "3": 1.4, "4": 1.0, "5": 1.0, "6": 1.3}
der = kat[c] * rer
ilosc = der * 100 / float(a)
witmin1 = float(d) * der / 1000
witmax1 = float(e) * der / 1000
witmin2 = witmin1 * 1000 / ilosc
witmax2 = witmax1 * 1000 / ilosc
print("RER: {0} kcal/dzień".format(round(rer, 2)))
print("DER: {0} kcal/dzień".format(round(der, 2)))
print("Ilość karmy: {0} g/dzień".format(round(ilosc, 2)))
print("Minimalna zawartość witaminy B12 w karmie [ug/kg]: \
{0}".format(round(witmin2, 2)))
print("Maksymalna zawartość witaminy B12 w karmie [ug/kg]: \
{0}".format(round(witmax2, 2)))

```

Tabela 6.5 Zalecane dzienne pobranie witaminy B₁₂

Zapotrzebowanie na witaminę B ₁₂	min	max
zalecane dzienne pobranie [µg/1000 kcal]	5,50	8,75

Zad. 6.6

W Pythonie utwórz skrypt pozwalający na obliczanie dziennego zapotrzebowania na energię metaboliczną (DER) ciężarnych suk. Masę ciała suki i tydzień ciąży wprowadza użytkownik, zaś DER wyliczane jest przez skrypt na podstawie tabeli 6.6. Przykładowe obliczenia wykonaj dla suki o masie ciała 10 kg w 5. tygodniu ciąży.

Tabela 6.6 Dzielne zapotrzebowanie energetyczne suki (DER) wyliczane zależnie od tygodnia ciąży

Tydzień ciąży	Wzór do wyliczania DER ciężarnej suki
1 - 4	1,9·RER
5	DER do 4 tygodnia + 18 kcal/kg masy ciała
6 - 8	DER do 4 tygodnia + 36 kcal/kg masy ciała
9	DER do 4 tygodnia + 18 kcal/kg masy ciała

Zad. 6.7

Utwórz podobny skrypt dla suk w okresie laktacji w zależności od liczebności miotu (tab. 6.7). Przykładowe obliczenia wykonaj dla suki o masie ciała 15 kg i czterech szczeniętach w miocie.

Tabela 6.7 Dzielne zapotrzebowanie energetyczne suk (DER) w okresie laktacji w zależności od liczebności miotu

Liczba szczeniąt w miocie	Wzór do wyliczania DER suki w okresie laktacji
1	DER = 3,0 * RER
2	DER = 3,5 * RER
3 - 4	DER = 4,0 * RER
5 - 6	DER = 5,0 * RER
7 - 8	DER = 5,5 * RER
9	DER > 6,0 * RER

Zad. 6.8

Zaimplementuj skrypt Pythona z zad. 6.1 na stronie WWW korzystając z poniższego kodu (skorzystaj np. z notatnika systemowego zapisując plik jako HTML).

```
<!DOCTYPE html>
<html lang="pl">
  <meta charset="UTF-8">
  <title>Implementacja Pythona</title>
  <script src="https://cdnjs.cloudflare.com/ajax/libs/brython/3.8.8/brython.js"
integrity="sha256-rA89wPrTJJQFWJaZveKW8jpdmC3t5F9rRkPyBjz8G04="
crossorigin="anonymous"></script>
  <script
src="https://cdnjs.cloudflare.com/ajax/libs/brython/3.8.8/brython_stdlib.js"
```

```

integrity="sha256-Gnrw9tIjrsXcZSCh/wos5Jrpn0bNVNFJuNJI9d71TDs="
crossorigin="anonymous"></script>

</head>

<body bgcolor="yellow" onload="brython()">
  <script type="text/python">
    from browser import document

    def click(ev):
        a = document["bialko"].value
        b = document["tluszcz"].value
        c = document["wegl"].value
        d = 3.5 * float(a) + 8.5 * float(b) + 3.5 * float(c)
        e = 14.64 * float(a) + 35.56 * float(b) + 14.64 * float(c)
        document["wynik1"].textContent = round(d, 2)
        document["wynik2"].textContent = round(e, 2)

    document["ok"].bind("click", click)
  </script>
  <label for="bialko">Podaj zawartość białka surowego [%]: </label>
  <input id="bialko"><br><br>
  <label for="tluszcz">Podaj zawartość tłuszczu surowego [%]: </label>
  <input id="tluszcz"><br><br>
  <label for="wegl">Podaj zawartość węglowodanów [%]: </label>
  <input id="wegl"><br><br>
  <button id="ok">OK</button>
  <p>Kaloryczność karmy wynosi: <b><span id="wynik1"></span></b> kcal/100 g</p>
  <p>Kaloryczność karmy wynosi: <b><span id="wynik2"></span></b> kJ/100 g</p>
</body>
</html>

```

Zad. 6.9

W podobny sposób jak w zad. 6.8 zaimplementuj skrypt Pythona z zad. 6.2.

Źródła:

<https://towardsdatascience.com/run-python-code-on-websites-exploring-brython-83c43fb7ac5f>

https://brython.info/static_doc/en/intro.html