

# Wstęp do Sztucznej Inteligencji - rok akademicki 2022/2023

Przed rozpoczęciem pracy z notatnikiem zmień jego nazwę zgodnie z wzorem:  
NrAlbumu\_Nazwisko\_Imie\_PoprzedniaNazwa.

Przed wystąpieniem notatnika upewnij się, że rozwiązałeś wszystkie zadania/ćwiczenia.

## Zadanie 1 (obowiązkowe, 10pkt.)

Aby wygenerować treść zadania wprowadź swój nr albumu i wykonaj poniższe komórki (Shift + Enter)

```
nr_albumu = '142706'

#import sys
#from google.colab import drive
#drive.mount('/content/drive')
# upewnij się że poniższa ścieżka jest poprawna
#path_nb = r'/content/drive/My Drive/Colab
Notebooks/WdSI_2022/T1_Lab_Prolog/'
#sys.path.append(path_nb)

import pickle
from IPython.display import HTML

#zadania = pickle.load(open(path_nb+'zadania.pkl', 'rb'))
zadania = pickle.load(open('zadania.pkl', 'rb'))
nr = int(nr_albumu) % len(zadania)
przyjaciele = zadania[nr]['przyjaciele']
zwierzeta = zadania[nr]['zwierzeta']
fakty = ['{}. {}'.format(i+1, s) for i,s in enumerate(zadania[nr]
['fakty'])]

tresc = """<HTML><BODY><H3>Zadanie dla numeru albumu: {}.<br>Zadanie
nr {}.</H3><br>
Przyjaciele:<br>
<b>{}</b> <br> mieszkają w jednym budynku, każdy na innym piętrze (0,
1, 2, ...).
<br><br>
Każdy z nich ma inne zwierzę:<br>
<b>{}</b><br><br>
Na którym piętrze mieszka każdy z nich i jakie ma zwierzę, jeśli
następujące zdania są prawdą?<br>
<b>{}</b></BODY></HTML>""".format(nr_albumu, nr+1, '
```

```
'.join(przyjaciele), ' '.join(zwierzeta), '<br>'.join(fakty))
HTML(tresc)
```

```
-----
-----
NameError                                Traceback (most recent call
last)
```

```
Cell In[1], line 6
```

```
    4 #zadania = pickle.load(open(path_nb+'zadania.pkl', 'rb'))
    5 zadania = pickle.load(open('zadania.pkl', 'rb'))
----> 6 nr = int(nr_albumu) % len(zadania)
      7 przyjaciele = zadania[nr]['przyjaciele']
      8 zwierzeta = zadania[nr]['zwierzeta']
```

```
NameError: name 'nr_albumu' is not defined
```

## Instrukcje

- Rozwiąż przydzielone sobie zadanie za pomocą odpowiedniego programu w Prologu.
- W komórce TWÓJ PROGRAM przedstaw ten program.
- W komórce ODPOWIEDŹ podaj rozwiązanie w postaci:

```
Jarek 0 kot
Franek 3 chomik
Stefan 1 rybka
...
```

## TWÓJ PROGRAM:

```
floor(0).
floor(1).
floor(2).
floor(3).
floor(4).
floor(5).
floor(6).
floor(Jarek, Franek, Stefan, Albert, Szymon, Robert, Marcin) :-
    floor(J),
    floor(F),
    floor(S),
    floor(A),
    floor(Sz),
    floor(R),
    floor(M),
    Jarek = J,
    Franek = F,
    Stefan = S,
    Albert = A,
    Szymon = Sz,
    Robert = R,
```

```
Marcin = M,  
S < J,  
Sz > F,  
S > R,  
R < Sz,  
M < F,  
A > M,  
J > A,  
Sz < S,  
F > R,  
F < S,  
R < J,  
F > A,  
J > M,  
M > R,  
M < S,  
A < Sz,  
S > A,  
Sz > M,  
A > R,  
J > Sz,  
J > F.
```

```
person(jarek).  
person(franek).  
person(stefan).  
person(albert).  
person(szymon).  
person(robert).  
person(marcin).
```

```
animal(pies).  
animal(kot).  
animal(rybka).  
animal(chomik).  
animal(papuga).  
animal(waz).  
animal(kanarek).
```

```
condition(Item) :-  
    Item = [X, 5, pies],  
    X \= szymon,  
    X \= jarek,  
    X \= franek,  
    X \= robert,  
    X \= marcin.  
condition(Item) :-  
    Item = [X, 6, chomik],  
    X \= robert,  
    X \= marcin,
```

```

    X \= szymon,
    X \= franek.
condition(Item) :-
    Item = [X, 0, papuga],
    X \= szymon,
    X \= marcin,
    X \= jarek,
    X \= stefan,
    X \= franek.
condition(Item) :-
    Item = [X, 2, waz],
    X \= jarek,
    X \= marcin,
    X \= szymon,
    X \= stefan,
    X \= robert,
    X \= franek.
condition(Item) :-
    Item = [X, 3, kanarek],
    X \= marcin,
    X \= stefan,
    X \= robert,
    X \= szymon,
    X \= jarek.
condition(Item) :-
    Item = [X, 4, rybka],
    X \= marcin,
    X \= robert,
    X \= jarek,
    X \= albert,
    X \= franek.
condition(Item) :-
    Item = [X, 1, kot],
    X \= robert,
    X \= stefan,
    X \= szymon,
    X \= albert,
    X \= jarek,
    X \= franek.

apartment(List, Final_Result) :-
    List = [Person, Floor, Animal],
    person(Person),
    floor(Floor),
    animal(Animal),
    condition(List),
    Final_Result = List.

finalResult(Results) :-

```

```
floor(Ja, Fr, St, Al, Szy, Ro, Ma),  
apartment([jarek, Ja, _], Result_1),  
apartment([frank, Fr, _], Result_2),  
apartment([stefan, St, _], Result_3),  
apartment([albert, Al, _], Result_4),  
apartment([szymon, Szy, _], Result_5),  
apartment([robert, Ro, _], Result_6),  
apartment([marcin, Ma, _], Result_7),  
Results = [Result_1, Result_2, Result_3, Result_4, Result_5,  
Result_6, Result_7].
```

## ODPOWIEDŹ:

```
? - finalResult(R).
```

```
R = [[jarek, 6, chomik], [frank, 3, kanarek], [stefan, 5, pies],  
[albert, 2, waz], [szymon, 4, rybka], [robert, 0, papuga], [marcin, 1,  
kot]]
```